



Grant Agreement no. 643529



iManageCancer

*Empowering patients and strengthening
self-management in cancer diseases*

Research and Innovation Action

**PHC-26-2014: Self management of health and disease:
citizen engagement and mHealth**

D3.4 Extended integrated prototype of iManageCancer platform

Contractual Due Date: 31 July 2017
Actual Submission Date: 15 September 2017

Lead partner for deliverable: FORTH

Dissemination Level: Public

Revision: v1.6

COVER AND CONTROL PAGE OF DOCUMENT	
Project Acronym:	<i>iManageCancer</i>
Project Full Name:	Empowering patients and strengthening self-management in cancer diseases
Project Duration	1 February 2015 - 31 July 2018
Deliverable No.:	D3.4
Deliverable Name:	Extended integrated prototype of iManageCancer platform
Nature (R, DEM) ¹	DEM
Dissemination Level (PU, CO) ²	PU
Version:	1.6
Actual Submission Date:	15 September 2017
Editor: Institution: E-Mail:	Haridimos Kondylakis FORTH kondylak@ics.forth.gr
Contributors (Institution)	<ul style="list-style-type: none"> - Fatima Schera, Michael Schäfer, Stephan Kiefer (FRAU) - Lefteris Koumakis (FORTH) - Zhikun Deng (BED) - Stefan Hoffmann (SGS) - Eric Herve Ngantchjon (PHILIPS-NL)
Reviewers (Institution)	<ul style="list-style-type: none"> • Norbert Graf (USAAR) • Fatima Schera (FRAU)

Copyright

© Copyright 2017 iManageCancer Consortium

The research leading to these results has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 643529.

The author(s) is (are) solely responsible for the content of this document, it does not represent the opinion of the European Commission and the Commission is not responsible for any use that might be made of the information it contains.

¹ **R** = Document, report (excluding the periodic and final reports), **DEM** = Demonstrator, pilot, prototype, plan designs

² **PU** = Public, fully open, e.g. web, **CO** = Confidential, restricted under conditions set out in Model Grant Agreement

Document History

Issue Date	Version	Changes Made / Reason for this Issue
31.07.17	0.9	Version for internal review procedure
31.08.17	1.0	Updates according to reviewers comments
14.09.17	1.6	Document ready for submission

Table of Contents

1. Executive summary	8
2. Introduction and overview	9
3. iManageCancer Portal iPHR	12
3.1. Intended Purpose	13
3.2. Provided functionality	13
3.3. Technical implementation	14
3.3.1. System concept and internal architecture	14
3.3.2. Interfaces	15
3.3.3. Database	17
3.3.4. Data protection	17
3.4. User manual	17
3.5. Installation and configuration guidelines	17
4. MyHealthAvatar for iManageCancer (Prostate Cancer)	19
4.1. Intended Purpose	19
4.2. Provided functionality	19
4.2.1. Function Suite	19
4.2.2. Goals	26
4.2.3. Location	27
4.2.4. Journal	28
4.2.5. Calendar	31
4.2.6. Profile	31
4.2.7. Sharing	33
4.3. Technical implementation	34
4.3.1. System concept and internal architecture	34
4.3.4. Data protection	46
4.4. User manual	48
4.4.1. Starting the MyHealthAvatar-iManageCancer App	48
4.4.2. Accessing the Main page from the Start page	48
4.4.3. Accessing the Location page from the Start page	49
4.4.4. Access Left Menu	51
4.4.5. Access Settings Page	51
4.4.6. Access Overview page and Function Suite	57
5. MyHealthAvatar for iManageCancer (Breast Cancer)	85
5.1. Intended Purpose	85
5.2. Provided functionality	85
5.2.1. Function Suite	85
5.2.2. Goals	93
5.2.3. Location	94
5.2.4. Journal	95
5.2.5. Calendar	98
5.2.6. Profile	99
5.2.7. Sharing	100
5.3. Technical implementation	101
5.3.1. System concept and internal architecture	101
5.3.2. Interfaces	106
5.3.3. Database	110
5.3.4. Data protection	114
5.4. User manual	115
5.4.1. Starting the MyHealthAvatar-iManageCancer App	115

5.4.2.	Accessing the Main page from the Start page.....	116
5.4.3.	Accessing the Location page from the Start page	116
5.4.4.	Access Left Menu	118
5.4.5.	Access Settings Page.....	118
5.4.6.	Access Overview page and Function Suite	125
5.4.7.	Access Journal Function	140
5.4.8.	Access Calendar Function.....	145
5.4.9.	Access Profile Function	146
5.4.10.	Access Sharing Function	149
5.5.	Installation and configuration guidelines	150
6.	Serious game for adult cancer patients	152
6.1.	Intended Purpose	152
6.2.	Provided functionality	152
6.2.1.	Multi Language Support	152
6.2.2.	Account Management	152
6.2.3.	Customisable Avatars.....	153
6.2.4.	Social Interactions	153
6.2.5.	iMC Rewards System.....	154
6.2.6.	3D Game world	154
6.2.7.	Main Game Mechanics	155
6.2.8.	Trivia Mini-game	155
6.2.9.	Feedback System.....	156
6.2.10.	Game Back-end Systems	156
6.3.	Technical implementation	156
6.3.1.	Introduction	156
6.3.2.	Interfaces	158
6.3.3.	Databases and structures	159
6.3.4.	Data Protection.....	160
6.4.	User manual	160
6.4.1.	Application launching	160
6.4.2.	Language Selection	161
6.4.3.	Login & Account Management.....	161
6.4.4.	Character Creation	162
6.4.5.	Character Customisation	162
6.4.6.	Level Selection.....	163
6.4.7.	Game Introduction	163
6.4.8.	Main Game Mechanics	164
6.4.9.	Progression & Winning.....	164
6.4.10.	Building new Buildings	165
6.4.11.	Social Interface	165
6.4.12.	iMC Rewards Interface.....	166
6.4.13.	Trivia Mini-game.....	166
6.4.14.	Feedback Dialogs	167
6.5.	Installation and configuration guidelines	167
7.	Serious game for paediatric cancer patients.....	167
7.1.	Intended Purpose	167
7.2.	Provided functionality	167
7.2.1.	Decision menu.....	167
7.2.2.	Patient shooter game	168
7.2.3.	Supporter shooter game.....	168
7.2.4.	Game backend	168

7.3.	Technical implementation	169
7.3.1.	System concept and internal architecture	169
7.3.2.	Interfaces	169
7.3.3.	Database	171
7.3.4.	Data protection	172
7.4.	User manual	173
7.4.1.	Starting the Serious Game for Kids (First Time)	173
7.4.2.	Login Screen Test	173
7.4.3.	Choose Game – Mode	174
7.4.4.	Dashboard Test.....	174
7.4.5.	Main Buttons – Dashboard: Option Test	175
7.4.6.	Main buttons – Dashboard: Catwalk Test	176
7.4.7.	Level Selection Map Test.....	177
7.4.8.	Special Weapon Menu Test	178
7.4.9.	Shooter Game.....	179
7.4.10.	Present Transfer between Supporter and Patient.....	181
7.5.	Installation and configuration guidelines	181
8.	iManageMyHealth.....	182
8.1.	Intended Purpose	182
8.2.	Provided functionality	182
8.2.1.	Drug management	182
8.2.2.	Managing health documents	183
8.2.3.	Vital sign monitoring	183
8.2.4.	Care Flow Engine triggered management services	184
8.3.	Technical implementation	184
8.3.1.	System concept and internal architecture	184
8.3.2.	Interfaces	186
8.3.3.	Database	186
8.3.4.	Data protection	193
8.4.	User manual	194
8.4.1.	Module <i>My Drugs</i>	198
8.4.2.	Module <i>Measurements</i>	207
8.4.3.	Module Health documents	209
8.4.4.	Module ‘ <i>Health Management</i> ’ (in combination with app iSupportMyPatients) ..	213
8.5.	Installation and configuration guidelines	217
8.5.1.	Configuration	219
9.	iSupportMyPatients	220
9.1.	Intended Purpose	220
9.2.	Provided functionality	220
9.3.	Technical implementation	220
9.3.1.	System concept and internal architecture	220
9.3.2.	Interfaces	220
9.3.3.	Database	220
9.3.4.	Data protection	220
9.4.	User manual	221
9.5.	Installation and configuration guidelines	222
10.	CareFlowEngine.....	223
10.1.	Intended Purpose	223
10.2.	Provided functionality	223
10.3.	Technical implementation	223
10.3.1.	System concept and internal architecture	223

10.3.2.	Interfaces	226
10.3.3.	Database	226
10.3.4.	Data protection	226
10.4.	User manual	226
10.5.	Installation and configuration guidelines	234
11.	Model Repository	235
11.1.	Intended Purpose	235
11.2.	Provided functionality	235
11.3.	Technical implementation	236
11.3.1.	System concept and internal architecture	236
11.3.2.	Interfaces	236
11.3.3.	Database	238
11.3.4.	Data protection	238
11.4.	User manual	238
11.4.1.	Available models	238
11.4.2.	Rest API.....	239
11.5.	Installation and configuration guidelines	242
11.5.1.	Setting up Wildfly 10.0.0	242
11.6.	Database	248
12.	Conclusions	249

1. Executive summary

This deliverable presents the technical documentation of the **extended** iManageCancer platform prototype and is a successor of D3.2 the technical documentation of the initial iManageCancer platform prototype. The initial prototype became available during January 2017 and the extended version became available on July 2017. It consists of the applications iPHR, MyHealthAvatar for iManageCancer, iSupportMyPatients and iManageMyHealth for cancer patients and/or health professionals and the game for kids, all connected to a backbone of services comprising a data store with the personal health records, the Care Flow Engine, a model repository and an authentication service. This deliverable describes the extended iManageCancer platform, following several cycles of improvements after an initial validation presented at D3.2 and small scale user studies.

With the provision of the extended version of the iManageCancer apps and games, the milestone MS3³ of the project is achieved.

The intended purpose of the apps and game has been outlined in a way that the use of the tools for diagnosis and therapy is excluded. Those features that are only useful in such a context are provided for test and demonstration purposes only.

³ MS2: Extended integrated prototype of iManageCancer platform. Software released for clinical validation. Related deliverables submitted.

2. Introduction and overview

In July 2017 the development team of iManageCancer provided the extended integrated version of the iManageCancer platform.

The extended iManageCancer platform comprises a set of four applications and two games for the end users of the platform. Figure 1 summarizes their purpose and main features.

Furthermore, this deliverable presents the Central Decision Support Framework of the platform with the Care Flow Engine and the Model Repository. Figure 2 shows the high-level architecture of the platform. More details on the architecture are given in D3.3. The Central Decision Support Framework is explained in deliverable D5.2.

In order not to overload this document we reference further deliverables here that describe already parts of this prototype:

- D3.2 Updated iManageCancer architecture including semantic interoperability methodology
- D3.3 Updated iManageCancer architecture including semantic interoperability methodology
- D4.2 Health Avatar and PHR Services
- D5.2 Initial decision support and patient guidance services integrated in iManageCancer Platform
- D6.2 Generic health enquiry tool
- D6.3 Initial versions of psycho-emotional monitoring instrument, family evaluation tool and monitoring tool for life style and vital signs

					
	MyHealth Avatar 4iMC	iManage MyHealth	iSupport MyPatients	iPHR	Game for Kids
	BED	FRAU	FRAU	FORTH	SGS
Users	patients	patients	Health professionals	patients, doctors, researchers	patients, supporters
Devices	Android 5, 6	Android 5, 6	Android 5, 6	Web app: all	Android, iOS
Gadgets	actimeters, scale	scale, BPM, thermometer			
Intended purpose	life style management	general health management	health management	health management	encouragement & education
Integrated tools	eDiary	drug management, vital signs monitoring, documents management, care flows	care flows: pain, neutropenia risk prediction	eDiary and PHR, data sharing, Health information Recommender, psycho-emotional monitoring	

Figure 1: Overview about the different end user applications provided by the initial iManageCancer platform prototype.

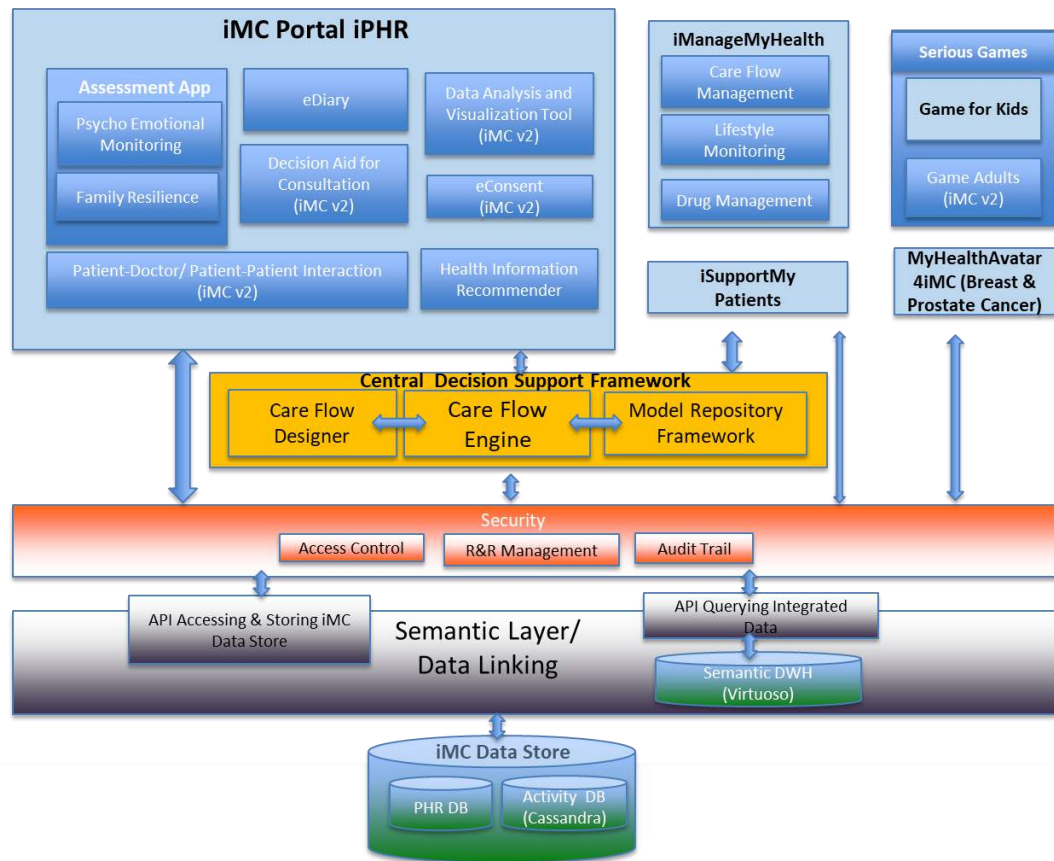


Figure 2: High-level architecture of initial iManageCancer platform prototype. (Components or features in the application layer that are part of the next version of the platform are marked with iMC v2).

3. iManageCancer Portal iPHR

The iManageCancer Portal (iPHR) with its related tools and services is developed by FORTH.

iPHR enables an individual to own and manage a complete, secure, digital copy of his/her health and wellness information. It integrates health information across sites of care and over time. The system is essentially an inversion of the current approach to medical records, in that the record resides with the patients and the patients grant permissions to institutions, clinicians, researchers, and other users of medical information. iPHR is a distributed, web-based, personally controlled electronic medical record system that is ubiquitously accessible to the nomadic user, built to public standards.

iPHR is composed of many applications including:

- ***eDiary - Calendar***: A timeline view of all available information showing medications, problems, appointments and procedures.
- ***Demographics***: View and update patient information such as gender, date of birth, contact information, name and surname.
- ***Labs***: View and update laboratory result values.
- ***Problems***: View and update diseases, illnesses, injuries, physiologic, mental or psychological condition or disorders.
- ***Procedures***: View and update medical treatments or surgeries of the patient.
- ***Allergies***: View and update abnormal reactions to encountered allergens.
- ***Medications***: View and update drugs or other substances received.
- ***Measurements***: View and update vital signs that indicate the status of the body's vital functions.
- ***Contact***: A form to communicate with other users of the system.
- ***Appointments***: View and update appointments between the patient and doctors.
- ***Upload Documents***: Upload your health data documents.
- ***Assessment app***: Psycho-Emotional and Family Resilience Questionnaires
- ***Personal Health Information Recommender***: Allow patients to search in a high-quality document repository for useful information.
- ***Smart analytical data services*** – A whole framework enabling the data analysis of patient data. Data can be visualized using multiple visualization paradigms. In addition multiple data analysis techniques such as feature selection and clustering are available whereas alerts can also be generated in order for the researchers/doctors to be notified when certain conditions are met. More information on the Smart Analytical Data services can be found in D8.1 and D8.2.
- ***Patient-Doctor and Patient-Patient interactions (forum, chat)*** – These tools enable patient to patient and patient to doctor online and offline communication.
- ***eConsent Tool*** – This enables researchers to make requests for data access which administrators should check and approve. Then the corresponding patients should give their consent in order for the data to be further anonymized and used by these researchers.
- ***Decision Aid for Consultation*** – This tool enables patients to perform a shared decision making with their doctors, to better understand the consequences of their choices and to select the therapy most appropriate for them.

3.1. *Intended Purpose*

A patient using an iPHR has the ability to run different applications as outlined above. In principle, iPHR is a user friendly way to store and view the individual clinical and activity data. The patient can enter all kind of data he/she creates. Such data are measurement results like blood glucose, temperature, weight, blood pressure, heart rate, pain or mood data on a scale range and others. Furthermore, he/she will be able to enter the main data of his/her tumour disease (passport screening) including treatment plan and other medications. In this regard the patient can enter any kind of information in an electronic diary, i.e. symptoms, problems with therapies, mood, level of pain, level of anxiety, and level of appetite. He/she can also provide a personal profile with habits, hobbies, favourite meals, etc. The patient can decide if he/she wants to link his social network with the PHR. He/she can view the documents on a timeline, or see the data as charts or in a tabular format. For each of the documents or data the patient is able to select those she/he wants to share with other people like other clinicians or relatives. He/she can grant access to them for a selected time period and he/she is able to withdraw such a granting at any time without giving an explanation. The information is entered in an intuitive way. The software structures the information in a way that further processing of data by the system is possible. If appointments to a clinical visit or schedules for medications are entered then a notification for the patient will be visually or acoustically provided by the system at the actual time point.

3.2. *Provided functionality*

The goal of iPHR system is to provide an innovative ecosystem and enhance the self-management principle of patients through the involvement of all stakeholders (family, friends, healthcare professionals, etc) participating in the therapeutic process.

The iPHR moves beyond the current state of the art of existing PHR systems and provides enhanced functionality in six main directions:

- **GUI:** It provides a nice, user friendly graphical user interface that is platform independent and optimized for mobile devices (laptops, tablets, mobile devices etc.).
- **Sharing & interoperability:** Multiple roles are supported such as patients, health professionals, companions and researchers allowing the secure and seamless sharing of selective information, enhancing the patient-doctor and patient-patient interaction and communication and enabling researches to access statistical information. In addition, researchers can request statistical data, provided by the system as long as the requests are approved by the administrator and the patients have given the corresponding consent.
- **e-Diary:** It further optimizes interactions of the participants using e-diaries, allowing the patients to enter and to view their activities and behaviors across different period of time
- **Multiple apps:** Besides legacy apps for managing and recording the individual health status (i.e. problems, allergies, medications, procedures, laboratory results etc.), novel apps have been implemented focusing on the psycho-emotional monitoring of the cancer patient, providing intelligent services (e.g. drug interactions and recommendations, alerts, patient profiling etc.) and managing medical documents.
- **Advanced data management:** The users are able to be connected through their iManageCancer account to external data sources such as activity trackers, sensors and social media. To this direction a novel big-data infrastructure has been designed and implemented allowing the uninterrupted addition of future data sources.
- **Intelligent Data Analytic services:** Data analysis and data mining in iManageCancer is an iterative approach, which combines data from the semantic layer of iManageCancer, pre-processes the data, performs the analysis and provides the results for visualization

based on the data distillation model. All these are implemented in an intelligent data analytic framework available to the doctors and researchers. A detailed presentation of the intelligent data analytic services is available at D8.1 Report on implemented data analysis and data mining services.

- **Involve stakeholders in the design and development process:** Right from the beginning of the development process all involved stakeholders were heavily involved in the development process. Besides the requirements elicitation phase where more than two hundred possible end-users were contacted, earlier versions of the system have already been evaluated by a diverse set of patients and physicians at three different places and time points and the results have been used to further improve the system.

3.3. Technical implementation

3.3.1. System concept and internal architecture

The architecture of the iPHR system is shown in the following figure and consists of three layers.

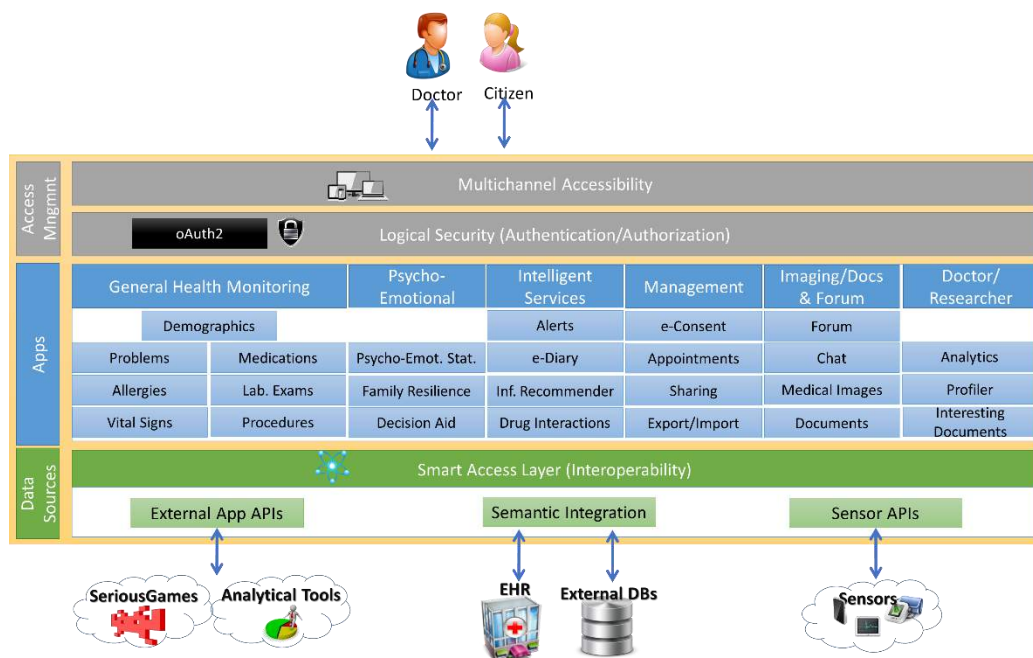


Figure 3: Architecture of iPHR.

- The data service layer offering APIs for integrating, importing and exporting data: Patients can access the iPHR functionalities from any kind of device (desktop, tablet, smartphone) by just using a web browser. By using the provided GUI, the patients can access the intelligent apps offered by the apps layer, enabling them to manage their data from anywhere with an active internet connection. The interface is automatically adapted to the screen analysis of the device and appropriate information is automatically hidden or visualized to optimize user experience. In addition, the interface is translated into four languages (English, German, Italian and Greek) whereas adding a new translation is only matter of translating a single text file.
- The app layer offering all available app functionalities: The apps layer allows patients to view, gather and manage their medical information in an accessible and secure method. The Platform is designed to encourage the patient to take a more active role of their health management, enhance clinician-patient communication, maximize compliance to therapy and produce intelligent recommendations. There are general healthh monitoring apps,

psycho-emotional monitoring apps, intelligent services, management apps, imaging/doc apps and apps specifically for doctors and researchers.

- The access management layer enabling secure role management and multi-device compatibility: The data service layer links the iPHR to external data sources such as activity trackers, sensors and electronic health records. Other types of data can also be pushed on demand to the infrastructure using their native data model. To store this vast amount of raw data in their native format including structured, semi-structured and unstructured data an Apache Cassandra⁴³ database is used as an instantiation of a data lake concept.

3.3.2. Interfaces

Three main public interfaces have been implemented within the iManageCancer portal, the API for iPHR, the OAuth2.0 and the API for the data store.

API for iPHR:

Provides the following procedures for accessing the iPHR data:

1 ACCOUNTS

- 1.1 POST <https://iphr.ics.forth.gr/api/accounts/>
- 1.2 GET <https://iphr.ics.forth.gr/api/accounts/>
- 1.3 POST https://iphr.ics.forth.gr/api/accounts/{ACCOUNT_EMAIL}/authsystems/
- 1.4 GET <https://iphr.ics.forth.gr/api/accounts/search>
- 1.5 POST
https://iphr.ics.forth.gr/api/accounts/{ACCOUNT_EMAIL}/authsystems/password/change
- 1.6 POST https://iphr.ics.forth.gr/api/accounts/{ACCOUNT_EMAIL}/forgot-password
- 1.7 GET https://iphr.ics.forth.gr/api/accounts/{ACCOUNT_EMAIL}
- 1.8 GET https://iphr.ics.forth.gr/api/accounts/{ACCOUNT_EMAIL}/inbox/
- 1.9 POST https://iphr.ics.forth.gr/api/accounts/{ACCOUNT_EMAIL}/inbox/
- 1.10 GET https://iphr.ics.forth.gr/api/accounts/{ACCOUNT_EMAIL}/inbox/{MESSAGE_ID}
- 1.11 POST
https://iphr.ics.forth.gr/api/accounts/{ACCOUNT_EMAIL}/inbox/{MESSAGE_ID}/archive
- 1.12 GET https://iphr.ics.forth.gr/api/accounts/{ACCOUNT_EMAIL}/notifications/

2 DEMOGRAPHICS

- 2.1 GET https://iphr.ics.forth.gr/api/records/{RECORD_ID}/demographics

3 RECORDS

- 3.1 POST <https://iphr.ics.forth.gr/api/records/>
- 3.2 POST https://iphr.ics.forth.gr/api/records/{RECORD_ID}/owner
- 3.3 GET https://iphr.ics.forth.gr/api/accounts/{ACCOUNT_EMAIL}/records/

4 APPS

- 4.1 GET <https://iphr.ics.forth.gr/api/apps/>
- 4.2 GET https://iphr.ics.forth.gr/api/apps/{PHA_EMAIL}

5 APPLICATION DOCUMENTS

- 5.1 POST https://iphr.ics.forth.gr/api/records/{RECORD_ID}/documents/

5.2 GET https://iphr.ics.forth.gr/api/records/{RECORD_ID}/documents/{DOCUMENT_ID}

5.3 POST

https://iphr.ics.forth.gr/api/records/{RECORD_ID}/documents/{DOCUMENT_ID}/set-status

5.4 GET https://iphr.ics.forth.gr/api/records/{RECORD_ID}/reports/Problem/

5.5 GET https://iphr.ics.forth.gr/api/records/{RECORD_ID}/reports/Allergy/

5.6 GET https://iphr.ics.forth.gr/api/records/{RECORD_ID}/reports/Medication/

5.7 GET https://iphr.ics.forth.gr/api/records/{RECORD_ID}/reports/LabResult/

5.8 GET https://iphr.ics.forth.gr/api/records/{RECORD_ID}/reports/Procedure/

5.9 GET https://iphr.ics.forth.gr/api/records/{RECORD_ID}/reports/Measurements/

5.10 POST

https://iphr.ics.forth.gr/api/records/{RECORD_ID}/documents/{DOCUMENT_ID}/replace

5.11 POST https://iphr.ics.forth.gr/documentsUpload/{RECORD_ID}

6 CARENETS -In order to share documents with an account, you have to add the account in a carenet and also share the desired documents with the carenet

6.1 GET https://iphr.ics.forth.gr/api/records/{RECORD_ID}/carenets/

6.2 POST https://iphr.ics.forth.gr/api/carenets/{CARENET_ID}/accounts/

6.3 GET https://iphr.ics.forth.gr/api/carenets/{CARENET_ID}/accounts/

6.4 DELETE https://iphr.ics.forth.gr/api/carenets/{CARENET_ID}/accounts/{ACCOUNT_ID}

6.5 POST https://iphr.ics.forth.gr/api/carenets/{CARENET_ID}/rename

6.6 DELETE https://iphr.ics.forth.gr/api/carenets/{CARENET_ID}

6.7 PUT

https://iphr.ics.forth.gr/api/records/{RECORD_ID}/documents/{DOCUMENT_ID}/carenets/{CARENET_ID}

6.8 DELETE

https://iphr.ics.forth.gr/api/records/{RECORD_ID}/documents/{DOCUMENT_ID}/carenets/{CARENET_ID}

6.9 GET

https://iphr.ics.forth.gr/api/records/{RECORD_ID}/documents/{DOCUMENT_ID}/carenets/

7 MODELS

7.1 Medication model

7.2 File model

API for Indivo can be found in the wiki pages of iManageCancer (internal – restricted access).

https://atlas.ics.forth.gr/iManageCancer/wiki/index.php/Indivo_API

API for OAuth2 (provider)

For accessing PHR services, using OAuth 2.0 authorization, the first step is to get an access token from the PHR provider. Details for the calls can be found in the wiki pages of iManageCancer (internal – restricted access).

API for data store:

The iPHR DB is based on IndivoX and a complete documentation for the currently existing API can be found at <http://docs.indivohealth.org/en/latest/api-reference.html>

More details can be found the D3.1 “Initial iManageCancer architecture document including state of the art report” section 4 Initial Architecture.

3.3.3. Database

The iPHR database is an efficient, effective and secure storage of all health-related information of patients. Initially the following core data models supported but as the project progresses we expect more to be added: (i) Allergy, (ii) Equipment, (iii) Immunization, (iv) Lab Result, (v) Medication, (vi) Problem, (vii) Procedure (viii) Vital Signs, and (ix) Simple Clinical Note.

The implementation of the database and the interaction with it is based on DJANGO, Python, PostgreSQL technologies.

More details can be found in the deliverable D3.1 “Initial iManageCancer architecture document including state of the art report”.

3.3.4. Data protection

Every patient registered to the platform can upload all kind of data into his PHR. As these data are private and sensible, specific requirements for data protection have been addressed. Specifically, the system supports:

1. A roles and rights management system, which controls access to the iManageCancer platform.
2. Secure upload of data via https protocol.
3. Data is stored in an encrypted way.
4. The patient has full control about his data. She/he can upload, edit and delete her/his data at any time without giving a reason.
5. The uploaded data belong to the patient.
6. Secondary usage of data is only possible after anonymisation of the data.
7. Contracts need to be signed for secondary usage of anonymised data between the patient (data provider) and the data user.
8. The patient can share his data with other people at any time. These people are not allowed to download, edit or delete the data. They gain only the allowance to read the data after signing a contract with the patient.
9. An audit trail system monitoring all the actions in the platform.
10. eConsent will be part of the platform.
11. iManageCancer guarantees the storage of the data for the lifetime of the project.

3.4. User manual

iPHR is a complex web site that incorporates various applications and services for the iManageCancer framework. For that reason a detailed documentation for the end users is already available online and can be found in the web page of the iPHR: <https://iphr.ics.forth.gr/apps/procedures/static/Tutorial.pdf>

3.5. Installation and configuration guidelines

In order to minimize the risk of security issues arising we set extra requirements in terms of technology and procedures including:

- data centers located in the EU

- compliant with EU data protection laws
- encrypted storage, the uploaded documents from the patients must be stored encrypted on the server
- Physical location server rooms must at least have ISO 27001 and ISO 27018 certifications.

Technical requirements of the iPHR server can be found in the following table:

	Hardware						Software		
	Cores		RAM		HD/SSD		OS	Software (Servers, etc)	DBs
	Min	Ideal	Min	Ideal	Min	Ideal			
iPHR (FORTH)		4		16	80	200	Ubuntu 16 LTS	Apache, Tomcat, Solr	PostgreSQL, MySQL, Virtuoso

Installation details:

- Ubuntu OS (latest version – currently 16.04 LTS)
- The backend server runs on port 8000, the UI server on port 80
- Python 2.6+
- Django version 1.3.1 and Django South DB
- Postgres
- Apache

4. MyHealthAvatar for iManageCancer (Prostate Cancer)

4.1. Intended Purpose

MyHealthAvatar (MHA) for iManageCancer is a solution that offers access, collection and sharing of long term and consistent personal health status data through an integrated digital representation, which helps prostate cancer patient with risk prediction, prevention and surgery recovery to individual citizens.

4.2. Provided functionality

4.2.1. Function Suite

A list of suite functions allow the management of information on the user's weight, activity, sleep, mood state, surgery recovery exercise and risk prediction questionnaire after the surgery.

4.2.1.1 Weight Suite

Weight Suite manage the user's weight information. User can access the weight information from Withings or MHA. User also can obtain the weekly/monthly weight summary in the Suite. Reference to D2.3-[3.2.1. use cases, Health Avatar GUI].

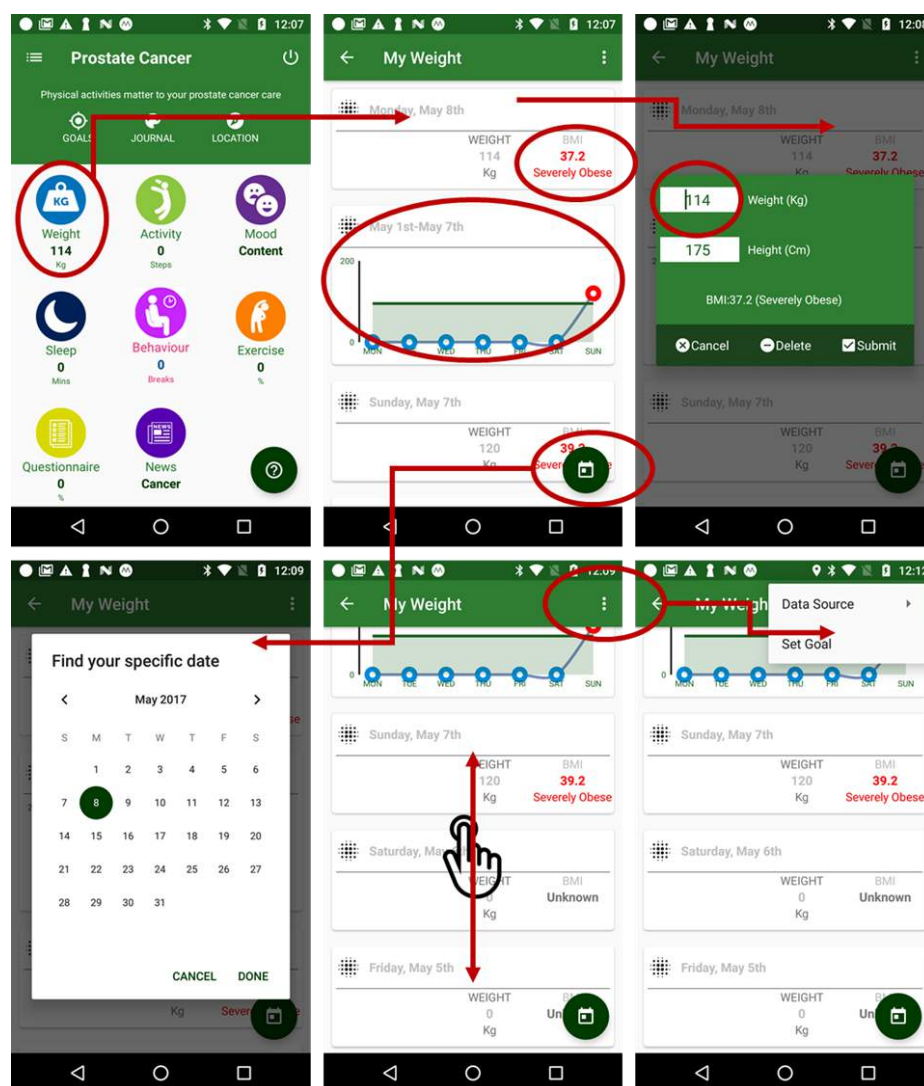


Figure 4: The Weight Suite function.

4.2.1.2 Activity Suite

Activity Suite manages the user's walk steps, distance and duration information. User can access the information from Withings, Fitbit, Moves or MHA data source. User also can obtain the weekly/monthly activity summary in the Suite. This function is reference to the requirement in D2.3-[3.2.1. use cases, Health Avatar GUI and 3.11.1. use cases, Lifestyle Monitoring].

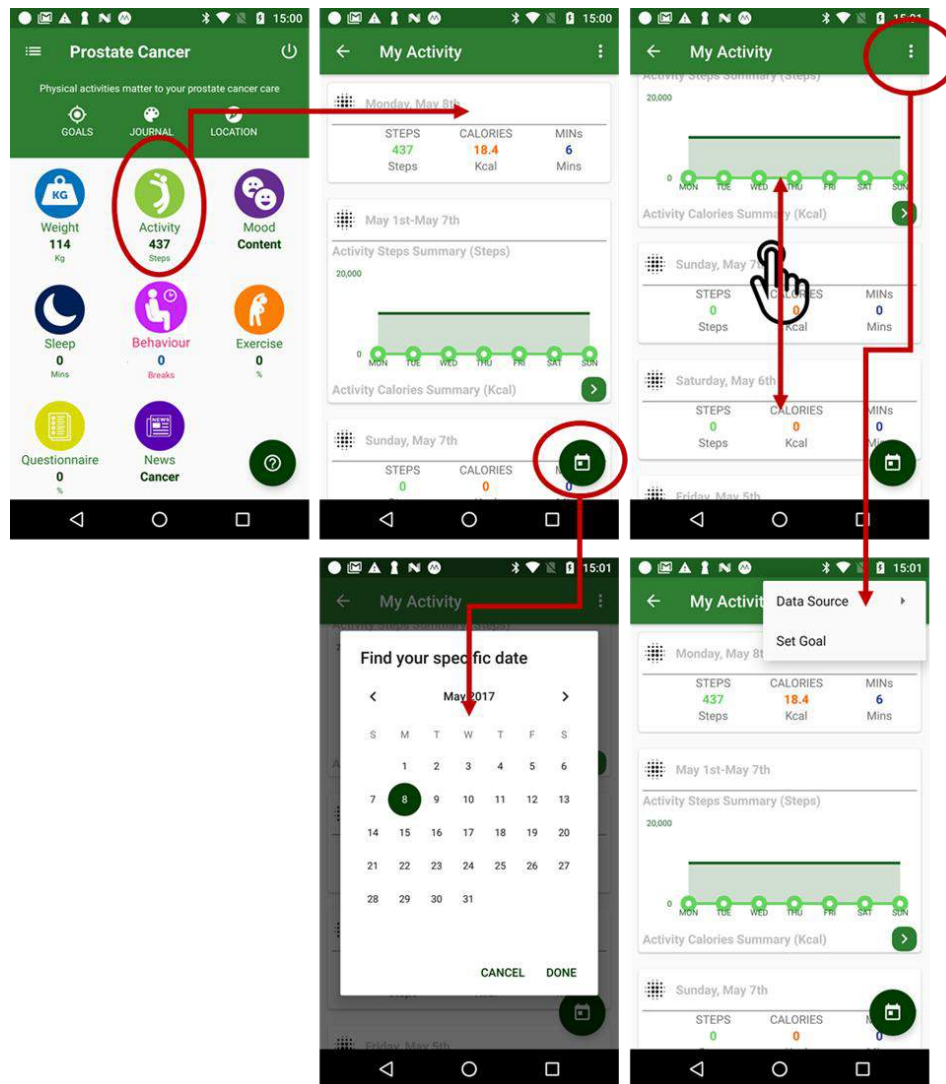


Figure 5: the Activity Suite function

4.2.1.3 Sleep Suite

Sleep Suite manages the user's sleep information. User can access the sleep duration, quality information from Fitbit or MHA data source. User also can obtain the weekly/monthly sleep summary in the Suite. This function is reference to the requirement in D2.3-[3.2.1. use cases, Health Avatar GUI and 3.11.1. use cases, Lifestyle Monitoring].

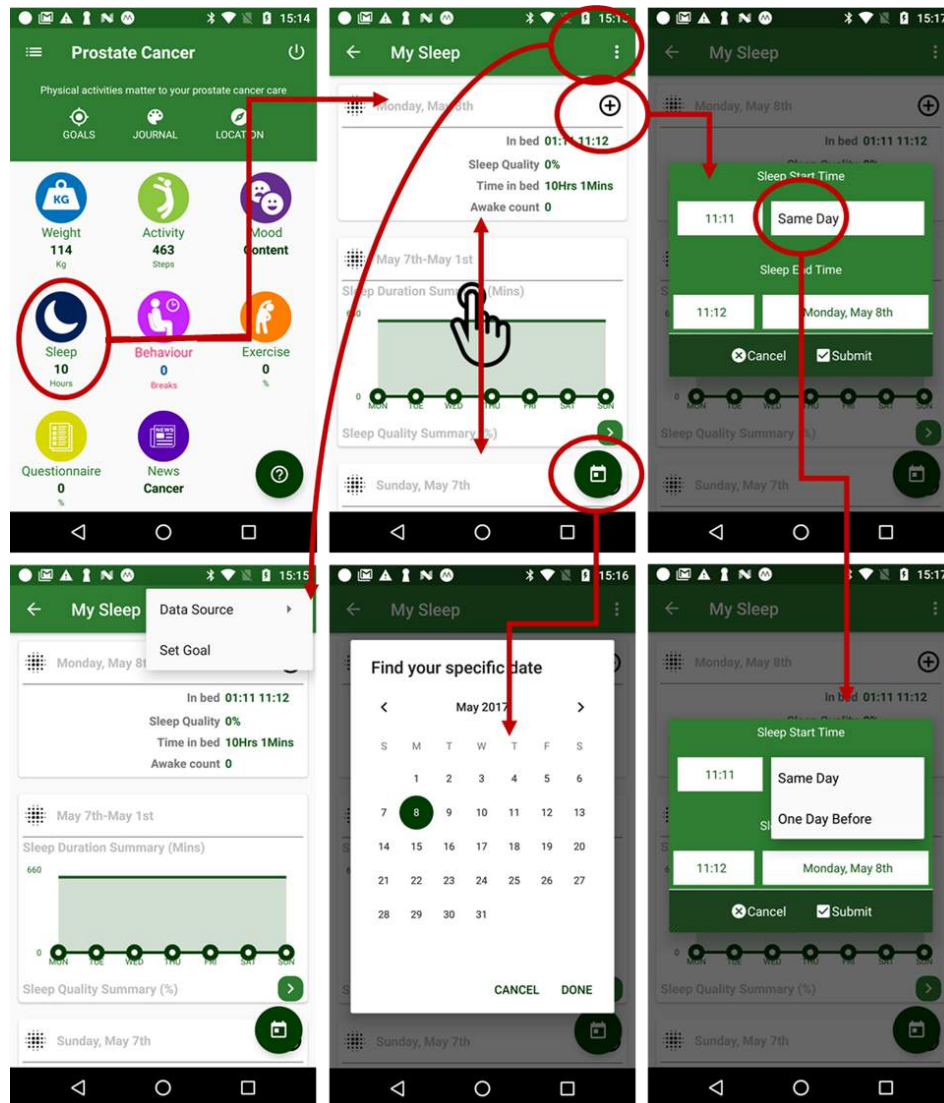


Figure 6: the Sleep Suite function

4.2.1.4 Mood Suite

Mood Suite manages the user's mood information. The user can obtain the weekly/monthly mood summary in the Suite. He/she also can use the relax program to relax herself/himself. This function is reference to the requirement in D2.3-[3.2.1. use cases, Health Avatar GUI and 3.11.1. use cases, Lifestyle Monitoring]

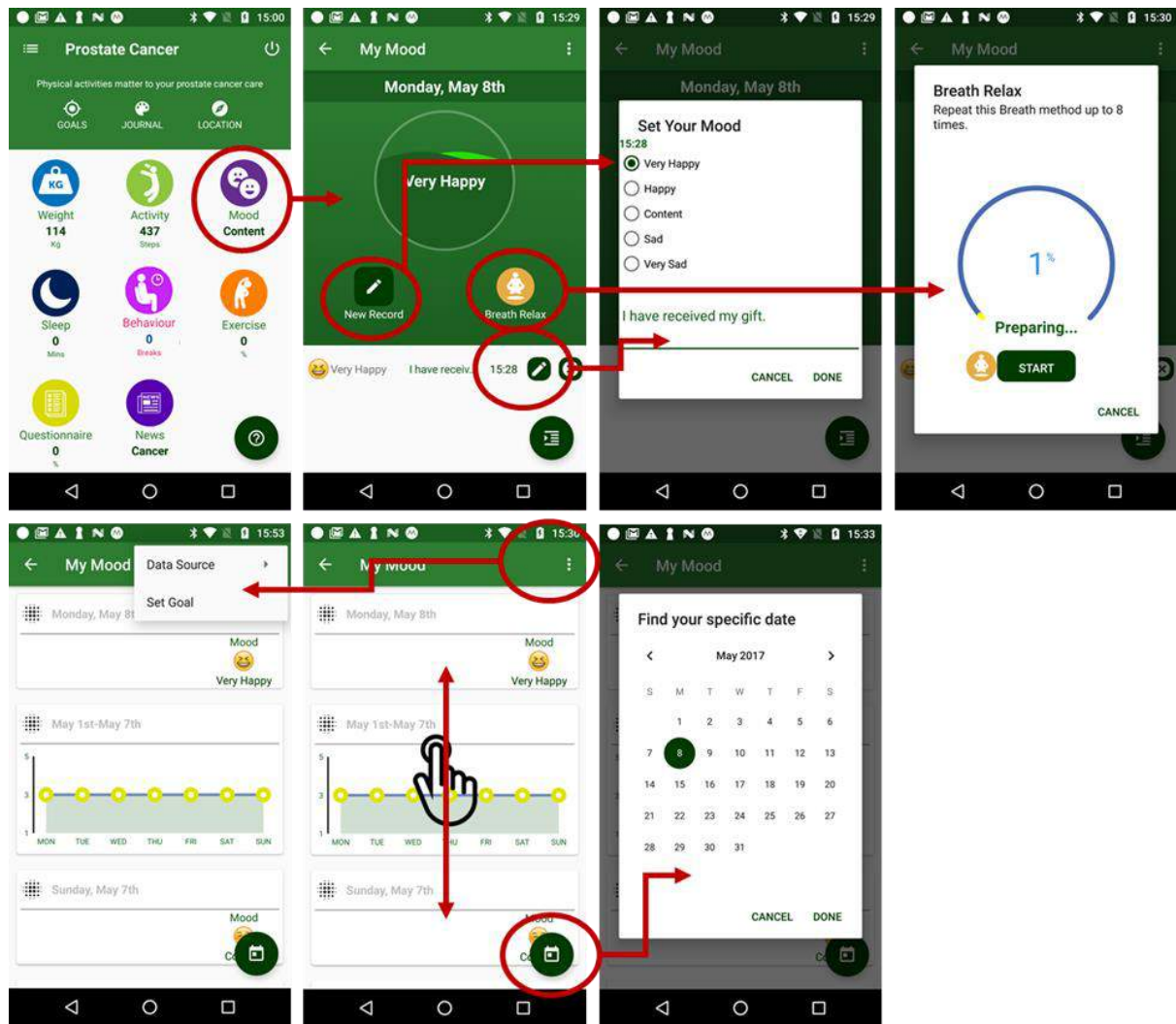


Figure 7: the Mood Suite function

4.2.1.5 Sitting Behaviour Suite

Studies have linked excessive sitting with being overweight and obese, some types of cancer like prostate and breast cancer. The behaviour suite helps the users monitor how long the sitting time and activity time during a day that due on the current day. User could help reduce long periods of sitting by standing on the train or bus, using the stairs instead of lifts, setting reminders to get up every 30 minutes, stand or walk while on the phone, walk to a co-worker's desk instead of emailing, and walk up the stairs during TV ad breaks. In this suite, Users would receive an alert every 30 mins to take an activity from sitting. Users also can choose the monitor duration to have the alerts. This function is reference to the requirement in D2.3-[3.2.1. use cases, Health Avatar GUI and 3.11.1. use cases, Lifestyle Monitoring]

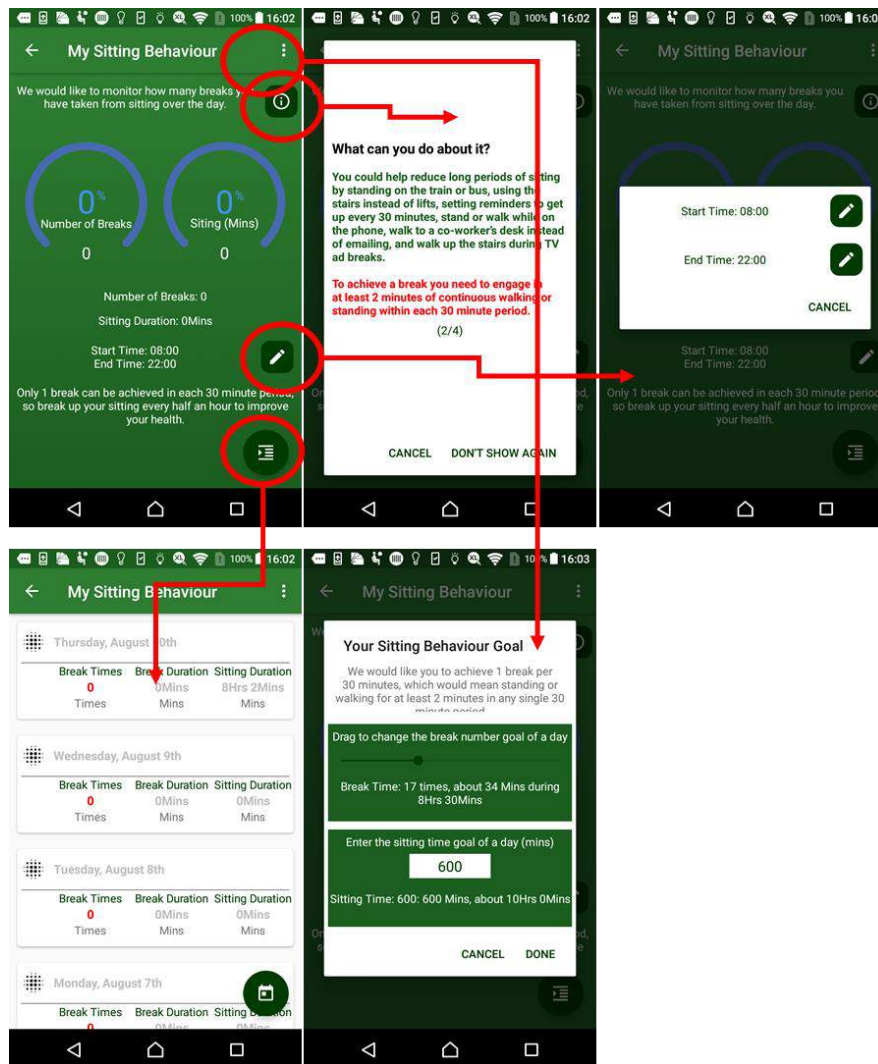


Figure 8: the Sitting Behaviour Suite function

4.2.1.6 Exercise Suite

Exercise Suite manages the patient's exercise information after surgery. There are four different exercises for the patient every day. The suite also provides the demo video for the patient. This function is reference to the requirement in D2.3-[3.2.1. use cases, Health Avatar GUI and 3.11.1. use cases, Lifestyle Monitoring].

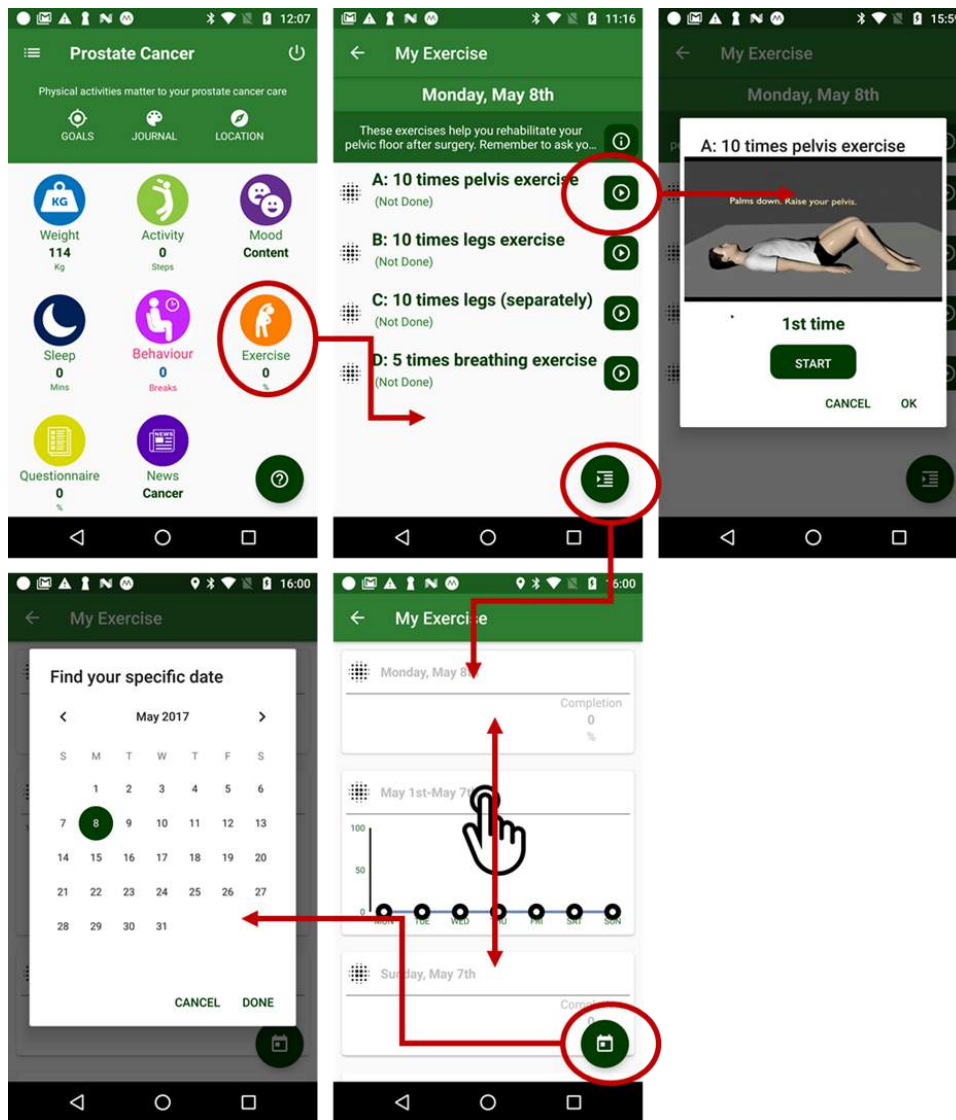


Figure 9: The Exercise Suite function.

4.2.1.7 Questionnaire Suite

Questionnaire Suite collects the patient's PSA information after surgery in order to let doctor monitor the PSA changes for the patient. Reference to D2.3-[3.2.1. use cases, Health Avatar GUI and 3.10.1. use cases, Health Enquiries].

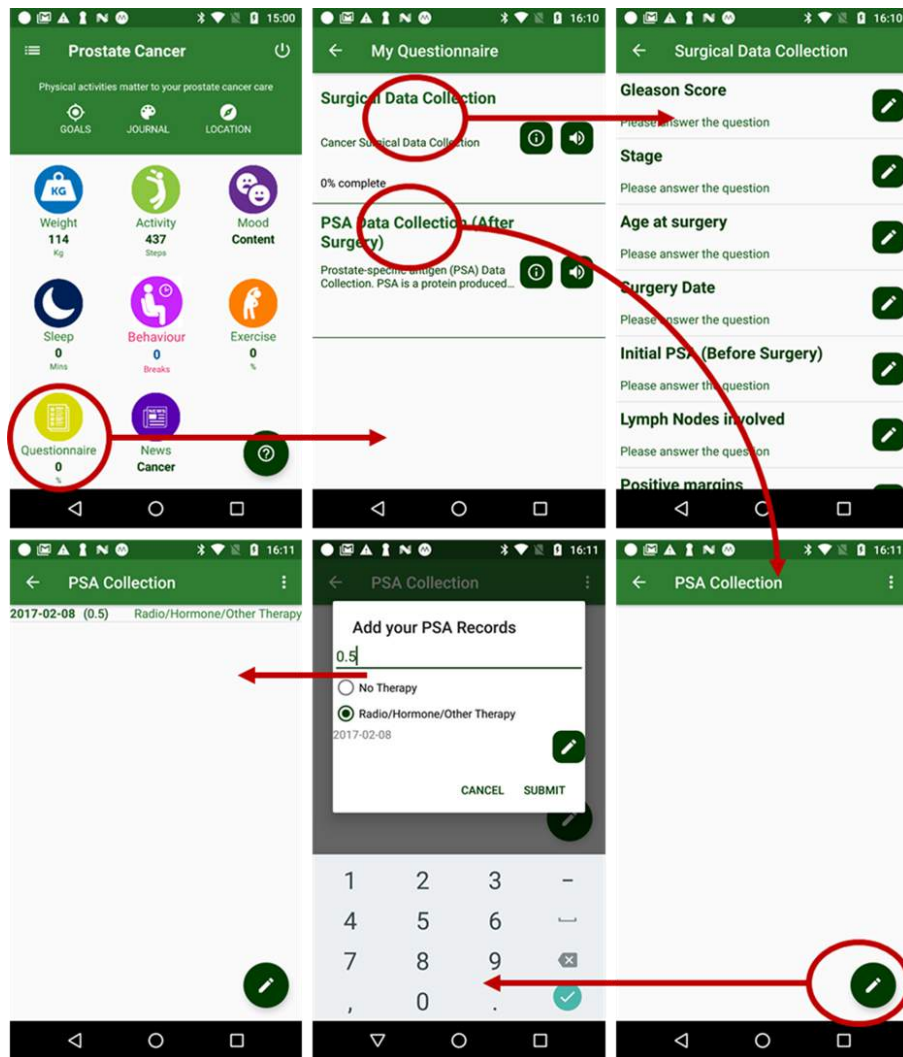


Figure 10: the Questionnaire Suite function

4.2.1.8 News Suite

This app provides guidance/news for patients with prostate cancer. By using the filter function, the patients can search and choose items on the list and view the details. For prostate cancer, patient can access the cancer information. Reference to D2.3-[3.6.1. use cases, Drug Self-Management]

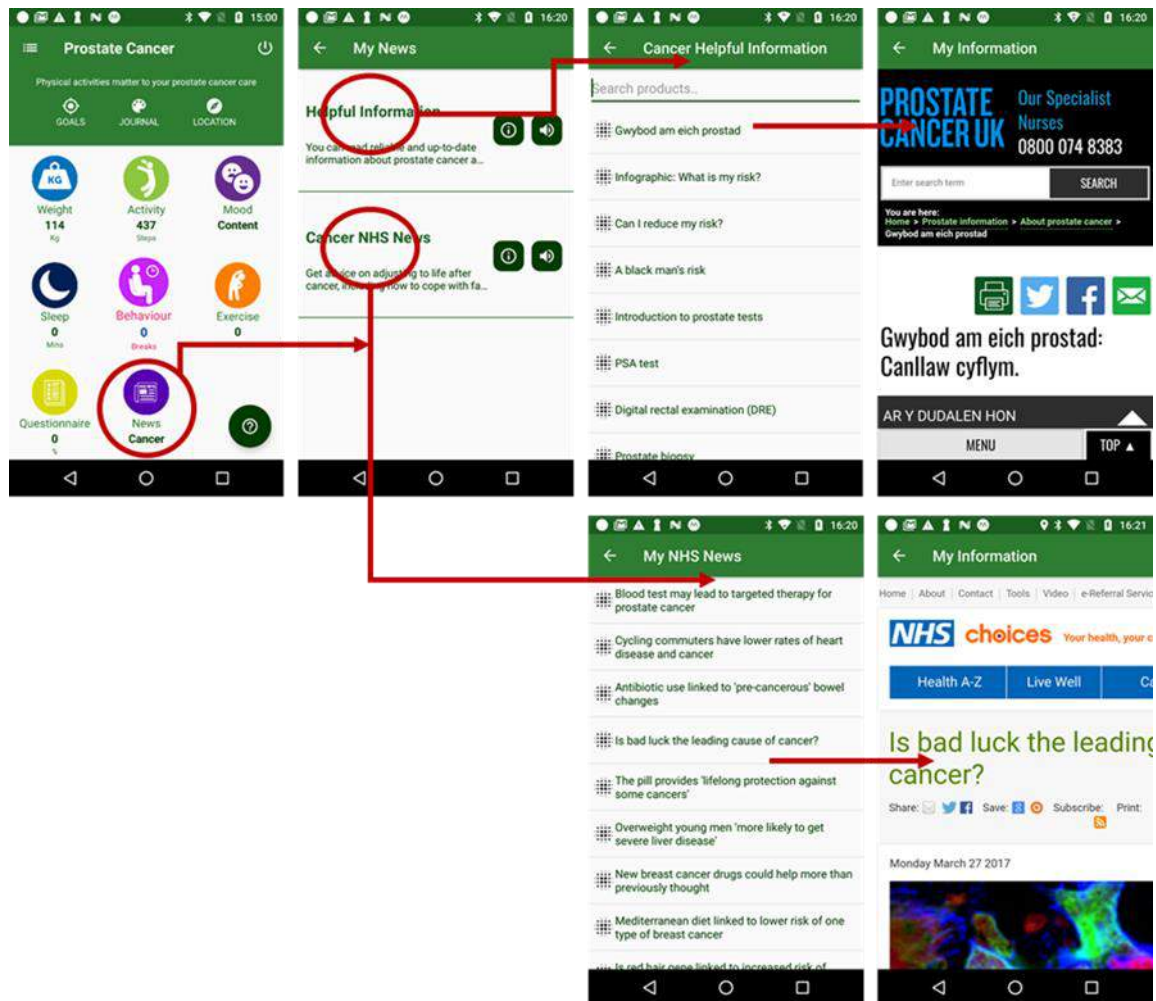


Figure 11: the News Suite function

4.2.2. Goals

User can also access to the goal setting page, the journal page and the location from the overview page on both prostate and breast cancer Apps. This function is reference to the requirement in D2.3-[3.13.1. use cases, visualisation scenario].

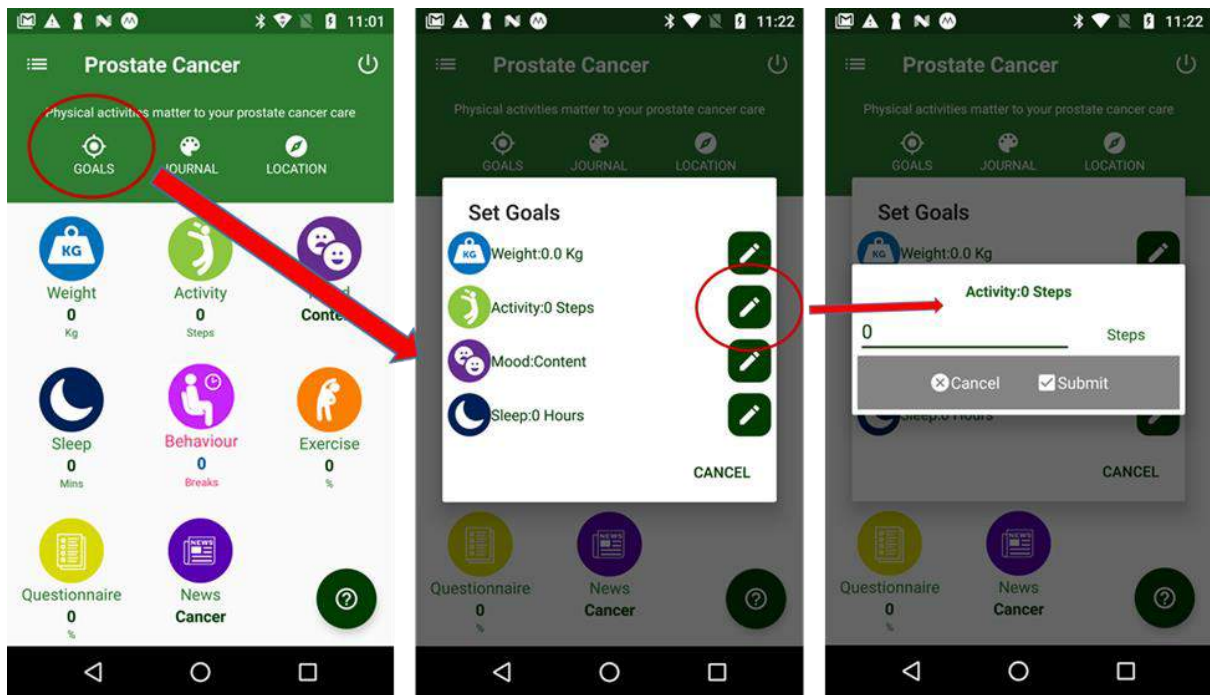


Figure 12: set the Goals

4.2.3. Location

Patient can check the daily location and tracking path by the function. Patient also can annotate the name of a location (e.g., home) and synchronise it with the server. Moreover, the user can turn on/off this function. This function is reference to the requirement in D2.3-[3.4.1. use cases, Motion and Exercise Planning, 3.4.1. use cases, Fatigue management and 3.12.1. use cases, Data Analysis and Data Mining].

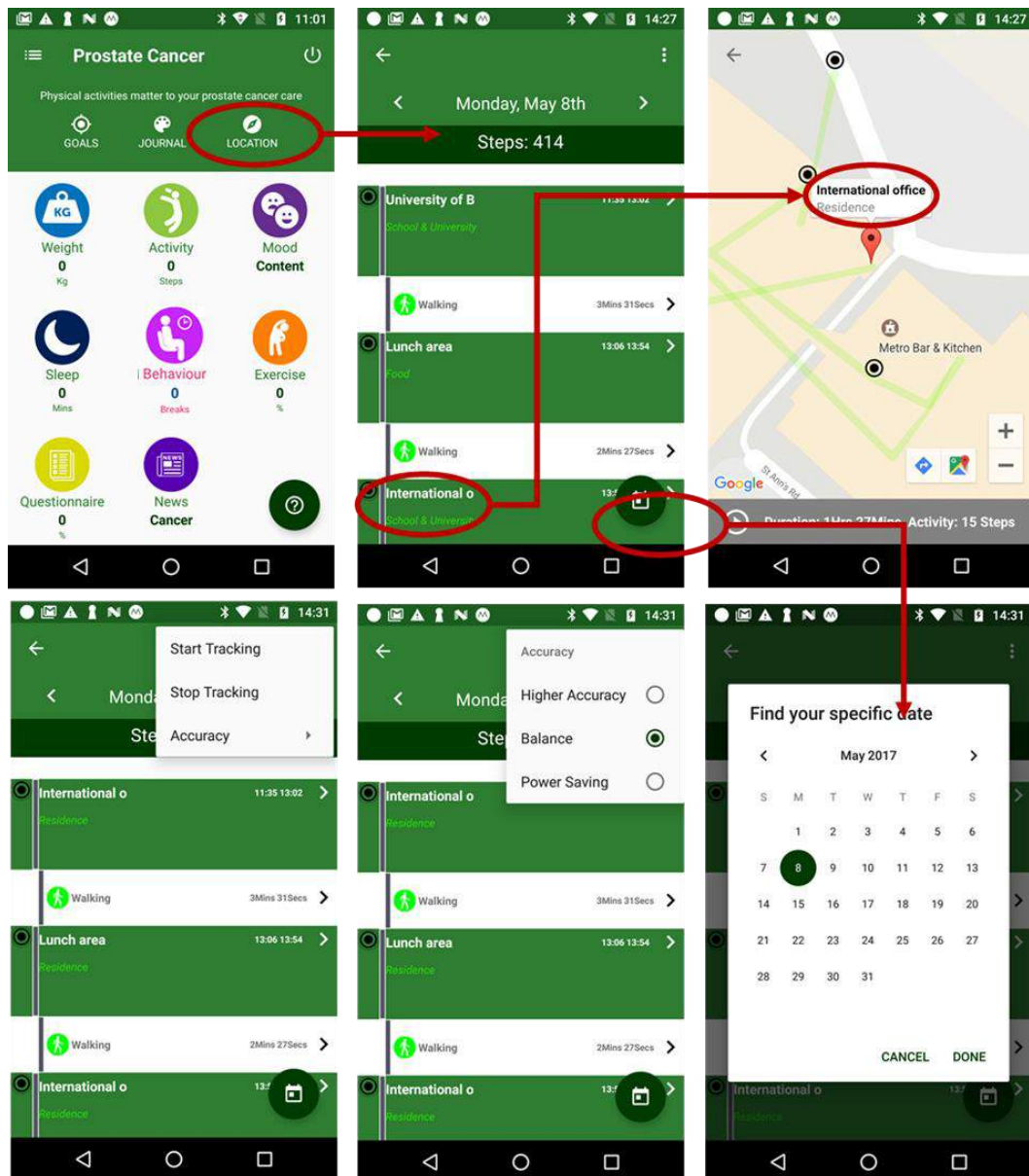


Figure 13: Location Tracking

4.2.4. Journal

A module for patients to communicate with the app and to check the cancer health tips.

4.2.2.1 Control panel

Control panel is easier for patient to manage the their inputs, such as answer for the questionnaire, take/pick phone, type message, voice input and so on. It also can help patient easily access NHS news, Suites function and weight-reduce-program. The output will display the main Journal page. This function is reference to the requirement in D2.3-[3.13.1. use cases, visualisation scenario and 3.5.1. use cases, Personal Health Information Recommender scenario].

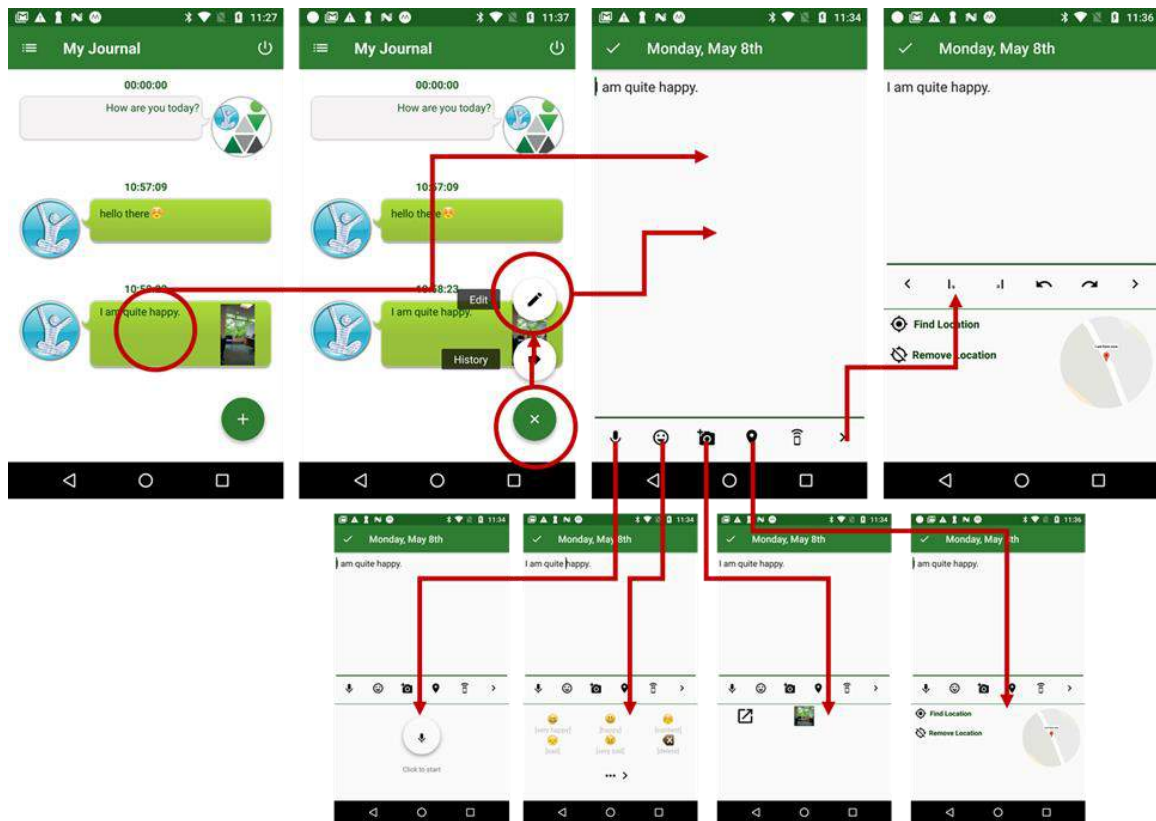


Figure 14: the control panel in Journal

4.2.2.2 Questionnaire

The questionnaire can be accessed quickly and easily via Journal and the patient can turn on/off the function. Reference to D2.3-[3.2.1. use cases, Health Avatar GUI and 3.10.1. use cases, Health Enquiries].

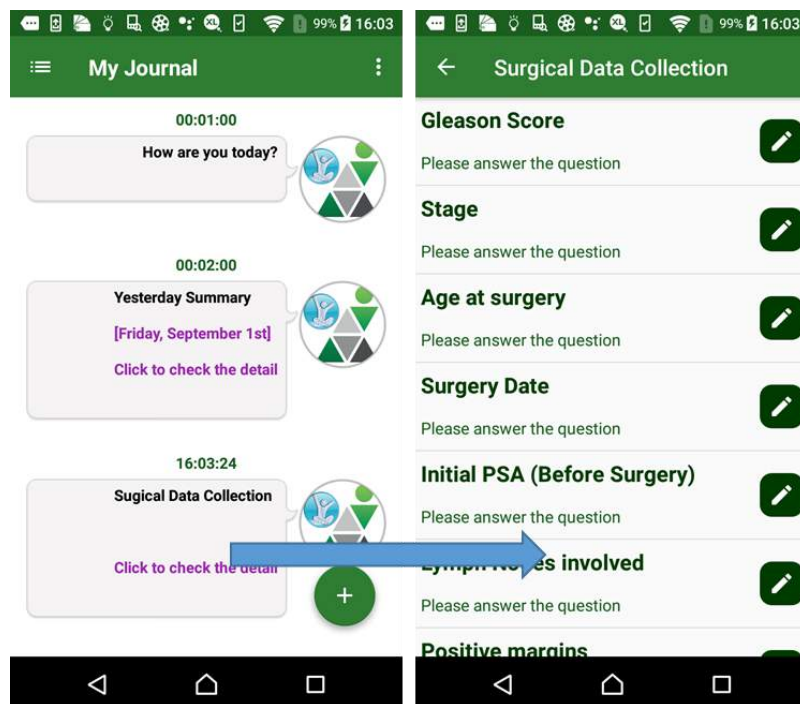


Figure 15: the Questionnaire in Journal

4.2.2.3 Mood Monitor

The users can quickly report their current mood in Journal page. This function is reference to the requirement in D2.3-[3.2.1. use cases, Health Avatar GUI and 3.11.1. use cases, Lifestyle Monitoring]

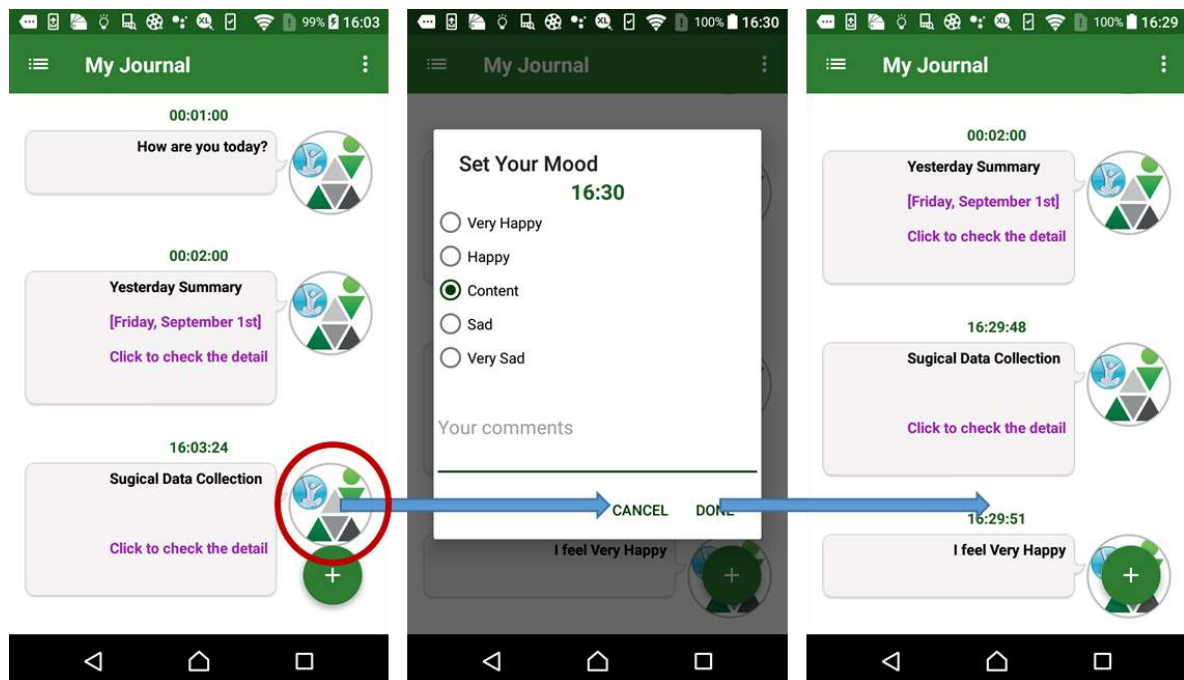


Figure 16: the Mood monitor in Journal

4.2.2.3 Summary

Summary is used to display the daily summary of the measurement value comparing to the goals. The patient can set the summary appears time. This function is reference to the requirement in D2.3-[3.13.1. use cases, visualisation scenario].

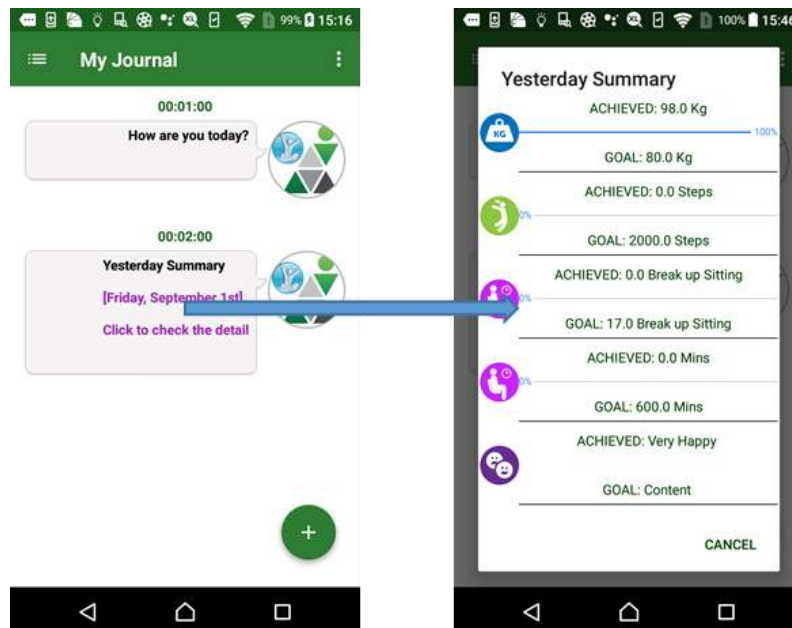


Figure 17: the Summary in Journal

4.2.5. Calendar

A module for patients to review the daily history information of a specific day. Reference to D2.3-[3.13.1. visualisation scenario].

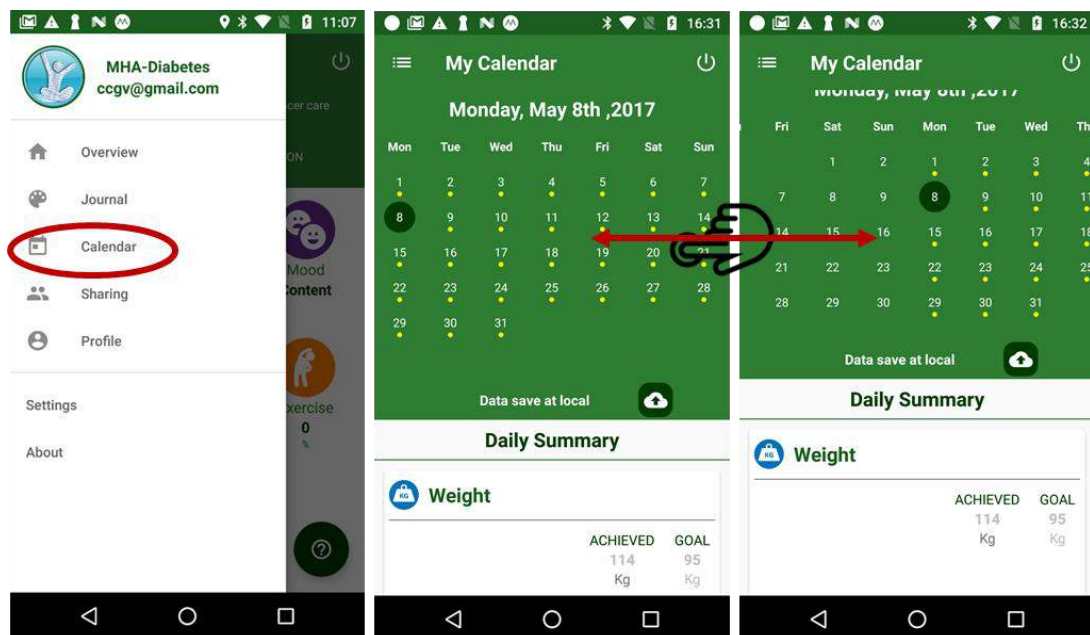


Figure 18: the Summary in Calendar.

4.2.6. Profile

The user profile allows the user to enter personal information in different categories. The profile covers common biochemistries information such as weight and height, as well as common

information such as name and age. The profile view allows the users to add and upload to the iMC platform important personal information and a detailed medical history.

4.2.6.1 Update the Profile Image

A function for patients to update the profile image. The image can be obtained by the device camera or from the image gallery. The new image will be synchronised with the server.

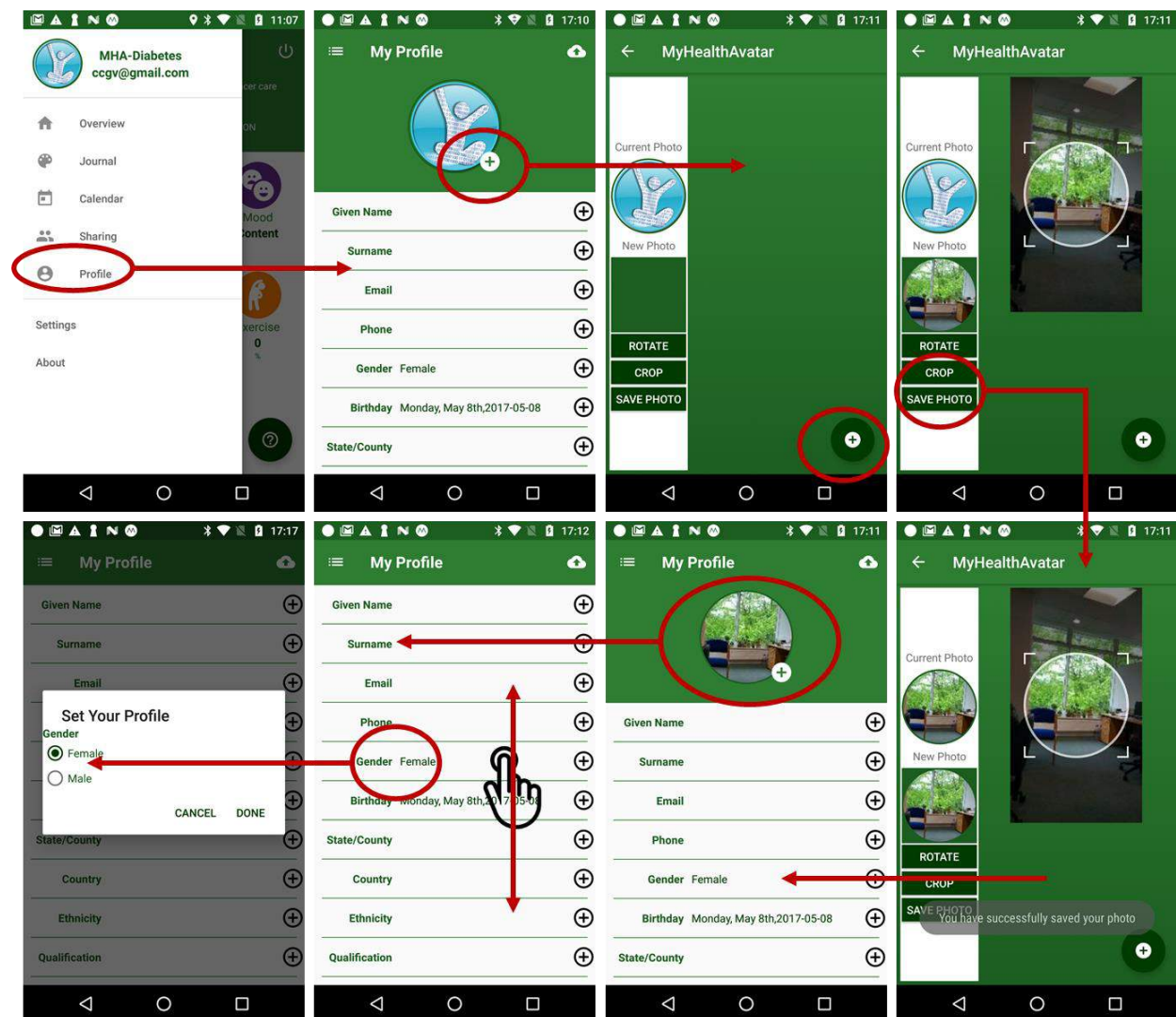


Figure 19: the profile image in Profile

4.2.4.1 Profile

A function for patients to record their general, health and medication profiles. The profile information will be synchronise to the server. This function is reference to the requirement in D2.3-[3.8.1. use cases, Psycho-Emotional Status and Management and 3.9.1. use cases, Family Resilience Evaluation].



Figure 20: My Profile

4.2.7. Sharing

The social page provides users with the opportunity to search for friends within the platform and send them a friend request. This kind of social interaction is intended to help motivate and encourage users to participate with the platform. The social page allows users to connect with other platform users to form social groups and allow group activities, such as the game. This function is reference to the requirement in D2.3-[3.14.1. use cases, Patient - Doctor and Patient - Patient Interaction].

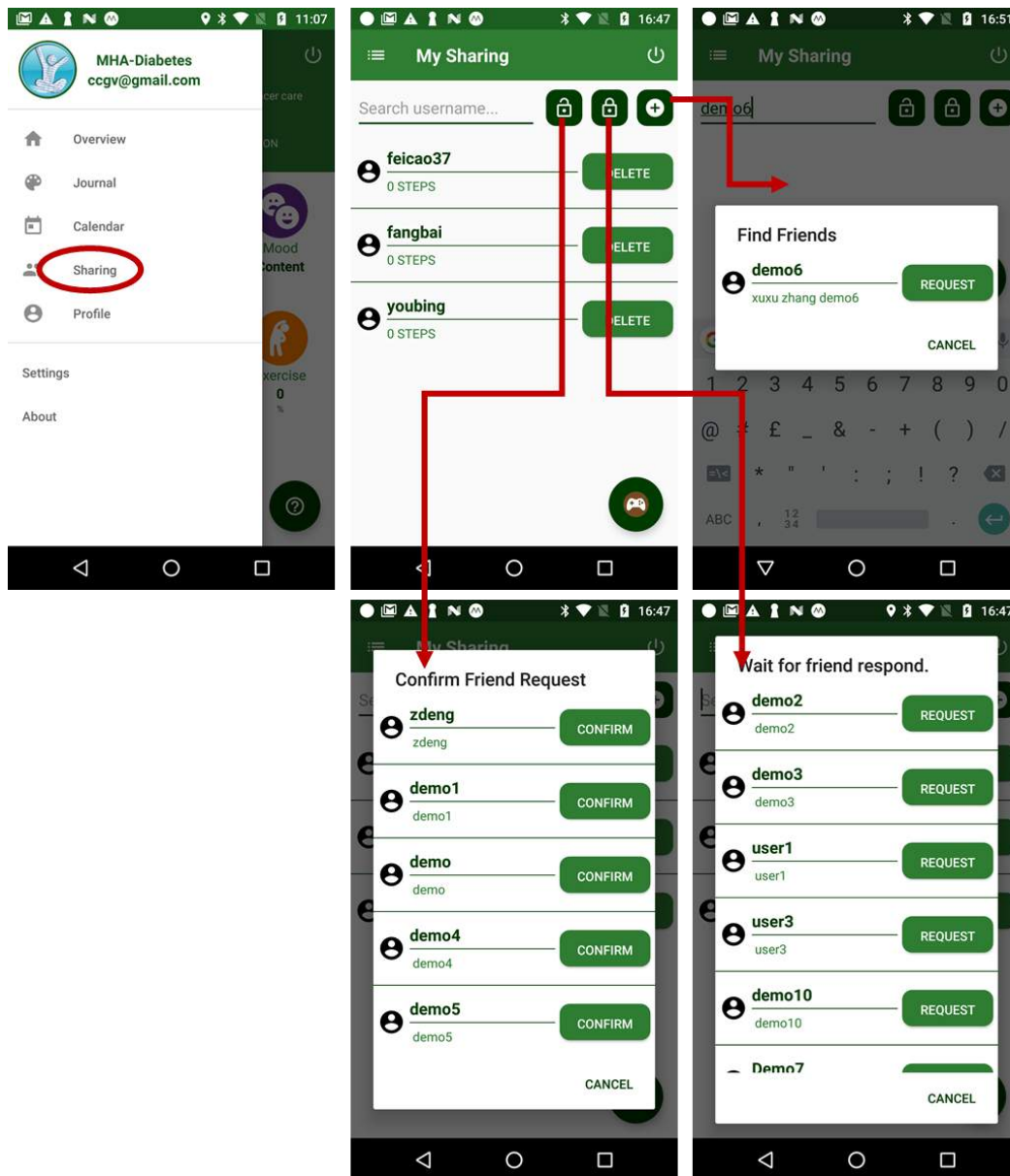


Figure 21: Sharing of activity information with friends

4.3. Technical implementation

4.3.1. System concept and internal architecture

In order to provide the function of access, collection and sharing of long term and consistent personal health status data, the MyHealthAvatar for iManageCancer App system has design as:

There are four data sources, such as MHA, Moves, Fitbit and Withings. Only MHA collects data through the MHA App, the other data sources are collected from the third party API.

There are six main function groups in MHA4IMC App, such as Overview (Suite), Journal, Calendar, Profile, Sharing and Settings.

Overview includes various function suites, such as weight, behaviour, activity, sleep, exercise, questionnaire and cancer news.

Journal is the gate that communicates between patient and App, it includes the diary, summary, mood monitor, daily journal record and questionnaire.

Calendar includes daily history summary.

Profile includes the user profile image and profile data management.

Sharing includes various functions, such as friend list, add new friend, delete friend, friend profile.

Settings includes various functions, such as active/deactivate function in Journal, Calendar and Suites.

4.3.1.1 Overview of the internal architecture

The architecture is shown in Figure 22. The data of the function suite is obtained from Database, if the data source is not MHA, the data will be download from the third party API to the local Database.

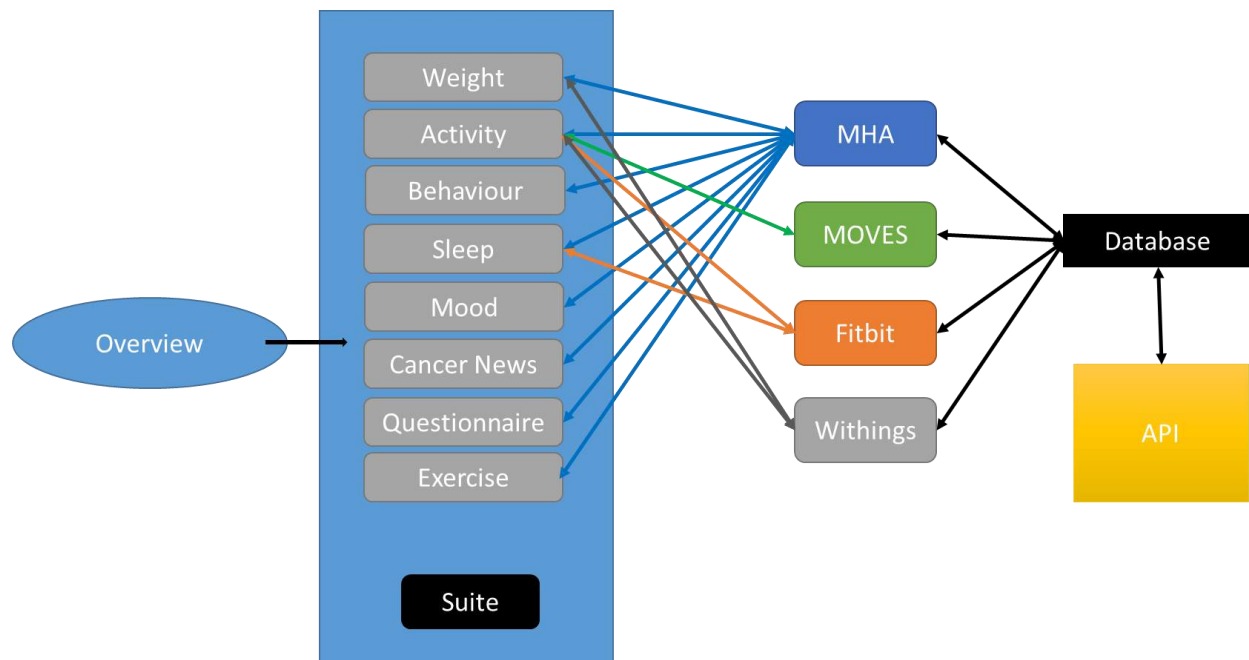


Figure 22: Internal architecture of app MyHealthAvatar for iManageCancer

4.3.1.2 Journal internal architecture

The architecture is shown in Figure 23. The data of summary is obtained from the local Database, if the data source is not MHA, the data will be download from the third party API to local Database. The mood data, diary and answer of questionnaire can be saved in the Database and uploaded to the server via MHA API.

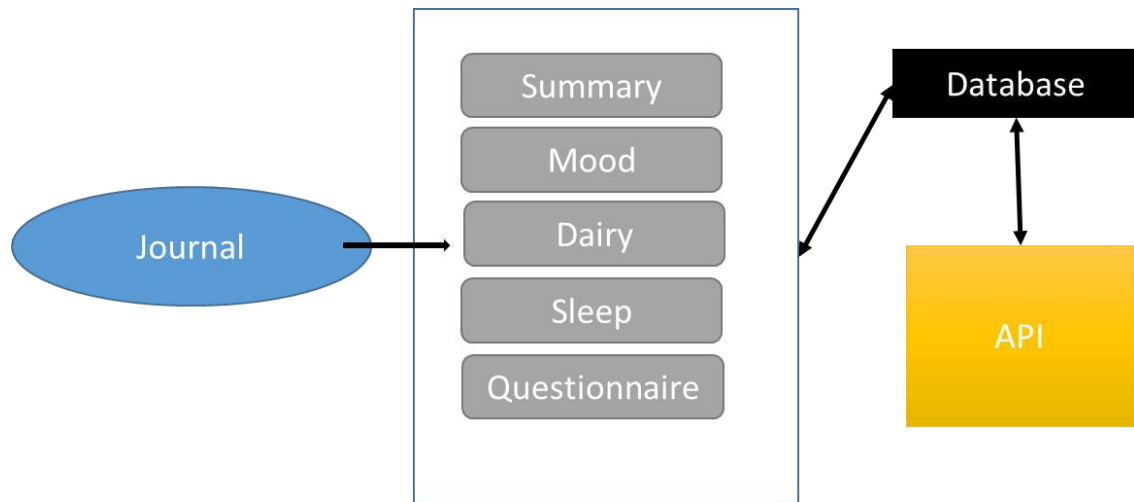


Figure 23: Internal architecture of journal functionality of app MyHealthAvatar for iManageCancer

4.3.1.3 Location architecture

The location function architecture is shown in Figure 24. The data of daily location is obtained from the GPS data processing. It includes GPS data collection, location data mining and segment generation. The Location will be saved in Database and upload to the server via API.

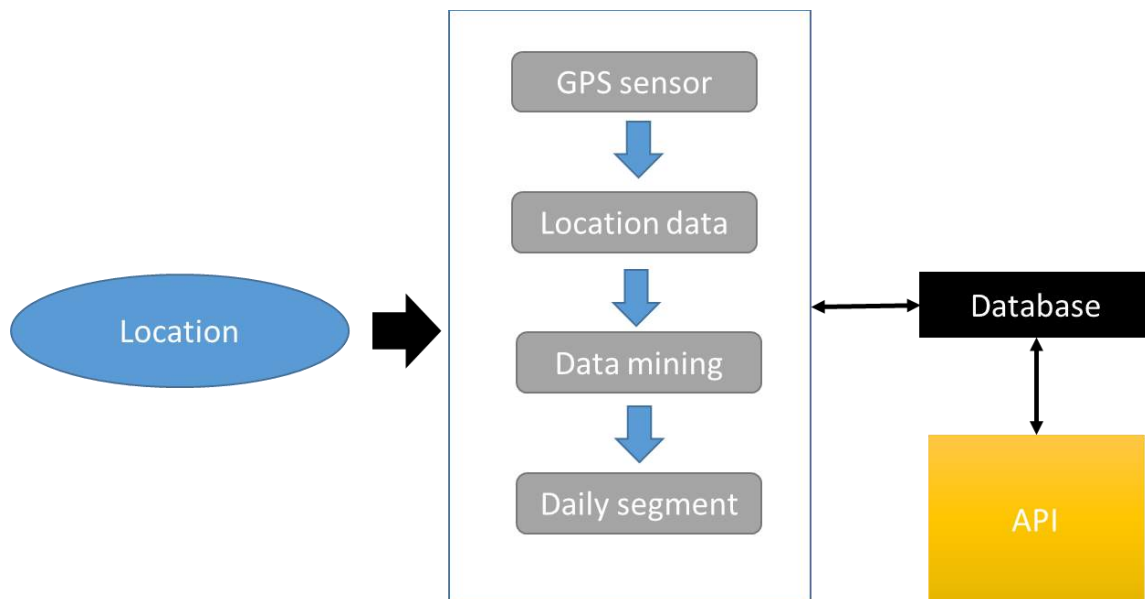


Figure 24: Internal architecture of location functionality of app MyHealthAvatar for iManageCancer

4.3.1.4 Calendar architecture

The architecture is shown in Figure 24. The data of daily summary is obtained from Databas, if the data source is not MHA, the data will be download from third party API to local Database.

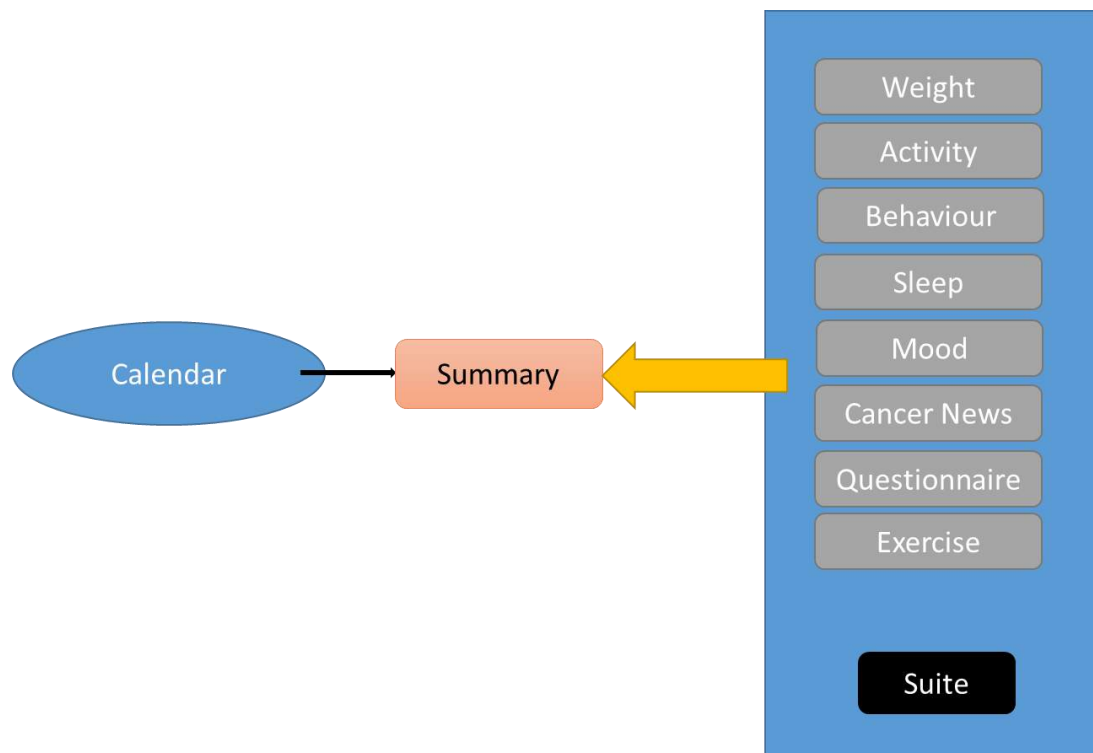


Figure 25: Internal architecture of calendar function of app MyHealthAvatar for iManageCancer

4.3.1.5 Profile internal architecture

The architecture is shown in Figure 26. The profile data is updated by MHA or other data source, such as Fitbit and Withings. The data is saved in the local Database and uploaded to the server via MHA API.

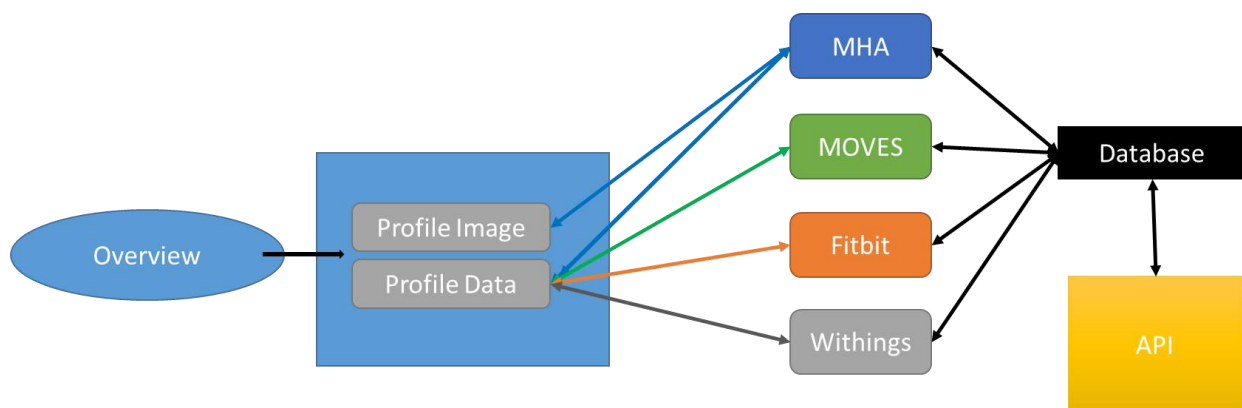


Figure 26: Internal architecture of Profile functionality of app MyHealthAvatar for iManageCance

4.3.1.6 Sharing internal architecture

The architecture is shown in Figure 27.

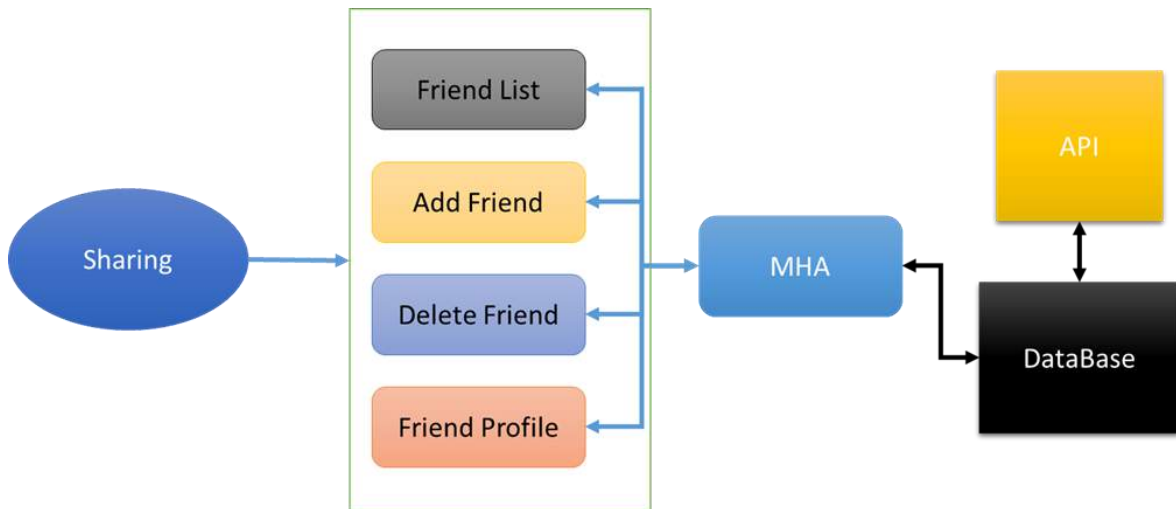


Figure 27: Internal architecture of Sharing functionality of App MyHealthAvatar for iManageCancer

4.3.1.7 Setting internal architecture

The architecture is shown in Figure 28. The Settings controls the functionality of other function groups.

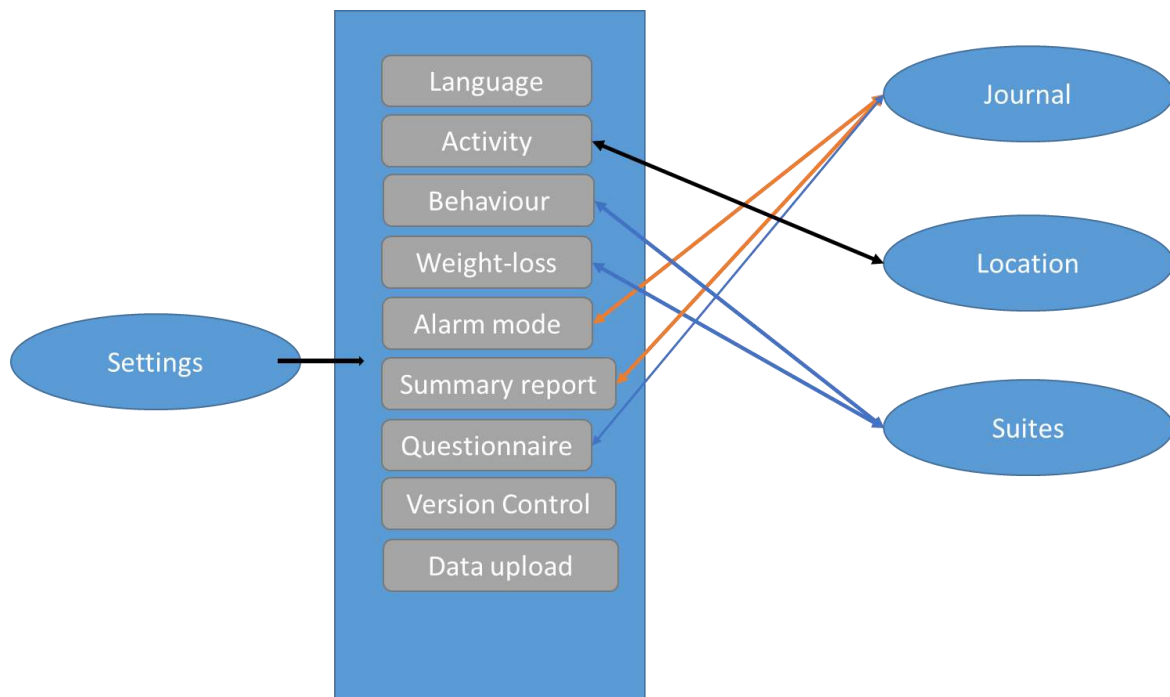


Figure 28: Internal architecture of Setting functionality of app MyHealthAvatar for iManageCancer

4.3.1.8 Main Internal Architecture

The architecture is shown in Figure 29.

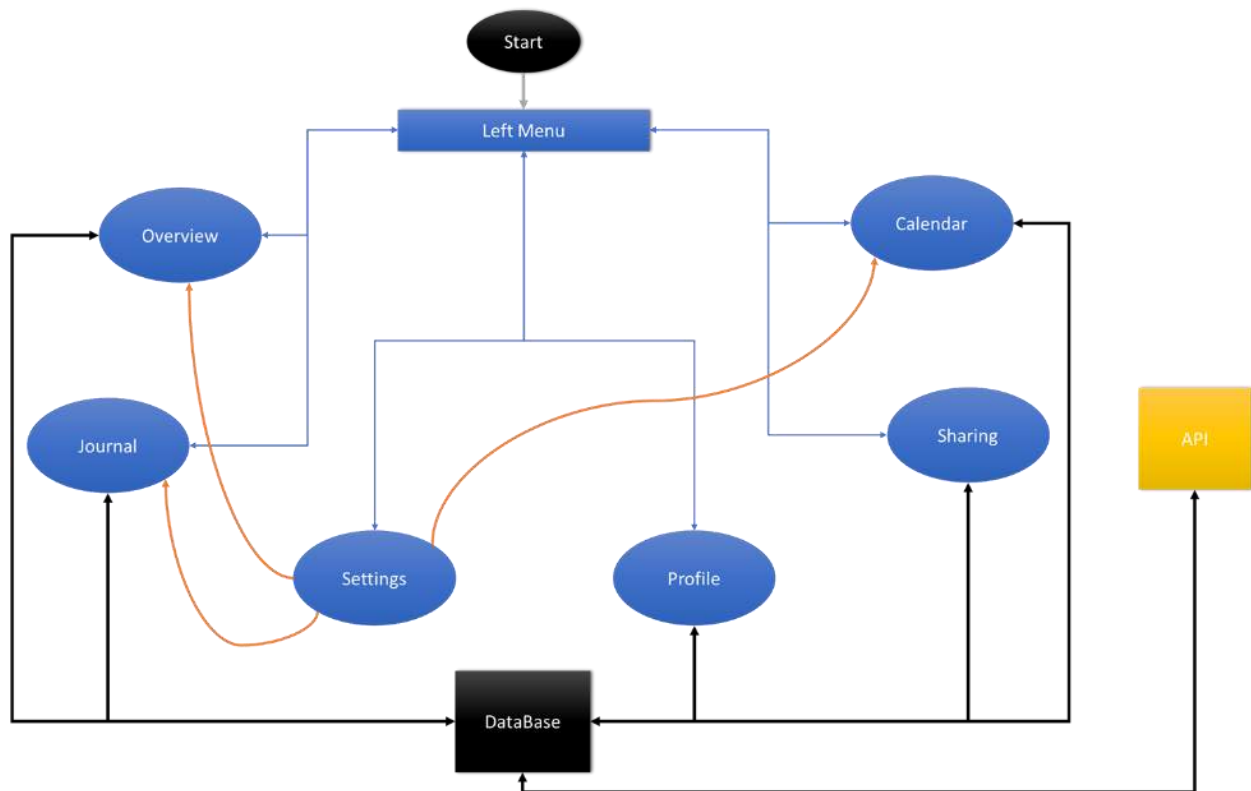


Figure 29: Internal architecture of app MyHealthAvatar for iManageCancer

4.3.2. Interfaces

4.3.2.1 API for Profile

- Get measurement profile——download the measurement profile data from server.
Link: <https://myhealthavatar.org/mha/api/v2/user/measurements>
Reference to 4.3.1.5 Profile.
- Get personal profile——download the personal profile data from server.
Link: https://myhealthavatar.org/mha/api/v2/user/personal_information
Reference to 4.3.1.5 Profile.
- Get/Post/Delete full profile——download/upload the data in general, health and medication profiles from/to server.
Link: <https://myhealthavatar.org/mha/api/v3/profile/full>
Reference to 4.3.1.5 Profile.
- Post profile image——upload the profile image to server.
Link: <https://myhealthavatar.org/mha/api/file/upload>
Reference to 4.3.1.5 Profile.

4.3.2.2 API for Journal

- Get/Post daily journal message——download/upload the message data from/to the server

Link: <https://myhealthavatar.org/mha/api/v3/moments>

Reference to 4.3.1.2 Journal.

- Get/Post journal Image—— download/upload the image data from/to server.

Link: <https://myhealthavatar.org/mha/api/v3/moments/image>

Reference to 4.3.1.2 Journal.

4.3.2.3 API for Activity

- Get Summary ——download the activity summary from the server.

Link: [https://myhealthavatar.org/mha/api/v2/user/activities/summary?from=\[start date\]&to=\[end date\]](https://myhealthavatar.org/mha/api/v2/user/activities/summary?from=[start date]&to=[end date]);

Reference to 4.3.1.1 Suite and 4.3.1.4 Calendar.

- Get Withings data (weight)——download the weight data (data source: Withings) from the server.

Link: [https://myhealthavatar.org/mha/api/app/measure/\[date\]](https://myhealthavatar.org/mha/api/app/measure/[date])

Reference to 4.3.1.1 Suite, 4.3.1.4 Calendar and 4.3.1.5 Profile.

- Get Fitbit data (heart rate)——download the heart rate data (data source: Fitbit) from the server.

Link: [https://myhealthavatar.org/mha/api/app/heart/\[date\]](https://myhealthavatar.org/mha/api/app/heart/[date])

Reference to 4.3.1.1 Suite, 4.3.1.4 Calendar and 4.3.1.5 Profile.

- Get fitbit data (sleep)——download the sleep data (data source: Fitbit) from the server.

Link: [https://myhealthavatar.org/mha/api/app/sleep/\[date\]](https://myhealthavatar.org/mha/api/app/sleep/[date])

Reference to 4.3.1.1 Suite.

- Get Activity level——download the activity level from the server.

Link: https://myhealthavatar.org/mha/api/v2/user/active_state

Reference to 4.3.1.1 Suite and 4.3.1.2 Journal.

- Get activity optimal values——download the activity optimal values from the server.

Link: [https://myhealthavatar.org/mha/api/v2/user/activities/segments?from=\[start date\]&to=\[end date\]](https://myhealthavatar.org/mha/api/v2/user/activities/segments?from=[start date]&to=[end date])

Reference to 4.3.1.1 Suite.

4.3.2.4 API for Location

- Get/Post name and category of the place——download the place name and category/ upload the changes of the place.

Link: https://myhealthavatar.org/mha/api/v2/user/my_places"

Reference to 4.3.1.2 Journal and 4.3.1.3 Location

- Get foursquare place list——download the foursquare place data from the foursquare data server.

Link: https://api.foursquare.com/v2/venues/search?client_id=%s&client_secret=%s&v=20130815&ll=%s&radius=100

Reference to 4.3.1.2 Journal and 4.3.1.3 Location

- Get Location Segment——download the location segments (data source: Moves) from the server.

Link: [https://myhealthavatar.org/mha/api/v2/user/activities/segments?date=\[date\]](https://myhealthavatar.org/mha/api/v2/user/activities/segments?date=[date]);

Link: [https://myhealthavatar.org/mha/api/v2/user/activities/segment?from=\[start date\]&to=\[end date\]](https://myhealthavatar.org/mha/api/v2/user/activities/segment?from=[start date]&to=[end date]);

Reference to 4.3.1.2 Journal and 4.3.1.3 Location

4.3.2.7 API for Goals

Get/post goals

Link: <https://myhealthavatar.org/mha/api/v2/user/goals>

Reference to 4.3.1.1 Suite.

4.3.2.8 API for Sharing

- Get user friend list

Link: <https://myhealthavatar.org/mha/api/friend/list>

Reference to 4.3.1.6 Sharing.

- Get user friend info

Link: [https://myhealthavatar.org/mha/api/friend/info?username=\[username\]](https://myhealthavatar.org/mha/api/friend/info?username=[username])

Reference to 4.3.1.6 Sharing.

- Get add friend request

Link: <https://myhealthavatar.org/mha/api/friend/request>

Reference to 4.3.1.6 Sharing.

- Approve users as friend

Link: [https://myhealthavatar.org/mha/api/friend/approve?username=\[username\]](https://myhealthavatar.org/mha/api/friend/approve?username=[username])

Reference to 4.3.1.6 Sharing.

- Add/remove user as friend

Link: [https://myhealthavatar.org/mha/api/friend/add?username=\[username\]](https://myhealthavatar.org/mha/api/friend/add?username=[username])

Link: [https://myhealthavatar.org/mha/api/friend/delete?username=\[username\]](https://myhealthavatar.org/mha/api/friend/delete?username=[username])

Reference to 4.3.1.6 Sharing.

4.3.2.9 API for News

- Get NHS news

Link: <https://myhealthavatar.org/nhs/news>

Reference to 4.3.1.2 Journal.

- Get cancer news

Link: <https://myhealthavatar.org/imc/news>

Reference to 4.3.1.1 Suite.

4.3.2.10 API for evaluation and audit

Post IMC Evaluation Questionnaire

Link: <http://live.ccg.gov.uk/mha/app/questionnaire>

Post IMC Audit

Link: <https://www.iphr.care/mha/api/iphr/audit>

4.3.2.11 API for upload

Post IMC data to the server

Link: <https://www.iphr.care/mha/api/app/data/update>

4.3.2.12 API for version control

- Get IMC version

Link: https://www.iphr.care/mha/api/app/update/get?name=mha_imc

- Get IMC new version App

Link: <https://www.iphr.care/mha/api/app/update/download/>

4.3.3. Database

4.3.3.1 Location Table

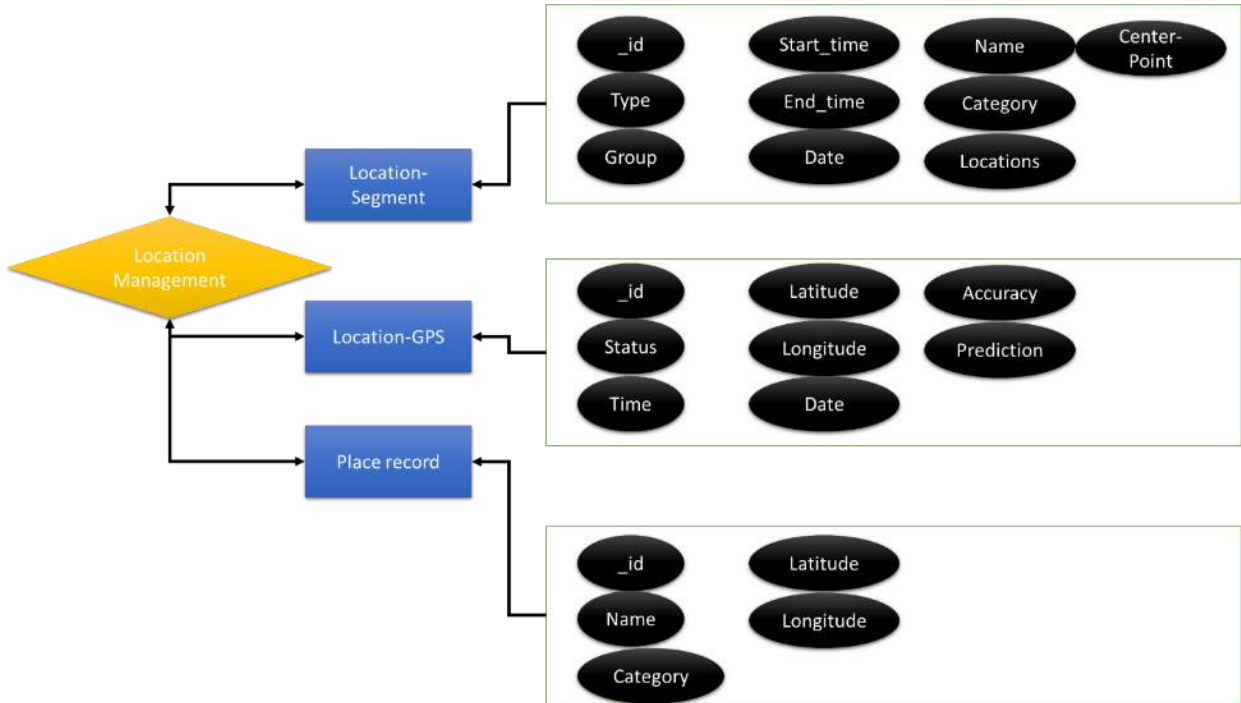


Figure 30: Location table

4.3.3.2 Sharing Table

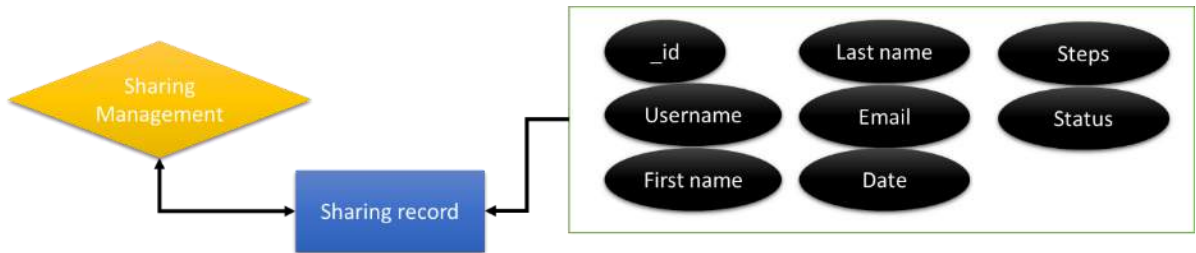


Figure 31: Sharing table

4.3.3.3 Journal Table

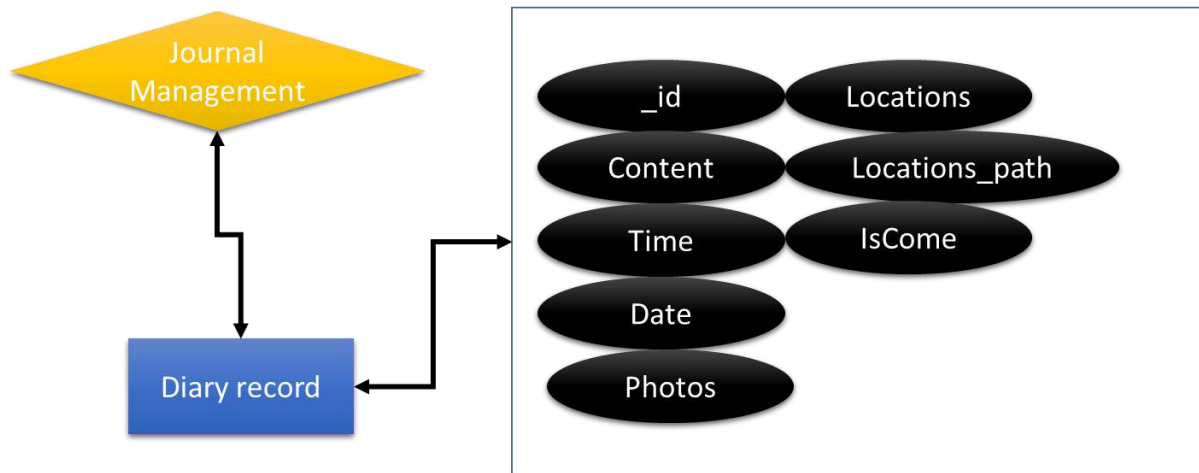


Figure 32: Journal table

4.3.3.4 Questionnaire Table

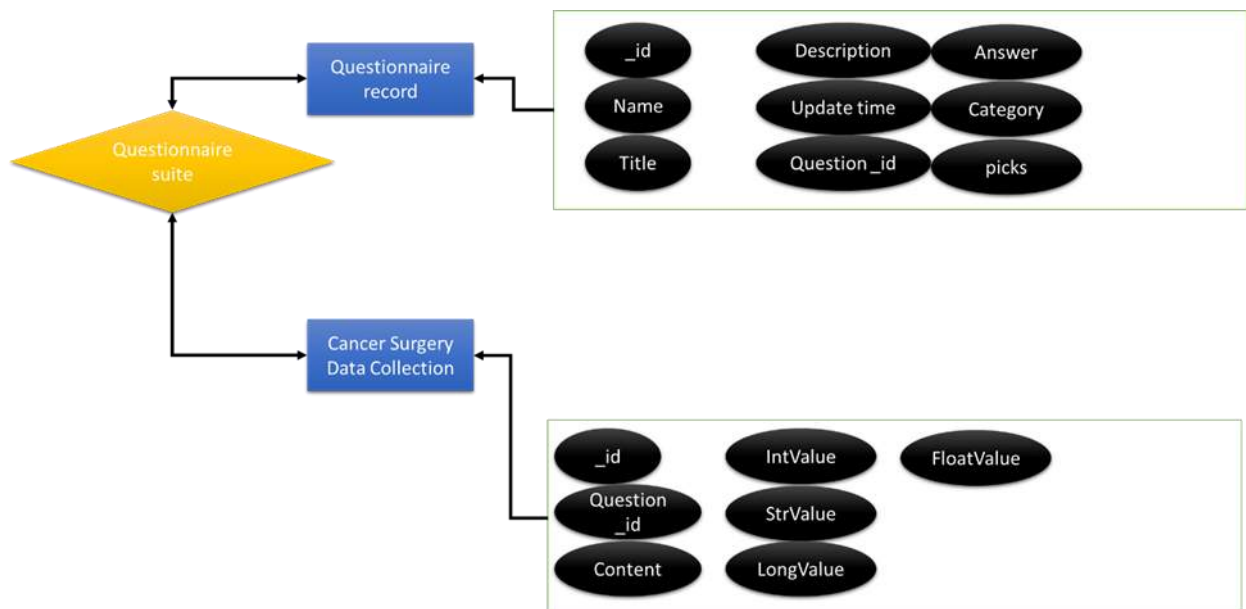


Figure 33: Questionnaire table

4.3.3.5 Prostate Cancer Exercise Table

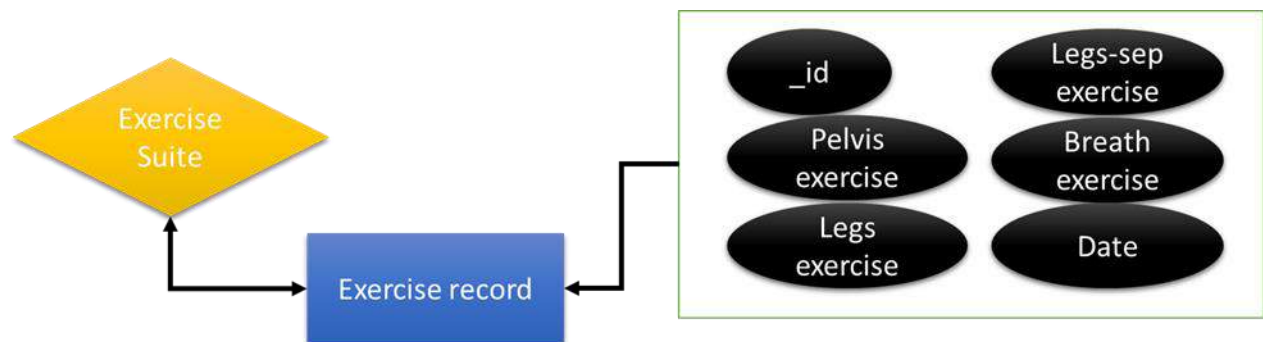


Figure 34: Prostate Cancer Exercise table

4.3.3.6 Mood Table

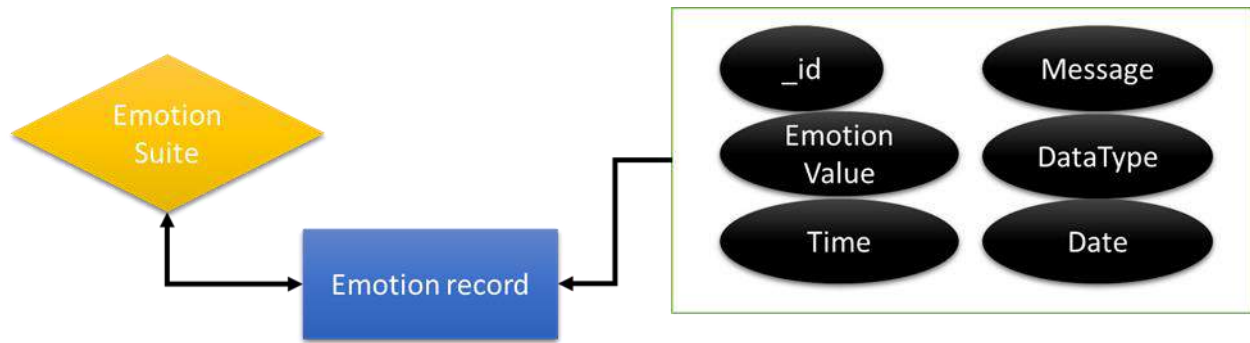


Figure 35: Emotion (Mood) table

4.3.3.7 Weight Table

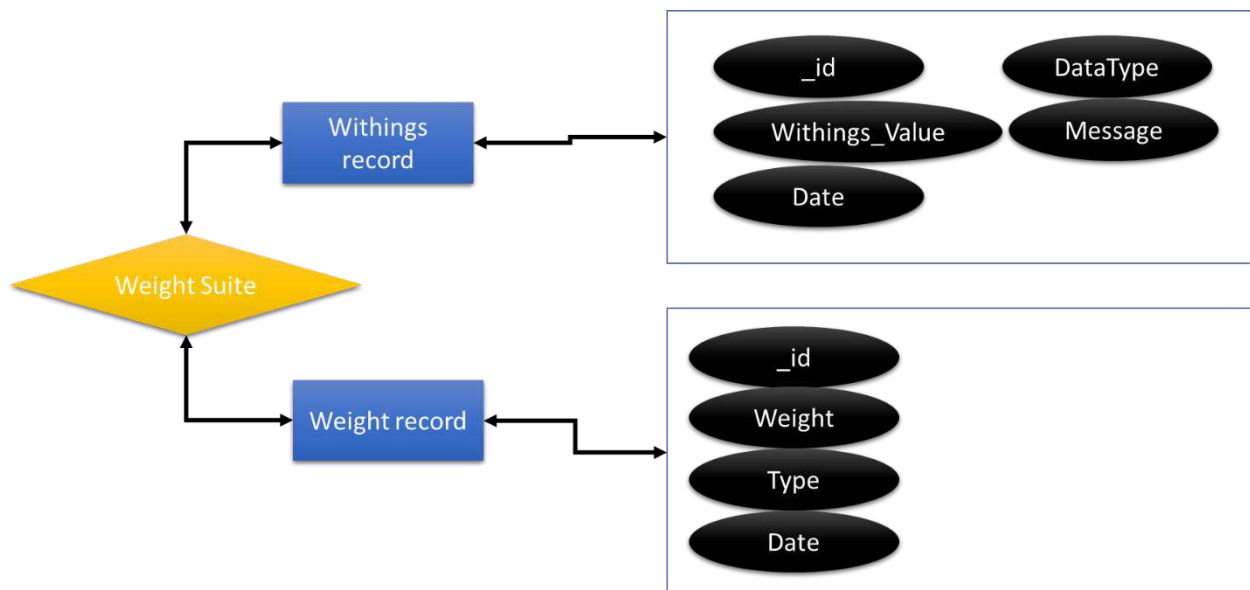


Figure 36: Weight table

4.3.3.9 Activity Table

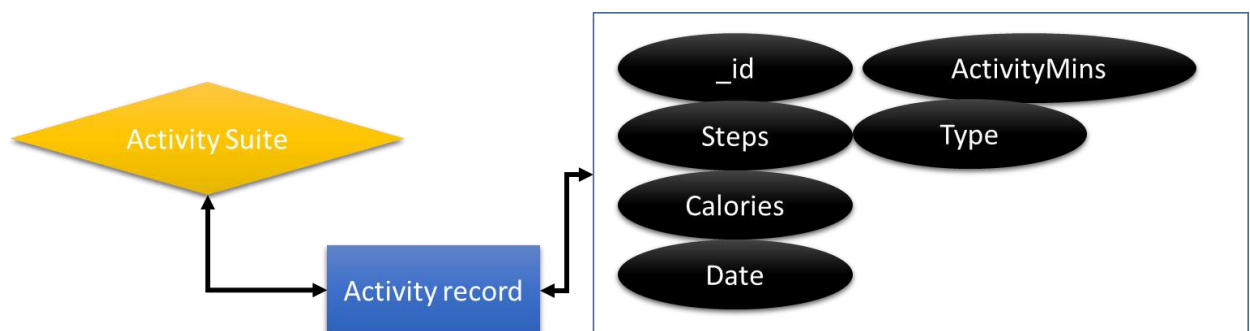


Figure 37: Activity table

4.3.3.10 Sleep Table

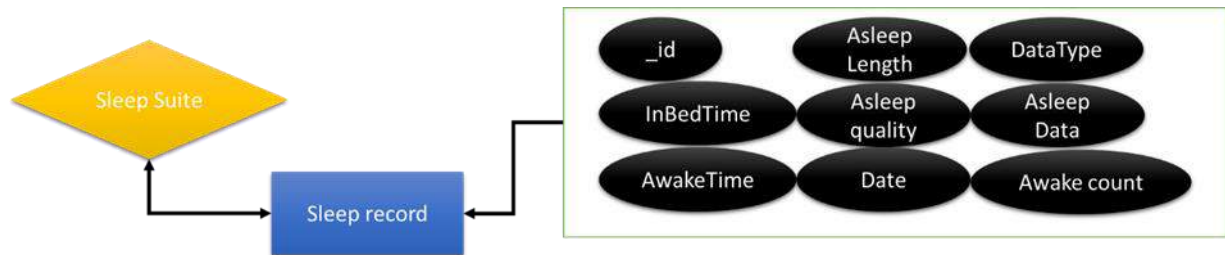


Figure 38: Sleep table

4.3.3.11 Goals Table

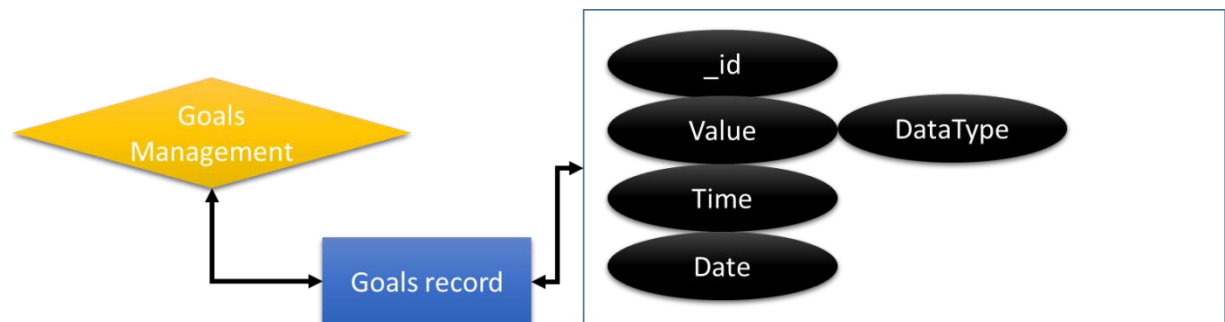


Figure 39: Goals table

4.3.3.12 Weight-loss Table

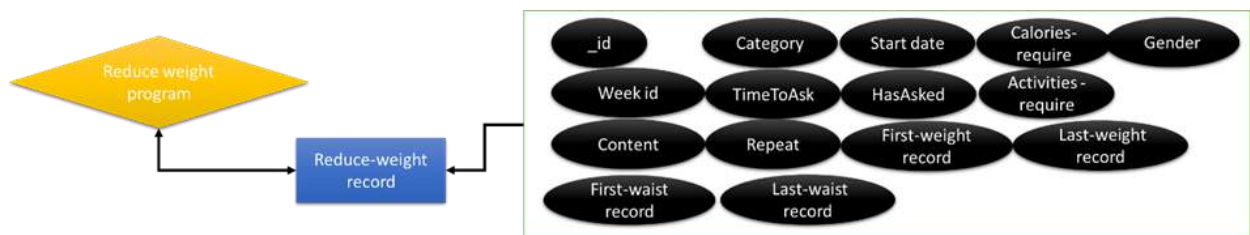


Figure 40: Weight-loss table

4.3.3.13 Evaluation Table

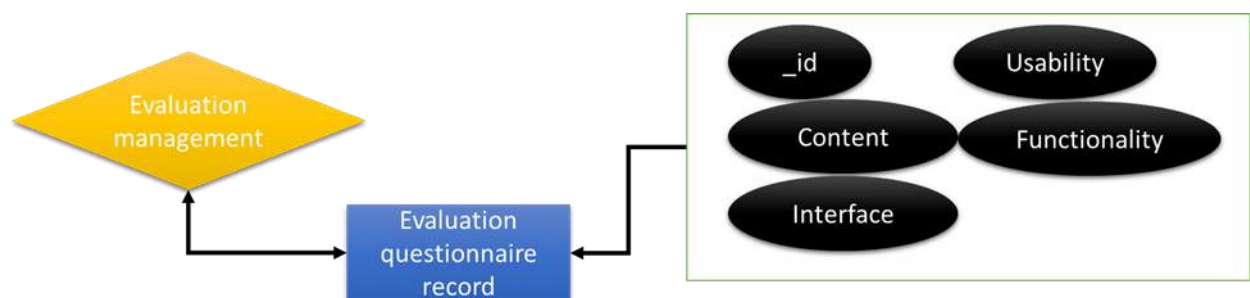


Figure 41: Evaluation table

4.3.4. Data protection

User will need to explicitly consent the terms and conditions, and the privacy statement in order to create an account and access/store data. Users are not required to provide real identifiable information of themselves, and the app facilitate users' with this privacy preserving mechanism.

The data generated within the MyHealthAvatar for iManageCancer is stored in a physically and

technically secure server, provided by a commercial company called Linode (<https://www.linode.com/>). The data generated from the app is stored in databases hosted by Linode, and the services employs industry-standard technical safeguards, such as cryptographic key-based authentication controls on servers which house personal information, browser-based SSL (secure sockets layer) encryption and other storage system access control mechanisms to protect the integrity of internal databases and prevent unauthorized access.

For data security and privacy protect, the application references the ISO/IEC 27002:2013 code of practices for comprehensive information security control and risk management. The system will incorporate controls, which aligned to ISO/IEC 27018:2014 best practices for protection of Personal Identifiable Information (PII). Service Organization Controls (SOC) 2 (security controls) will guide the trust principles of security, availability and processing integrity, which allows the system processing to be accurate, complete, fast and authorized.

Only user will have full access to the data generated by himself/herself within the app, and doctors only have access to patient data with patient's explicit informed consent. Furthermore, patients may choose to enter a game within the app where they compete with other users in a physical activity (step count) challenge. Patients can add other users to their game by searching for other users' names within the app and then send a request to compete in the activity competition with them. The peer user would then need to accept this request before any activity data is shared, but no personal information other than the number of steps accumulated will be visible to other users.

All data generated by the app belongs to the user, and if the user no longer wants to use the app, they can close their account and request that all of their data generated within the app is destroyed.

For authentication, the Universal Second Factor (U2F) protocol is proposed, which allows the system to augment the security of password infrastructure by adding a strong second factor to user login. Google Authenticator [6] on Android mobile device could serve as the second factor, which is a free application which gains popularity recently.

The integration utilizes JavaScript Object Notation (JSON) based Identity protocol in its security model, the JSON data format carries information with defined ways to encode tokens, symmetric / asymmetric keys and digital signatures. JSON Web Token (JWT) specification defines the way to encode token in this JavaScript format, these lightweight tokens can be used in HTTP headers and query strings. JSON Web Key (JWK) and JSON Web Signature (JWS) specifications define the way on encryption keys and digital signatures.

Federation and Single Sign On (SSO) is guided by OpenID Connect, which is essentially the third version of OpenID (however it is a complete rewrite, and not compatible with previous versions). OpenID connect is an HTTP-based protocol that provides SSO, it is built atop of OAuth 2 and achieve higher Levels of Assurance (LoA) compared to other similar purpose protocols like SAML and WS-Federation.

OAuth 2 is a "meta protocol" which provides a useful foundation for other protocols (e.g. OpenID Connect, NAPS and UMA). OAuth is very important in API system because it features:

- Delegated access to third parties applications
- Reduce password sharing between users and third-parties
- Enable the revocation of access

OAuth 2 is proposed to be used by the system for delegated access to the API.

HTTPS is always preferred in all system and third parties web applications, system maintains a list of allowed third parties, which includes key information:

- Callback URL which used by OAuth 2 and SSO
- Third-parties contact and permissions (controlled by team)

All subsystems have their own security model, and only give the least permission necessary to other subsystems and / or users. The permission is assigned and controlled by specific members of team, and will be audited and reviewed.

4.4. User manual

4.4.1. Starting the MyHealthAvatar-iManageCancer App

Action: Find the launcher logo for the MHA-iMC app in the Apps of your mobile device and press on it.

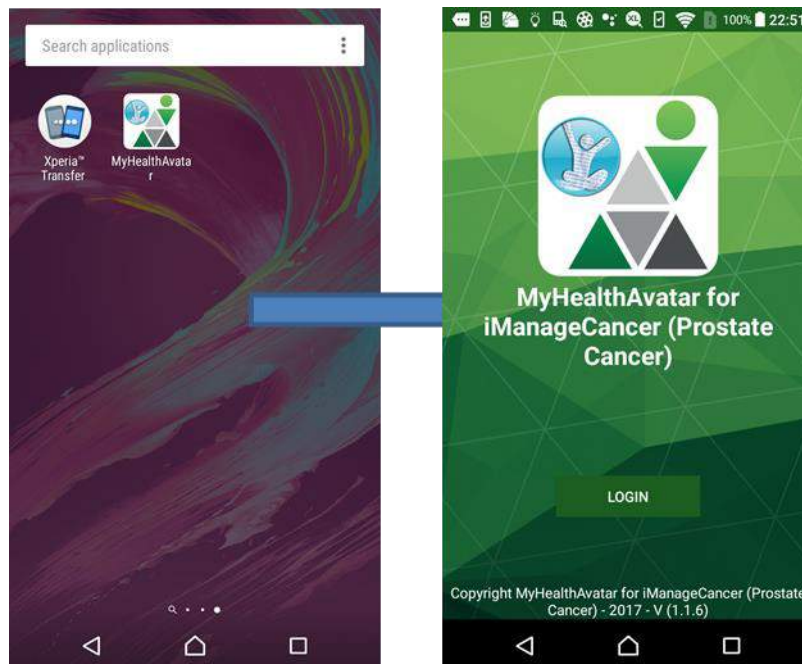


Figure 42: start MyHealthAvatar for iManageCancer

Result: The start page of the app is displayed.

4.4.2. Accessing the Main page from the Start page

Action: Insert your account and password and press on the Sign in button on the start page of the app.

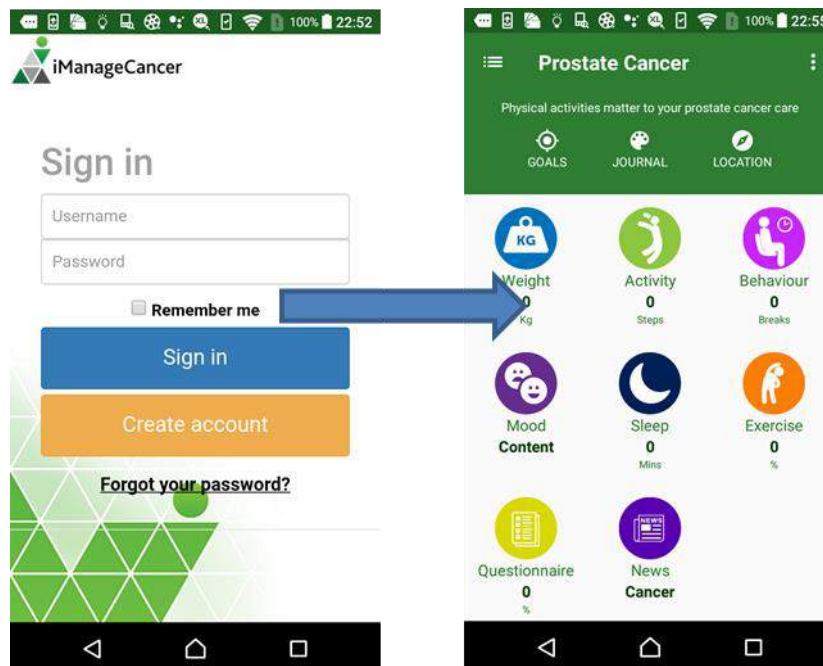


Figure 43: sign in MyHealthAvatar for iManageCancer

Result: After the successful login, the overview in the main page is displayed on the screen.

4.4.3. Accessing the Location page from the Start page

Action 1: Click the “Location” on the main page.

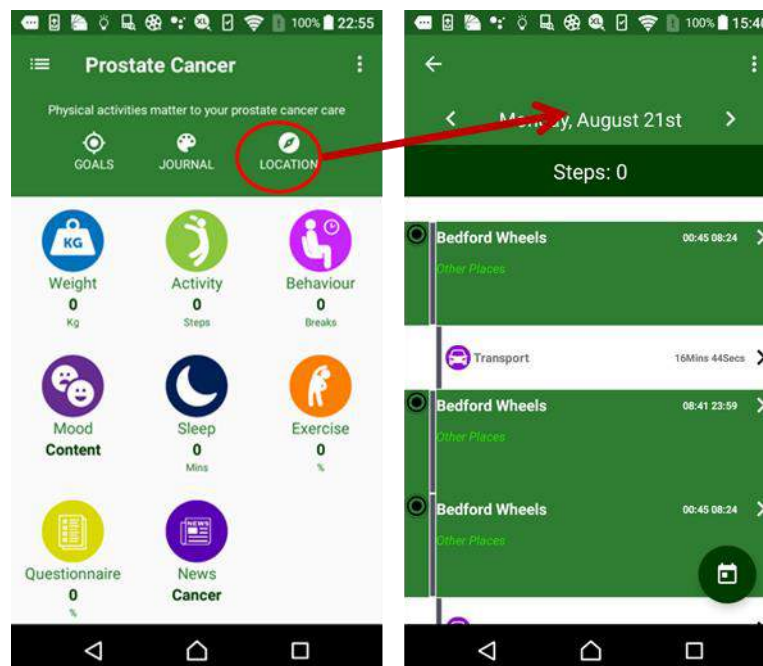


Figure 44: Location page overview

Action 2: Click the segment item and check the location on the map

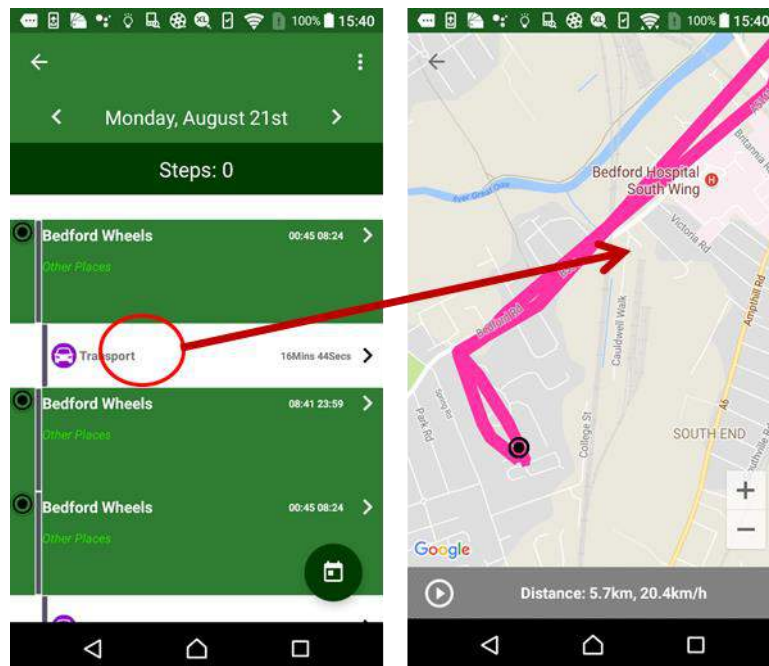


Figure 45: Location information on the map

Action 3: Annotate the segment item

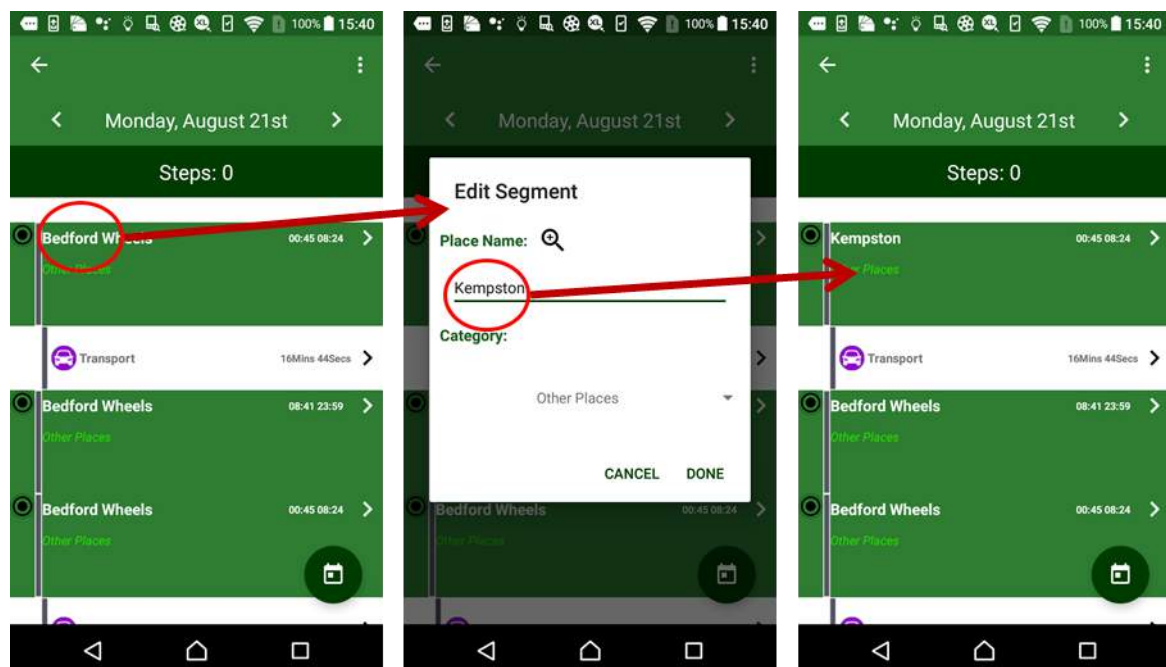


Figure 46: Location annotation

4.4.4. Access Left Menu

Action 1: Press the “menu“ button on the action bar to display left menu or use finger press the left edge of the screen and slide to right to display left menu.

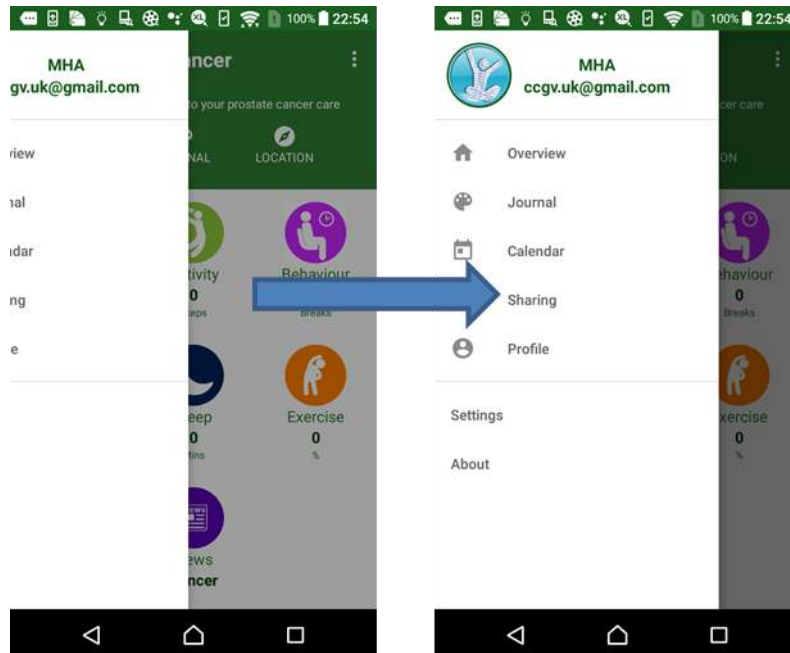


Figure 47: display left menu

Result: Left menu appears on the screen.

4.4.5. Access Settings Page

Before testing the functions, the user needs to turn on (enable) the functions first.

Action 1: Click the Settings in the left menu

4.4.4.1 Select a Language

The Language Mode – Switch the Language between English, German and Italian.

Action 1: Click the edit button on right side.

Action 2: Choose the data source.

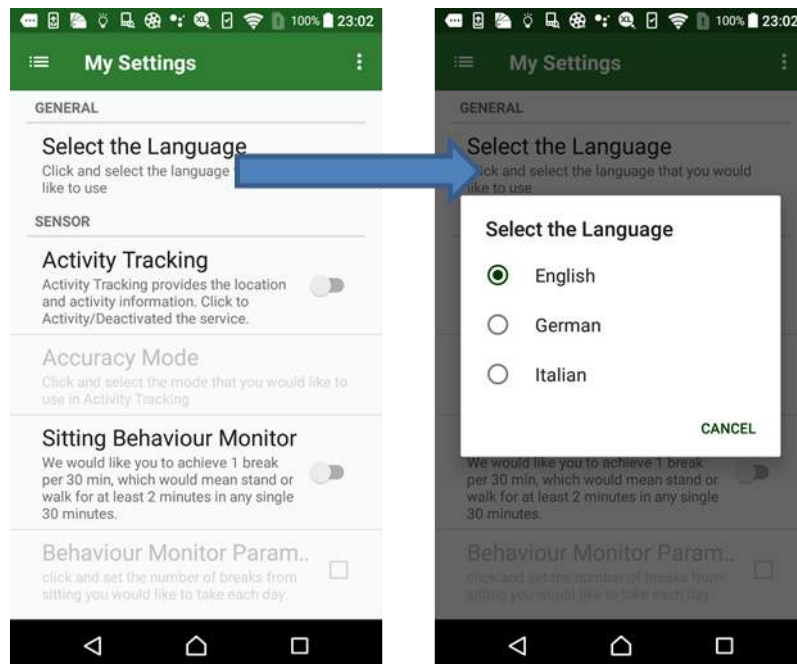


Figure 48: language settings page

Result: App Language is changed.

4.4.4.2 Enable the Location Activity Tracking

Location Sensor monitors the GPS location of the mobile. When user arrives at annotated place, the app will give the message which is related the place in Journal page.

Action 1: Turn on the Switch component.

Action 2: If Android 6.0 or above, allow the location permission

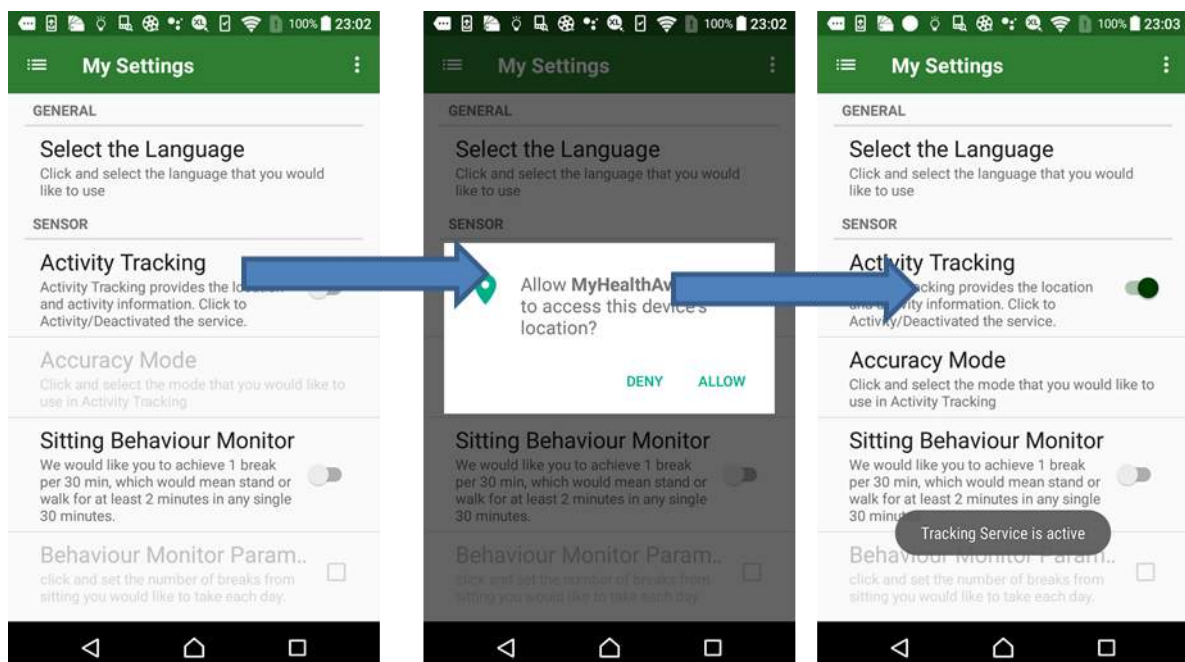


Figure 49: location sensor is activated.

Result: Location Sensor is activated.

4.4.4.3 Enable the Sitting Behaviour Monitor

Sitting Behaviour Monitor - This function monitors the user's break up sitting time per 30 mins.

Action 1: Turn on the Switch component.

Action 2: Once the user has turned on the function, an option will appear. Click the button and edit break up sitting and sitting time goals.

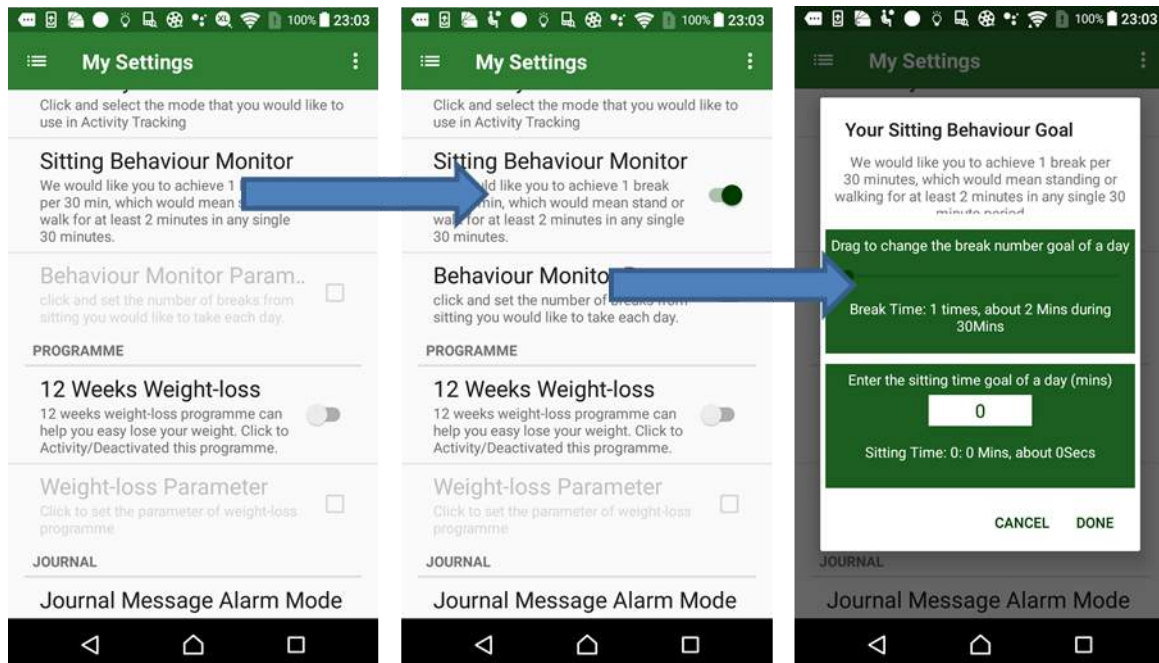


Figure 50: activity sensor is activated

Result: Sitting behavior monitor is activated.

4.4.4.4 Enable the 12 Weeks Weight-loss Program

12 weeks weight-loss program can help user reduce his weight during 12 weeks. This program will update the reduce weight goal week by week. During the program time period, the program will measure user's weight changes then summary user's activities every weekend. Moreover it gives suggestion tips to help user food and exercise.

Action 1: Turn on the Switch component.

Action 2: Choose your gender

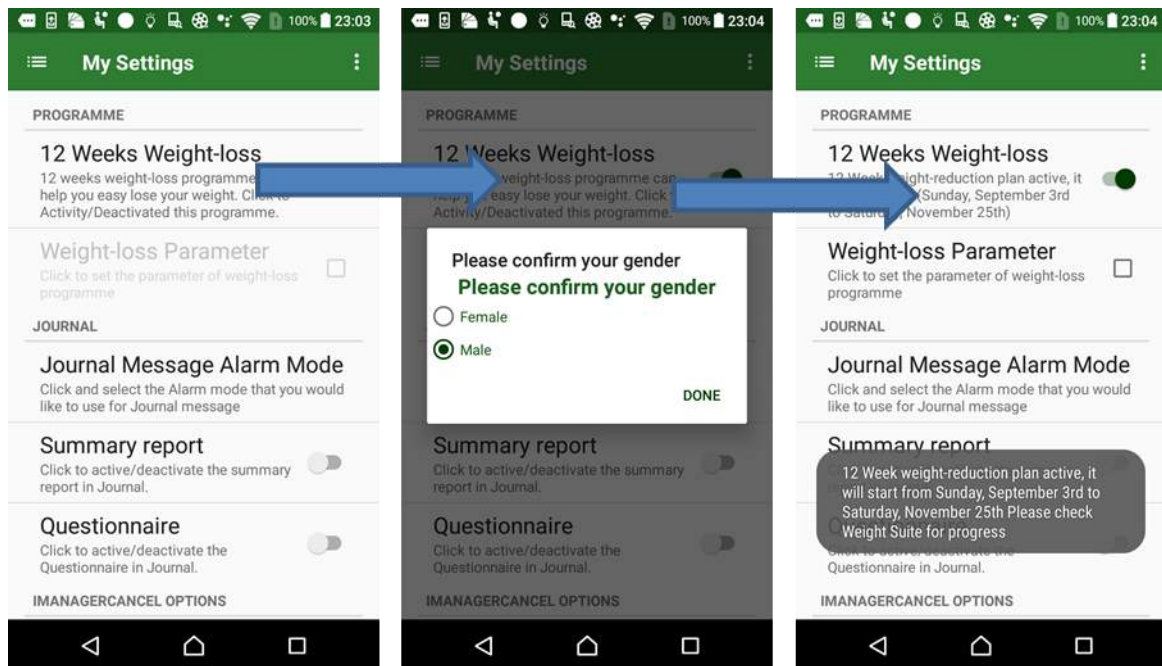


Figure 51: setup the weight-loss program

Action 2: Once the user has turned on the function, an option will appear. Click the button and choose the methods of showing the tips.

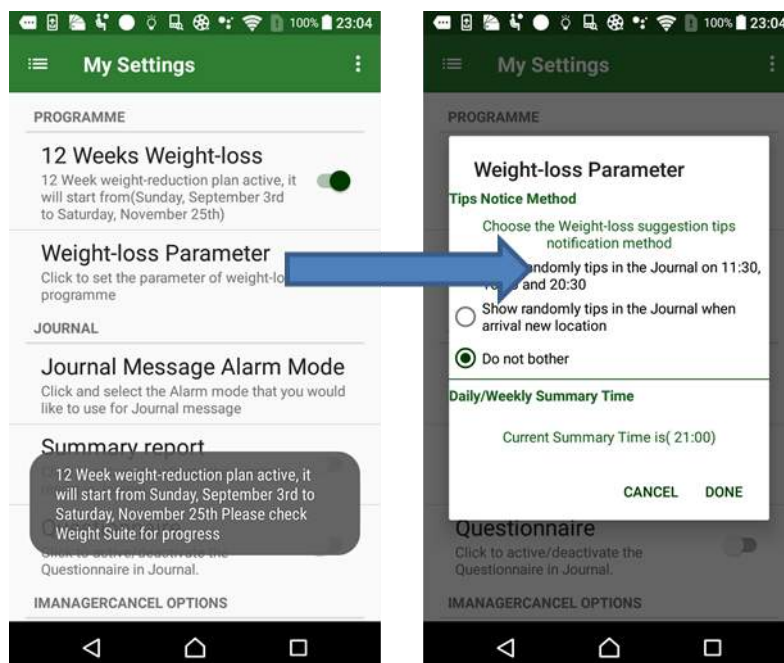


Figure 52: the set the method of showing tips for weight-loss

4.4.4.5 Enable the Journal Alarm

Journal Alarm-User can switch between the silence and alarm mode when the notice display in the Journal page.

Action 1: Turn on the Switch component.

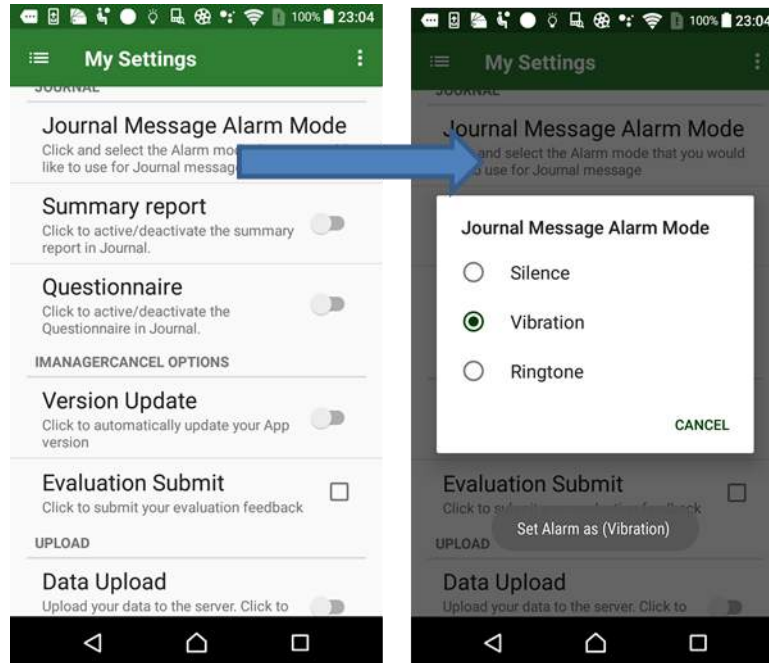


Figure 53: Journal alarm is activated

Result: Journal Alarm is activated.

4.4.4.6 Enable the Summary Report

Summary Report-Enable the summary of goals achievement report in Journal page.

Action 1: Turn on the Switch component.

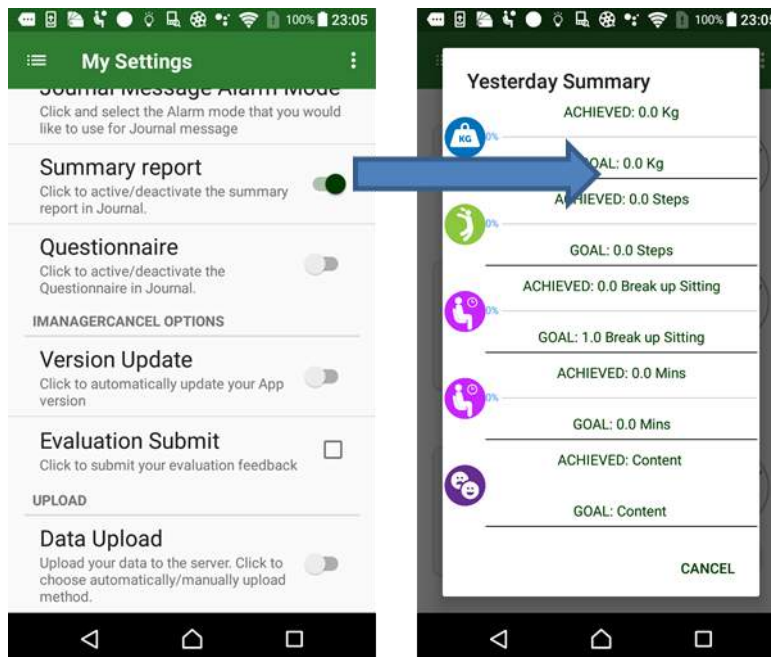


Figure 54: Summary report is activated

Result: Summary Report is activated.

4.4.4.7 Enable the Journal Questionnaire

Journal Questionnaire–Enable the questionnaires in Journal page.

Action 1: Turn on the Switch component.

Action 2: Once the user has turned on the function, the app will ask the questionnaires around at 12:30, 16:30 and 20:30 in Journal page.

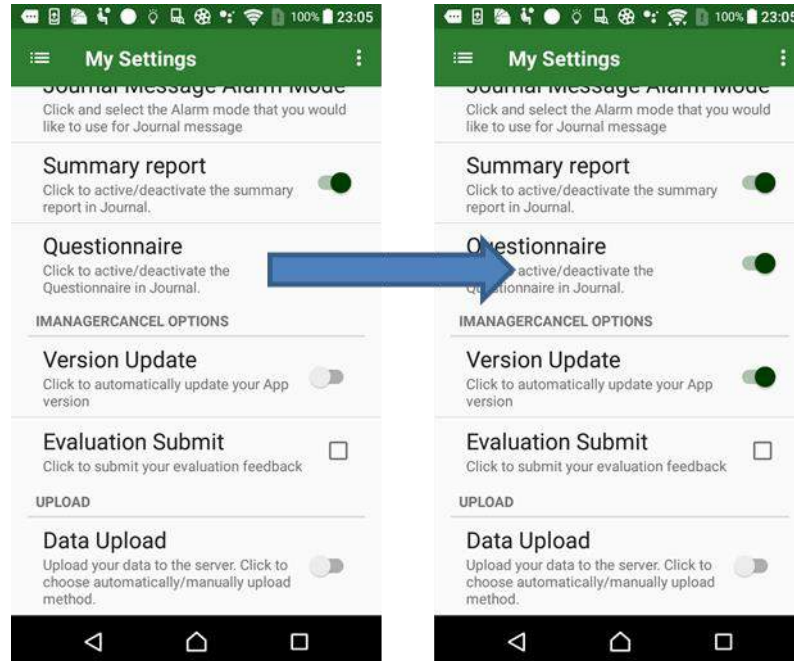


Figure 55: Journal questionnaires is activated

Result: Journal questionnaires are activated.

4.4.4.8 Enable the Version Update

Version Update–Enable the App can check the version number and upgrade itself.

Action 1: Turn on the Switch component.

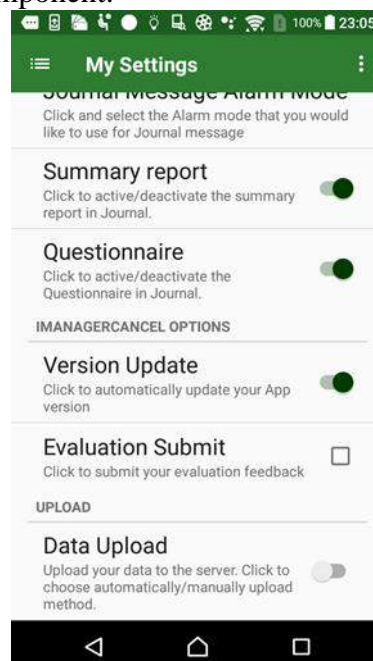


Figure 56: version update is activated.

Result: Version update is activated.

4.4.4.9 Enable the Data Upload

Data Upload–Enable the App can upload the data via internet.

Action 1: Turn on the Switch component.

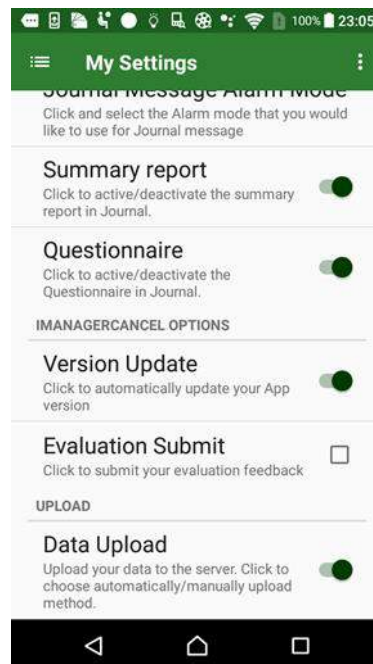


Figure 57: data upload is activated

Result: Data upload is activated.

4.4.6. Access Overview page and Function Suite

4.4.5.1 Access Weight Suite

Action 1: Click the Weight Suite icon in Overview to show the Weight Suite page.

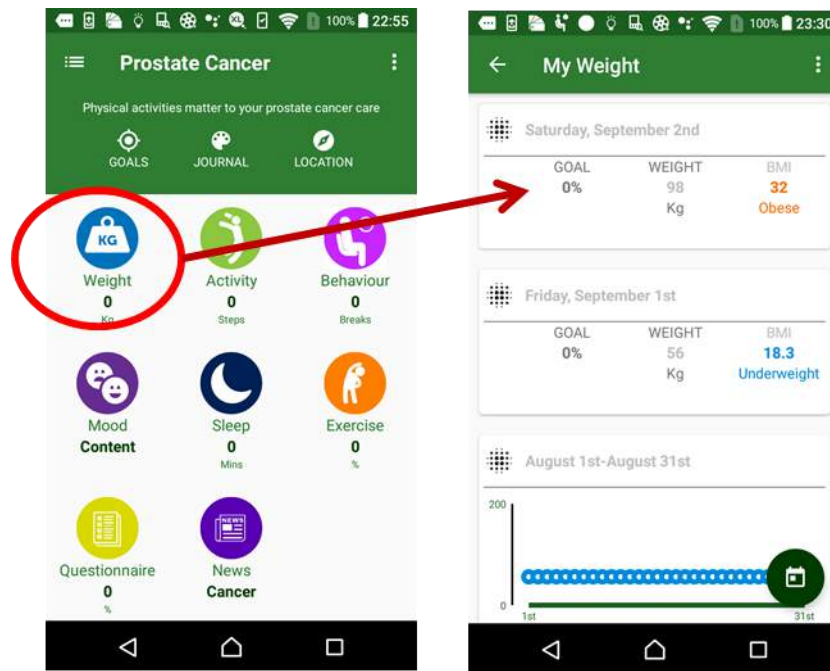


Figure 58: Weight suite overview

Action 2: Click the item to edit the weight and height values.

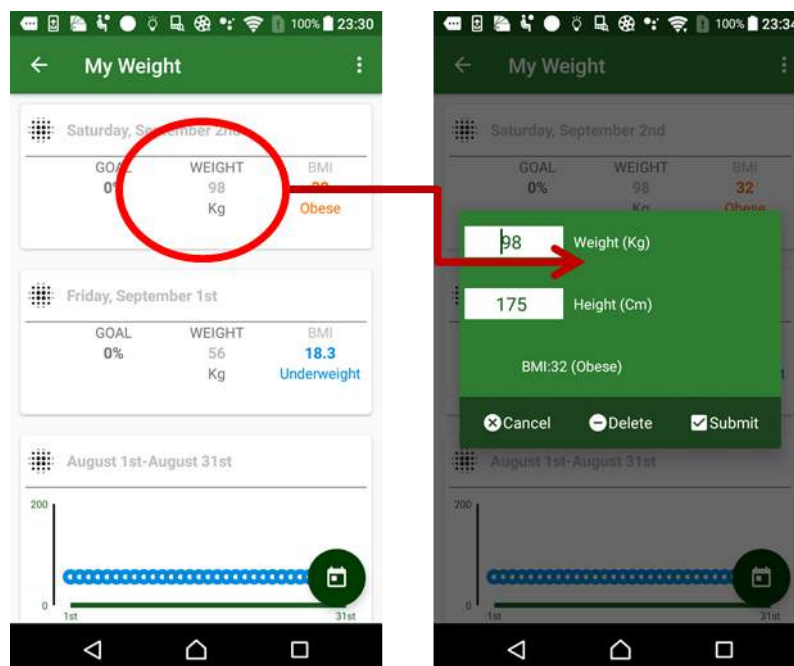


Figure 59: edit weight and height

Action 3: The different colors represent different BMI levels.

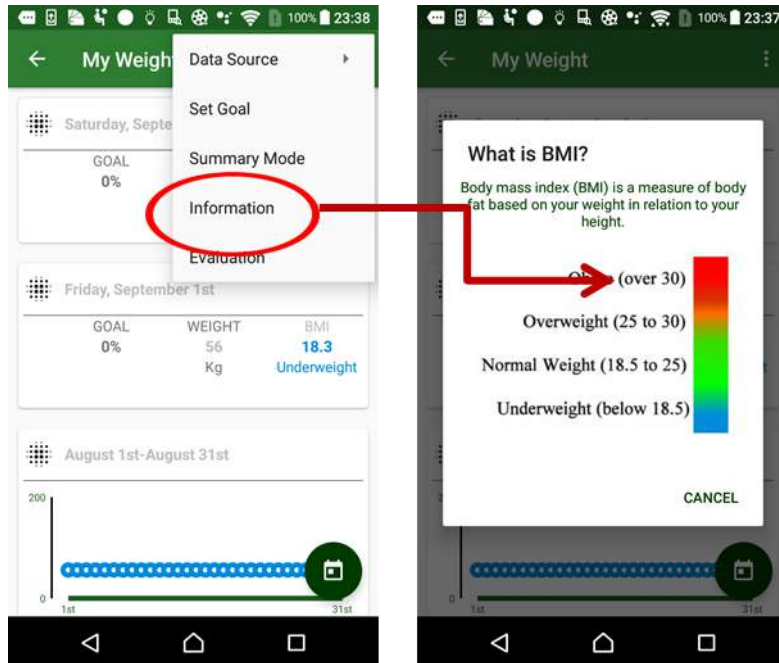


Figure 60: check weight BMI Level

Action 3: Click the “Data Mode” button to switch the data sources.

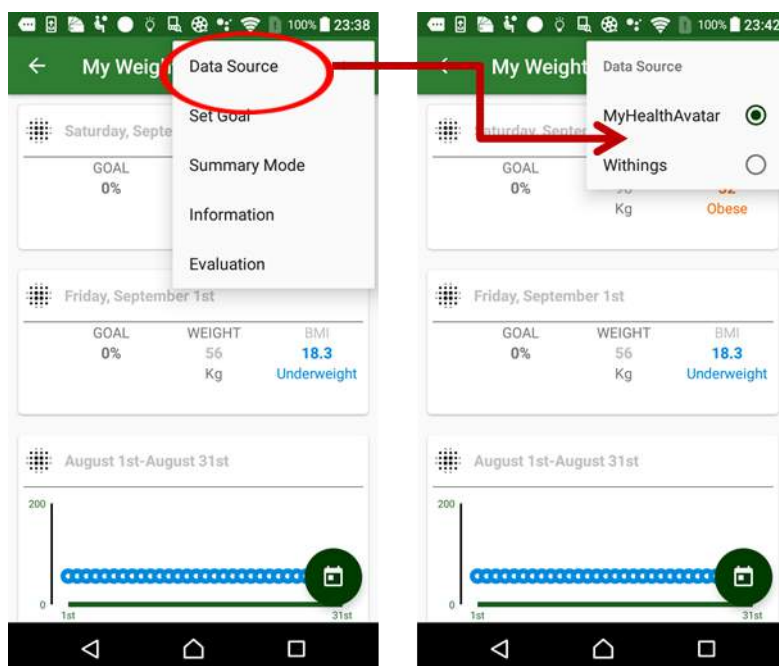


Figure 61: change the data source

Action 4: Click the “Set Goals” button to change the goals.

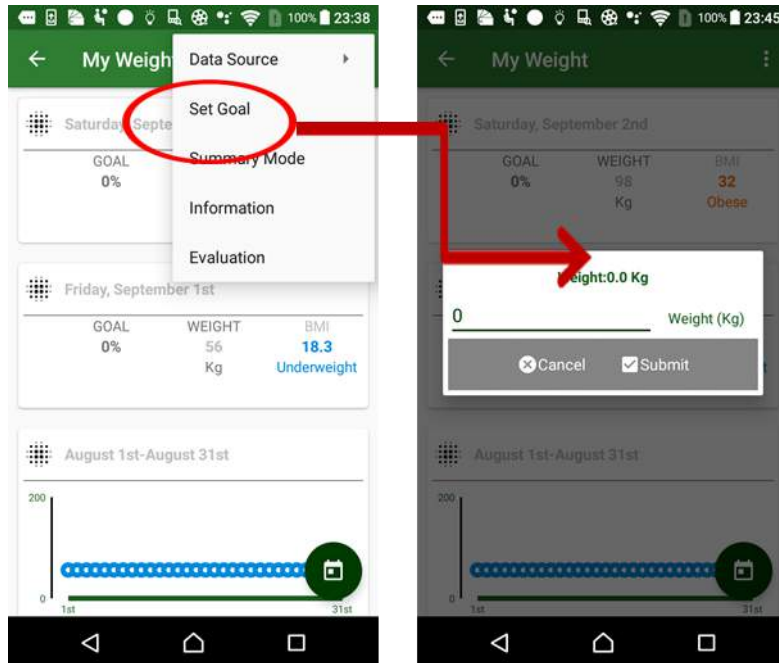


Figure 62: change the weight goal

Action 5: Swap the screen up and down, in order to check data and summary in different date.

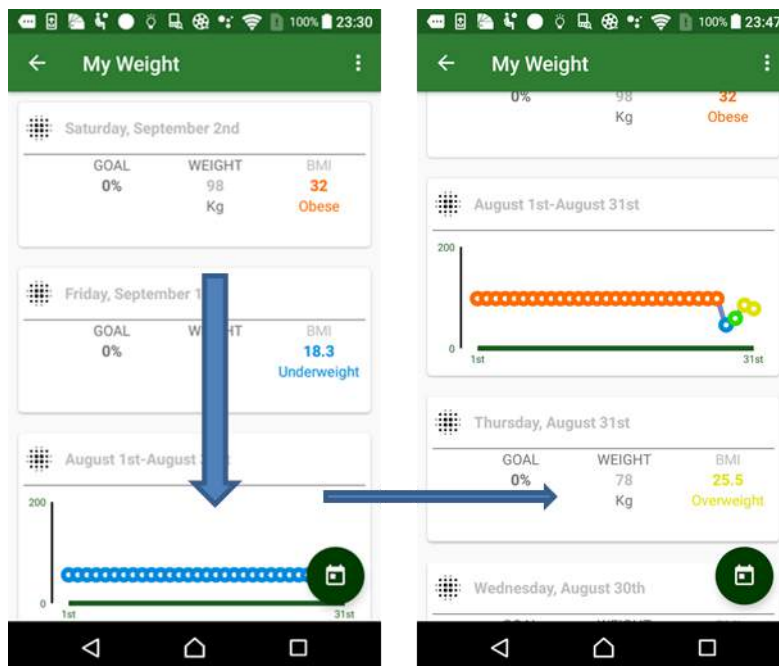


Figure 63: check the weigh on different date

4.4.5.2 Access Activity Suite

Action 1: Click the Activity Suite icon in Overview to show the Activity Suite page.

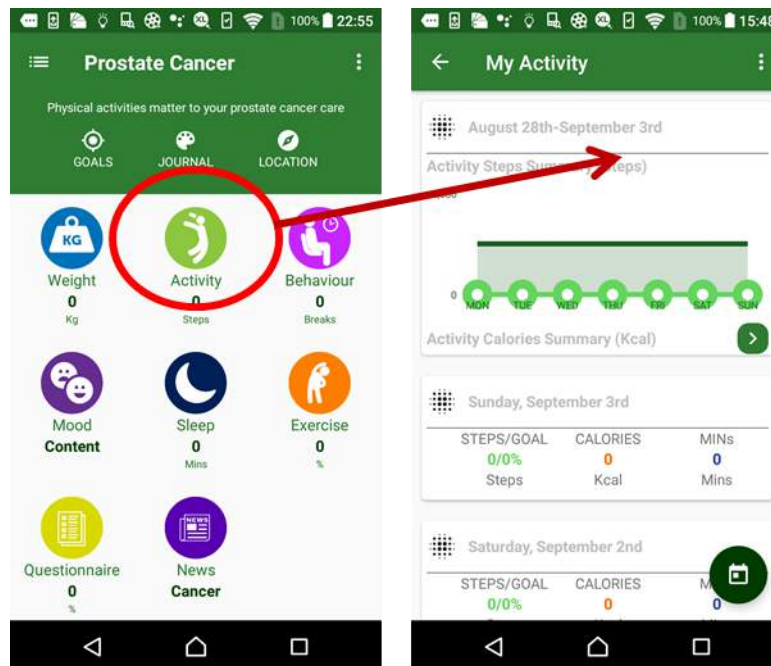


Figure 64: Activity suite overview

Action 2: Swap the screen to the up and down, in order to change the date and check the summary.

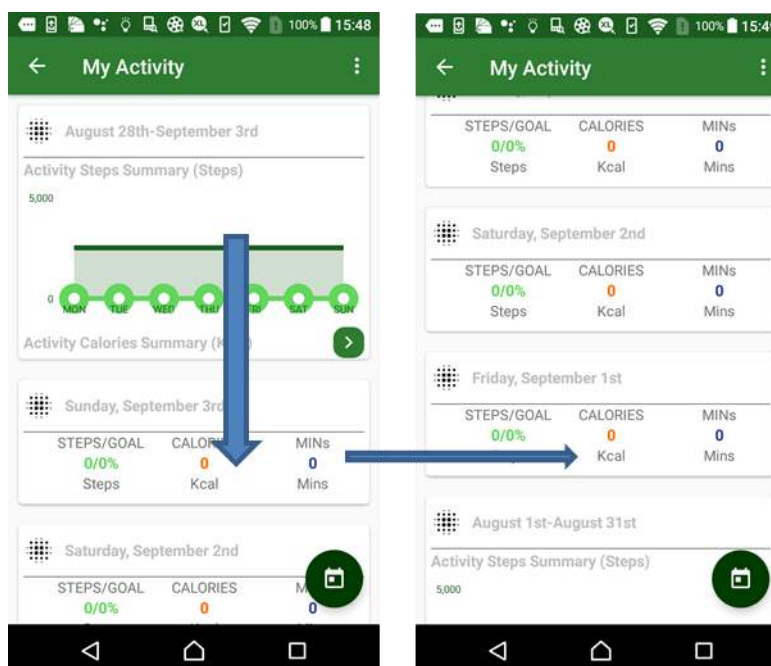


Figure 65: check the activity on different date

Action 3: Click the “Data Mode” button to switch the data sources.

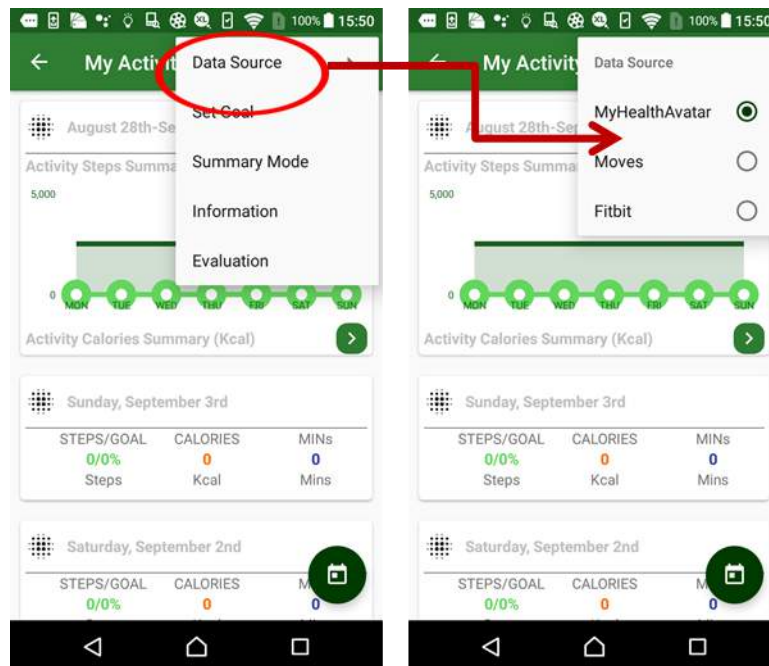


Figure 66: change the data source

Action 4: Click the “Set Goals” button to change the goals.

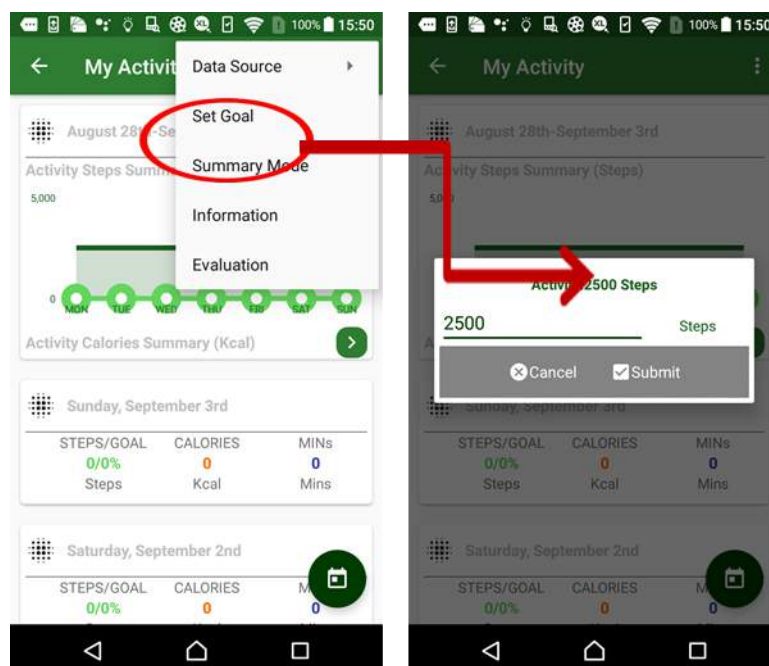


Figure 67: change the activity goal

4.4.5.3 Access Sitting Behaviour Suite

Action 1: Click the Behaviour Suite icon in Overview to show the Behaviour Suite page.

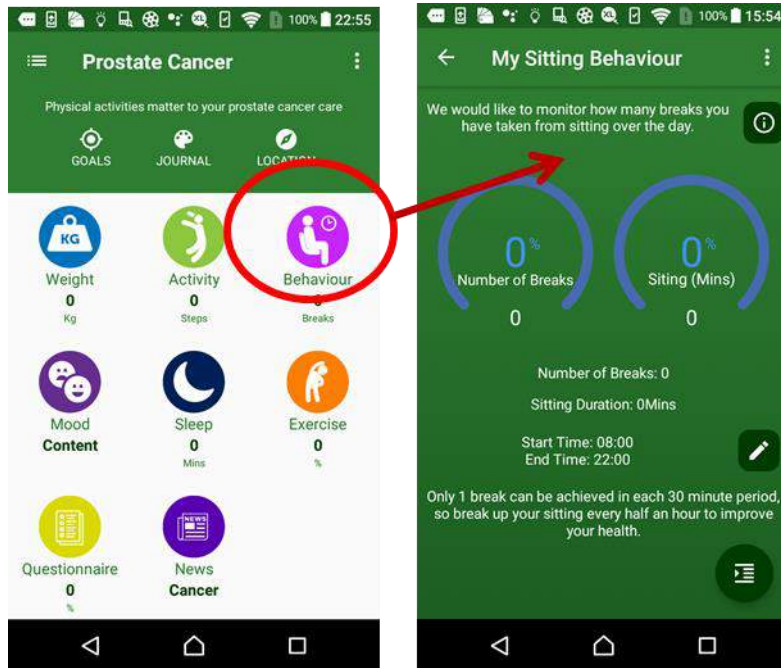


Figure 68: Behaviour suite overview

Action 2: Click the “edit” button to edit the duration time that you want to monitor the sitting behaviour.

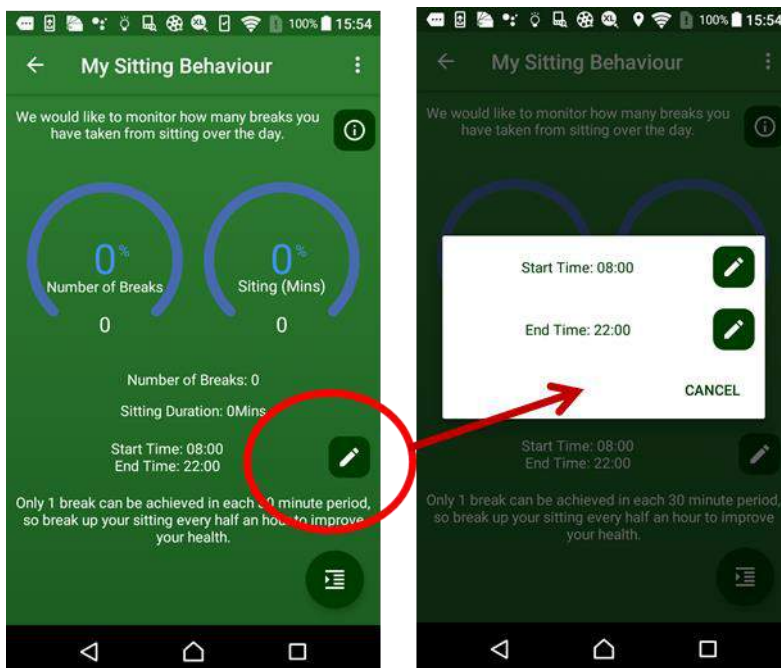


Figure 69: edit Behaviour monitor duration

Action 3: Click the “list” button to check the history records.

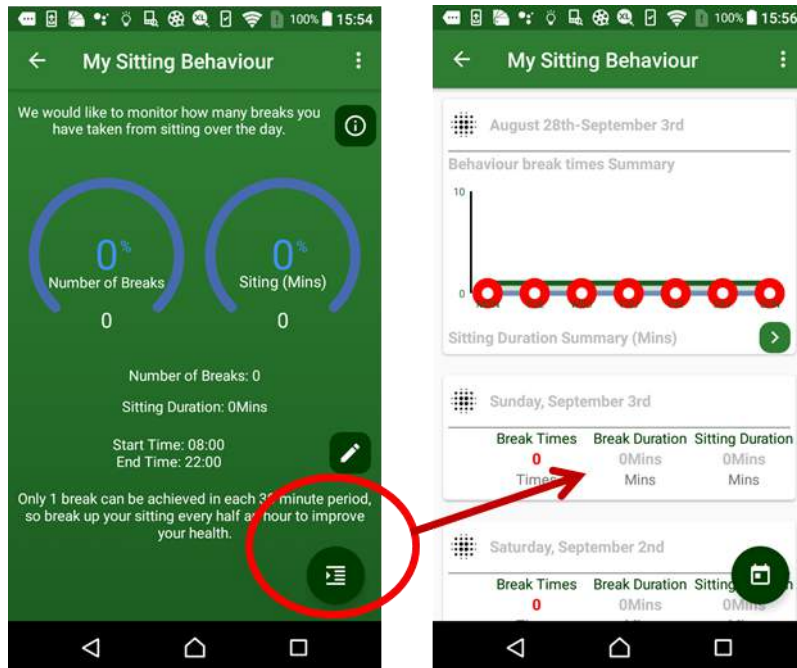


Figure 70: Behaviour history record

Action 4: Swap the screen to the up and down, in order to change the date and check the summary.

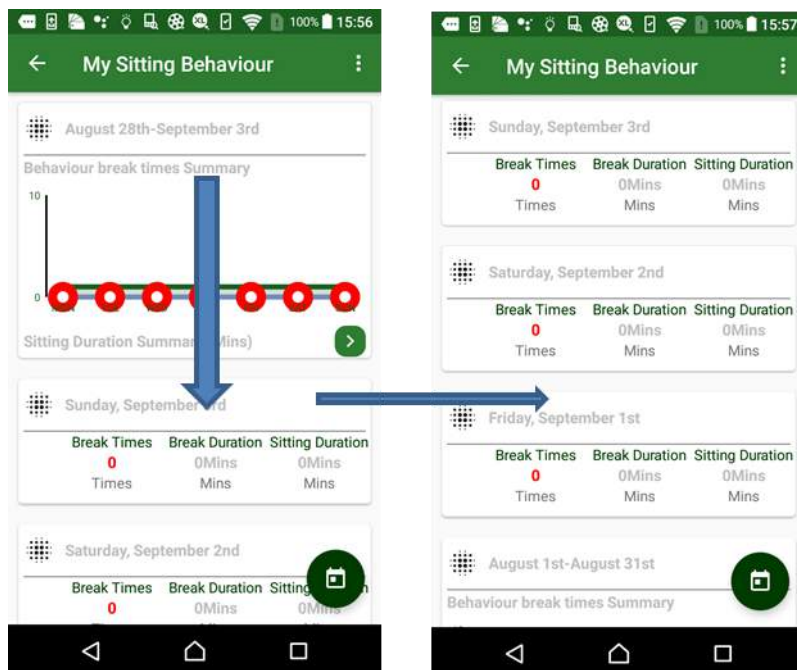


Figure 71: change the data source

Action 5: Click the “Set Goals” button to change the goals.

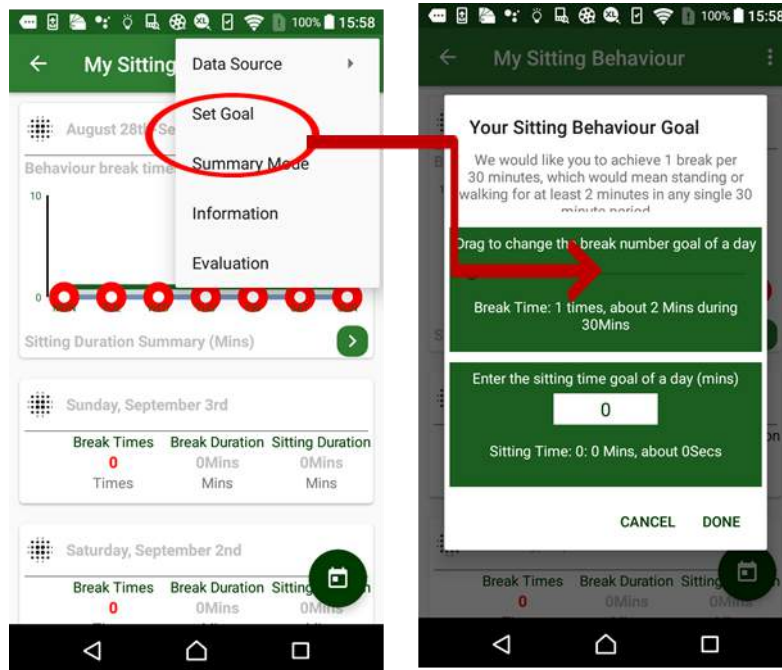


Figure 72: change the break up sitting time and sitting time goals

4.4.5.4 Access Sleep Suite

In this suite, patient can manually insert Sleep quality or get information on sleep quality from Fitbit automatically.

Action 1: Click the Sleep Suite icon in Overview to show the Sleep Suite page.

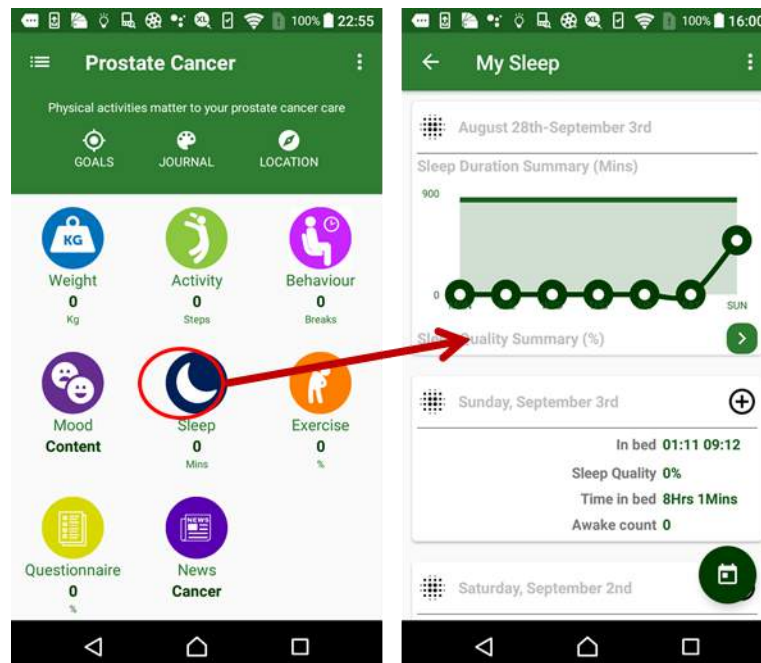


Figure 73: the Sleep suite overview

Action 2: Click the “+” button to add new sleep record.

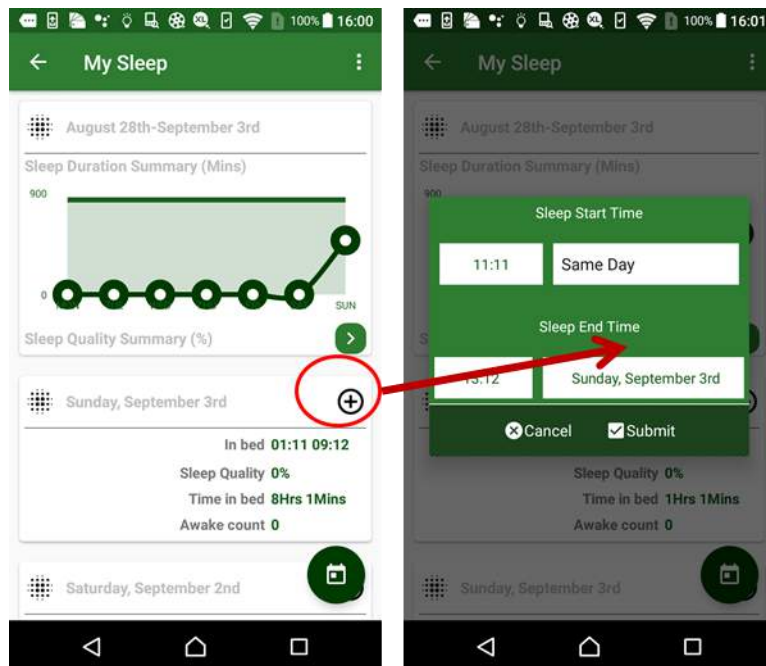


Figure 74: add new record to the Sleep suite (1)

Action 3: Swap the screen to the up and down, in order to change the date and check the summary.

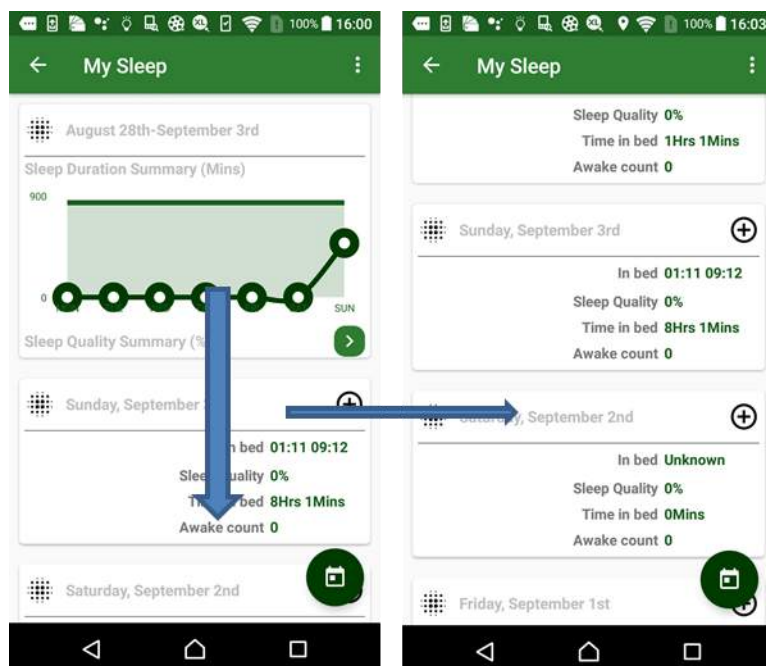


Figure 75: check the sleep information on different date

Action 4: Click the “Data Mode” button to switch the data sources.

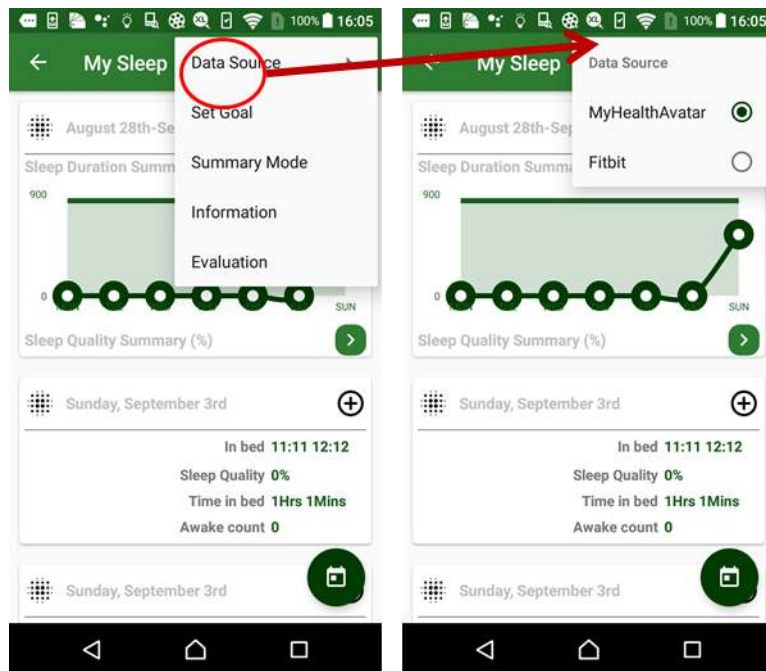


Figure 76: change the data source

Action 4: Click the “Set Goals” button to change the goals.

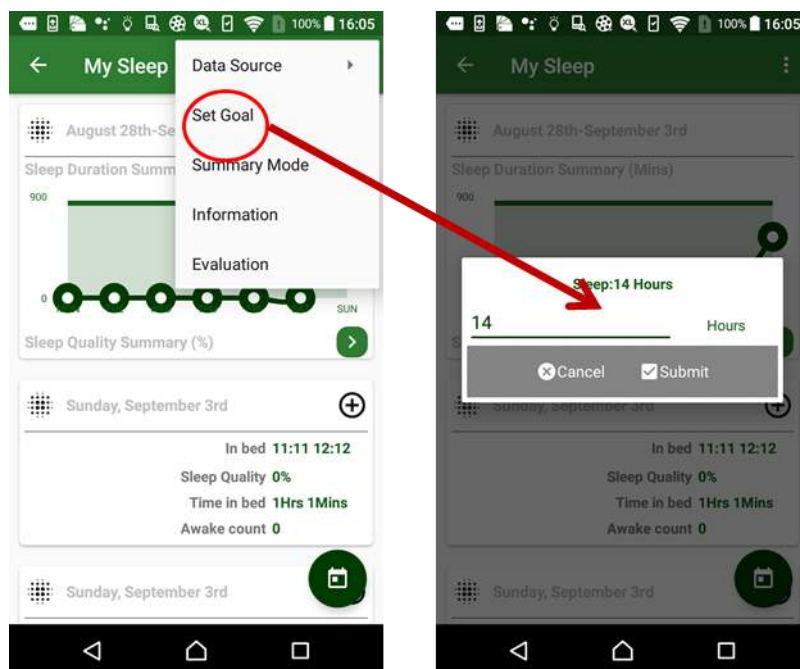


Figure 77: change the activity goal

4.4.5.5 Access Mood Suite

Action 1: Click the Mood Suite icon in Overview to show the Mood Suite page.

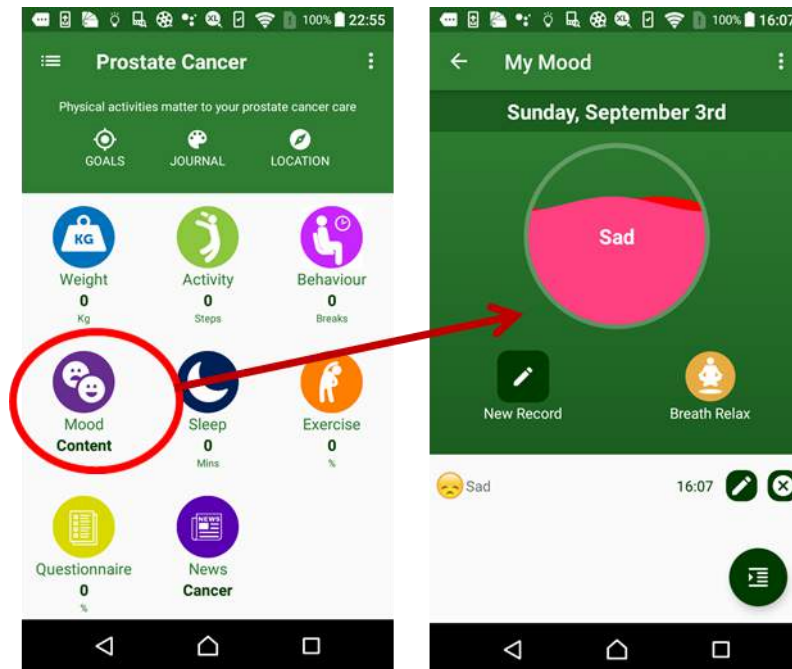


Figure 78: Mood suite overview

Action 2: Click the “edit” button to add new mood. The record of the new mood will affect the main mood status as shown in the water flow progress bar.

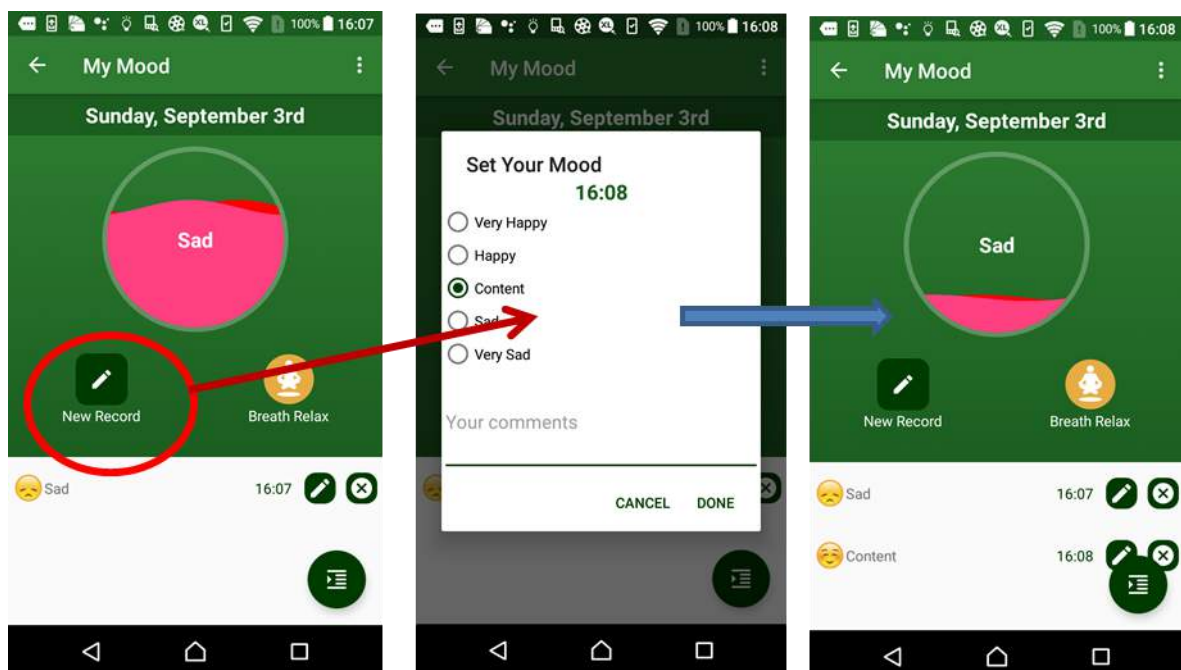


Figure 79: add new mood to Mood suite

Action 3: Click the “ok” button to submit new emotion and click Edit button to edit the emotions.

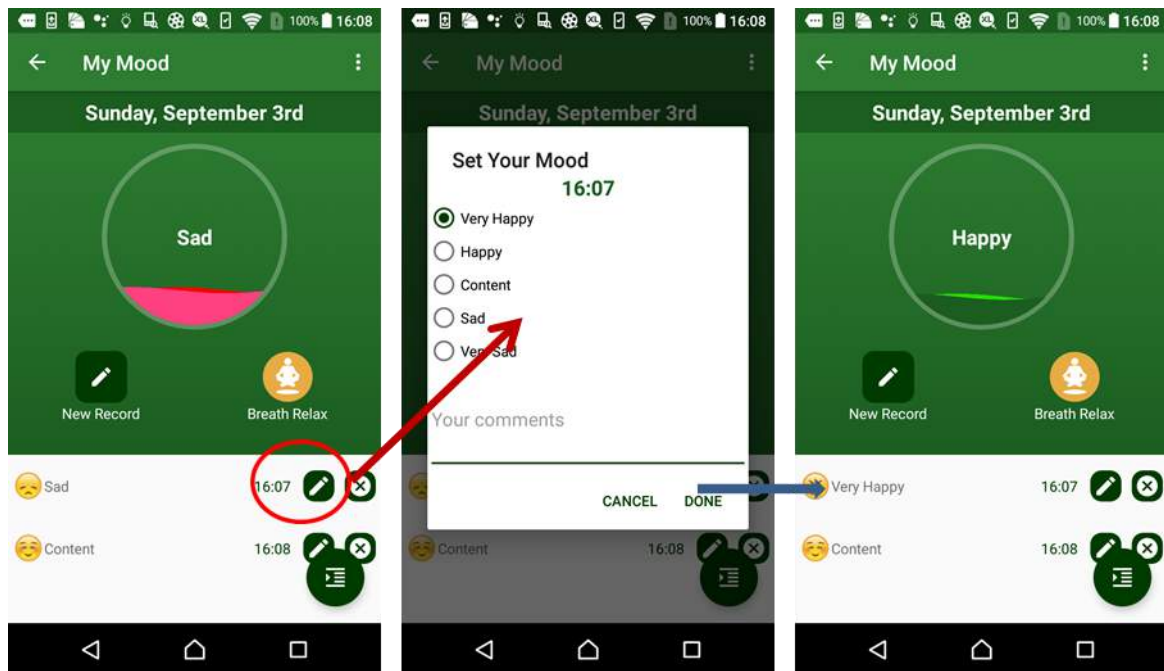


Figure 80: edit the mood in Mood suite

Action 4: Click the “list” button to check the history records.

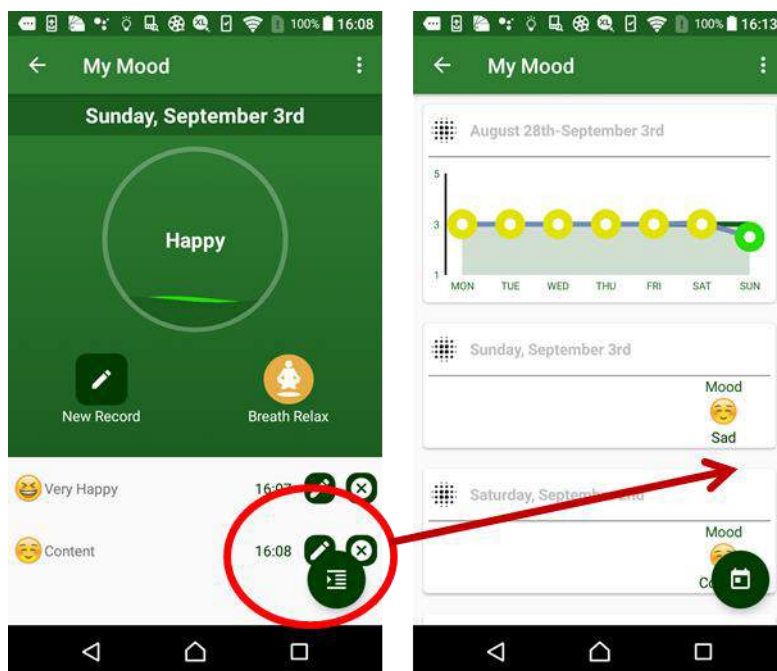


Figure 81: edit the mood in Mood suite

Action 5: Swap the screen to the up and down, in order to change the date and check the summary.

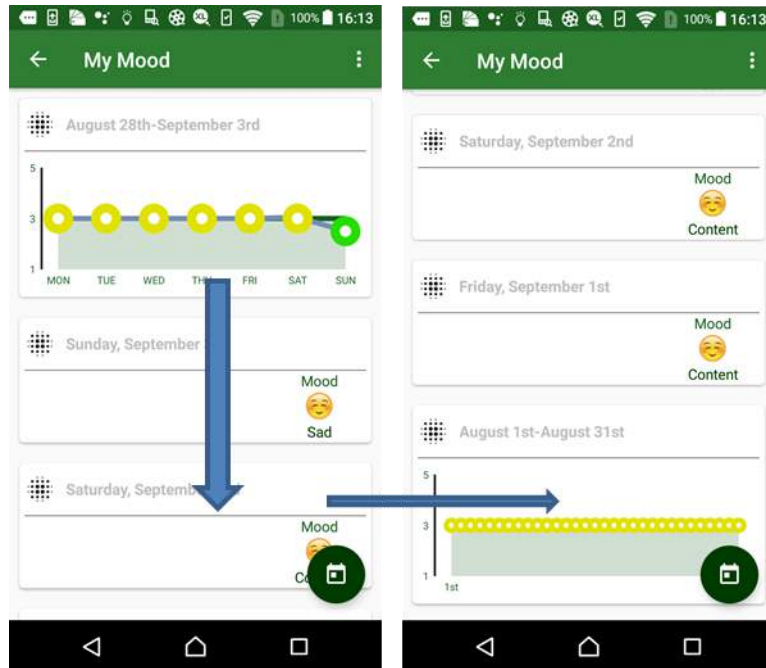


Figure 82: check the mood status on different date

Action 6: Click the relax button to start a relax program.

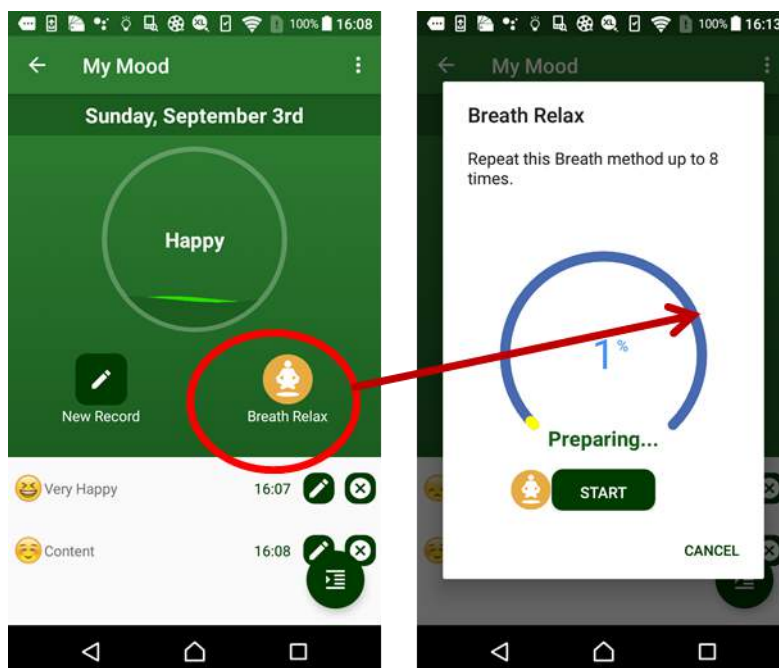


Figure 83: have the relax program

4.4.5.6 Access Exercise Suite

Action 1: Click the Exercise Suite icon in Overview to show the Exercise Suite page.

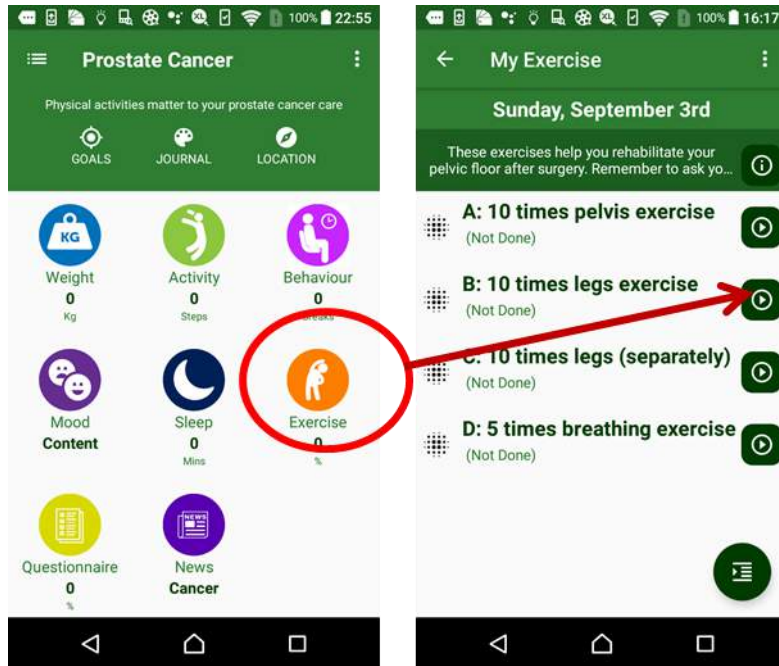


Figure 84: Exercise suite overview

Action 2: Click the item in the page to show the exercise panel.

Action 4: Click “Start” button to start the exercise program.

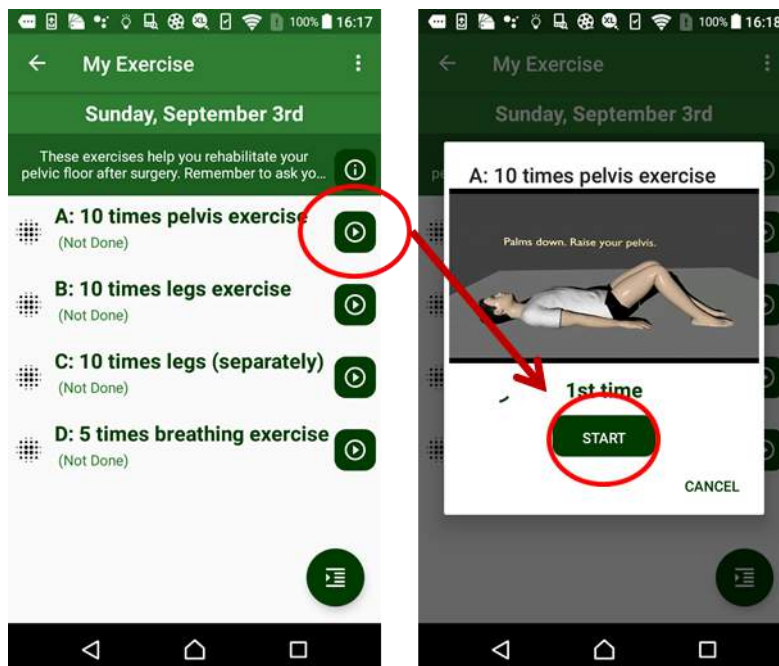


Figure 85: Exercise suite overview (1)

4.4.5.7 Access Questionnaire Suite

Action 1: Click the Questionnaire Suite icon in Overview to show the Questionnaire Suite page.

Action 2: If android 6.0 or above, allow the audio permission.

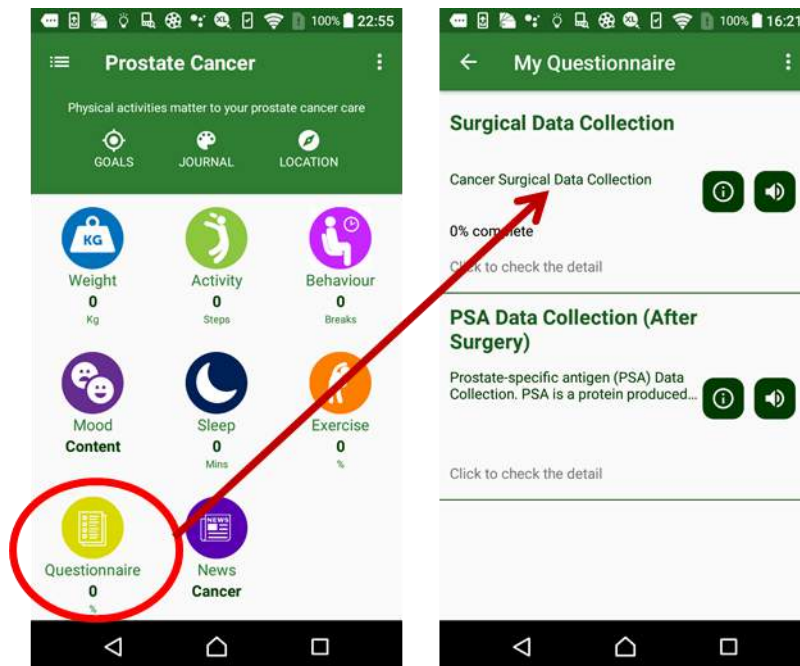


Figure 86: Questionnaire suite overview

Action 3: Click the Edit button to edit the value.

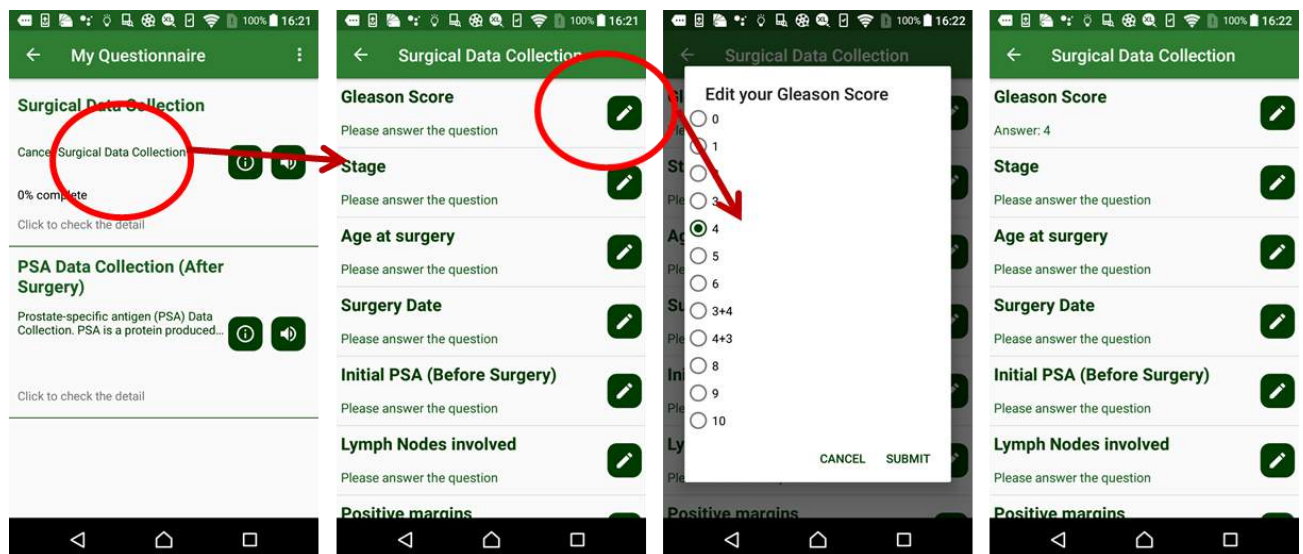


Figure 87: update the questionnaire

Action 4: Click the floating button to add the new record.

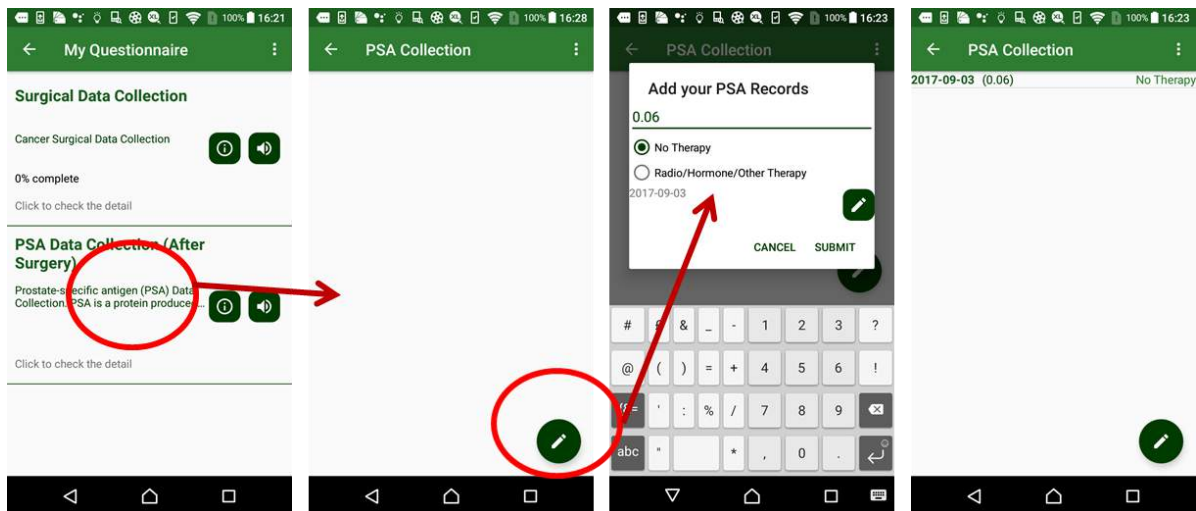


Figure 88: update the PSA record

4.4.5.9 Access Goals Setting

Action 1: Click the button “Set Goals”

Action 2: Click the item to edit the goals

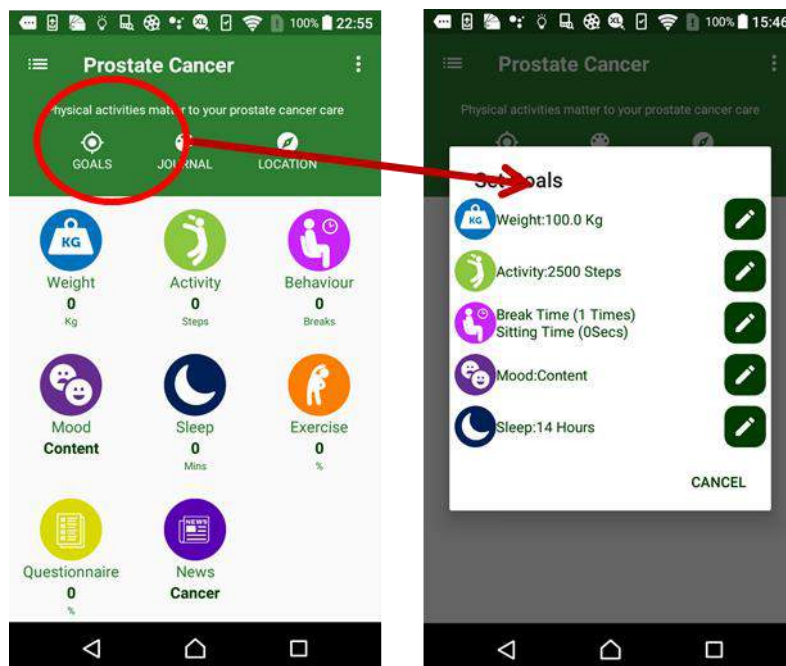


Figure 89: the Goals page overview

Result Goals Setting works fine.

4.4.7. Access Journal Function

Action 1: Click the Journal in the left menu

4.4.6.1 Access Journal Item Edit Panel

Action 1: Click “+” icon at the bottom of Journal modules.

Action 2: Click the “edit” icon in the appearing panel.

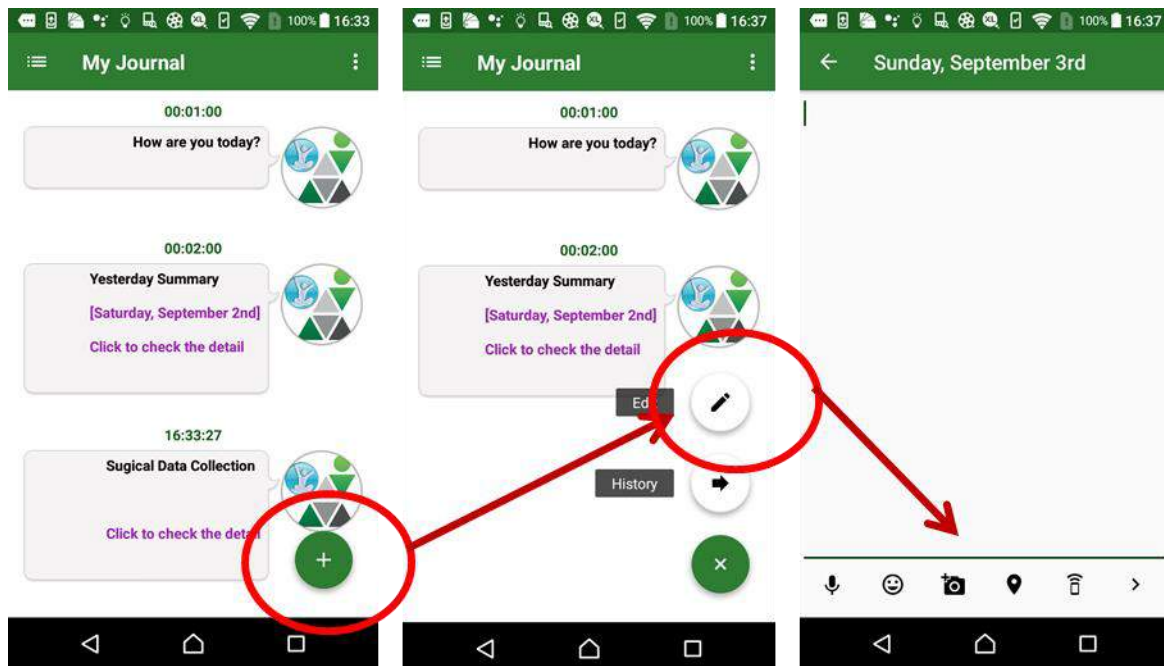


Figure 90: the control panel overview

Action 3: Send the icon to the Journal item edit screen.

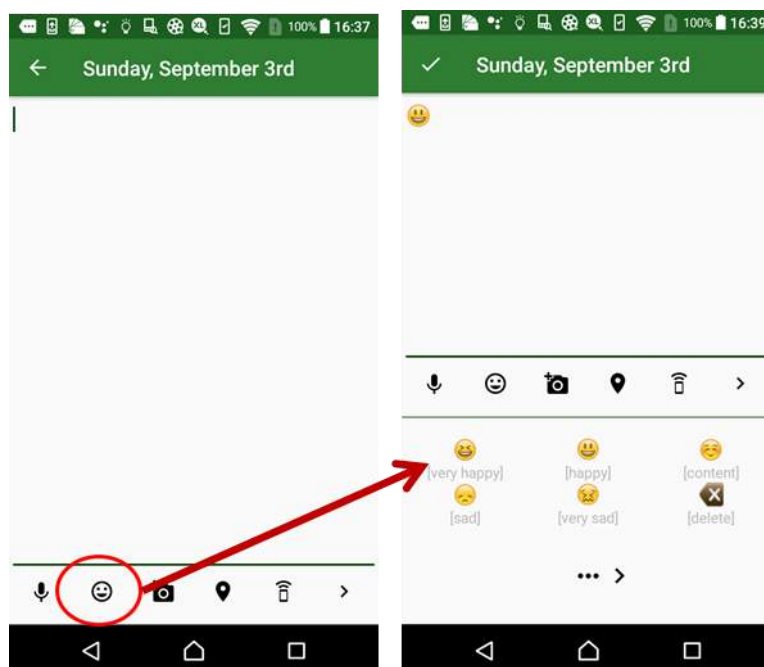


Figure 91: the control panel function (1)

Action 4: Send the message to the Journal item edit screen by using voice input.

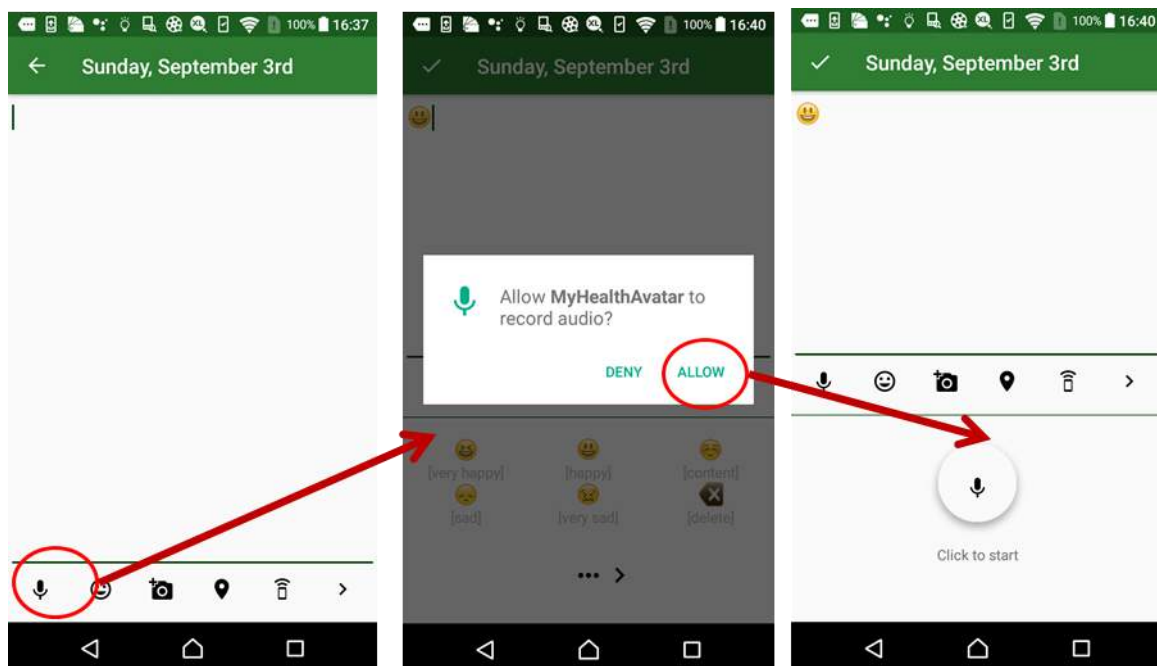


Figure 92: the control panel function (2)

Action 5: Take photo to the Journal item edit screen.

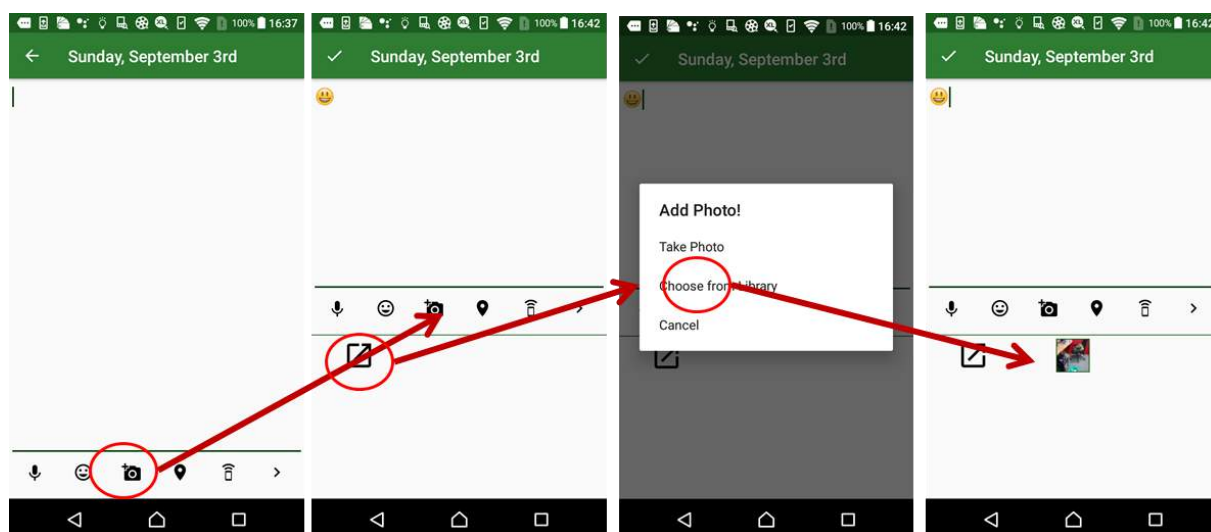


Figure 93: the control panel function (3)

Action 6: Take current location to the Journal item edit screen.

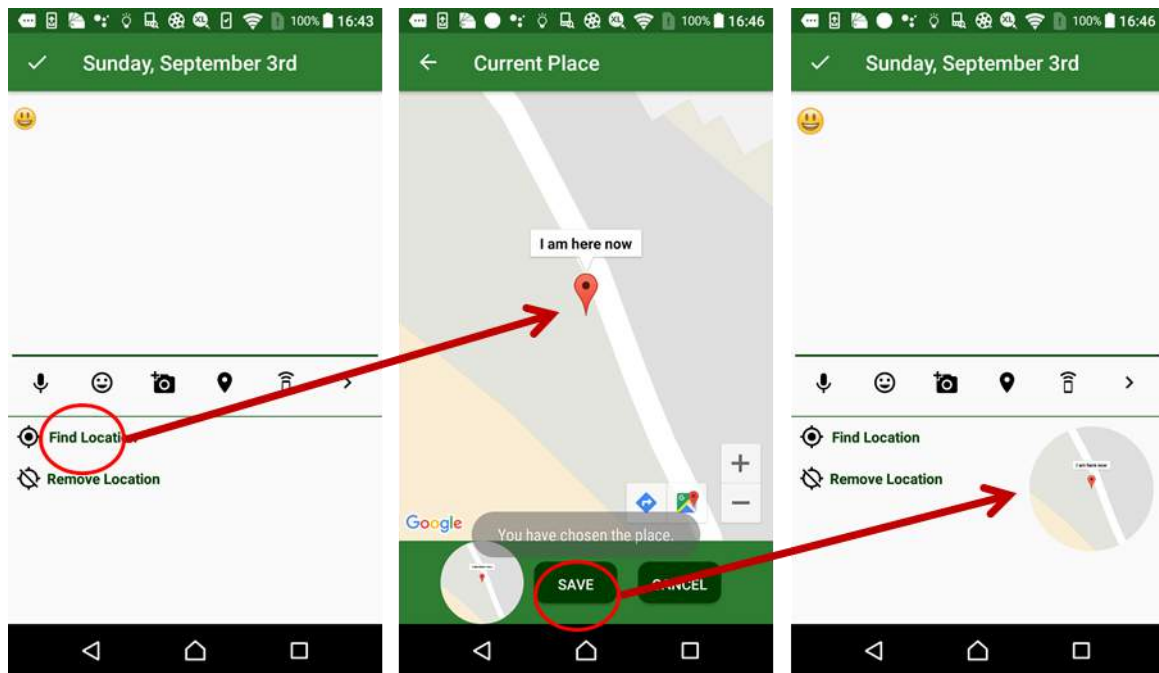


Figure 94: the control panel function (4)

Result icons control panel works fine.

4.4.6.2 Access Mood Monitor

Action 1: As mentioned in previous section, the Mood Questionnaires will appear around at 12:30, 16:30 and 20:30. Once the message appears in Journal page, user can pick the options.

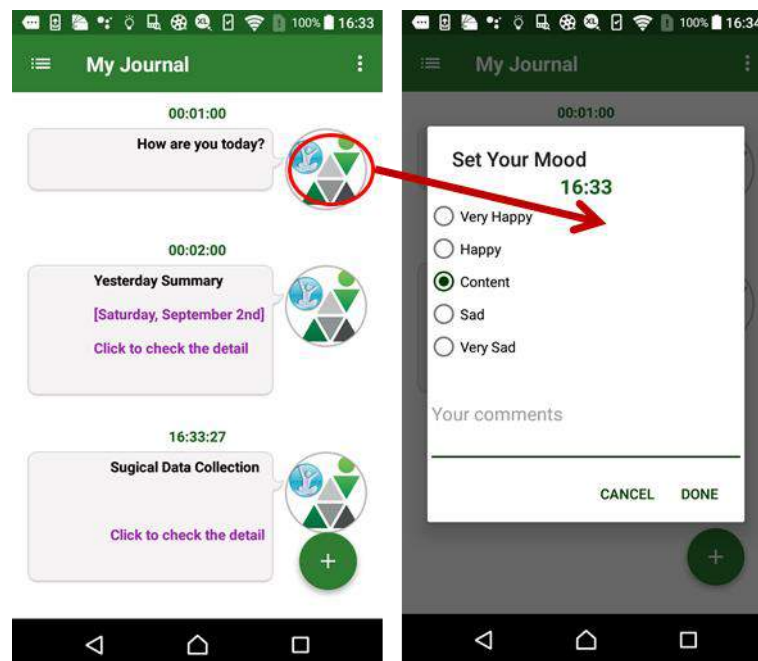


Figure 95: The Mood Journal

Result: Mood works fine.

4.4.6.3 Access Summary Function

Action 1: As mentioned in previous section, the Summary will appear according to the setting time. Once the message appears in Journal page, user can click the message and check the detail.

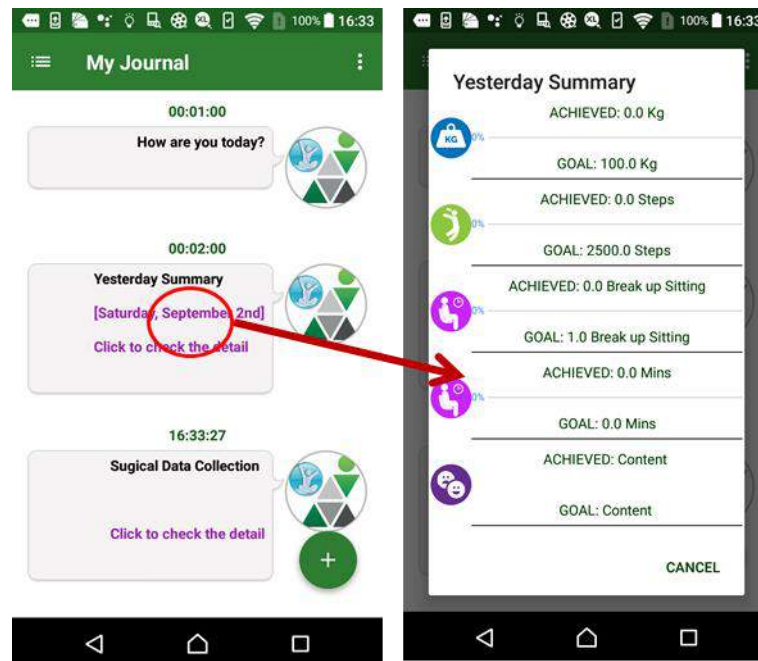


Figure 96: the Summary in Journal

Result: Summary function works fine.

4.4.6.4 Access Prostate Cancer Questionnaire

Action 1: As mentioned in previous section, the activity notice will appear according to the user's activity.

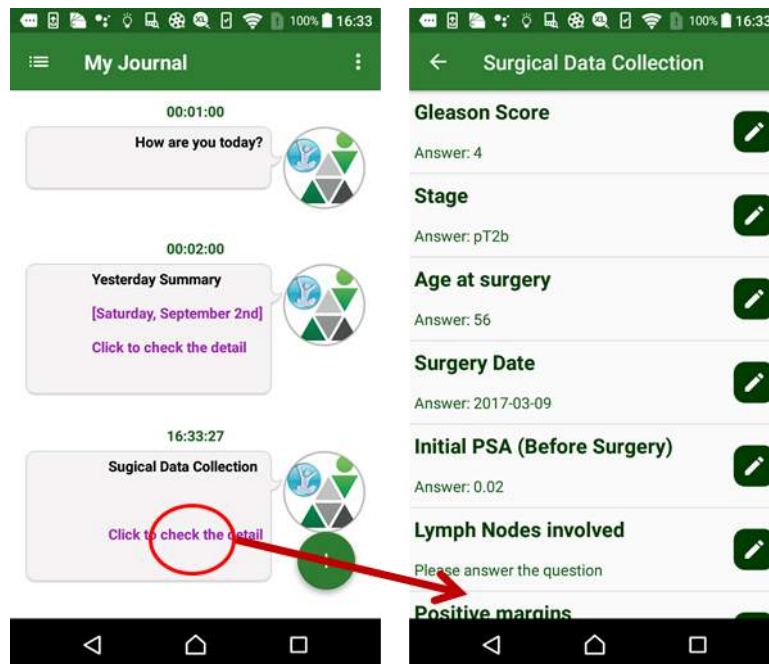


Figure 97: the Questionnaire in Journal

Result: Activity notice works fine.

4.4.8. Access Calendar Function

Action 1: Click the Calendar in the left menu

Action 2: Swap calendar to the left or right to change the month.

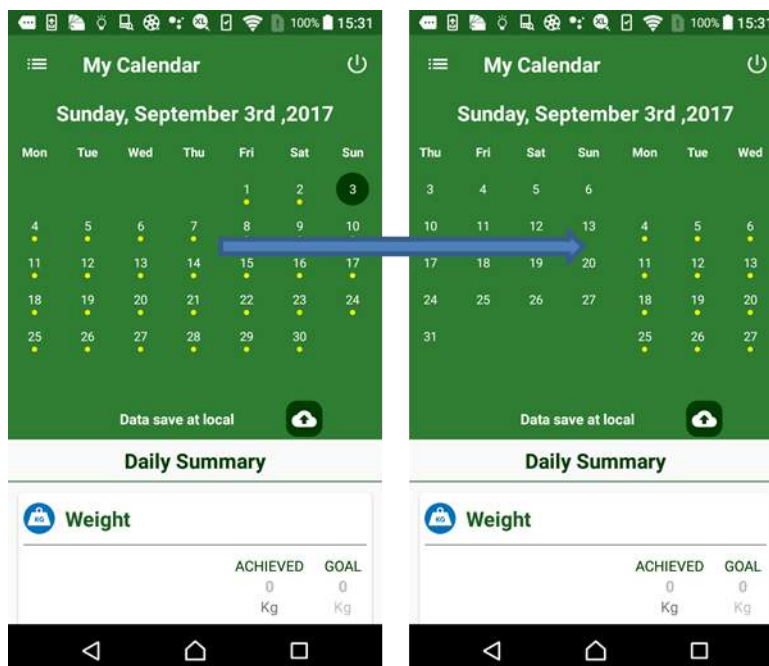


Figure 98: Calendar overview

Action 3: Click a date in the Calendar and check the daily summary

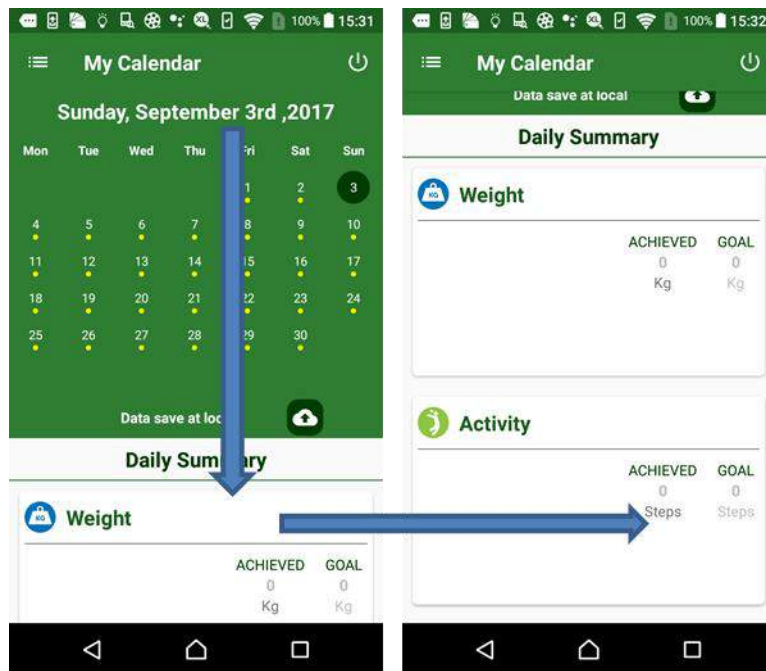


Figure 99: Calendar overview (2)

4.4.9. Access Profile Function

Action 1: Click the Profile in the left menu

Action 2: if any new changes on the Website, the profile will ask user to choose download the changes or use the local one.

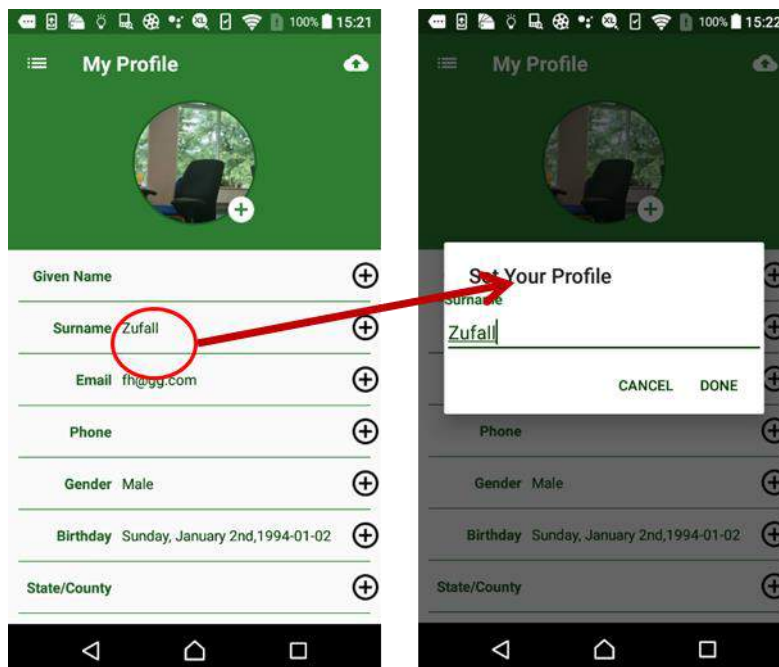


Figure 100: General Profile overview

Result: Sync function works fine.

4.4.8.1 Update Profile Picture

Action 1: Click “+” to add the user profile picture.

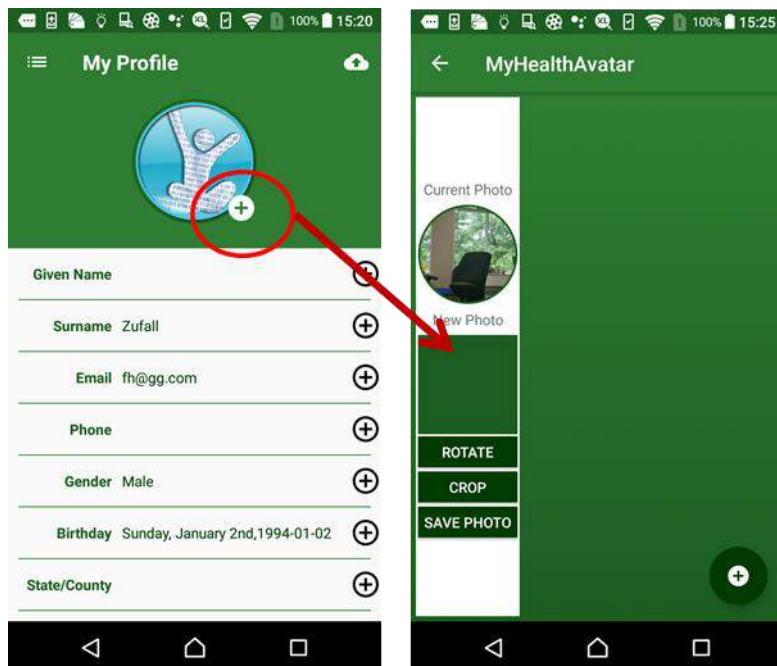


Figure 101: profile image updating (1)

Action 2: Click “+” button to load a picture and give permission to the App.

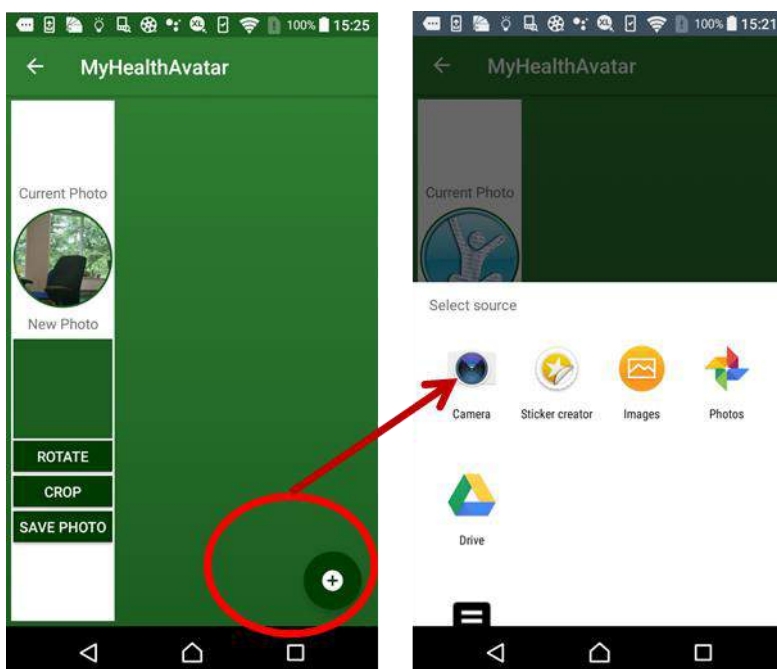


Figure 102: profile image updating (2)

Action 3: Click the cropped picture and save the photo. Then go back to the main page of Profile.

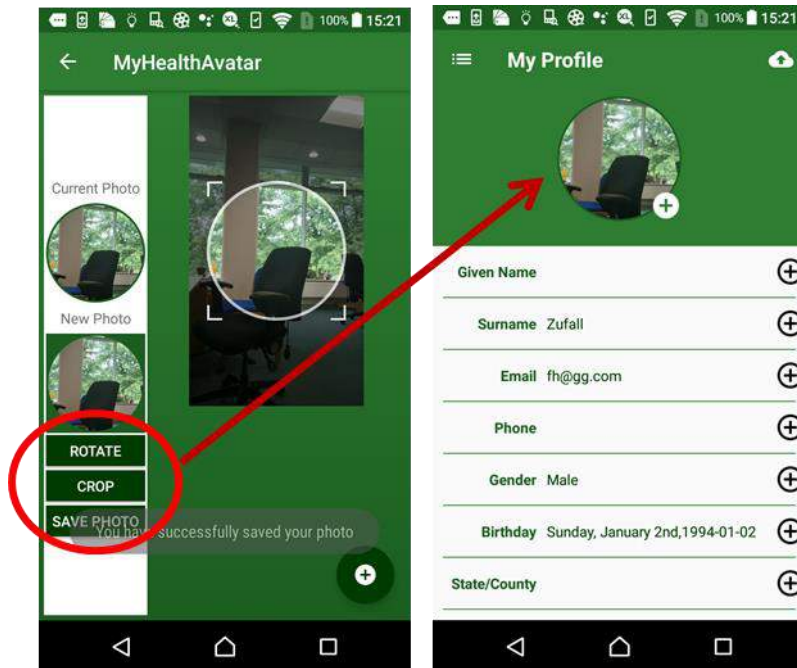


Figure 103: profile image updating (3)

Result: Profile Picture function works fine.

4.4.8.2 Update User Profile

Action 1: Click “+” to edit profile item.

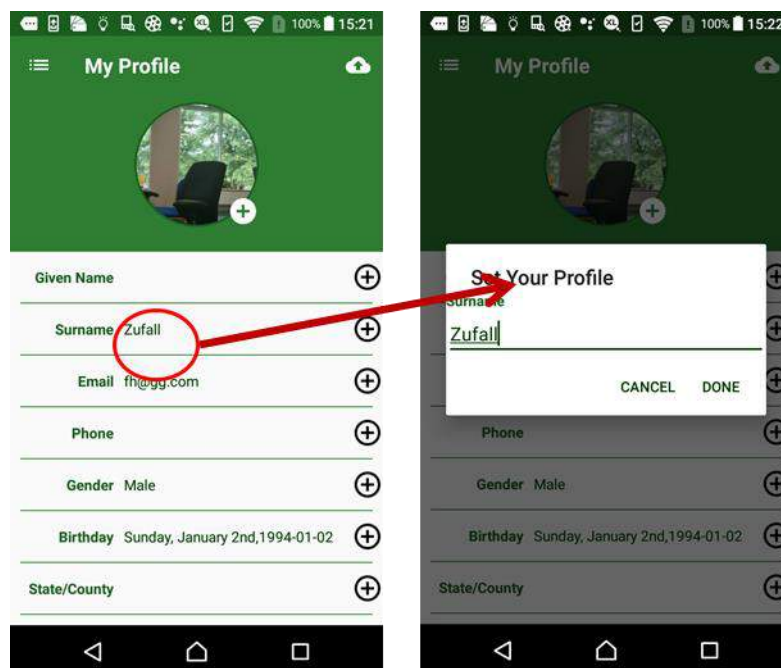


Figure 104: Profile updating (1)

Action 3: Upload the new changes to server by click the “Cloud” Button.

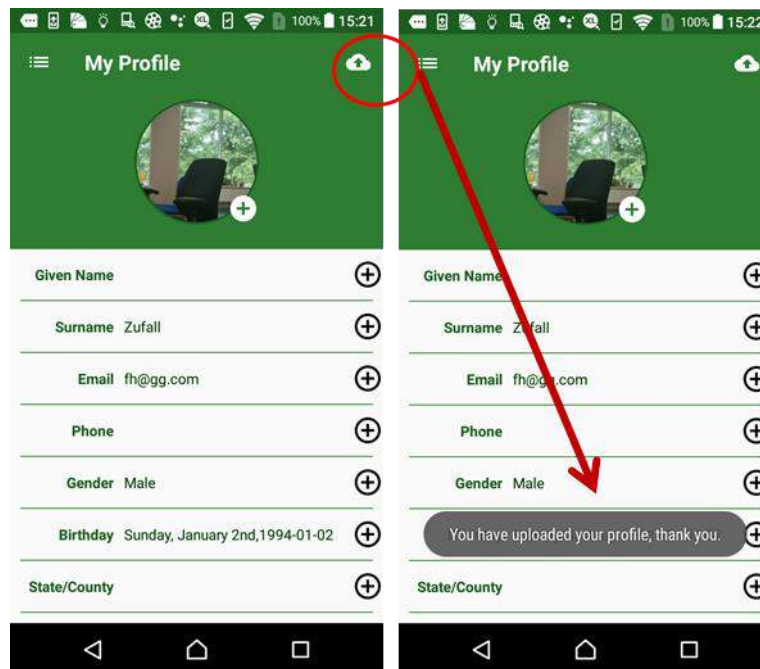


Figure 105: Profile upload

Result: Profile function works fine.

4.4.10. Access Sharing Function

Action 1: Click the Sharing in the left menu

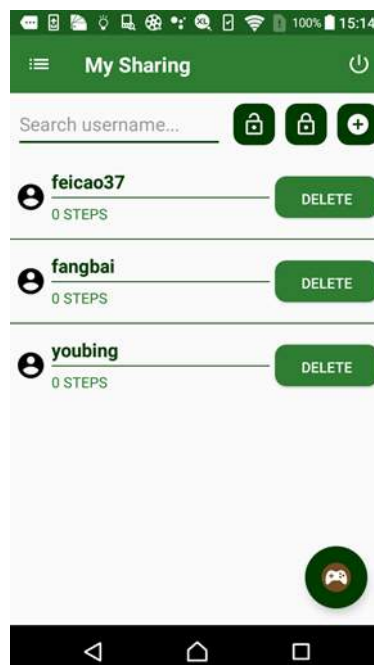


Figure 106: Sharing overview

Action 2: Enter the user name that you want to find and click find button.

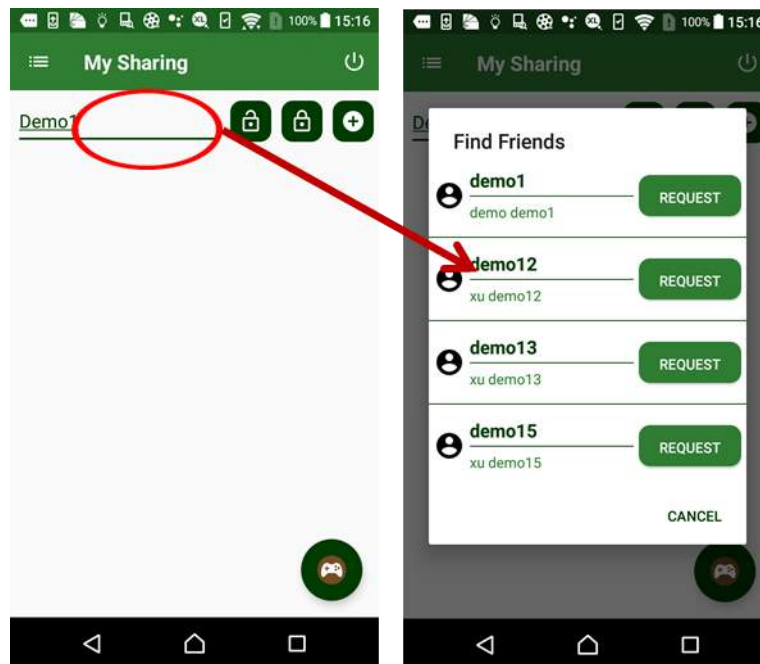


Figure 107: find the friend list

Action 3: Click “Add” button to send request to the user who you want to add as friend. Once the friend user confirm that request, she/he will be appear in the list.

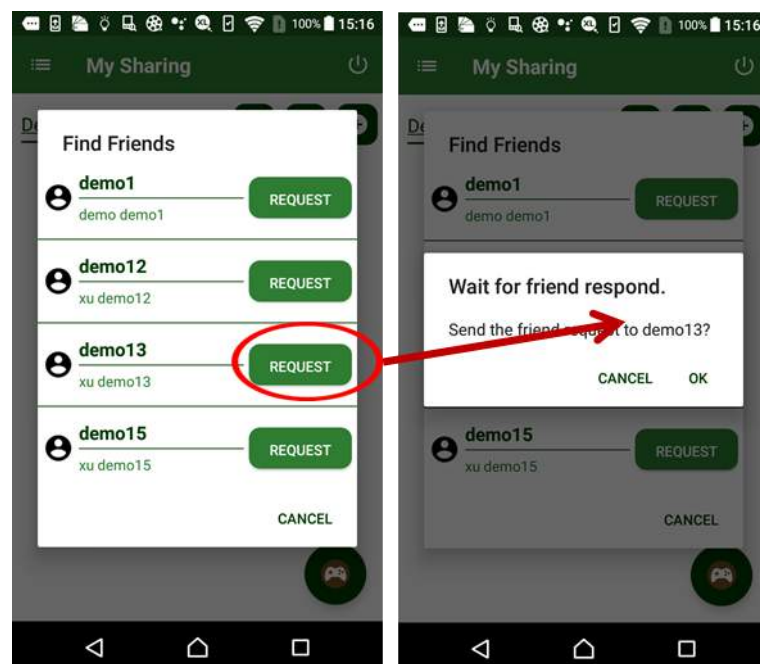


Figure 108: add new friend

Result: Sharing function works fine.

4.5. Installation and configuration guidelines

App correctly running with all features on all Android 5, 6, 7 devices.

General permission——Basic access permission, no need patients give the permission manually. The app can access the device features.

1. Device network access permission
2. Device internet access permission
3. Device receive boot completed access permission
4. Device vibrate access permission
5. Device wake lock access permission

Patient permission——App needs patients to give the permission manually in order to access the device features.

- App can access the device GPS sensor feature
- App can access the device storage feature
- App can access the device camera feature
- App can access the device audio record feature

5. MyHealthAvatar for iManageCancer (Breast Cancer)

5.1. *Intended Purpose*

MyHealthAvatar (MHA) for iManageCancer is a solution that offers access, collection and sharing of long term and consistent personal health status data through an integrated digital representation, which helps breast cancer patient with risk prediction, prevention and surgery recovery to individual citizens.

5.2. *Provided functionality*

5.2.1. *Function Suite*

A list of suite functions allow the management of information on the user's weight, activity, sleep, mood state, surgery recovery exercise and risk prediction questionnaire after the surgery.

5.2.1.1 Weight Suite

Weight Suite manage the user's weight information. User can access the weight information from Withings or MHA. User also can obtain the weekly/monthly weight summary in the Suite. Reference to D2.3-[3.2.1. use cases, Health Avatar GUI].

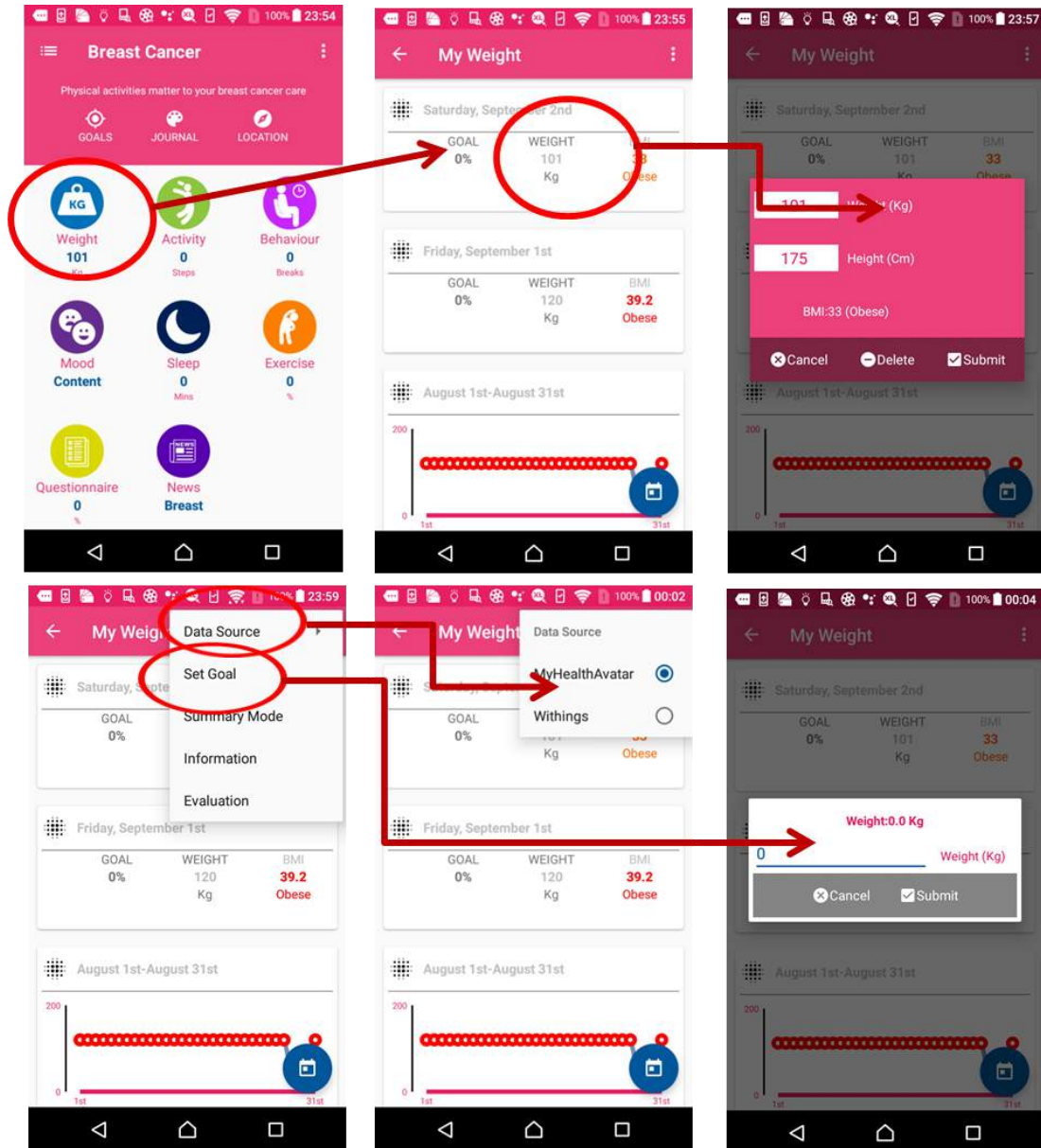


Figure 109: the Weight suite function

5.2.1.2 Activity Suite

Activity Suite manages the user's walk steps, distance and duration information. User can access the information from Withings, Fitbit, Moves or MHA data source. User also can obtain the weekly/monthly activity summary in the Suite. This function is reference to the requirement in D2.3-[3.2.1. use cases, Health Avatar GUI and 3.11.1. use cases, Lifestyle Monitoring].

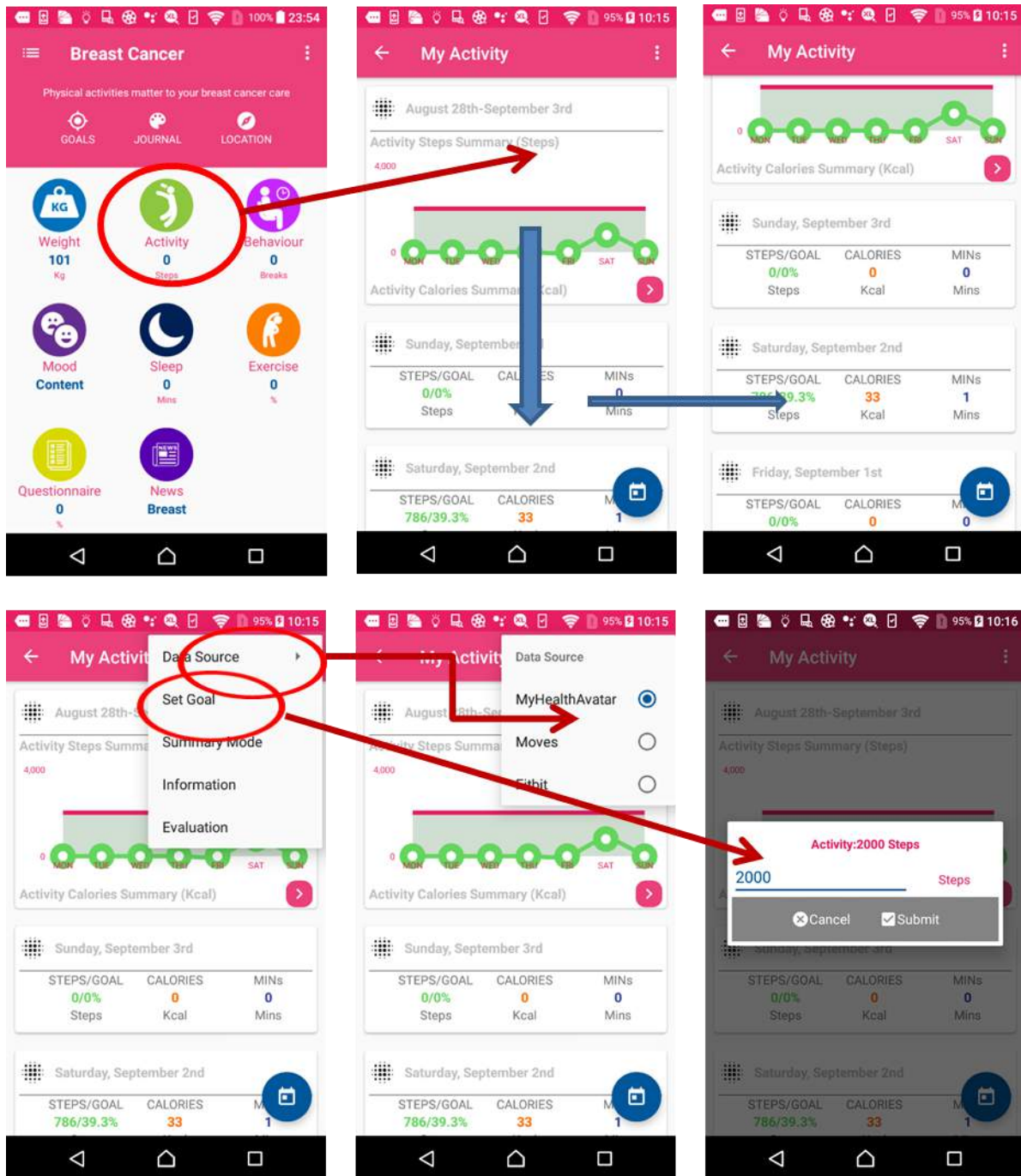


Figure 110: the Activity suite function

5.2.1.3 Sleep Suite

Sleep Suite manages the user's sleep information. User can access the sleep duration, quality information from Fitbit or MHA data source. User also can obtain the weekly/monthly sleep summary in the Suite. This function is reference to the requirement in D2.3-[3.2.1. use cases, Health Avatar GUI and 3.11.1. use cases, Lifestyle Monitoring].

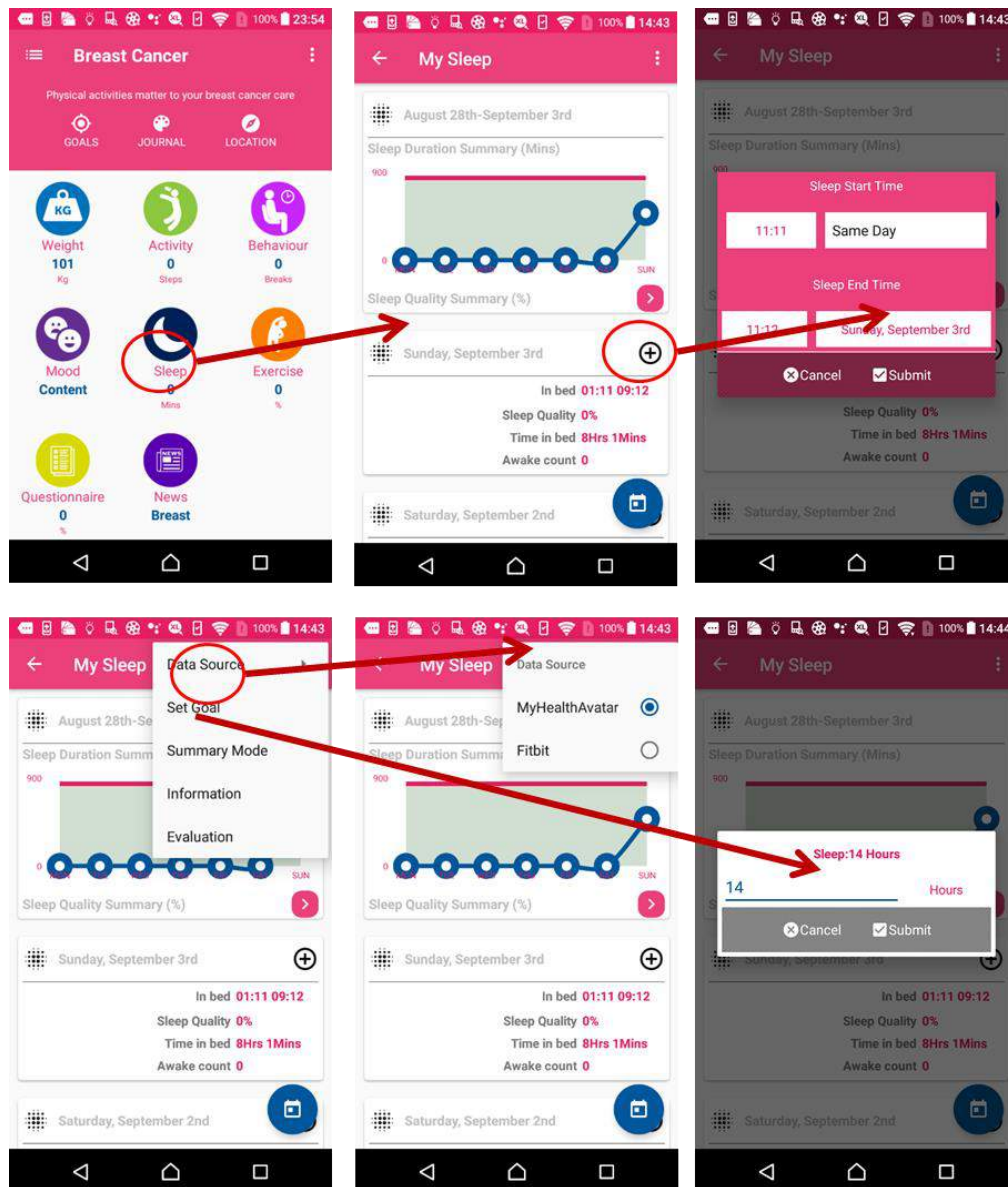


Figure 111: the Sleep suite function

5.2.1.4 Mood Suite

Mood Suite manages the user's mood information. The user can obtain the weekly/monthly mood summary in the Suite. He/she also can use the relax program to relax herself/himself. This function is reference to the requirement in D2.3-[3.2.1. use cases, Health Avatar GUI and 3.11.1. use cases, Lifestyle Monitoring]

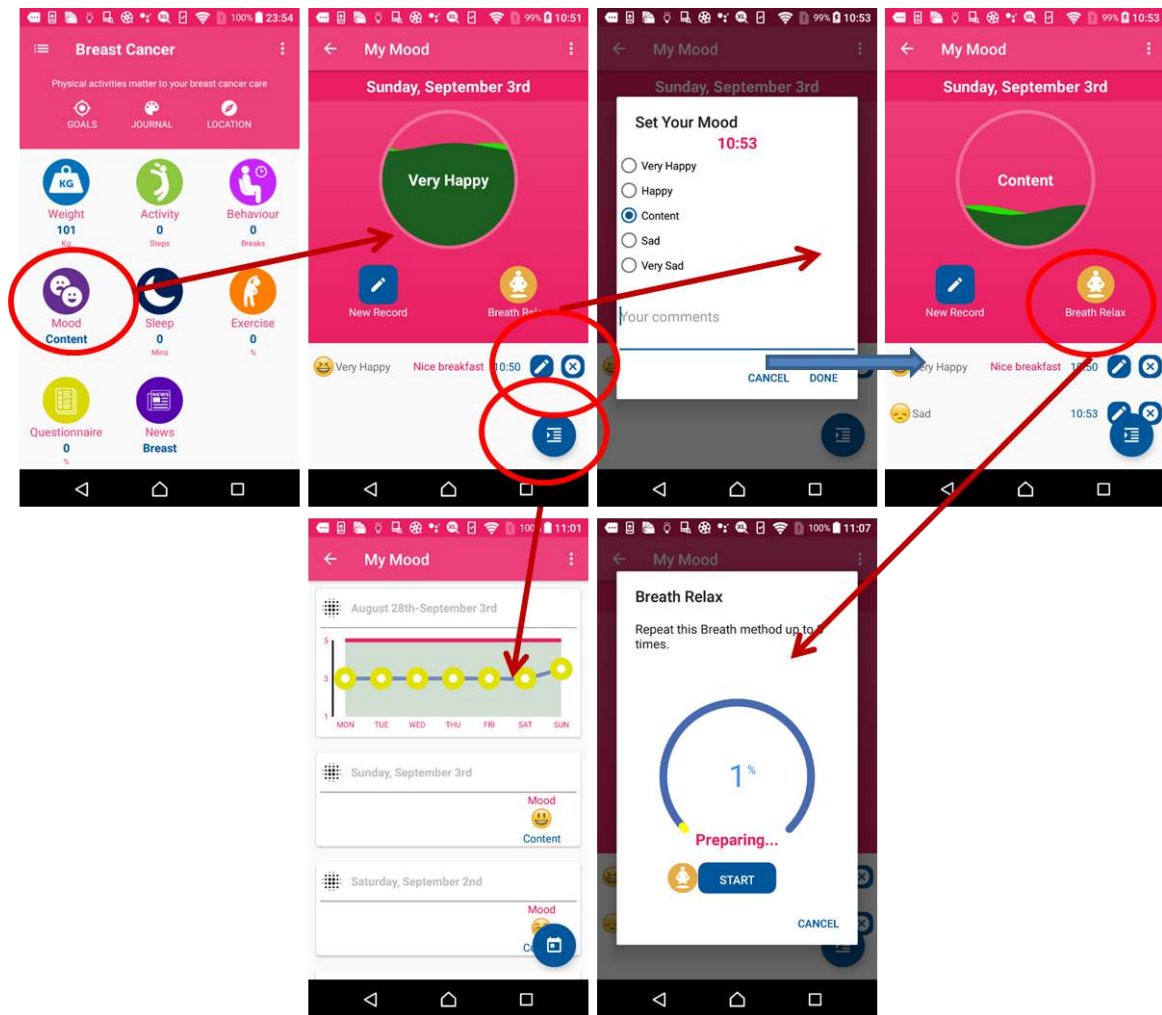


Figure 112: the Mood suite function

5.2.1.5 Sitting Behaviour Suite

Studies have linked excessive sitting with being overweight and obese, some types of cancer like breast cancer. The behaviour suite helps the users monitor how long the sitting time and activity time during a day that due on the current day. User could help reduce long periods of sitting by standing on the train or bus, using the stairs instead of lifts, setting reminders to get up every 30 minutes, stand or walk while on the phone, walk to a co-worker's desk instead of emailing, and walk up the stairs during TV ad breaks. In this suite, Users would receive an alert every 30 mins to take an activity from sitting. Users also can choose the monitor duration to have the alerts. This function is reference to the requirement in D2.3-[3.2.1. use cases, Health Avatar GUI and 3.11.1. use cases, Lifestyle Monitoring]

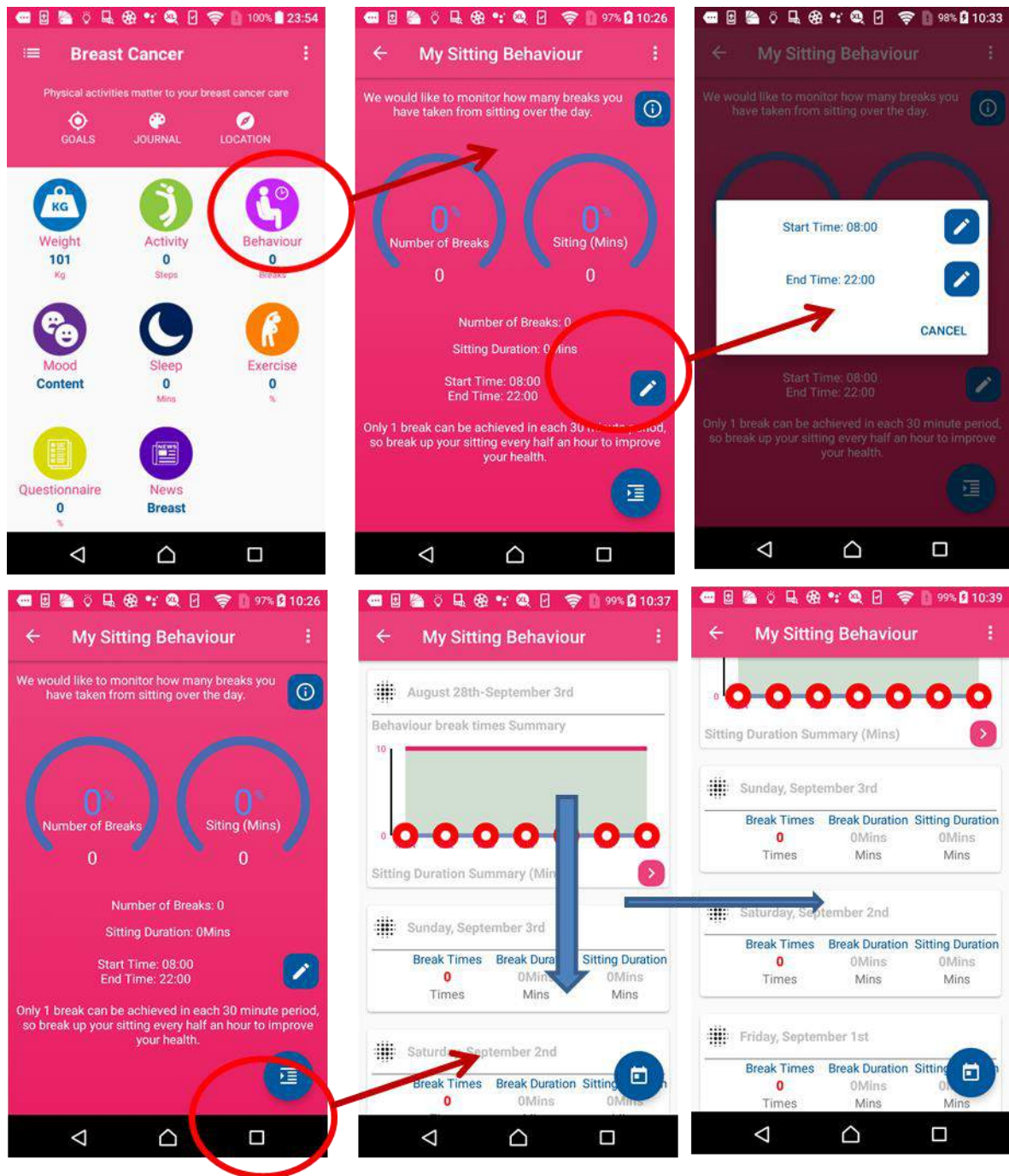


Figure 113: the Sitting Behaviour suite function

5.2.1.6 Exercise Suite

Exercise Suite manages the patient's exercise information after surgery. There are four different exercises for the patient every day. The suite also provides the demo video for the patient. This function is reference to the requirement in D2.3-[3.2.1. use cases, Health Avatar GUI and 3.11.1. use cases, Lifestyle Monitoring].

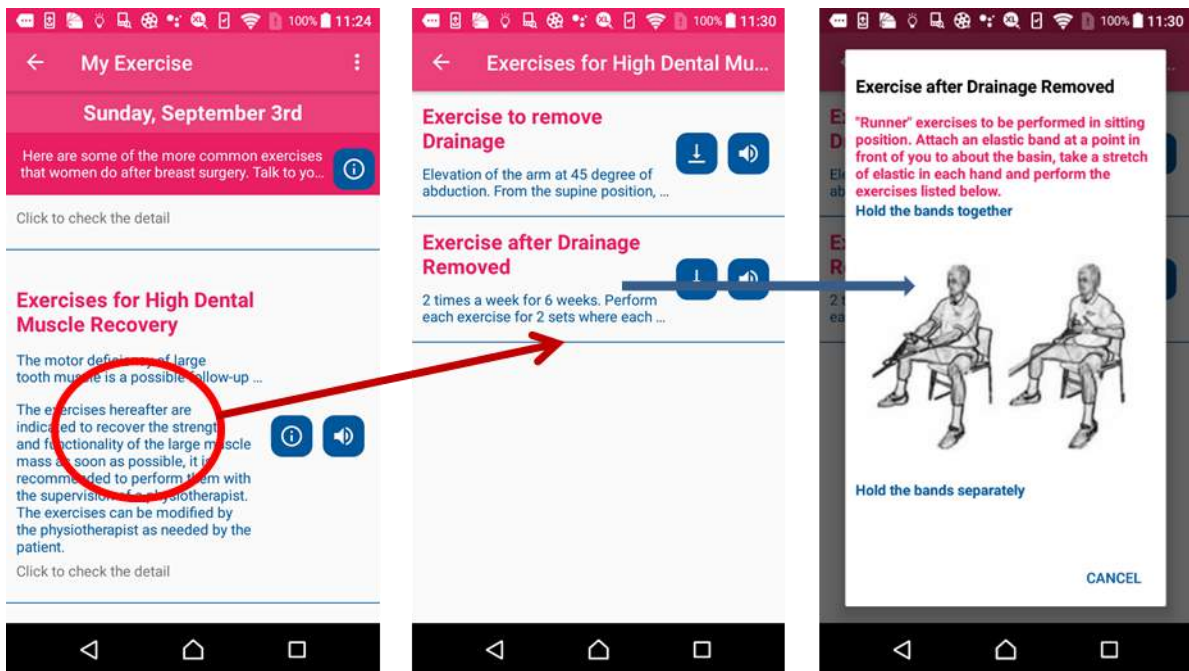


Figure 114: the Exercise suite function

5.2.1.7 Questionnaire Suite

Questionnaire Suite collects the patient's PSA information after surgery in order to let doctor monitor the PSA changes for the patient. Reference to D2.3-[3.2.1. use cases, Health Avatar GUI and 3.10.1. use cases, Health Enquiries].

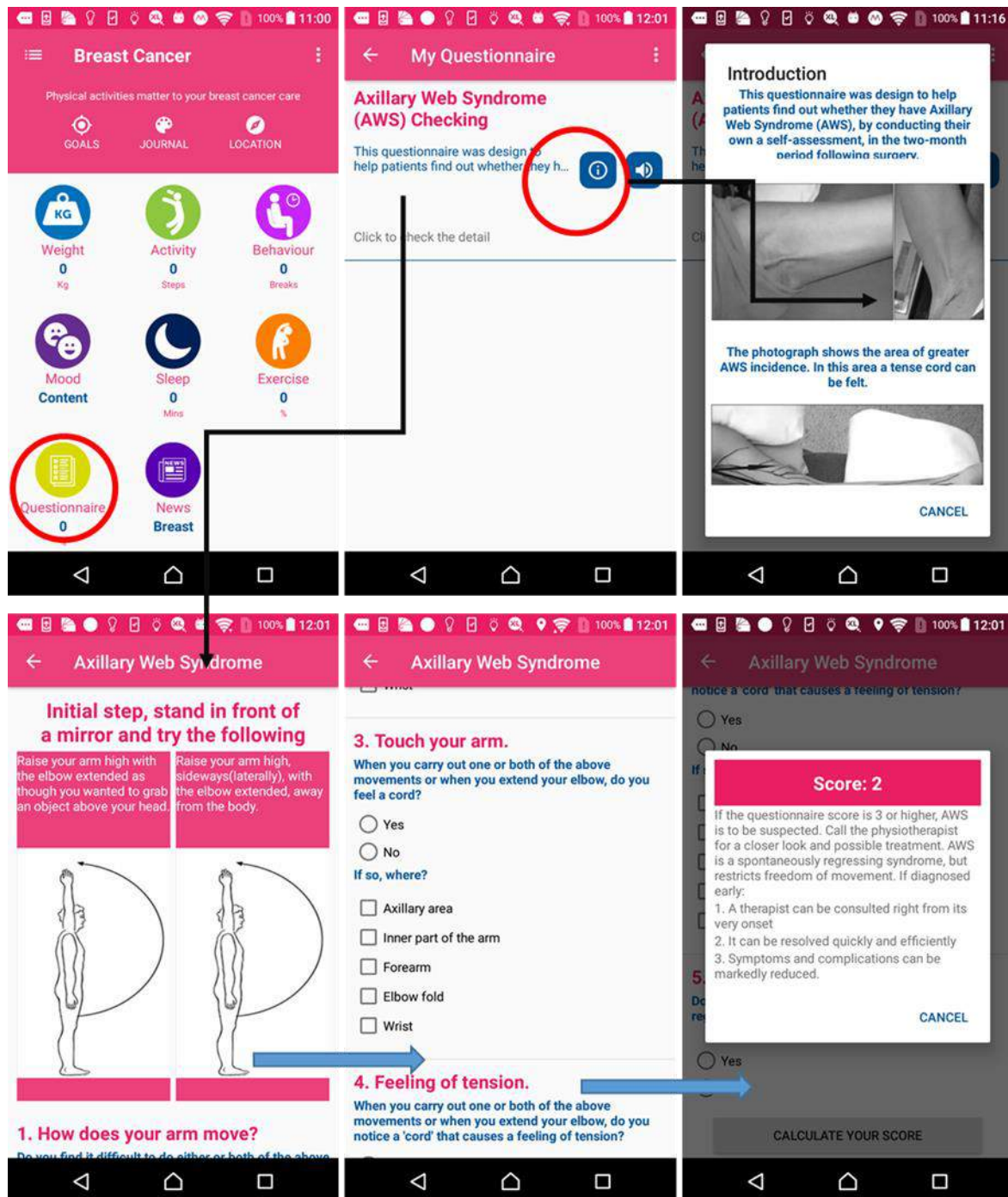


Figure 115: the Questionnaire suite function

5.2.1.8 News Suite

This app provides guidance/news for patients with breast cancer. By using the filter function, the patients can search and choose items on the list and view the details. For breast cancer, patient can access the cancer information. Reference to D2.3-[3.6.1. use cases, Drug Self-Management]

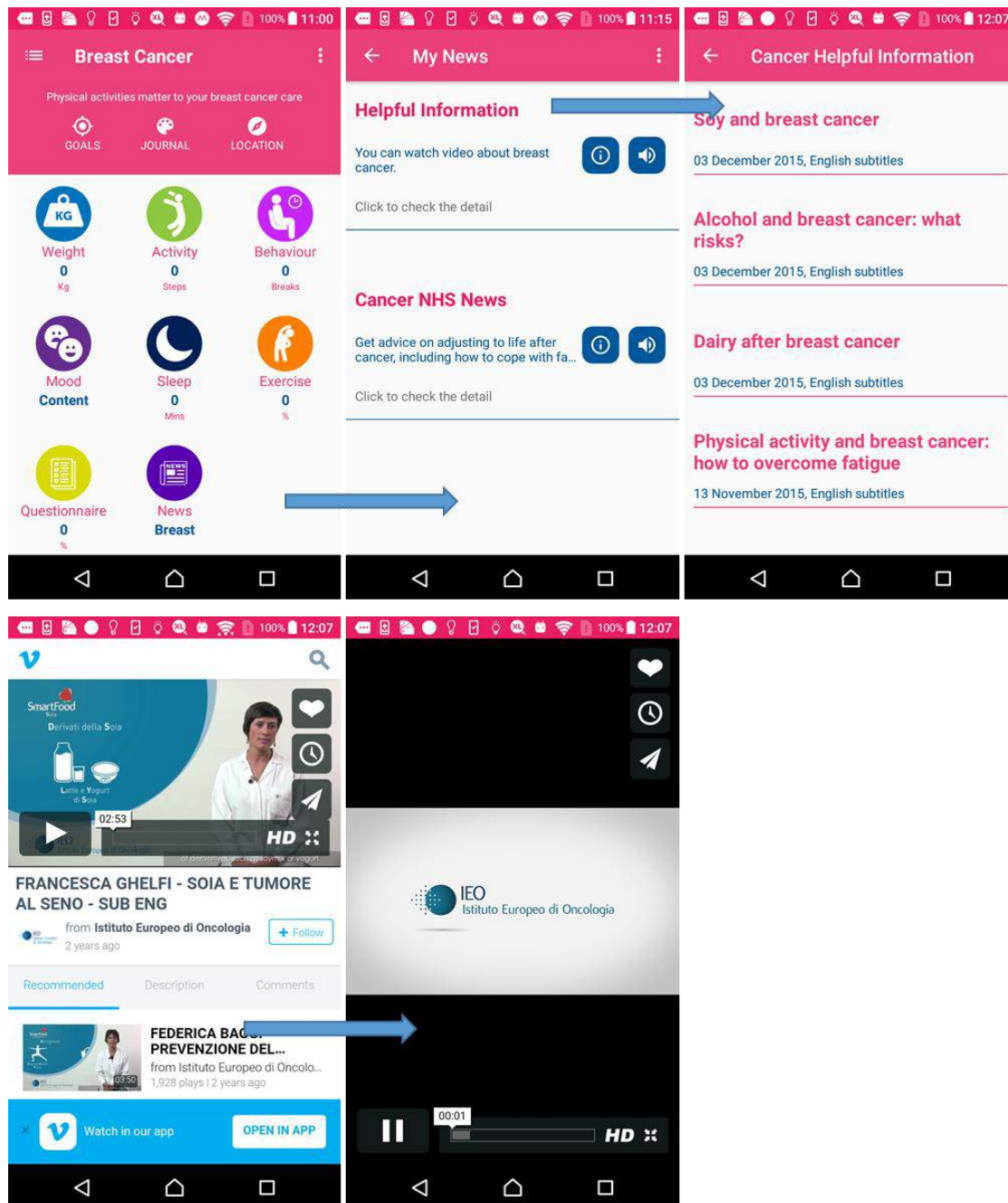


Figure 116: the News suite function

5.2.2. Goals

User can also access to the goal setting page, the journal page and the location from the overview page on both prostate and breast cancer Apps. This function is reference to the requirement in D2.3-[3.13.1. use cases, visualisation scenario].

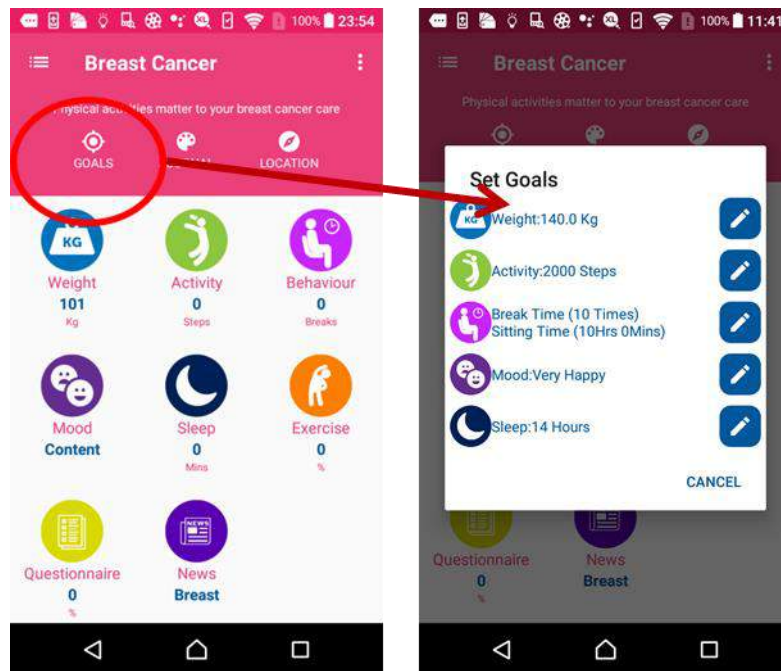


Figure 117: set the Goals

5.2.3. Location

Patient can check the daily location and tracking path by the function. Patient also can annotate the name of a location (e.g., home) and synchronise it with the server. Moreover, the user can turn on/off this function. This function is reference to the requirement in D2.3-[3.4.1. use cases, Motion and Exercise Planning, 3.4.1. use cases, Fatigue management and 3.12.1. use cases, Data Analysis and Data Mining].

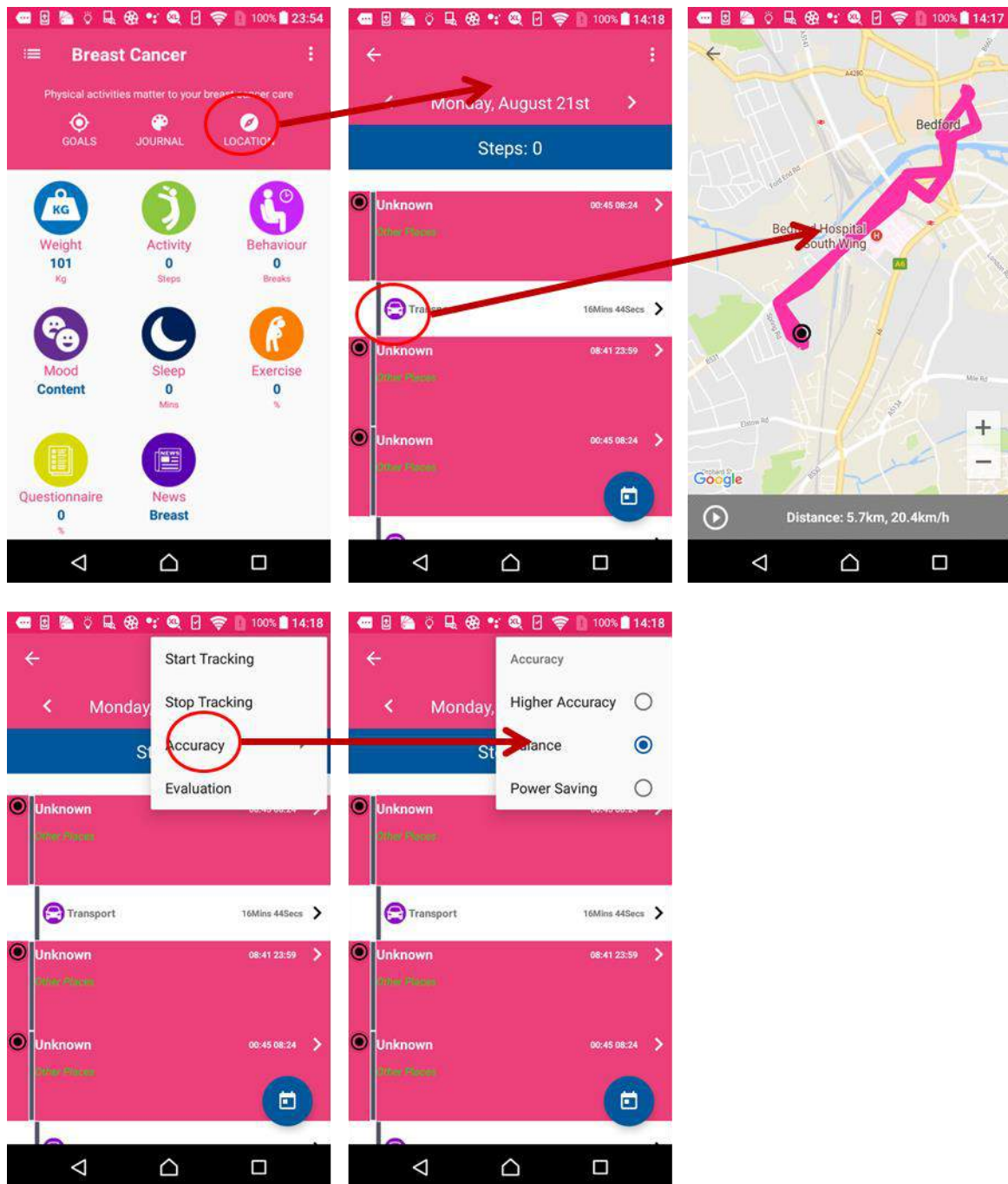


Figure 118: Location Tracking

5.2.4. Journal

A module for patients to communicate with the app and to check the cancer health tips.

5.2.2.1 Control panel

Control panel is easier for patient to manage the their inputs, such as answer for the questionnaire, take/pick phone, type message, voice input and so on. It also can help patient easily access NHS news, Suites function and weight-reduce-program. The output will display the main Journal page.

This function is reference to the requirement in D2.3-[3.13.1. use cases, visualisation scenario and 3.5.1. use cases, Personal Health Information Recommender scenario].

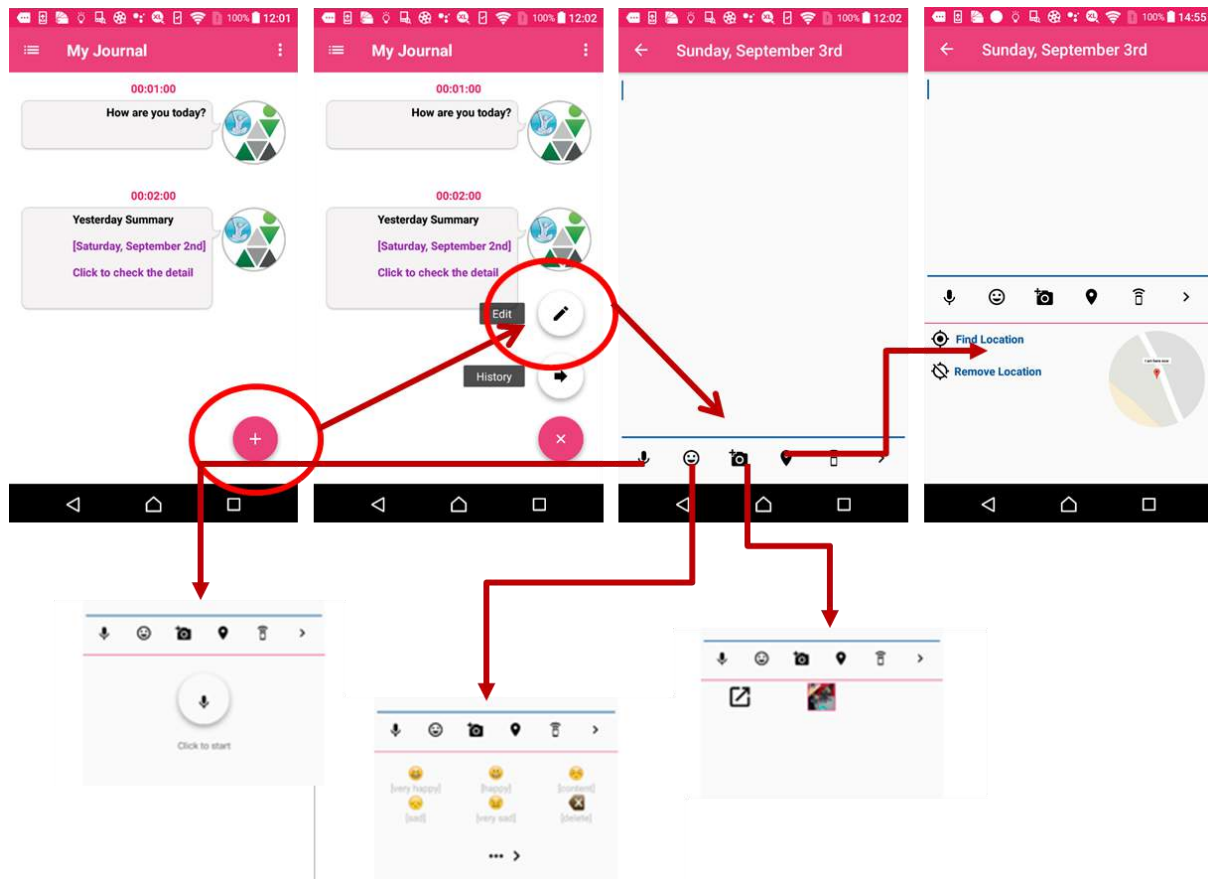


Figure 119: the control panel in Journal

5.2.2.2 Questionnaire

The questionnaire can be accessed quickly and easily via Journal and the patient can turn on/off the function. Reference to D2.3-[3.2.1. use cases, Health Avatar GUI and 3.10.1. use cases, Health Enquiries].

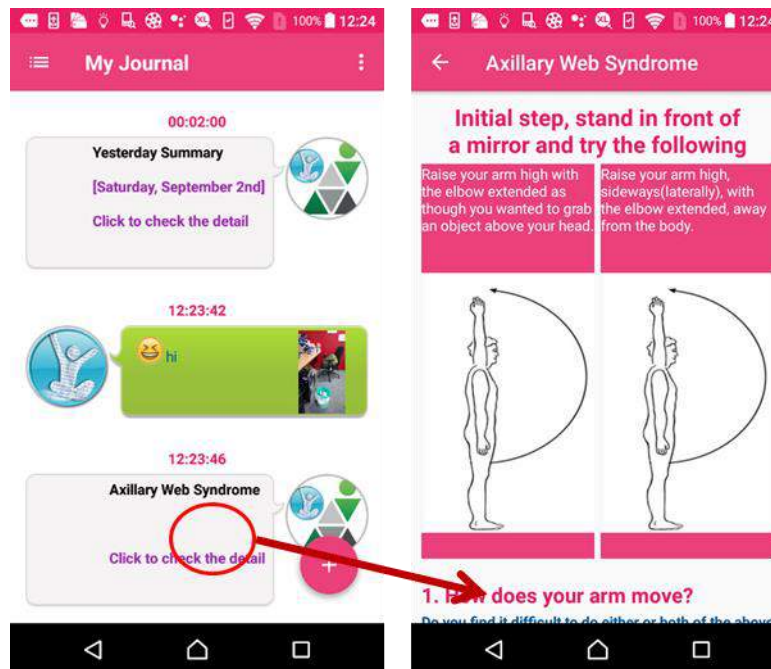


Figure 120: the Mood monitor in Journal

5.2.2.3 Mood Monitor

The users can quickly report their current mood in Journal page. This function is reference to the requirement in D2.3-[3.2.1. use cases, Health Avatar GUI and 3.11.1. use cases, Lifestyle Monitoring]

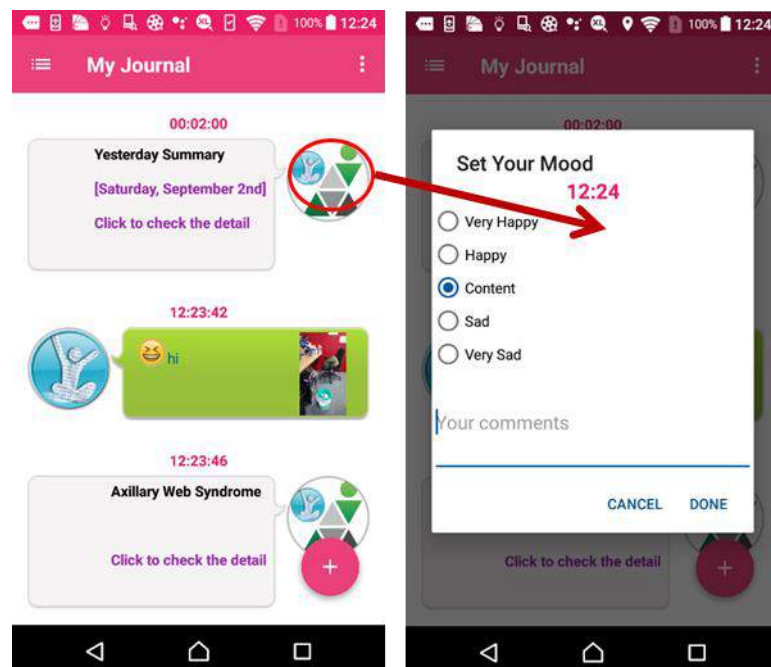


Figure 121: the Mood monitor in Journal

5.2.2.3 Summary

Summary is used to display the daily summary of the measurement value comparing to the goals. The patient can set the summary appears time. This function is reference to the requirement in D2.3-[3.13.1. use cases, visualisation scenario].

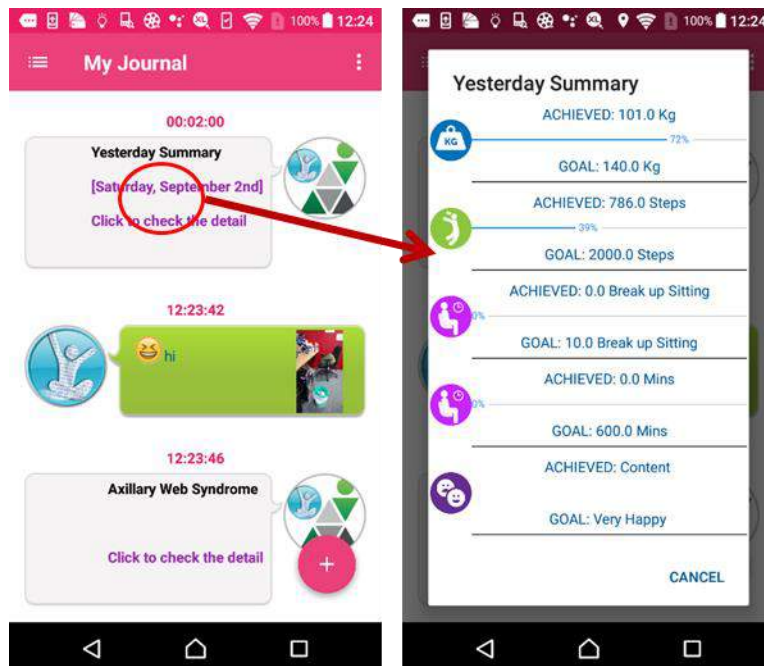


Figure 122: the Summary in Journal

5.2.5. Calendar

A module for patients to review the daily history information of a specific day. Reference to D2.3-[3.13.1. visualisation scenario].

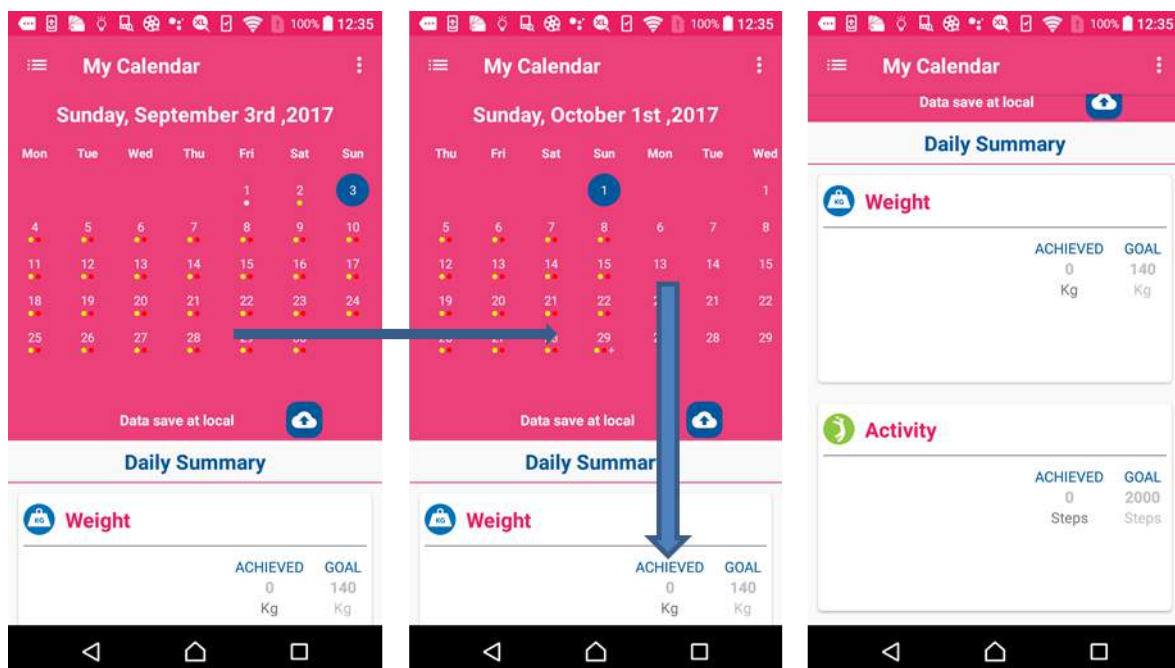


Figure 123: the Summary in Calendar

5.2.6. Profile

The user profile allows the user to enter personal information in different categories. The profile covers common biochemistries information such as weight and height, as well as common information such as name and age. The profile view allows the users to add and upload to the iMC platform important personal information and a detailed medical history.

4.2.6.1 Update the Profile Image

A function for patients to update the profile image. The image can be obtained by the device camera or from the image gallery. The new image will be synchronised with the server.

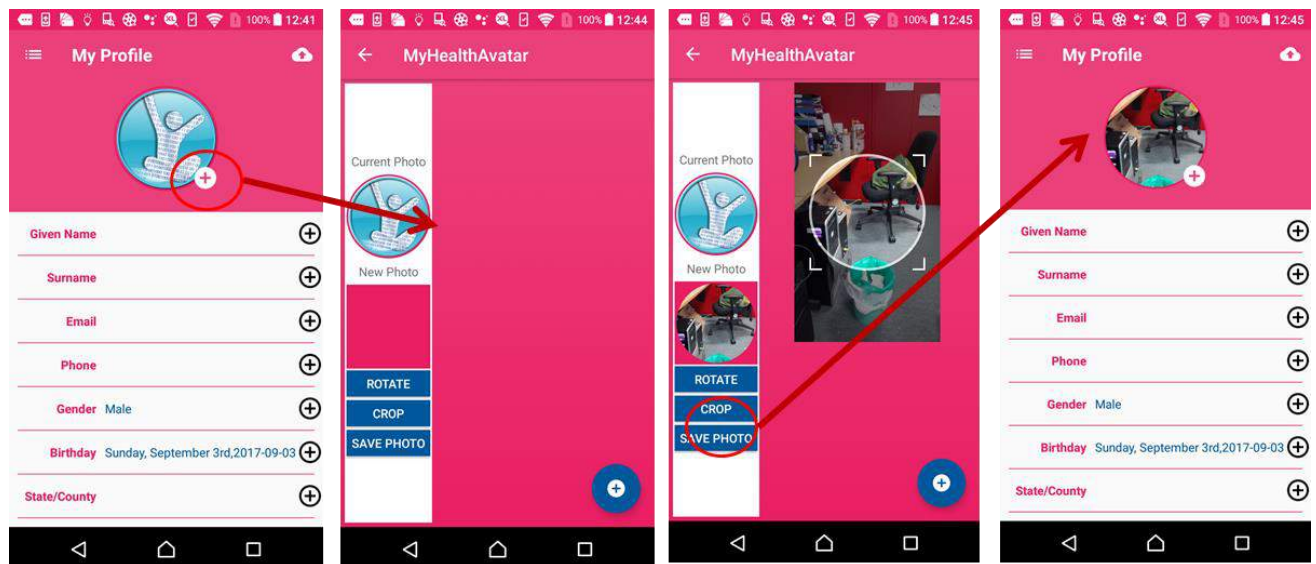


Figure 124: the profile image in Profile

4.2.4.1 Profile

A function for patients to record their general, health and medication profiles. The profile information will be synchronise to the server. This function is reference to the requirement in D2.3-[3.8.1. use cases, Psycho-Emotional Status and Management and 3.9.1. use cases, Family Resilience Evaluation].

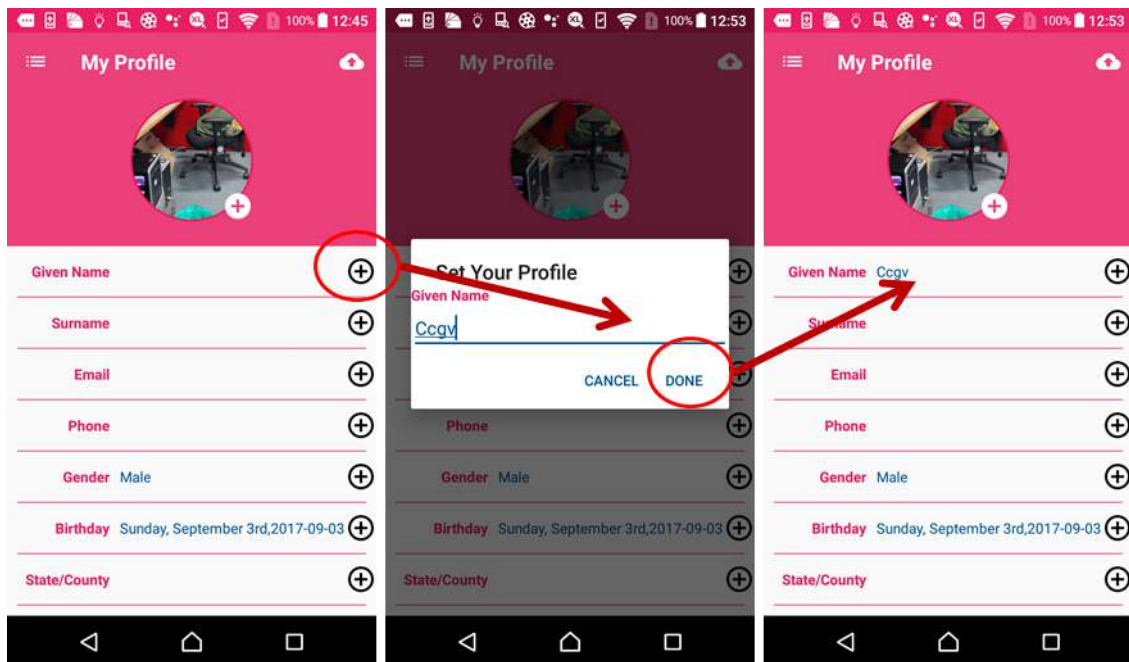


Figure 125: my Profile

5.2.7. Sharing

The social page provides users with the opportunity to search for friends within the platform and send them a friend request. This kind of social interaction is intended to help motivate and encourage users to participate with the platform. The social page allows users to connect with other platform users to form social groups and allow group activities, such as the game. This function is reference to the requirement in D2.3-[3.14.1. use cases, Patient - Doctor and Patient - Patient Interaction].

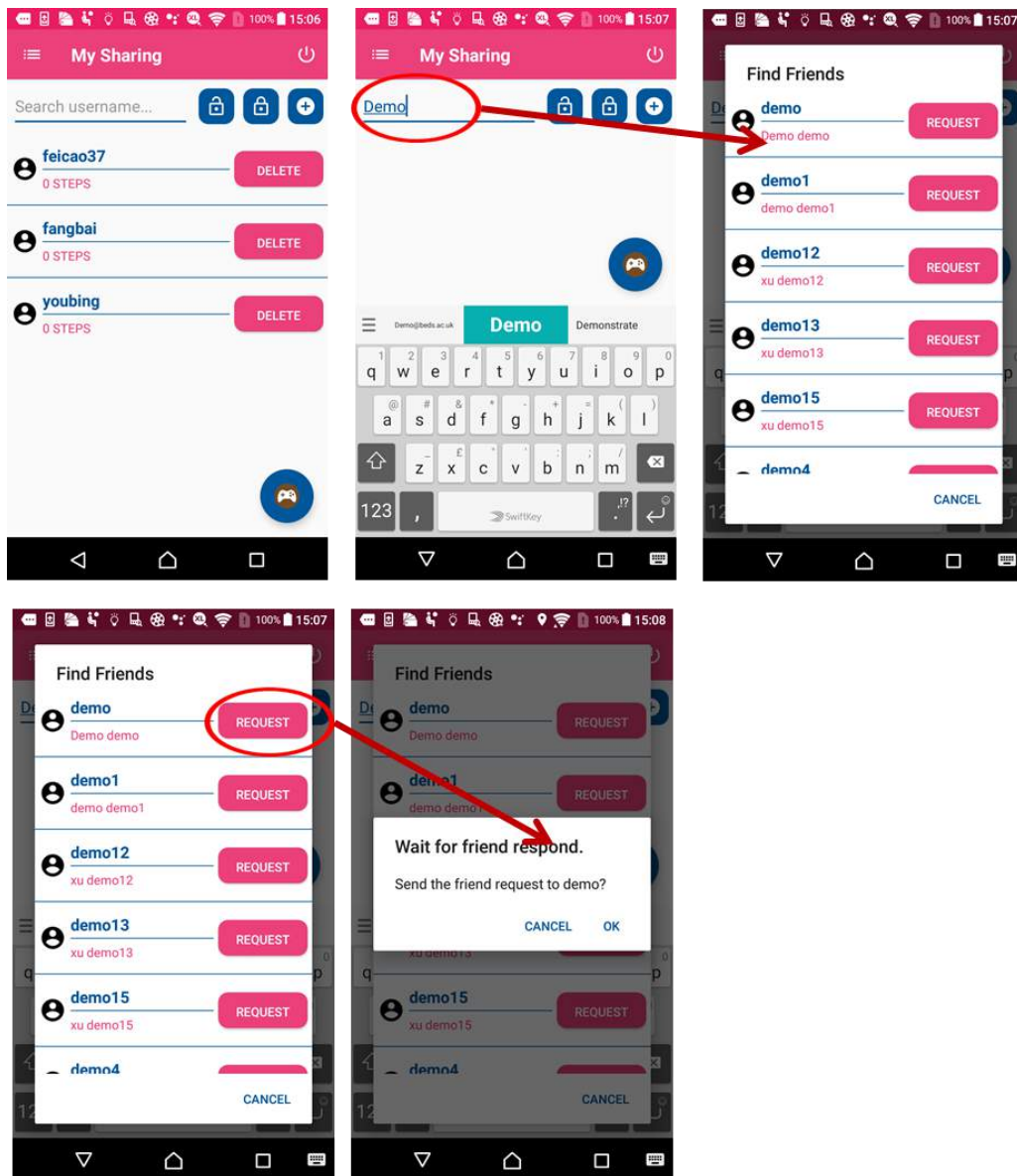


Figure 126: sharing of activity information with friends

5.3. Technical implementation

5.3.1. System concept and internal architecture

In order to provide the function of access, collection and sharing of long term and consistent personal health status data, the MyHealthAvatar for iManageCancer App system has design as:

There are four data sources, such as MHA, Moves, Fitbit and Withings. Only MHA collects data through the MHA App, the other data sources are collected from the third party API.

There are six main function groups in MHA4IMC App, such as Overview (Suite), Journal, Calendar, Profile, Sharing and Settings.

Overview includes various function suites, such as weight, Behaviour, activity, sleep, exercise, questionnaire and cancer news.

Journal is the gate that communicates between patient and App, it includes the diary, summary, mood monitor, daily journal record and questionnaire.

Calendar includes daily history summary.

Profile includes the user profile image and profile data management.

Sharing includes various functions, such as friend list, add new friend, delete friend, friend profile.

Settings includes various functions, such as active/deactivate function in Journal, Calendar and Suites.

5.3.1.1 Overview of the internal architecture

The architecture is shown in Figure 127. The data of the function suite is obtained from Database, if the data source is not MHA, the data will be download from the third party API to the local Database.

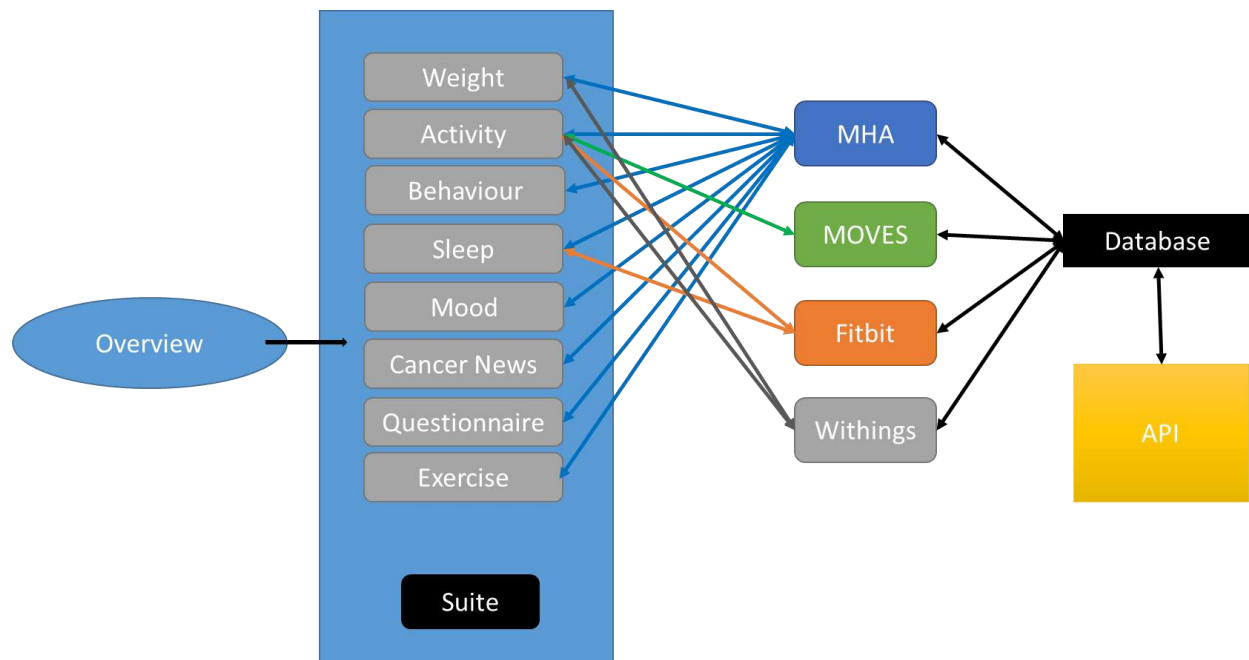


Figure 127: internal architecture of app MyHealthAvatar for iManageCancer

5.3.1.2 Journal internal architecture

The architecture is shown in Figure 128. The data of summary is obtained from the local Database, if the data source is not MHA, the data will be download from the third party API to local Database. The mood data, diary and answer of questionnaire can be saved in the Database and uploaded to the server via MHA API.

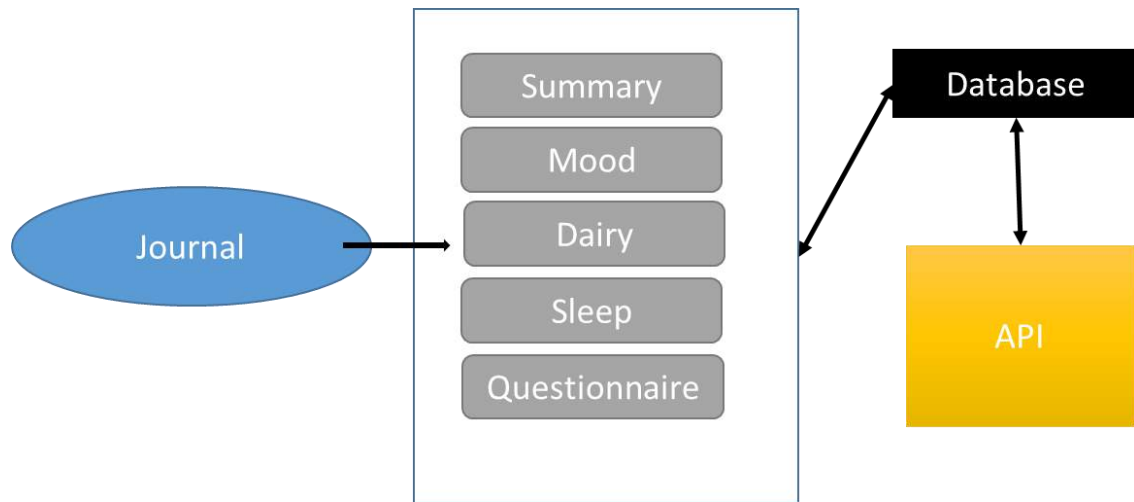


Figure 128: internal architecture of journal functionality of app MyHealthAvatar for iManageCancer

5.3.1.3 Location architecture

The location function architecture is shown in Figure 129. The data of daily location is obtained from the GPS data processing. It includes GPS data collection, location data mining and segment generation. The Location will be saved in Database and upload to the server via API.

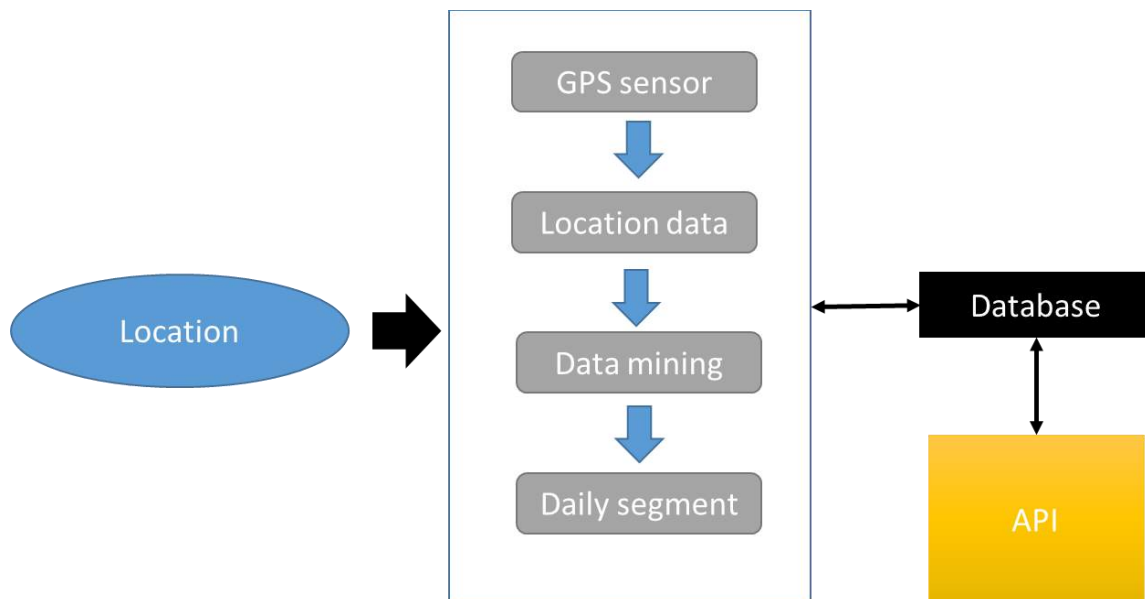


Figure 129: internal architecture of location functionality of app MyHealthAvatar for iManageCancer

5.3.1.4 Calendar architecture

The architecture is shown in Figure 130. The data of daily summary is obtained from Databas, if the data source is not MHA, the data will be download from third party API to local Database.

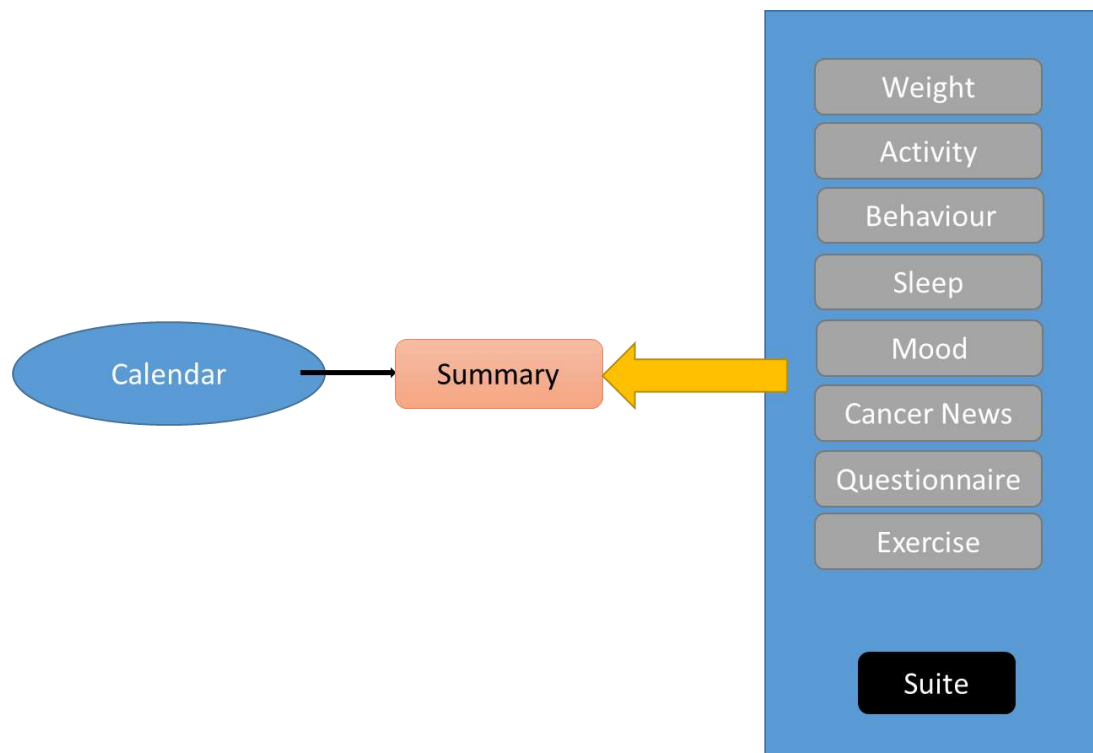


Figure 130: internal architecture of calendar function of app MyHealthAvatar for iManageCancer

5.3.1.5 Profile internal architecture

The architecture is shown in Figure 131. The profile data is updated by MHA or other data source, such as Fitbit and Withings. The data is saved in the local Database and uploaded to the server via MHA API.

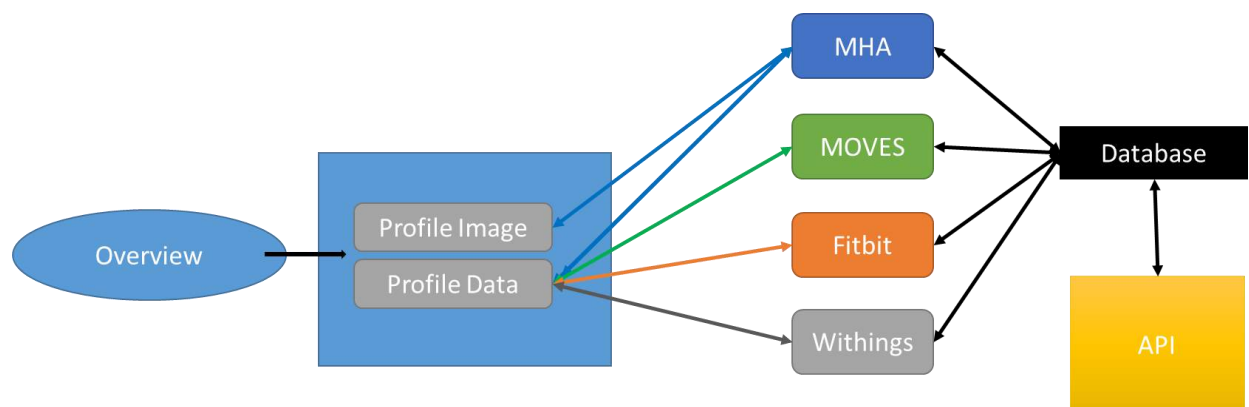


Figure 131: internal architecture of Profile functionality of app MyHealthAvatar for iManageCancer

5.3.1.6 Sharing internal architecture

The architecture is shown in Figure 132.

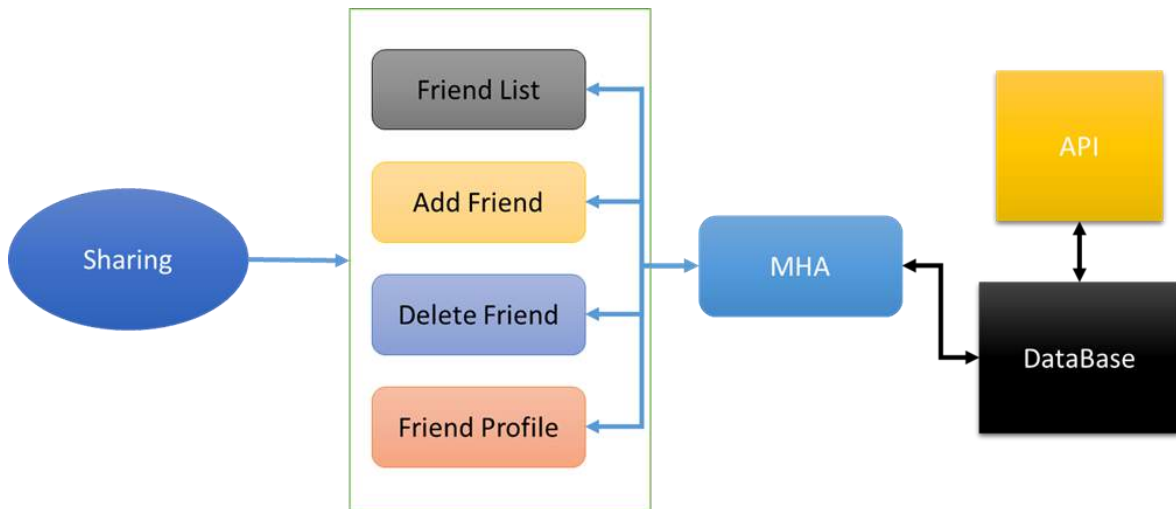


Figure 132: internal architecture of Sharing functionality of App MyHealthAvatar for iManageCancer

5.3.1.7 Setting internal architecture

The architecture is shown in Figure 133. The Settings controls the functionality of other function groups.

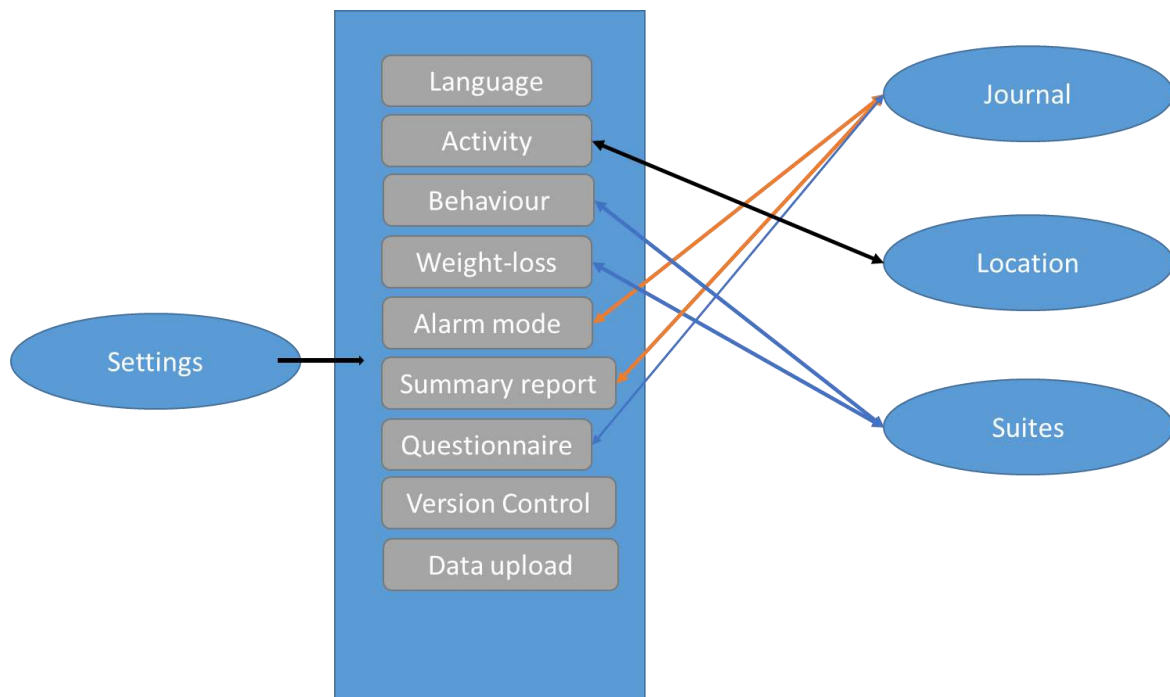


Figure 133: internal architecture of Setting functionality of app MyHealthAvatar for iManageCancer

5.3.1.8 Main Internal Architecture

The architecture is shown in Figure 134.

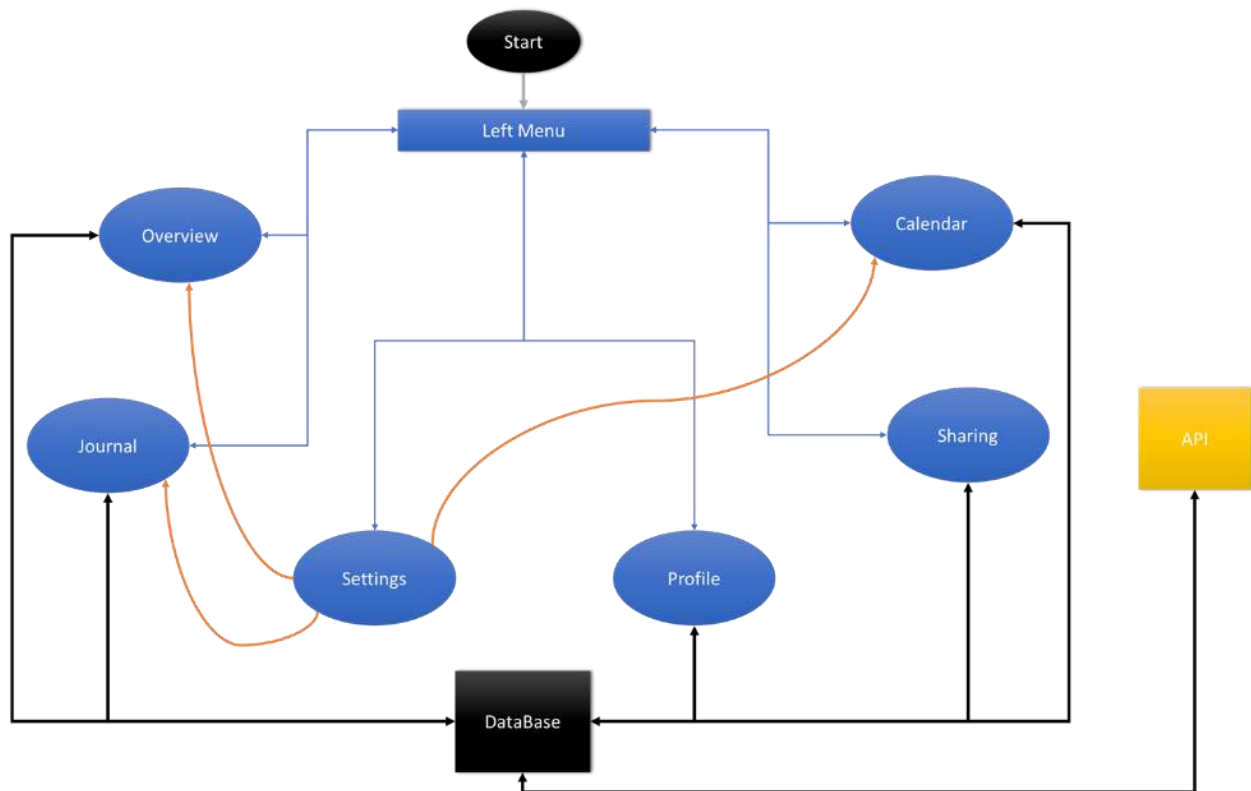


Figure 134: internal architecture of app MyHealthAvatar for iManageCancer

5.3.2. Interfaces

5.3.2.1 API for Profile

- Get measurement profile——download the measurement profile data from server.
Link: <https://myhealthavatar.org/mha/api/v2/user/measurements>
Reference to 5.3.1.5 Profile.
- Get personal profile——download the personal profile data from server.
Link: https://myhealthavatar.org/mha/api/v2/user/personal_information
Reference to 5.3.1.5 Profile.
- Get/Post/Delete full profile——download/upload the data in general, health and medication profiles from/to server.
Link: <https://myhealthavatar.org/mha/api/v3/profile/full>
Reference to 5.3.1.5 Profile.
- Post profile image——upload the profile image to server.
Link: <https://myhealthavatar.org/mha/api/file/upload>
Reference to 5.3.1.5 Profile.

5.3.2.2 API for Journal

- Get/Post daily journal message——download/upload the message data from/to the server

Link: <https://myhealthavatar.org/mha/api/v3/moments>

Reference to 5.3.1.2 Journal.

- Get/Post journal Image—— download/upload the image data from/to server.

Link: <https://myhealthavatar.org/mha/api/v3/moments/image>

Reference to 5.3.1.2 Journal.

5.3.2.3 API for Activity

- Get Summary ——download the activity summary from the server.

Link: [https://myhealthavatar.org/mha/api/v2/user/activities/summary?from=\[start date\]&to=\[end date\]](https://myhealthavatar.org/mha/api/v2/user/activities/summary?from=[start date]&to=[end date]);

Reference to 5.3.1.1 Suite and 5.3.1.4 Calendar.

- Get Withings data (weight)——download the weight data (data source: Withings) from the server.

Link: [https://myhealthavatar.org/mha/api/app/measure/\[date\]](https://myhealthavatar.org/mha/api/app/measure/[date])

Reference to 5.3.1.1 Suite, 5.3.1.4 Calendar and 5.3.1.5 Profile.

- Get Fitbit data (heart rate)——download the heart rate data (data source: Fitbit) from the server.

Link: [https://myhealthavatar.org/mha/api/app/heart/\[date\]](https://myhealthavatar.org/mha/api/app/heart/[date])

Reference to 5.3.1.1 Suite, 5.3.1.4 Calendar and 5.3.1.5 Profile.

- Get fitbit data (sleep)——download the sleep data (data source: Fitbit) from the server.

Link: [https://myhealthavatar.org/mha/api/app/sleep/\[date\]](https://myhealthavatar.org/mha/api/app/sleep/[date])

Reference to 5.3.1.1 Suite.

- Get Activity level——download the activity level from the server.

Link: https://myhealthavatar.org/mha/api/v2/user/active_state

Reference to 5.3.1.1 Suite and 5.3.1.2 Journal.

- Get activity optimal values——download the activity optimal values from the server.

Link: [https://myhealthavatar.org/mha/api/v2/user/activities/segments?from=\[start date\]&to=\[end date\]](https://myhealthavatar.org/mha/api/v2/user/activities/segments?from=[start date]&to=[end date])

Reference to 4.3.1.1 Suite.

5.3.2.4 API for Location

- Get/Post name and category of the place——download the place name and category/ upload the changes of the place.

Link: https://myhealthavatar.org/mha/api/v2/user/my_places"

Reference to 5.3.1.2 Journal and 5.3.1.3 Location

- Get foursquare place list——download the foursquare place data from the foursquare data server.

Link: https://api.foursquare.com/v2/venues/search?client_id=%s&client_secret=%s&v=20130815&ll=%s&radius=100

Reference to 5.3.1.2 Journal and 5.3.1.3 Location

- Get Location Segment——download the location segments (data source: Moves) from the server.

Link: [https://myhealthavatar.org/mha/api/v2/user/activities/segments?date=\[date\]](https://myhealthavatar.org/mha/api/v2/user/activities/segments?date=[date]);

Link: [https://myhealthavatar.org/mha/api/v2/user/activities/segment?from=\[start date\]&to=\[end date\]](https://myhealthavatar.org/mha/api/v2/user/activities/segment?from=[start date]&to=[end date]);

Reference to 5.3.1.2 Journal and 5.3.1.3 Location

5.3.2.7 API for Goals

Get/post goals

Link: <https://myhealthavatar.org/mha/api/v2/user/goals>

Reference to 5.3.1.1 Suite.

5.3.2.8 API for Sharing

- Get user friend list

Link: <https://myhealthavatar.org/mha/api/friend/list>

Reference to 5.3.1.6 Sharing.

- Get user friend info

Link: [https://myhealthavatar.org/mha/api/friend/info?username=\[username\]](https://myhealthavatar.org/mha/api/friend/info?username=[username])

Reference to 5.3.1.6 Sharing.

- Get add friend request

Link: <https://myhealthavatar.org/mha/api/friend/request>

Reference to 5.3.1.6 Sharing.

- Approve users as friend

Link: [https://myhealthavatar.org/mha/api/friend/approve?username=\[username\]](https://myhealthavatar.org/mha/api/friend/approve?username=[username])

Reference to 5.3.1.6 Sharing.

- Add/remove user as friend

Link: [https://myhealthavatar.org/mha/api/friend/add?username=\[username\]](https://myhealthavatar.org/mha/api/friend/add?username=[username])

Link: [https://myhealthavatar.org/mha/api/friend/delete?username=\[username\]](https://myhealthavatar.org/mha/api/friend/delete?username=[username])

Reference to 5.3.1.6 Sharing.

5.3.2.9 API for News

- Get NHS news

Link: <https://myhealthavatar.org/nhs/news>

Reference to 5.3.1.2 Journal.

- Get cancer news

Link: <https://myhealthavatar.org/imc/news>

Reference to 5.3.1.1 Suite.

5.3.2.10 API for evaluation and audit

Post IMC Evaluation Questionnaire

Link: <http://live.ccg.gov.uk/mha/app/questionnaire>

Post IMC Audit

Link: <https://www.iphr.care/mha/api/iphr/audit>

5.3.2.11 API for upload

Post IMC data to the server

Link: <https://www.iphr.care/mha/api/app/data/update>

5.3.2.12 API for version control

- Get IMC version

Link: https://www.iphr.care/mha/api/app/update/get?name=mha_imc

- Get IMC new version App

Link: <https://www.iphr.care/mha/api/app/update/download/>

5.3.3. Database

5.3.3.1 Location Table

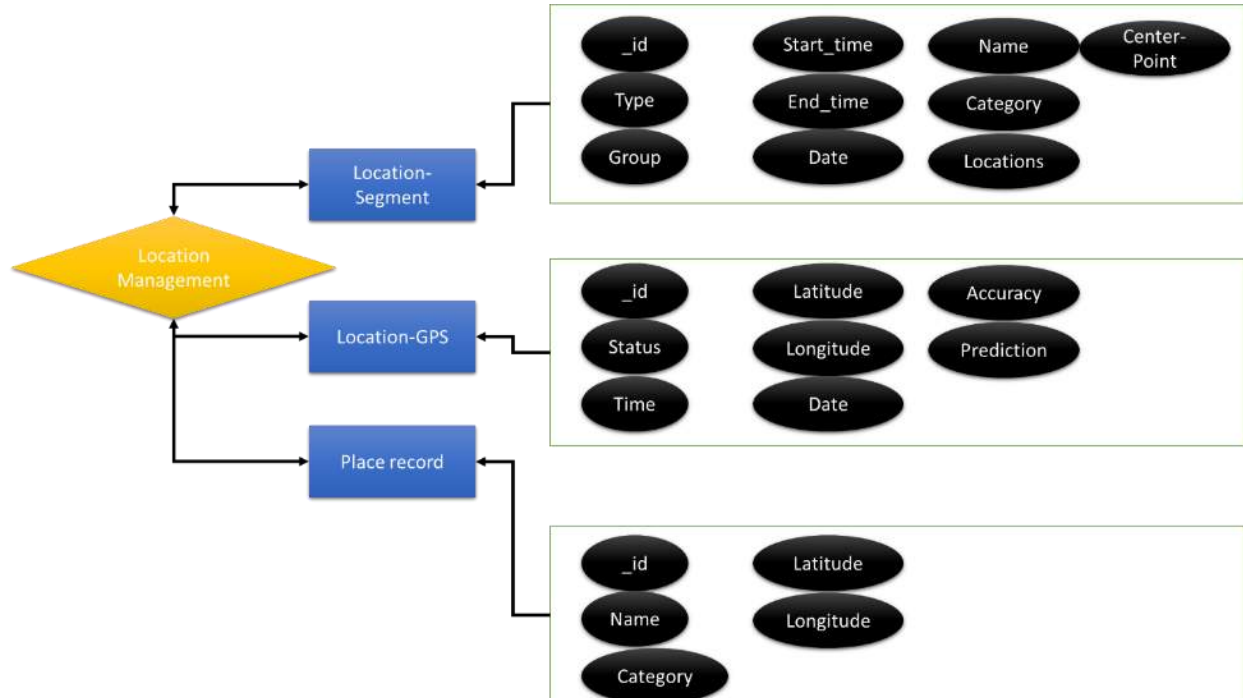


Figure 135: Location table

5.3.3.2 Sharing Table

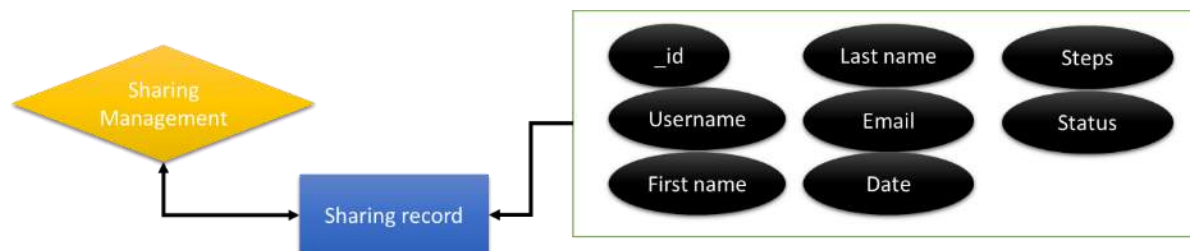


Figure 136: Sharing table

5.3.3.3 Journal Table

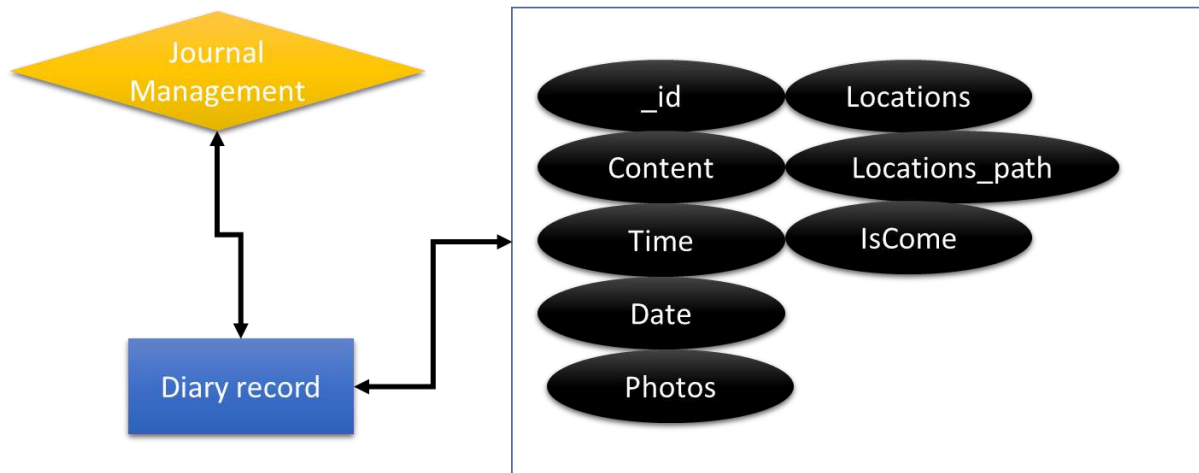


Figure 137: Journal table

5.3.3.4 Questionnaire Table

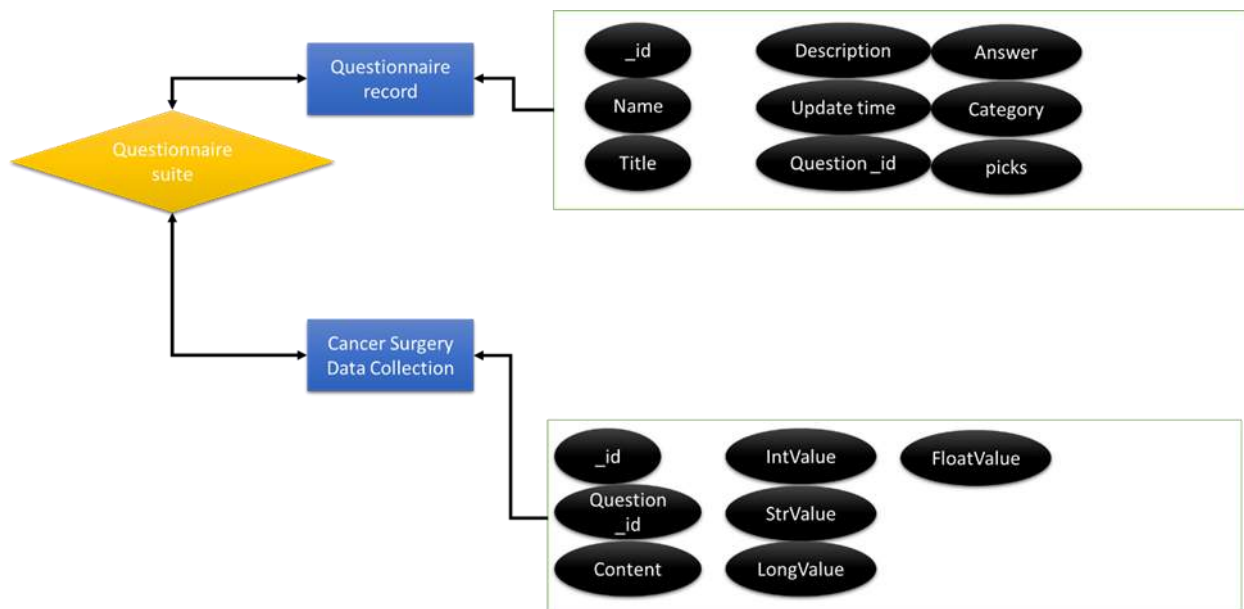


Figure 138: Questionnaire table

5.3.3.5 Breast Cancer Exercise Table

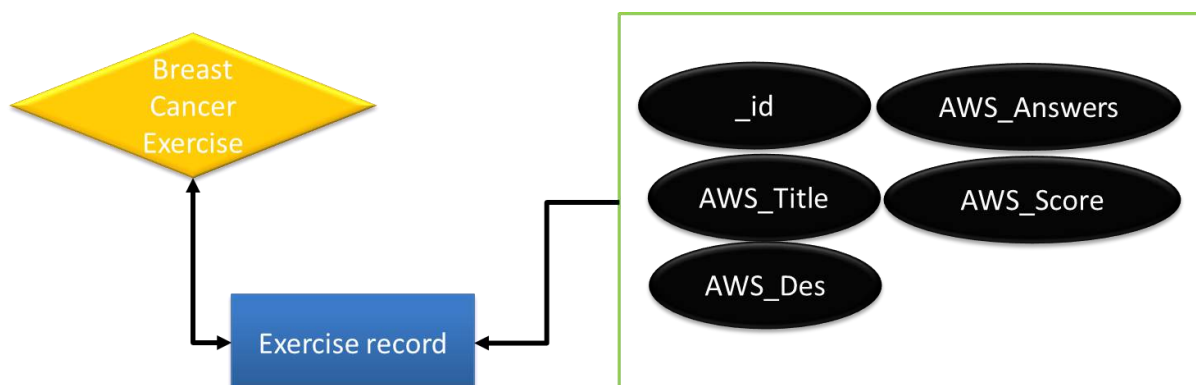


Figure 139: Breast Cancer Exercise table

5.3.3.6 Mood Table

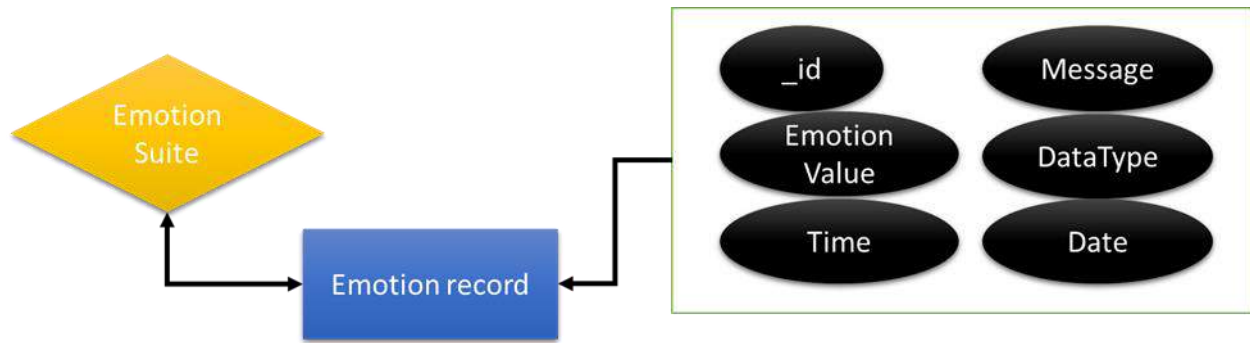


Figure 140: Emotion table

5.3.3.7 Weight Table

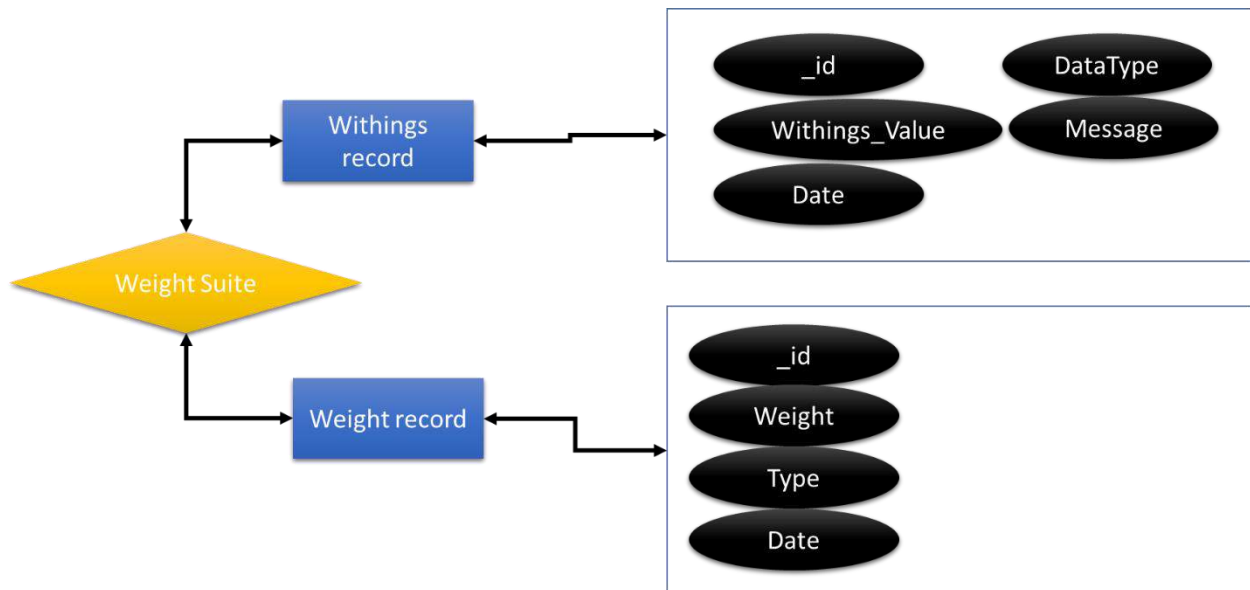


Figure 141: Weight table

5.3.3.8 Activity Table

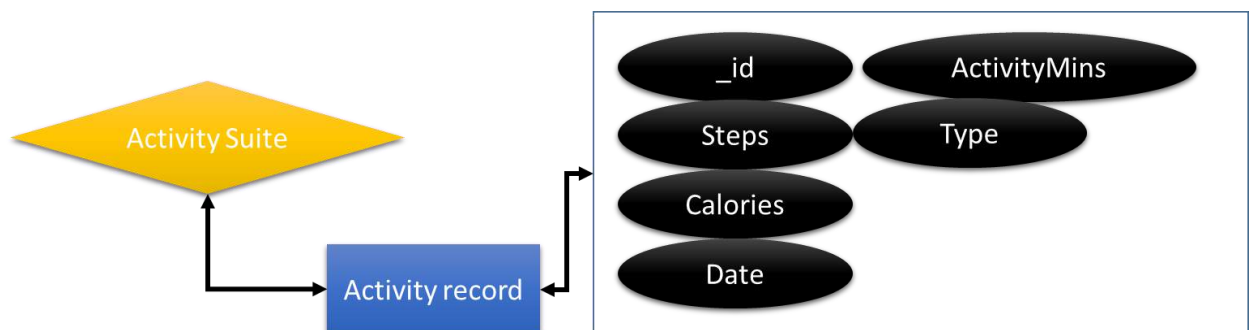


Figure 142: Activity table

5.3.3.9 Sleep Table

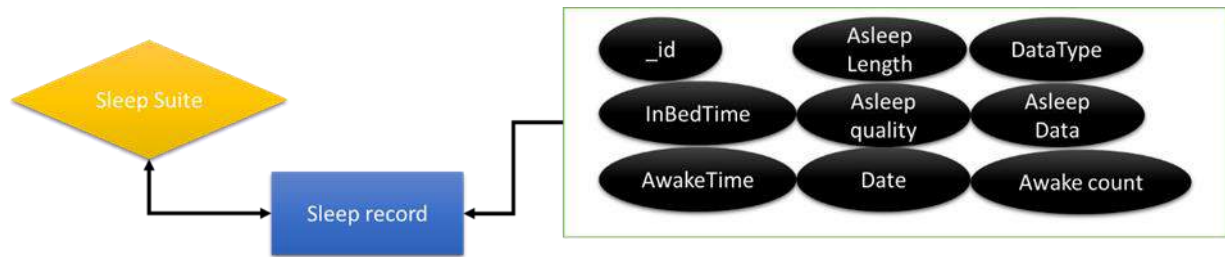


Figure 143: Sleep table

5.3.3.10 Goals Table

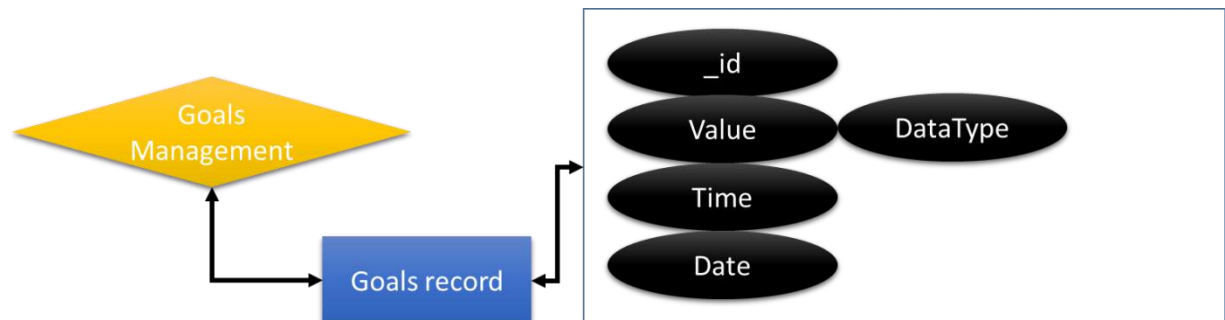


Figure 144: Goals table

5.3.3.11 Weight-loss Table

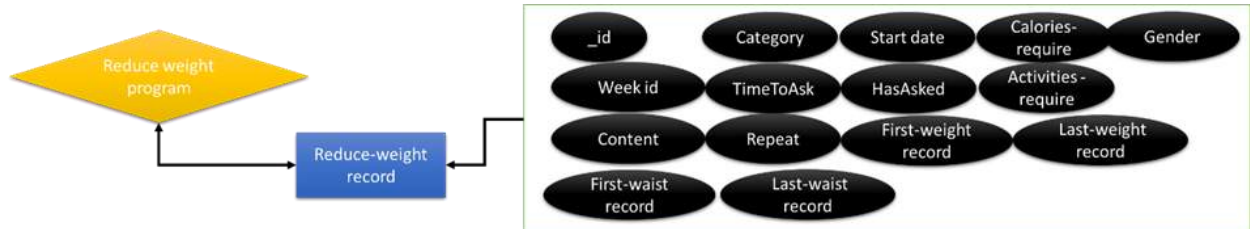


Figure 145: Weight-loss table

5.3.3.12 Evaluation Table

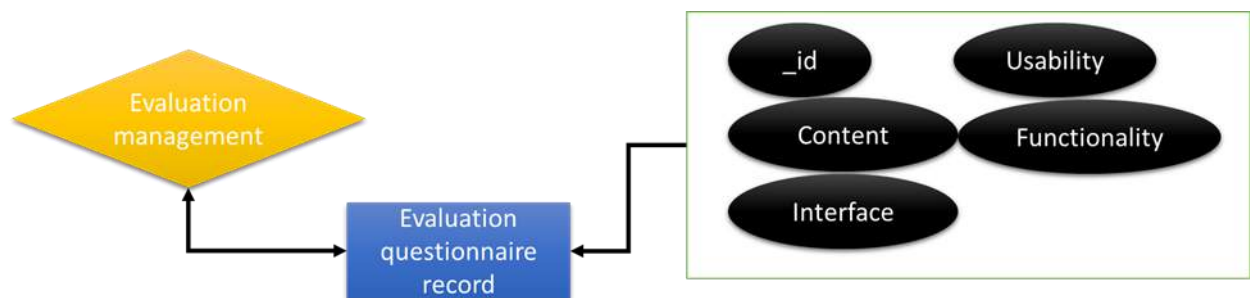
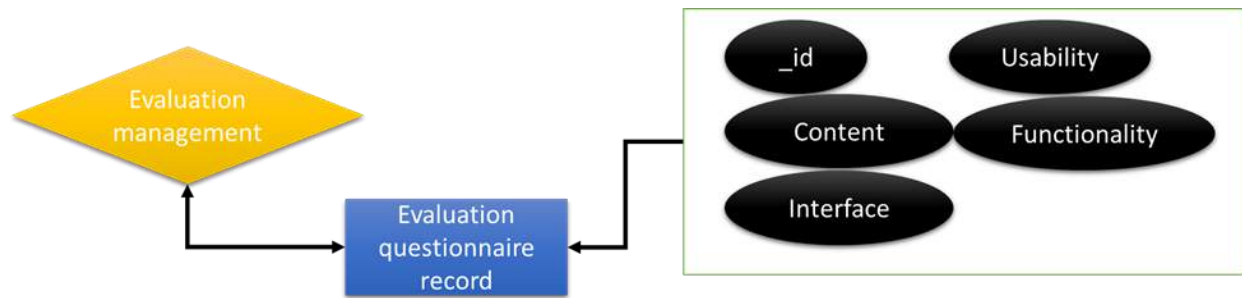


Figure 146: Evaluation table



5.3.4. Data protection

User will need to explicitly consent the terms and conditions, and the privacy statement in order to create an account and access/store data. Users are not required to provide real identifiable information of themselves, and the app facilitate users' with this privacy preserving mechanism.

The data generated within the MyHealthAvatar for iManageCancer is stored in a physically and technically secure server, provided by a commercial company called Linode (<https://www.linode.com/>). The data generated from the app is stored in databases hosted by Linode, and the services employs industry-standard technical safeguards, such as cryptographic key-based authentication controls on servers which house personal information, browser-based SSL (secure sockets layer) encryption and other storage system access control mechanisms to protect the integrity of internal databases and prevent unauthorized access.

For data security and privacy protect, the application references the ISO/IEC 27002:2013 code of practices for comprehensive information security control and risk management. The system will incorporate controls, which aligned to ISO/IEC 27018:2014 best practices for protection of Personal Identifiable Information (PII). Service Organization Controls (SOC) 2 (security controls) will guide the trust principles of security, availability and processing integrity, which allows the system processing to be accurate, complete, fast and authorized.

Only user will have full access to the data generated by himself/herself within the app, and doctors only have access to patient data with patient's explicit informed consent. Furthermore, patients may choose to enter a game within the app where they compete with other users in a physical activity (step count) challenge. Patients can add other users to their game by searching for other users' names within the app and then send a request to compete in the activity competition with them. The peer user would then need to accept this request before any activity data is shared, but no personal information other than the number of steps accumulated will be visible to other users.

All data generated by the app belongs to the user, and if the user no longer wants to use the app, they can close their account and request that all of their data generated within the app is destroyed.

For authentication, the Universal Second Factor (U2F) protocol is proposed, which allows the system to augment the security of password infrastructure by adding a strong second factor to user login. Google Authenticator [6] on Android mobile device could serve as the second factor, which is a free application which gains popularity recently.

The integration utilizes JavaScript Object Notation (JSON) based Identity protocol in its security model, the JSON data format carries information with defined ways to encode tokens, symmetric / asymmetric keys and digital signatures. JSON Web Token (JWT) specification defines the way to encode token in this JavaScript format, these lightweight tokens can be used in HTTP headers

and query strings. JSON Web Key (JWK) and JSON Web Signature (JWS) specifications define the way on encryption keys and digital signatures.

Federation and Single Sign On (SSO) is guided by OpenID Connect, which is essentially the third version of OpenID (however it is a complete rewrite, and not compatible with previous versions). OpenID connect is an HTTP-based protocol that provides SSO, it is built atop of OAuth 2 and achieve higher Levels of Assurance (LoA) compared to other similar purpose protocols like SAML and WS-Federation.

OAuth 2 is a “meta protocol” which provides a useful foundation for other protocols (e.g. OpenID Connect, NAPS and UMA). OAuth is very important in API system because it features:

- Delegated access to third parties applications
- Reduce password sharing between users and third-parties
- Enable the revocation of access

OAuth 2 is proposed to be used by the system for delegated access to the API.

HTTPS is always preferred in all system and third parties web applications, system maintains a list of allowed third parties, which includes key information:

- Callback URL which used by OAuth 2 and SSO
- Third-parties contact and permissions (controlled by team)

All subsystems have their own security model, and only give the least permission necessary to other subsystems and / or users. The permission is assigned and controlled by specific members of team, and will be audited and reviewed.

5.4. User manual

5.4.1. Starting the MyHealthAvatar-iManageCancer App

Action: Find the launcher logo for the MHA-iMC app in the Apps of your mobile device and press on it.

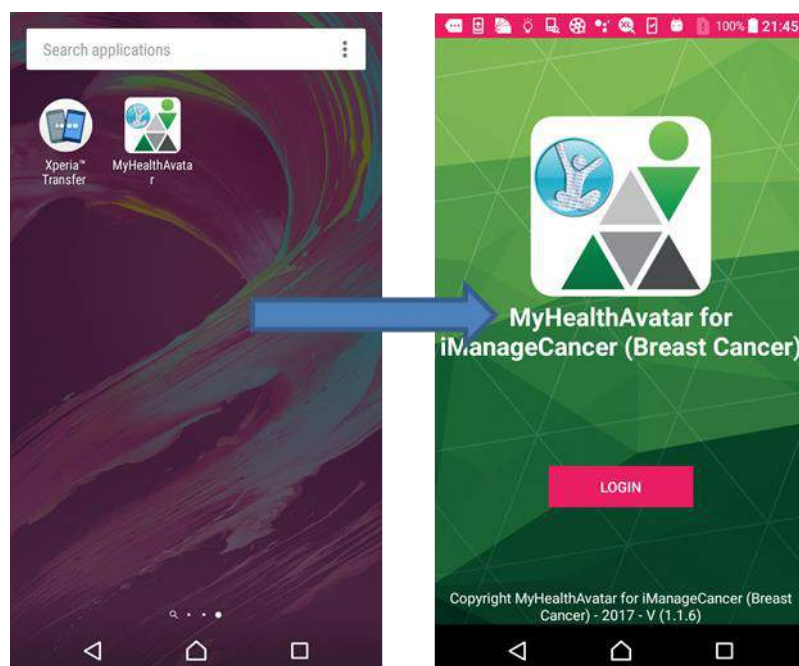


Figure 147: start MyHealthAvatar for iManageCancer

Result: The start page of the app is displayed.

5.4.2. Accessing the Main page from the Start page

Action: Insert your account and password and press on the Sign in button on the start page of the app.

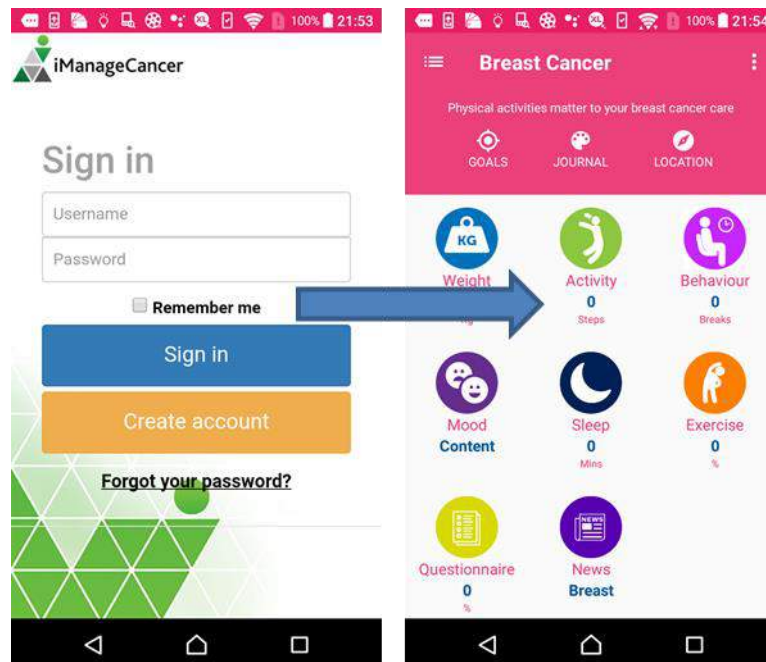


Figure 148: sign in MyHealthAvatar for iManageCancer

Result: After the successful login, the overview in the main page is displayed on the screen.

5.4.3. Accessing the Location page from the Start page

Action 1: Click the “Location” on the main page.

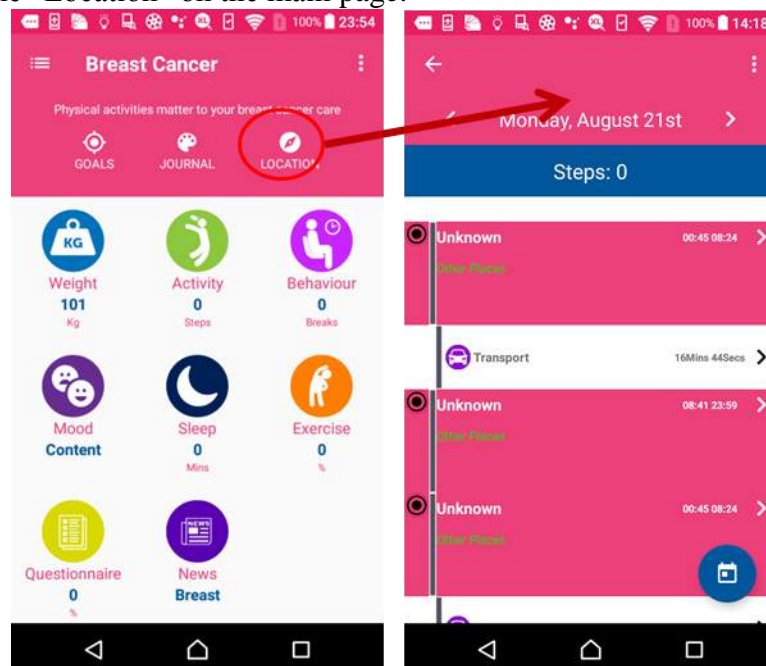


Figure 149: Location page overview

Action 2: Click the segment item and check the location on the map

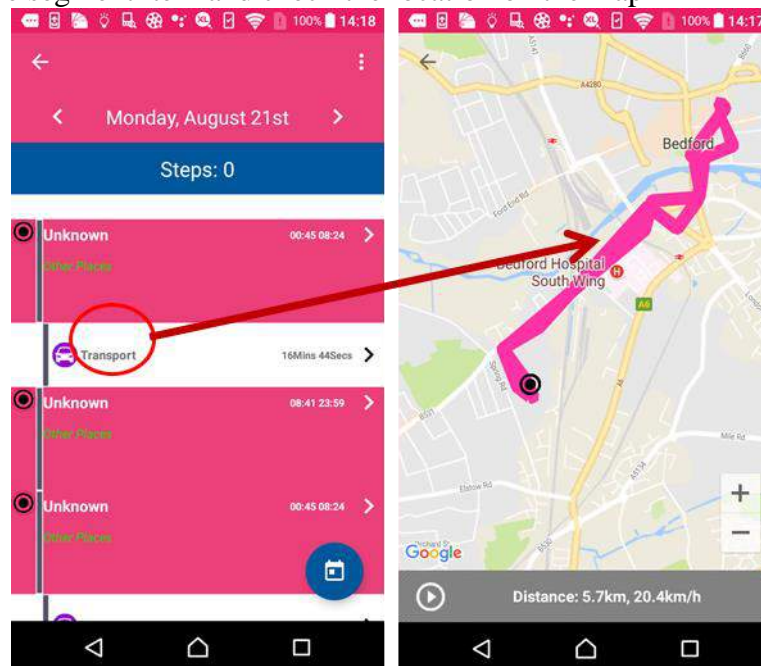


Figure 150: Location information on the map

Action 3: Annotate the segment item

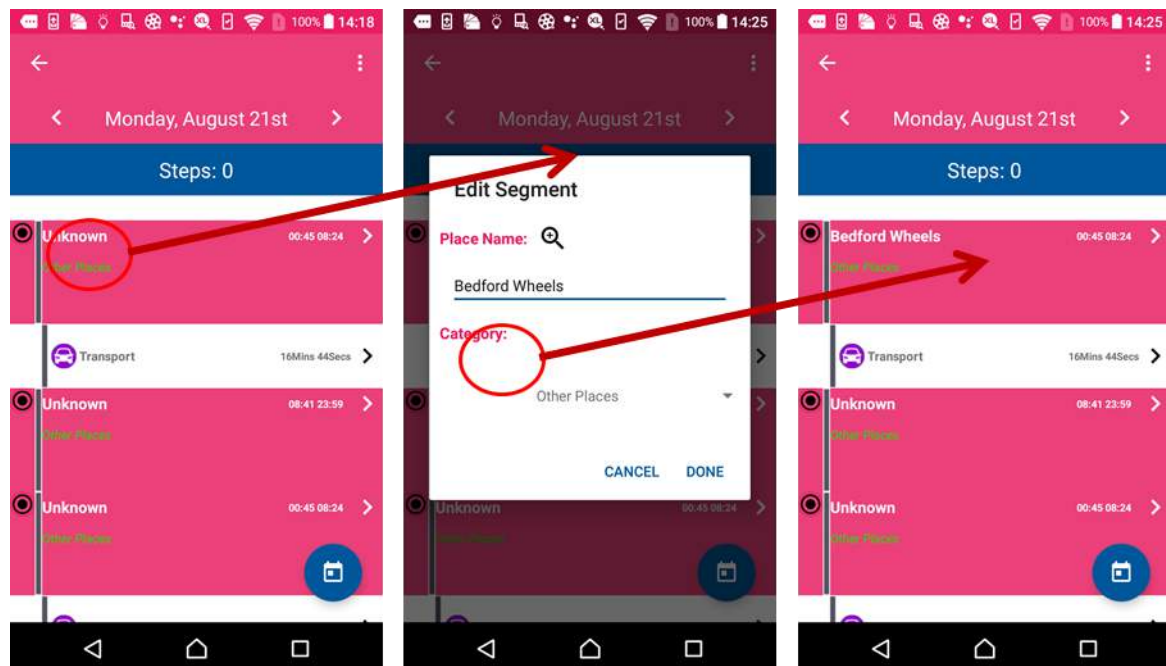


Figure 151: Location annotation

5.4.4. Access Left Menu

Action 1: Press the “menu“ button on the action bar to display left menu or use finger press the left edge of the screen and slide to right to display left menu.

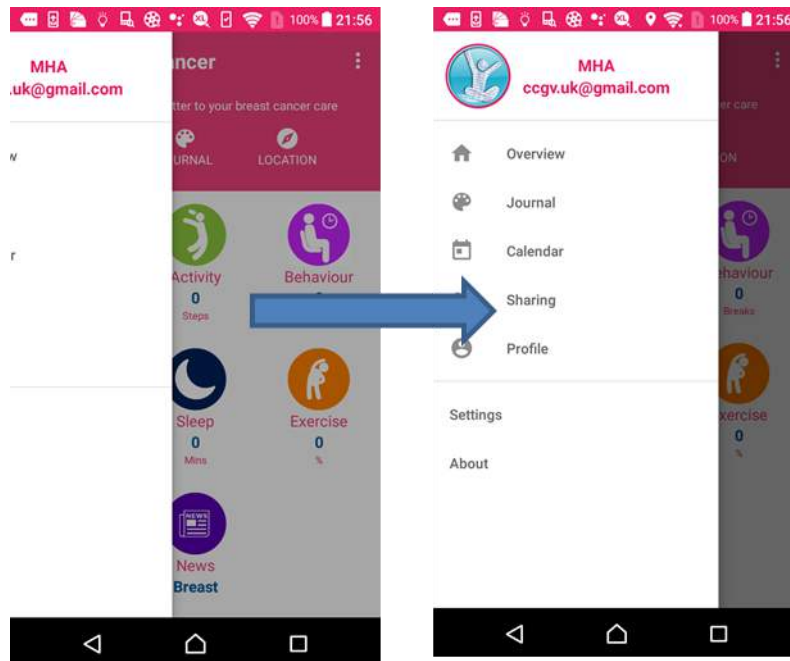


Figure 152: display left menu

Result: Left menu appears on the screen.

5.4.5. Access Settings Page

Before testing the functions, the user needs to turn on (enable) the functions first.

Action 1: Click the Settings in the left menu

5.4.4.1 Select a Language

The Language Mode – Switch the Language between English, German and Italian.

Action 1: Click the edit button on right side.

Action 2: Choose the data source.

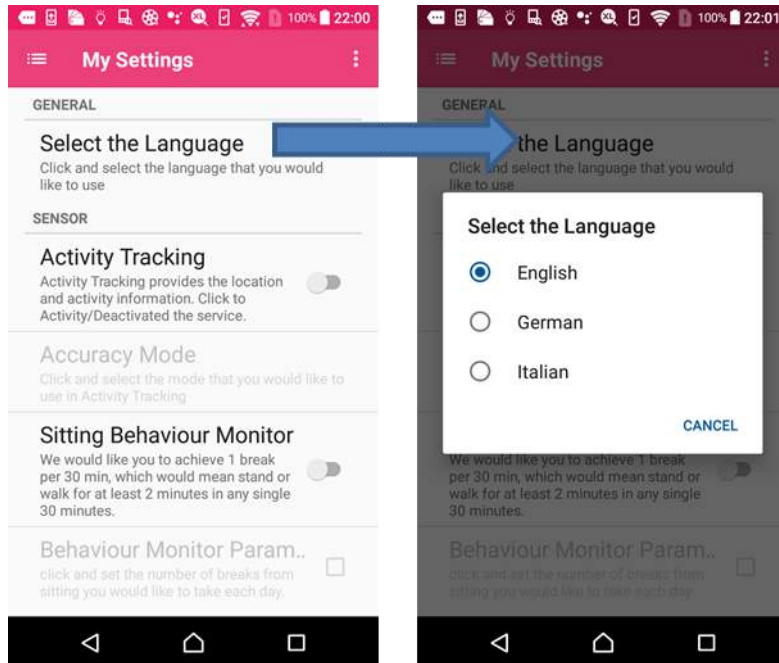


Figure 153: language settings page

Result: App Language is changed.

5.4.4.2 Enable the Location Activity Tracking

Location Sensor monitors the GPS location of the mobile. When user arrives at annotated place, the app will give the message which is related the place in Journal page.

Action 1: Turn on the Switch component.

Action 2: If Android 6.0 or above, allow the location permission

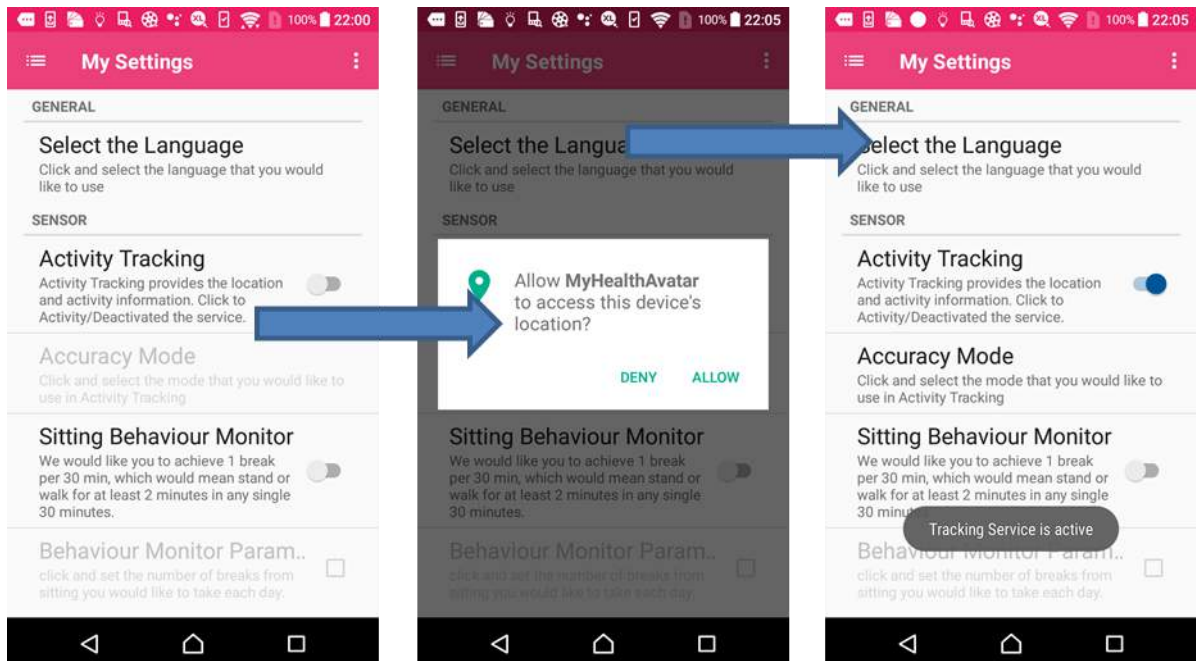


Figure 154: Location sensor is activated

Result: Location Sensor is activated.

5.4.4.3 Enable the Sitting Behaviour Monitor

Sitting Behaviour Monitor - This function monitors the user's break up sitting time per 30 mins.

Action 1: Turn on the Switch component.

Action 2: Once the user has turned on the function, an option will appear. Click the button and edit break up sitting and sitting time goals.

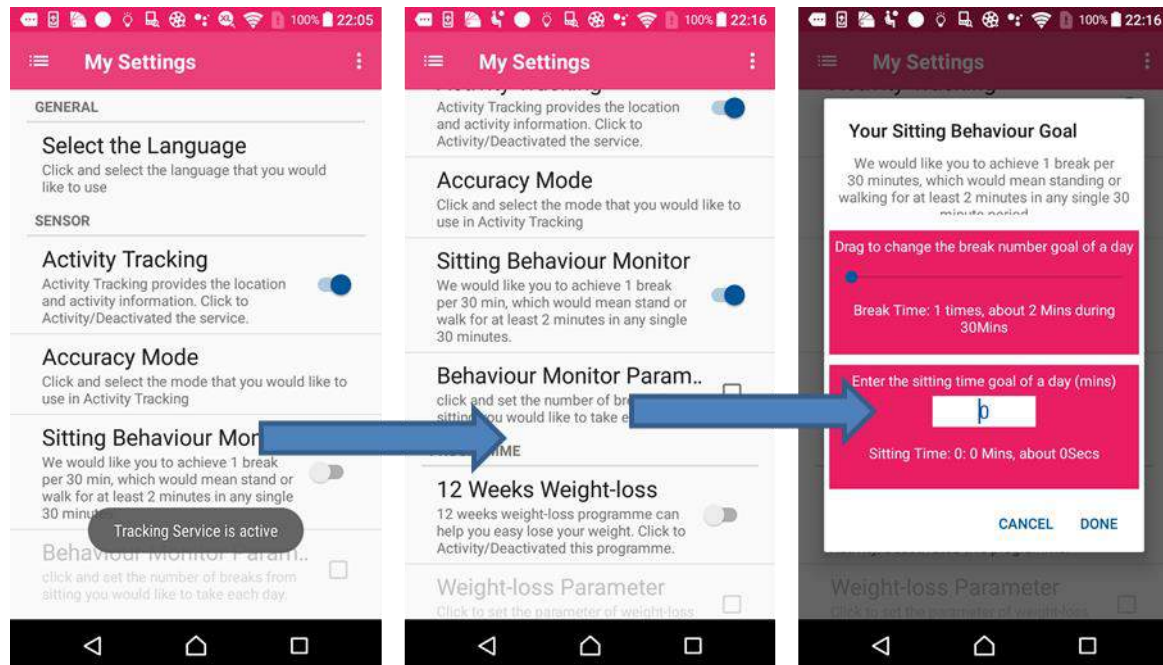


Figure 155: Activity sensor is activated

Result: Sitting behavior monitor is activated.

5.4.4.4 Enable the 12 Weeks Weight-loss Program

12 weeks weight-loss program can help user reduce his weight during 12 weeks. This program will update the reduce weight goal week by week. During the program time period, the program will measure user's weight changes then summary user's activities every weekend. Moreover it gives suggestion tips to help user food and exercise.

Action 1: Turn on the Switch component.

Action 2: Choose your gender

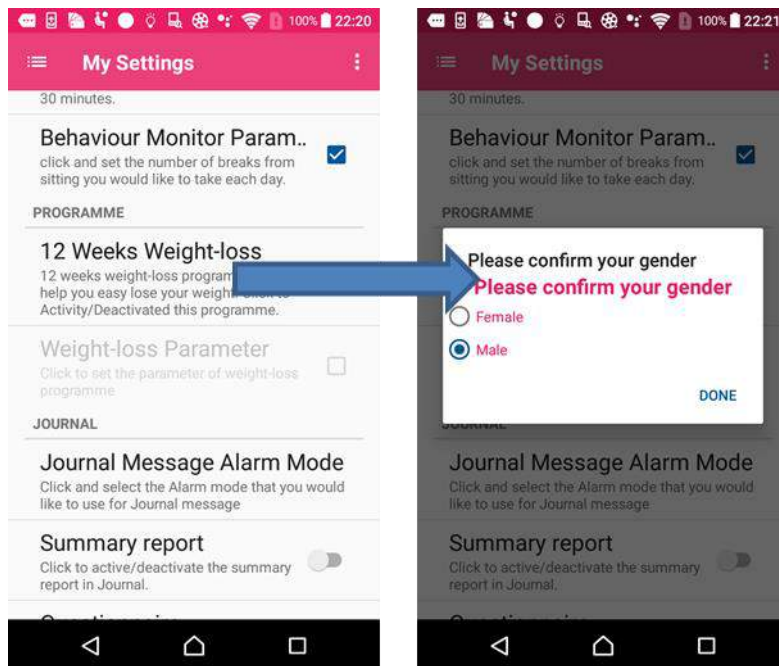


Figure 156: setup the weight-loss program

Action 2: Once the user has turned on the function, an option will appear. Click the button and choose the methods of showing the tips.

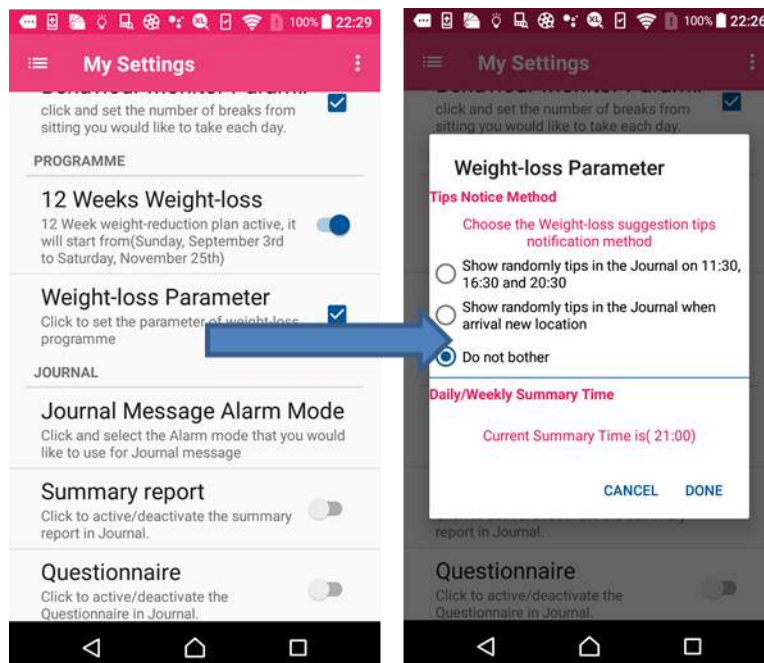


Figure 157: the set the method of showing tips

5.4.4.5 Enable the Journal Alarm

Journal Alarm-User can switch between the silence and alarm mode when the notice display in the Journal page.

Action 1: Turn on the Switch component.

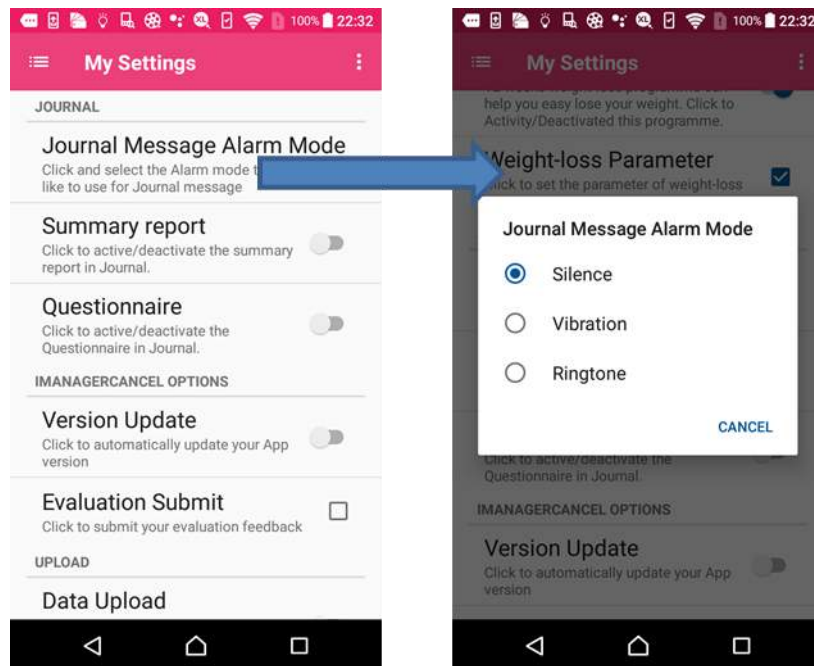


Figure 158: Journal alarm is activated

Result: Journal Alarm is activated.

5.4.4.6 Enable the Summary Report

Summary Report-Enable the summary of goals achievement report in Journal page.

Action 1: Turn on the Switch component.

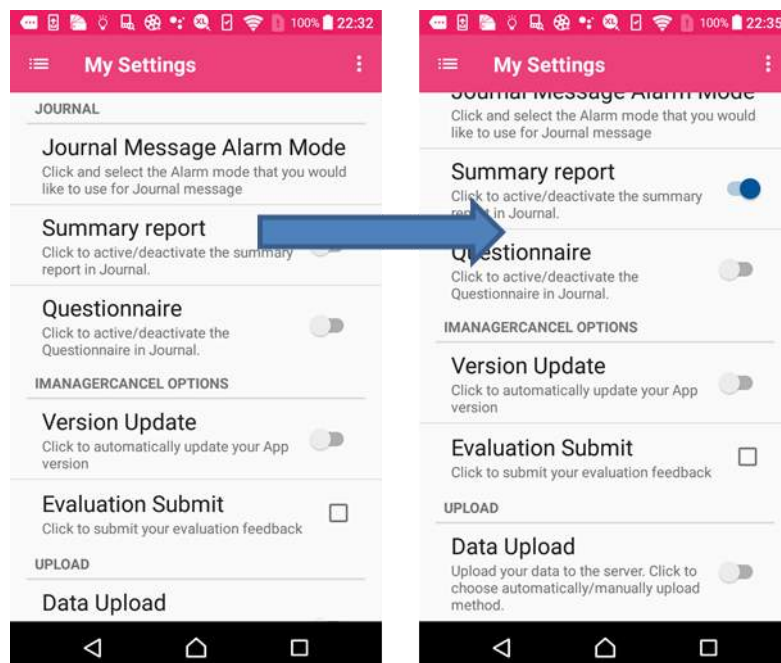


Figure 159: Summary report is activated

Result: Summary Report is activated.

5.4.4.7 Enable the Journal Questionnaire

Journal Questionnaire–Enable the questionnaires in Journal page.

Action 1: Turn on the Switch component.

Action 2: Once the user has turned on the function, the app will ask the questionnaires around at 12:30, 16:30 and 20:30 in Journal page.

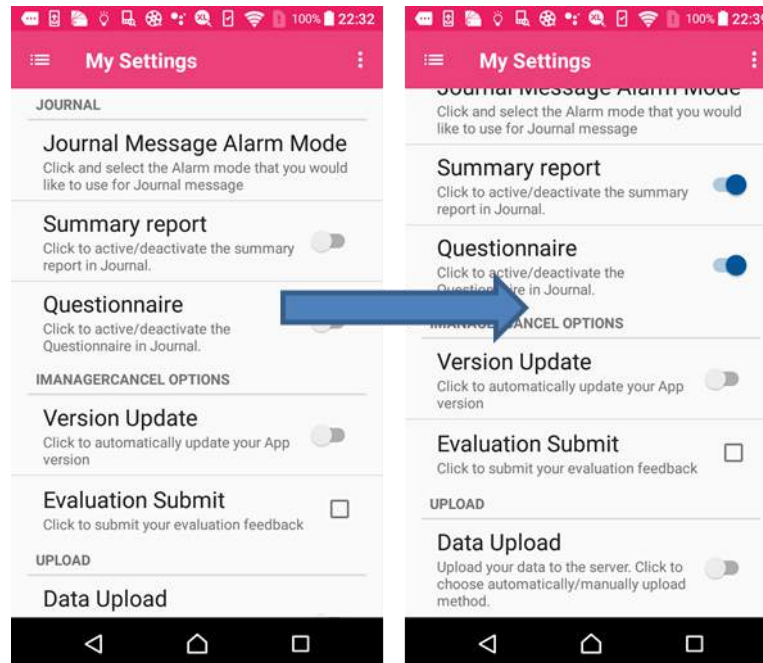


Figure 160: Journal Questionnaires is activated

Result: Journal questionnaires are activated.

5.4.4.8 Enable the Version Update

Version Update–Enable the App can check the version number and upgrade itself.

Action 1: Turn on the Switch component.

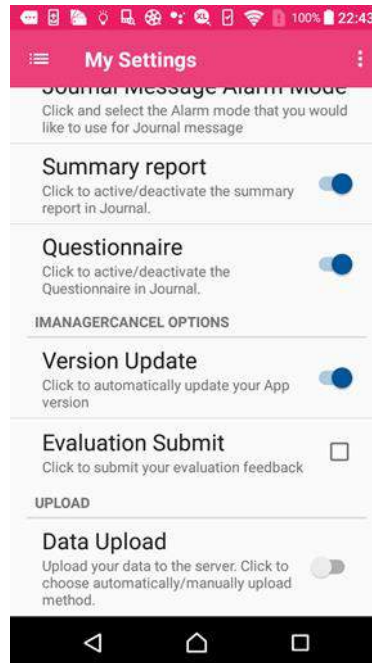


Figure 161: version update is activated

Result: Version update is activated.

5.4.4.9 Enable the Data Upload

Data Upload–Enable the App can upload the data via internet.

Action 1: Turn on the Switch component.

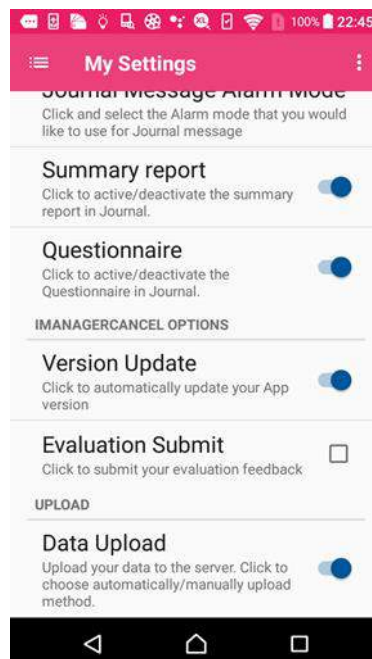


Figure 162: data upload is activated

Result: Data upload is activated.

5.4.6. Access Overview page and Function Suite

5.4.5.1 Access Weight Suite

Action 1: Click the Weight Suite icon in Overview to show the Weight Suite page.

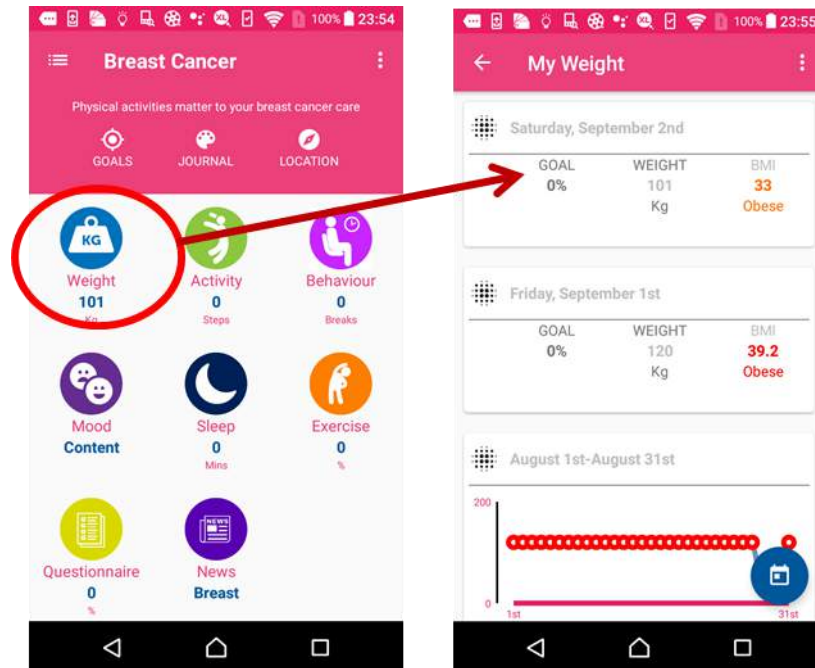


Figure 163: Weight suite overview

Action 2: Click the item to edit the weight and height values.

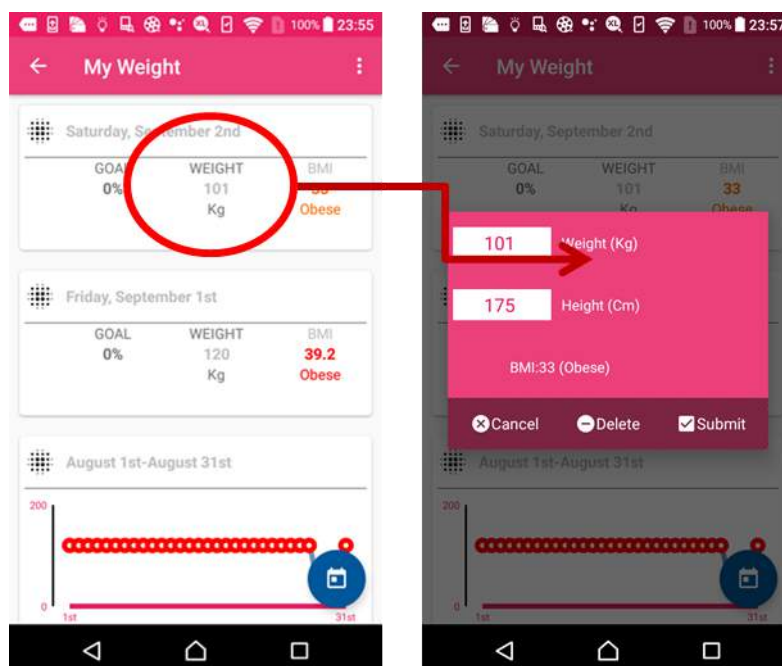


Figure 164: edit weight and height

Action 3: The different colors represent different BMI levels.

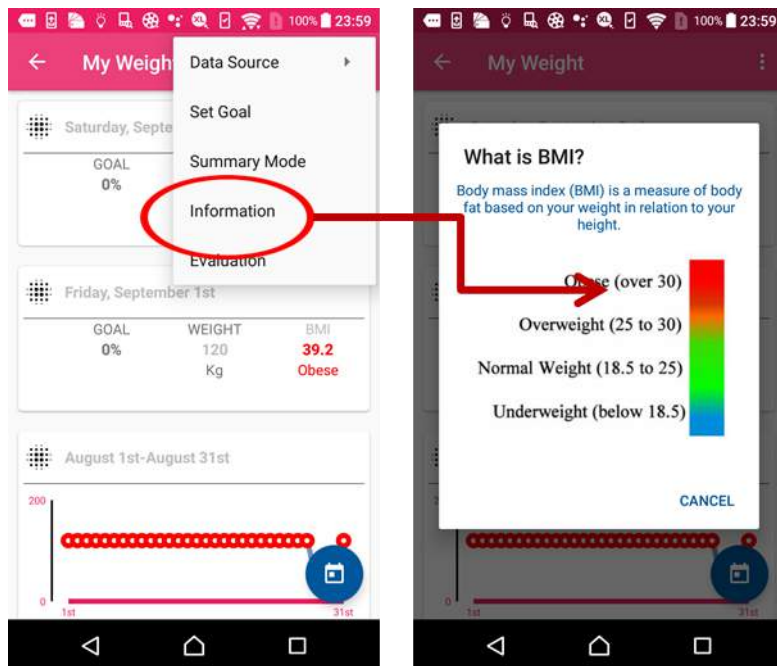


Figure 165: check weight BMI Level

Action 3: Click the “Data Mode” button to switch the data sources.

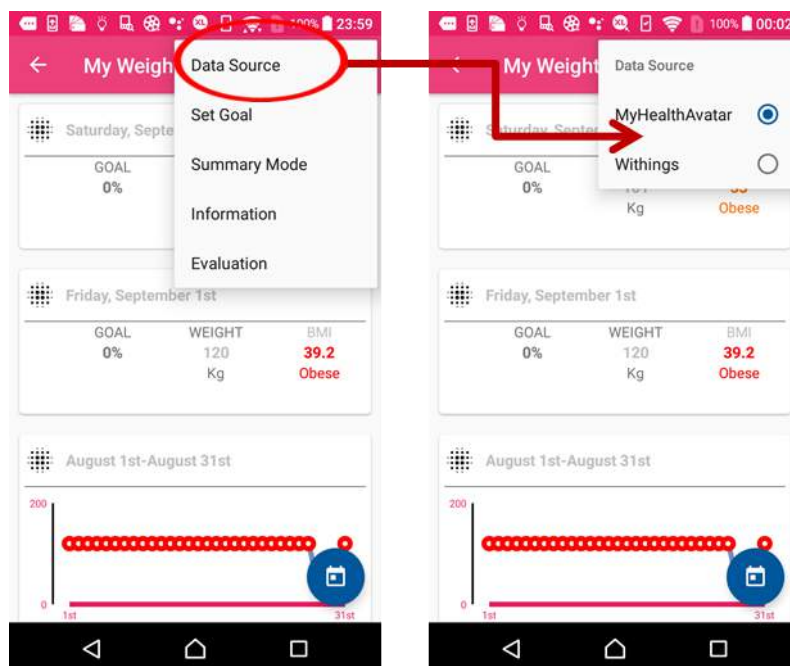


Figure 166: change the data source

Action 4: Click the “Set Goals” button to change the goals.

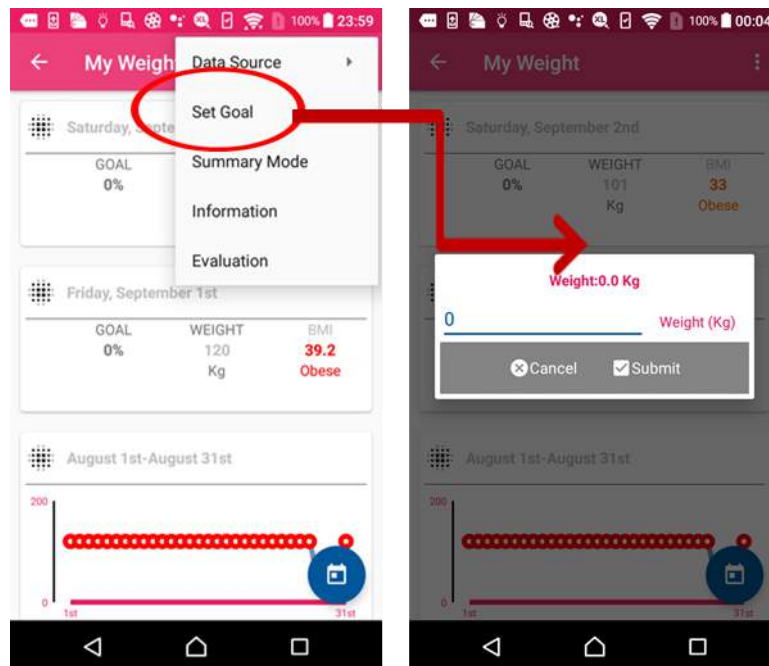


Figure 167: change the weight goal

Action 5: Swap the screen up and down, in order to check data and summary in different date.

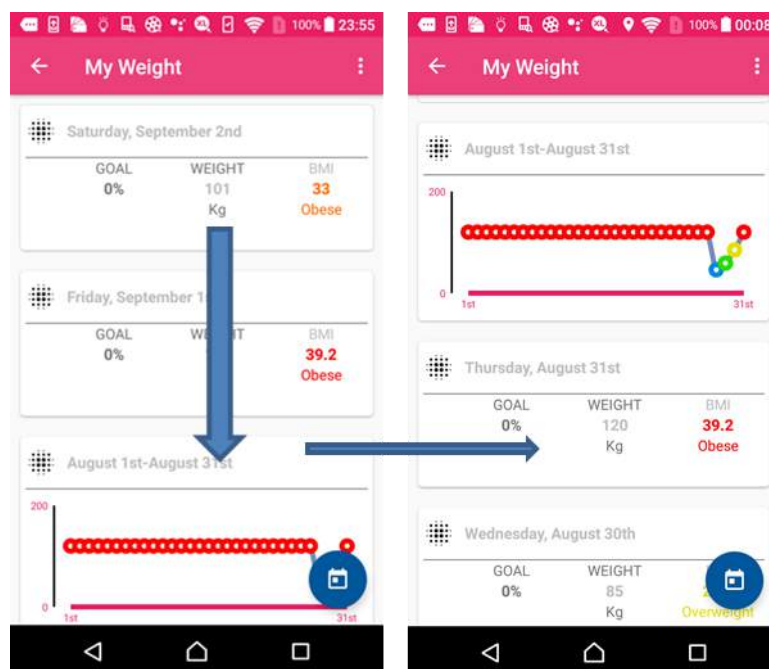


Figure 168: check the weigh on different date

5.4.5.2 Access Activity Suite

Action 1: Click the Activity Suite icon in Overview to show the Activity Suite page.

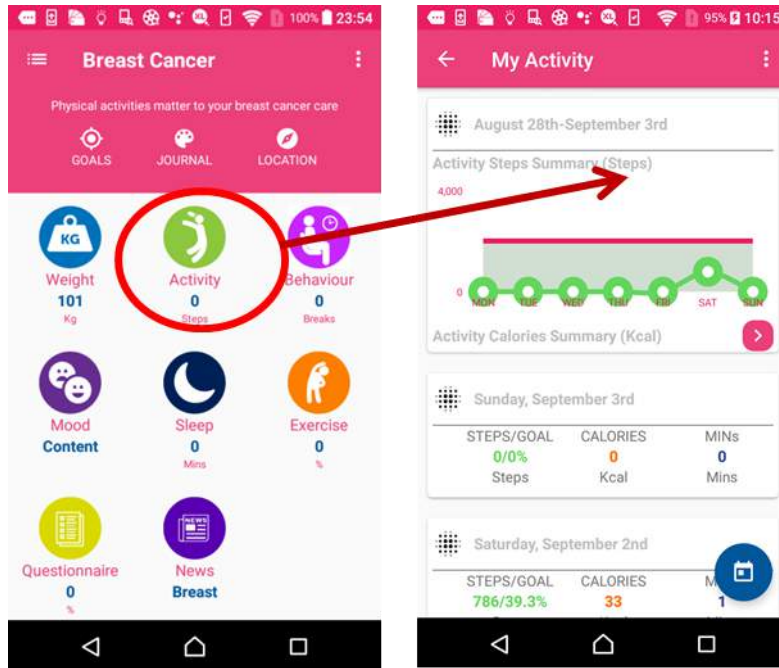


Figure 169: Activity suite overview

Action 2: Swap the screen to the up and down, in order to change the date and check the summary.

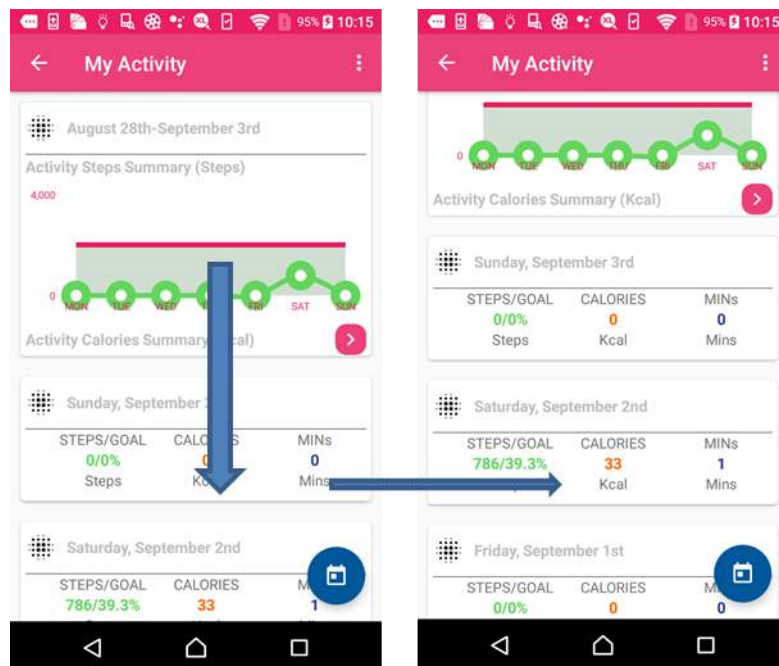


Figure 170: check the activity on different date

Action 3: Click the “Data Mode” button to switch the data sources.

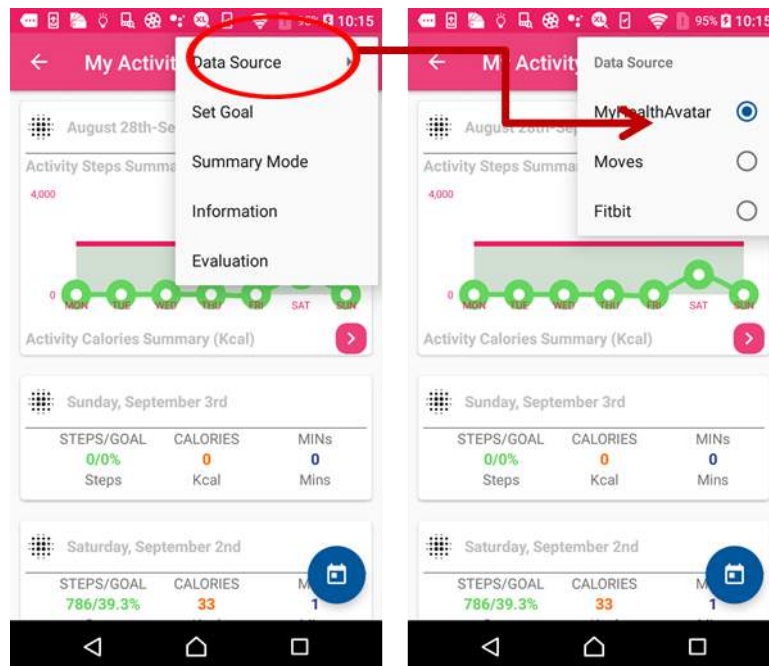


Figure 171: change the data source

Action 4: Click the “Set Goals” button to change the goals.

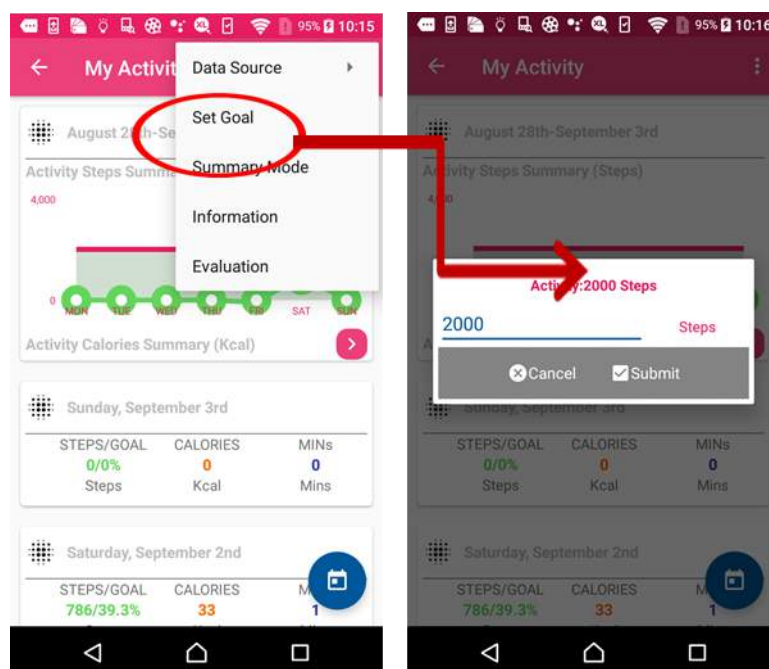


Figure 172: change the activity goal

5.4.5.3 Access Sitting Behaviour Suite

Action 1: Click the Behaviour Suite icon in Overview to show the Behaviour Suite page.

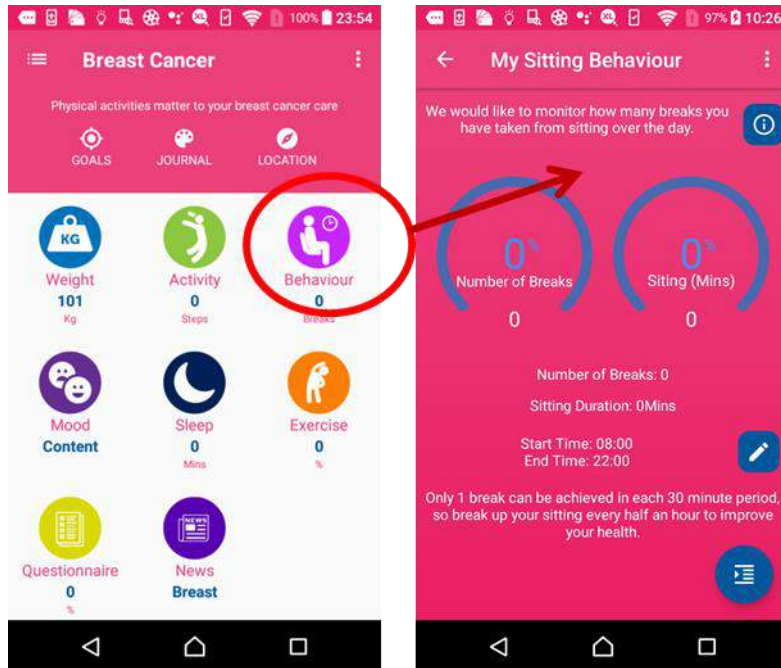


Figure 173: Behaviour suite overview

Action 2: Click the “edit” button to edit the duration time that you want to monitor the sitting behaviour.

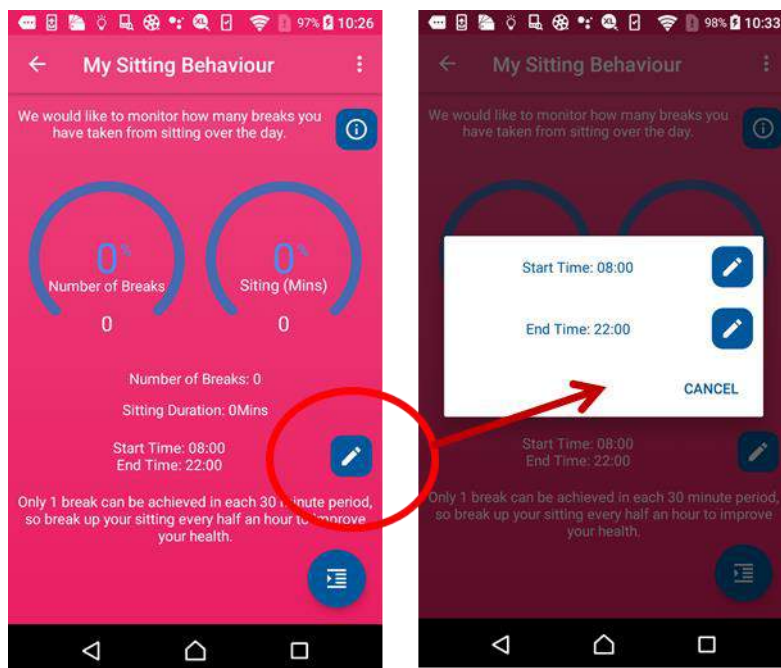


Figure 174: edit behaviour monitor duration

Action 3: Click the “list” button to check the history records.

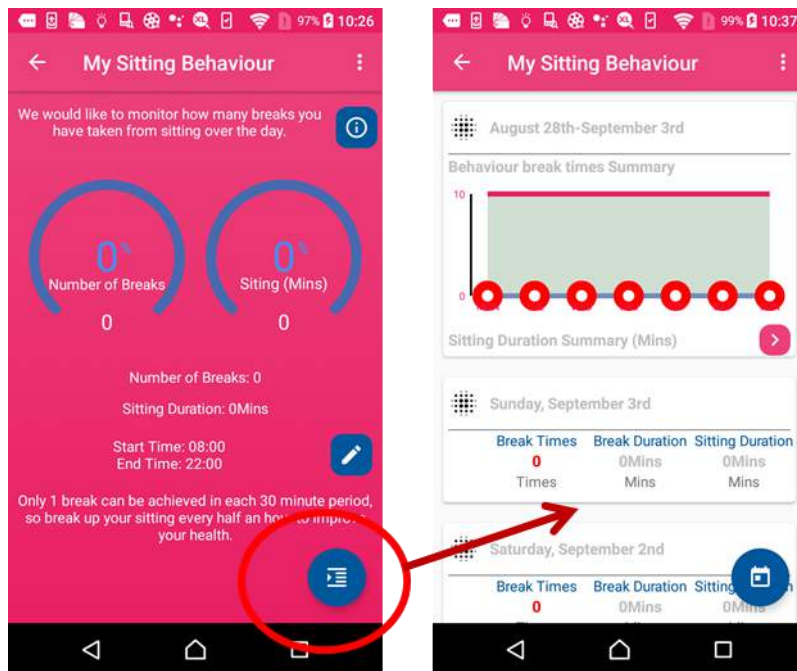


Figure 175: Behaviour history record

Action 4: Swap the screen to the up and down, in order to change the date and check the summary.

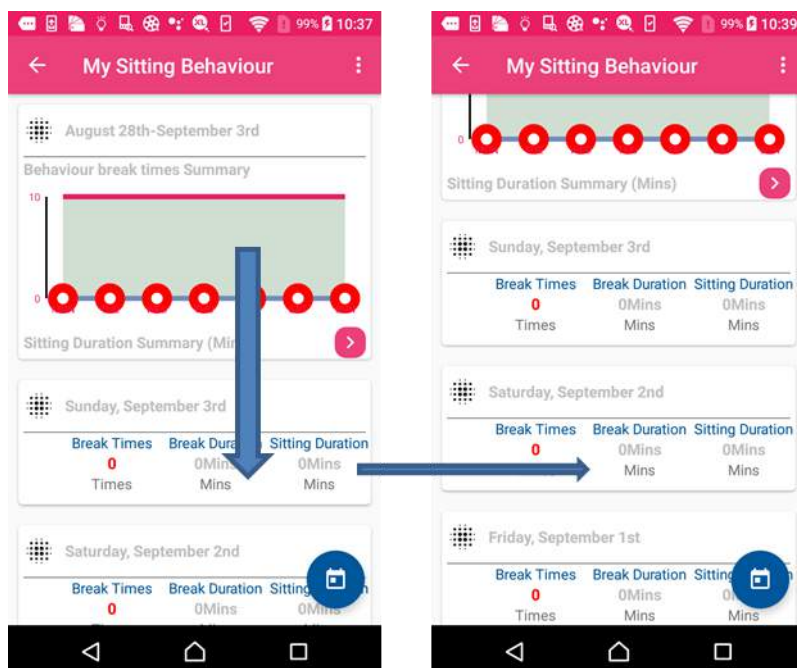


Figure 176: change the data source

Action 5: Click the “Set Goals” button to change the goals.

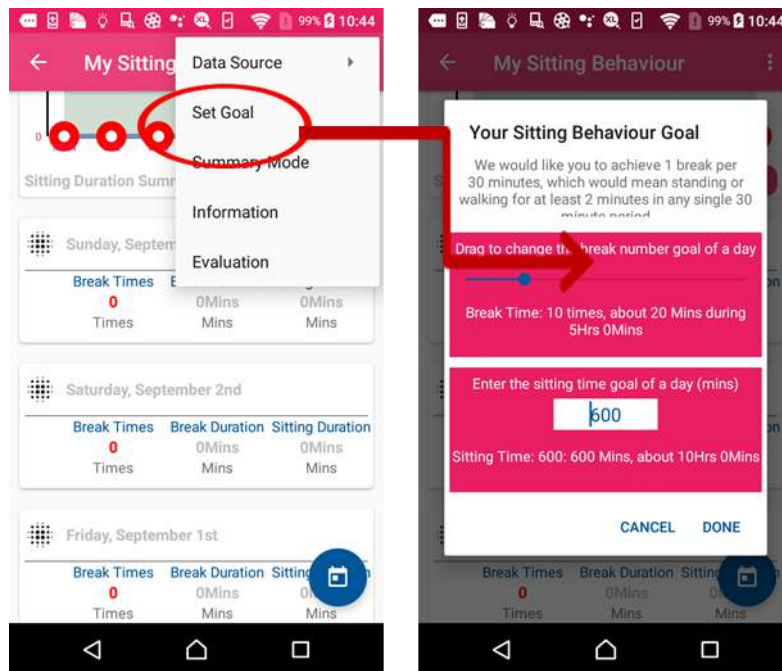


Figure 177: change the break up sitting and sitting time goals

5.4.5.4 Access Sleep Suite

In this suite, patient can manually insert Sleep quality or get information on sleep quality from Fitbit automatically.

Action 1: Click the Sleep Suite icon in Overview to show the Sleep Suite page.

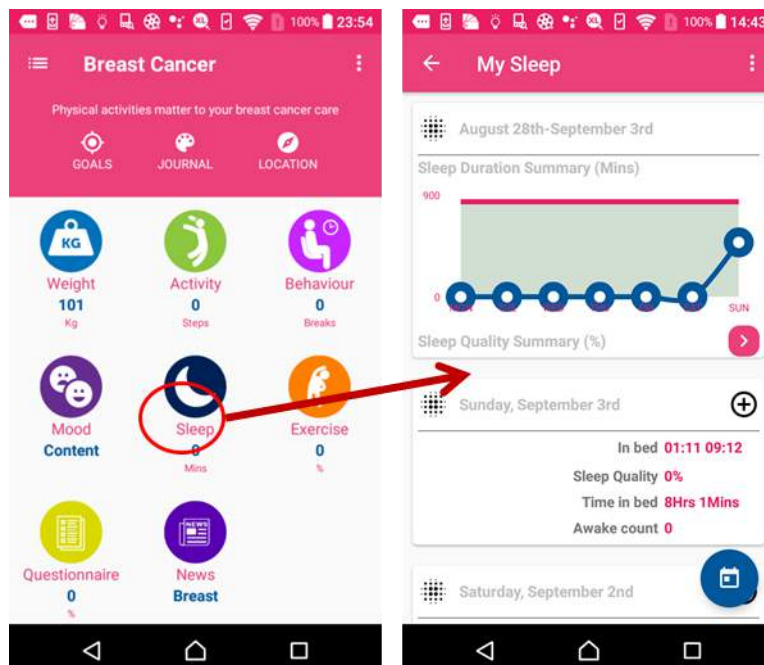


Figure 178: The Sleep suite overview

Action 2: Click the “+” button to add new sleep record.

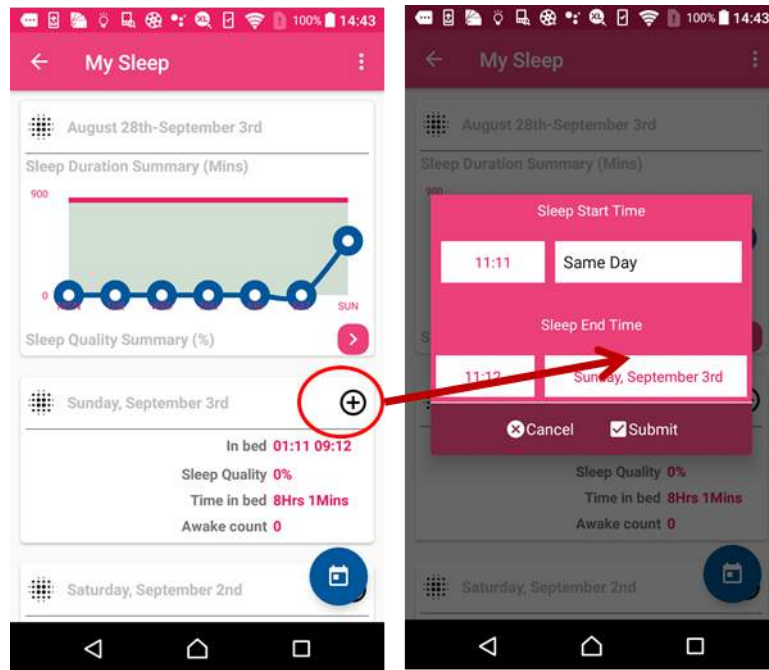


Figure 179: add new record to the Sleep suite

Action 3: Swap the screen to the up and down, in order to change the date and check the summary.

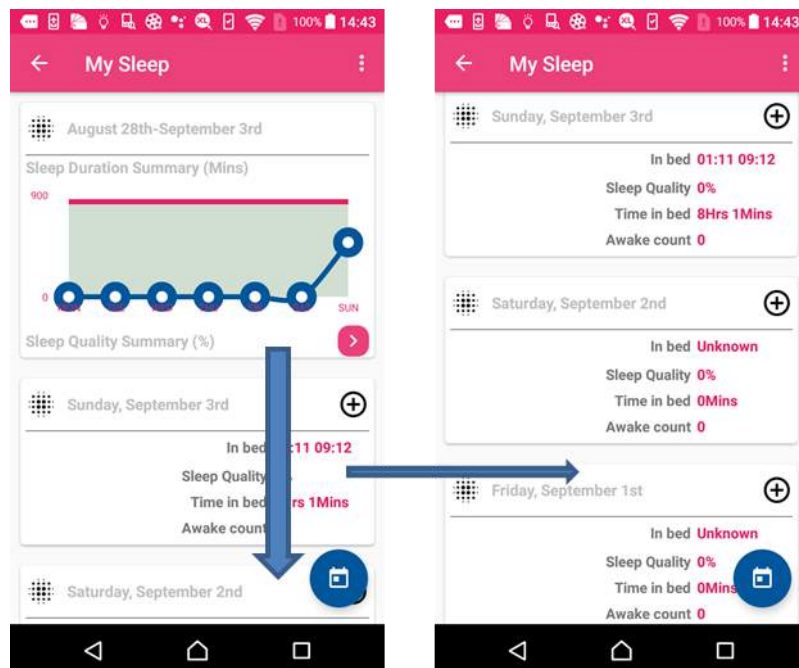


Figure 180: check the sleep information on different date

Action 4: Click the “Data Mode” button to switch the data sources.

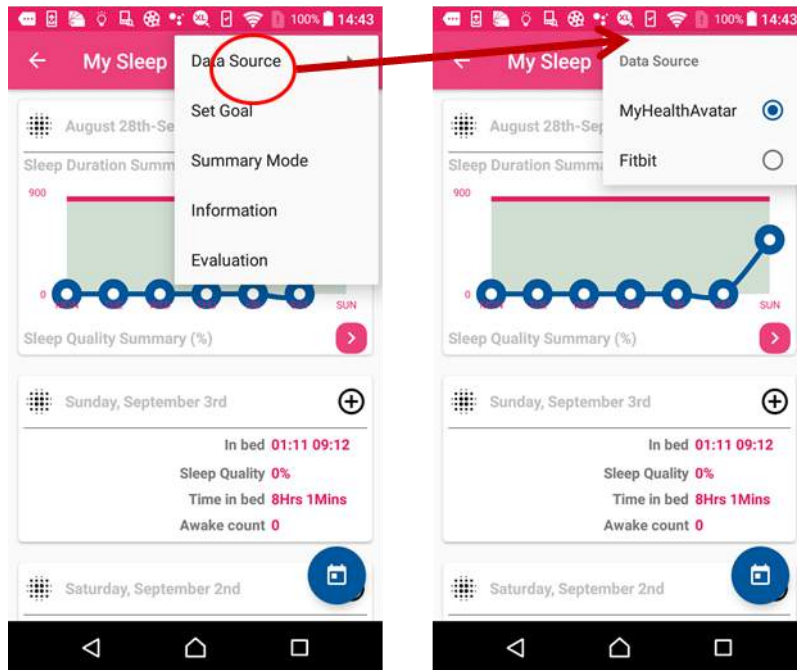


Figure 181: change the data source

Action 4: Click the “Set Goals” button to change the goals.

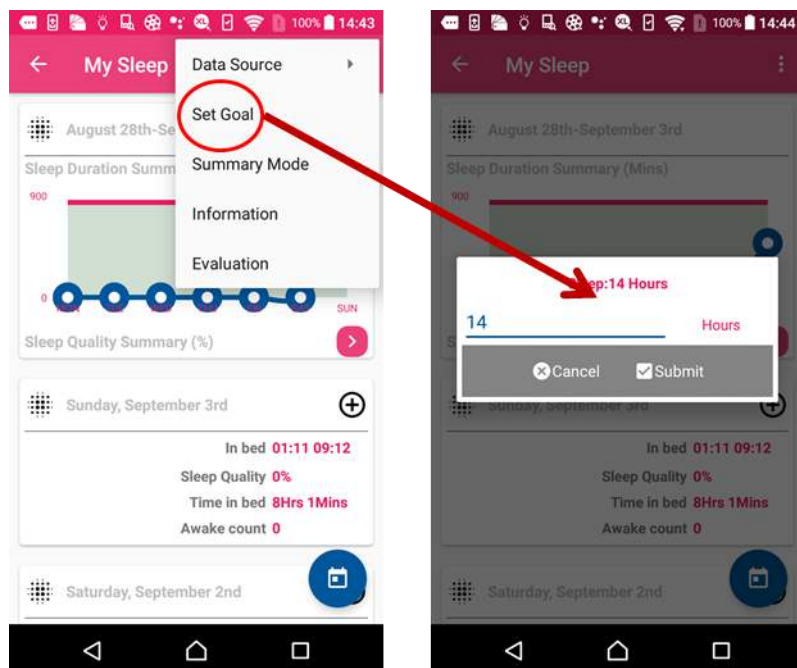


Figure 182: change the activity goal

5.4.5.5 Access Mood Suite

Action 1: Click the Mood Suite icon in Overview to show the Mood Suite page.

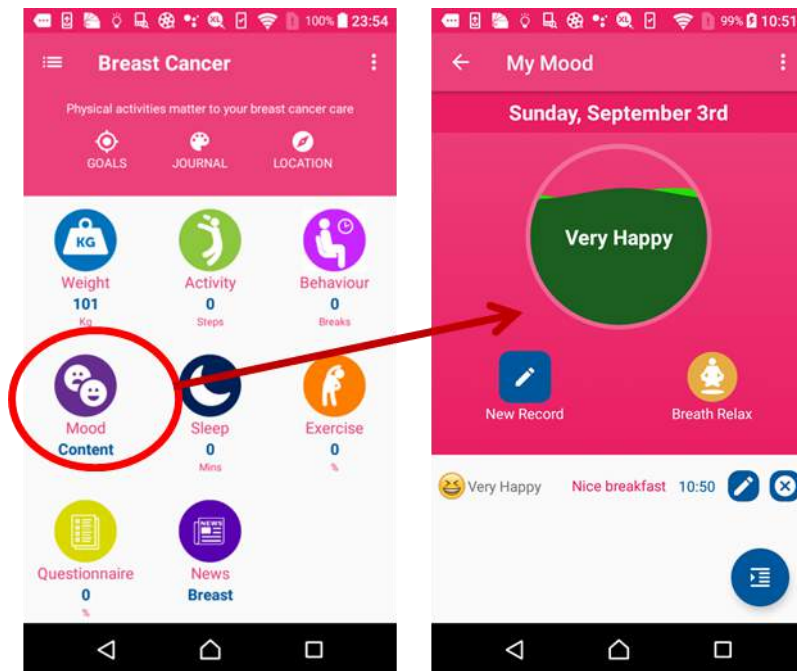


Figure 183: Mood suite overview

Action 2: Click the “edit” button to add new mood. The record of the new mood will affect the main mood status as shown in the water flow progress bar.

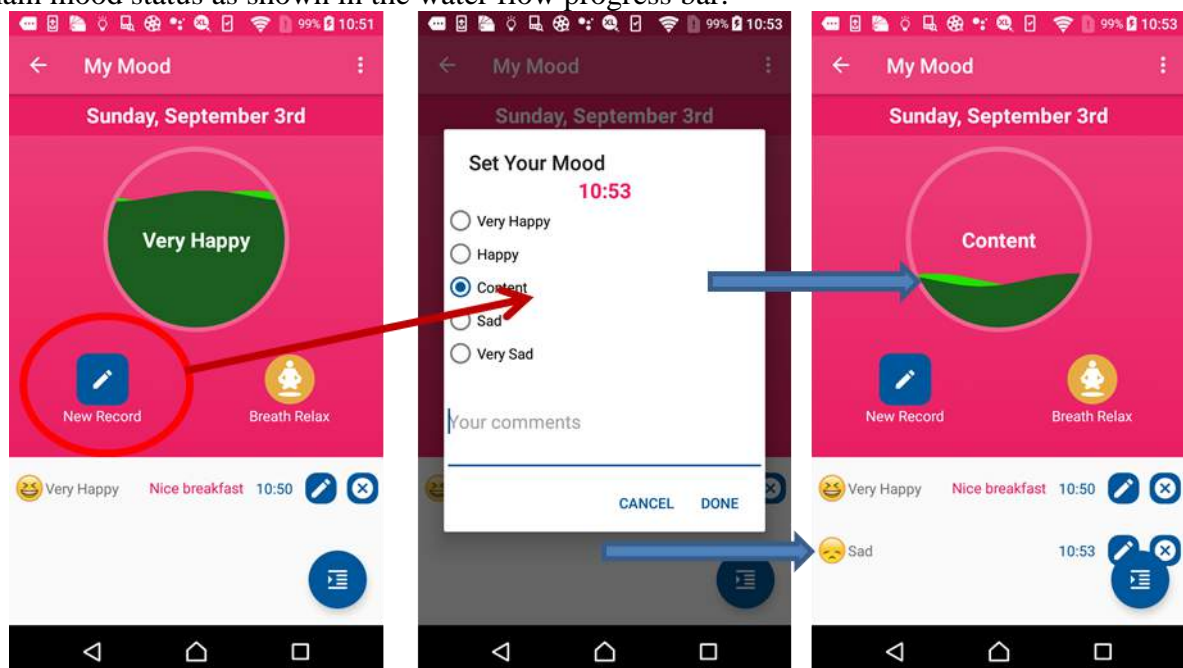


Figure 184: add new mood to Mood suite

Action 3: Click the “ok” button to submit new emotion and click Edit button to edit the emotions.

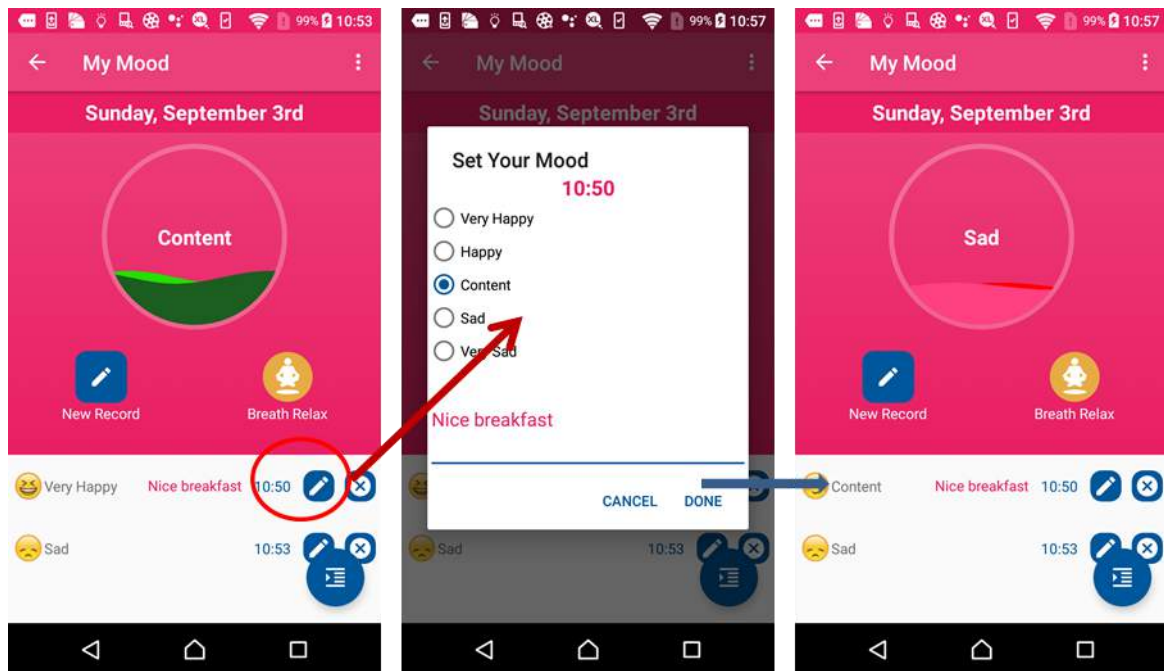


Figure 185: edit the mood in Mood suite

Action 4: Click the “list” button to check the history records.

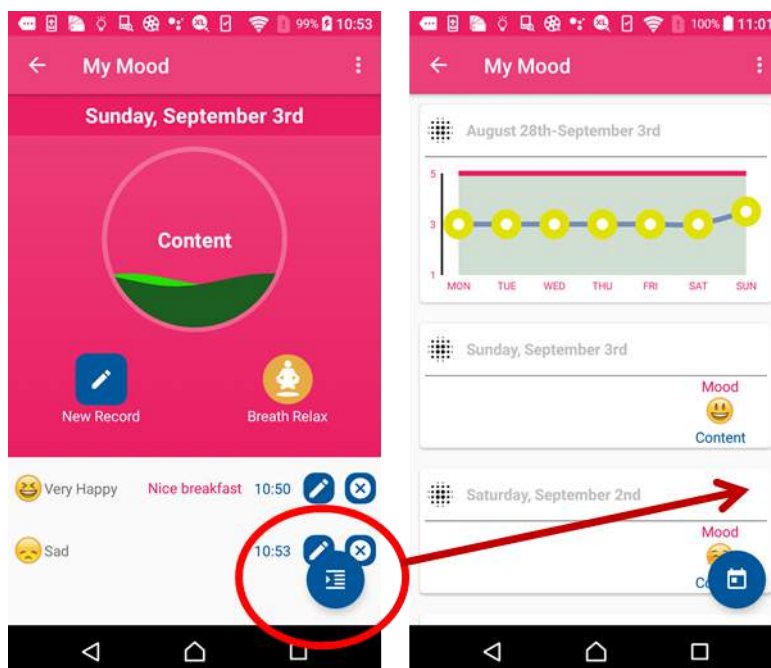


Figure 186: edit the mood in Mood suite

Action 5: Swap the screen to the up and down, in order to change the date and check the summary.

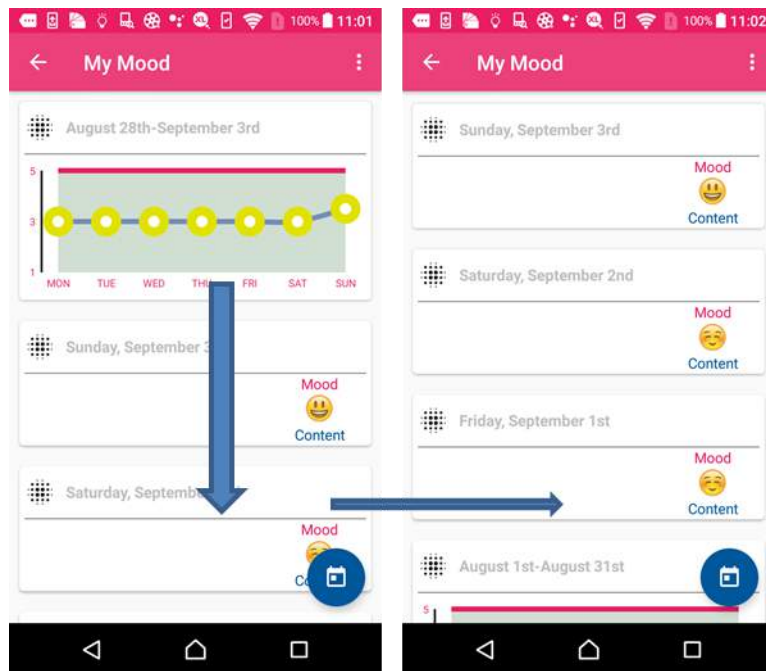


Figure 187: check the mood status on different date

Action 6: Click the relax button to start a relax program.

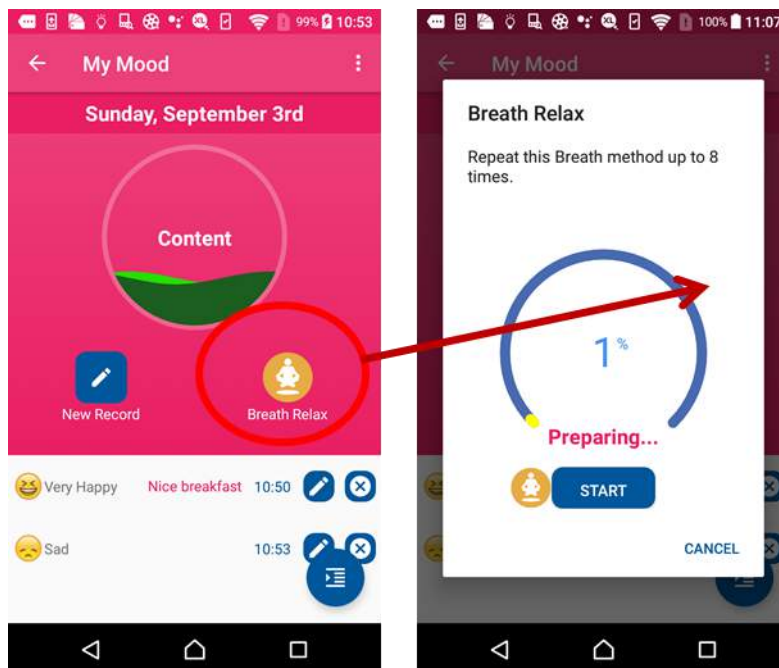


Figure 188: have the relax program

5.4.5.6 Access Exercise Suite

Action 1: Click the Exercise Suite icon in Overview to show the Exercise Suite page.

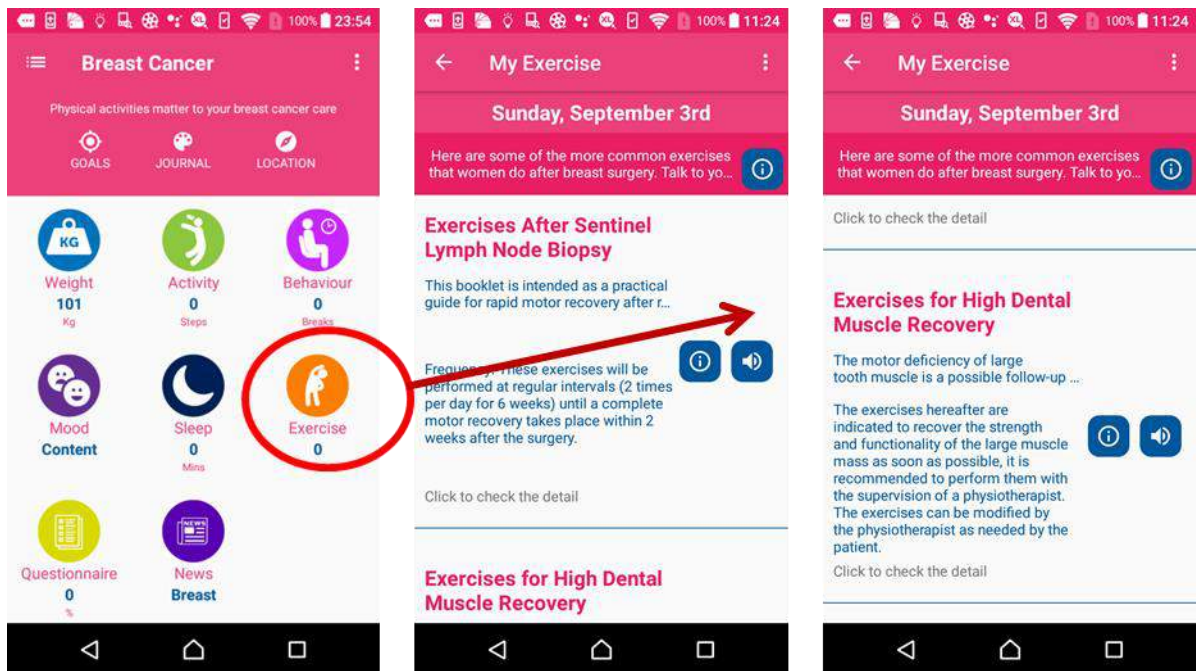


Figure 189: Exercise suite overview

Action 2: Click the item in the page to show the exercise panel.

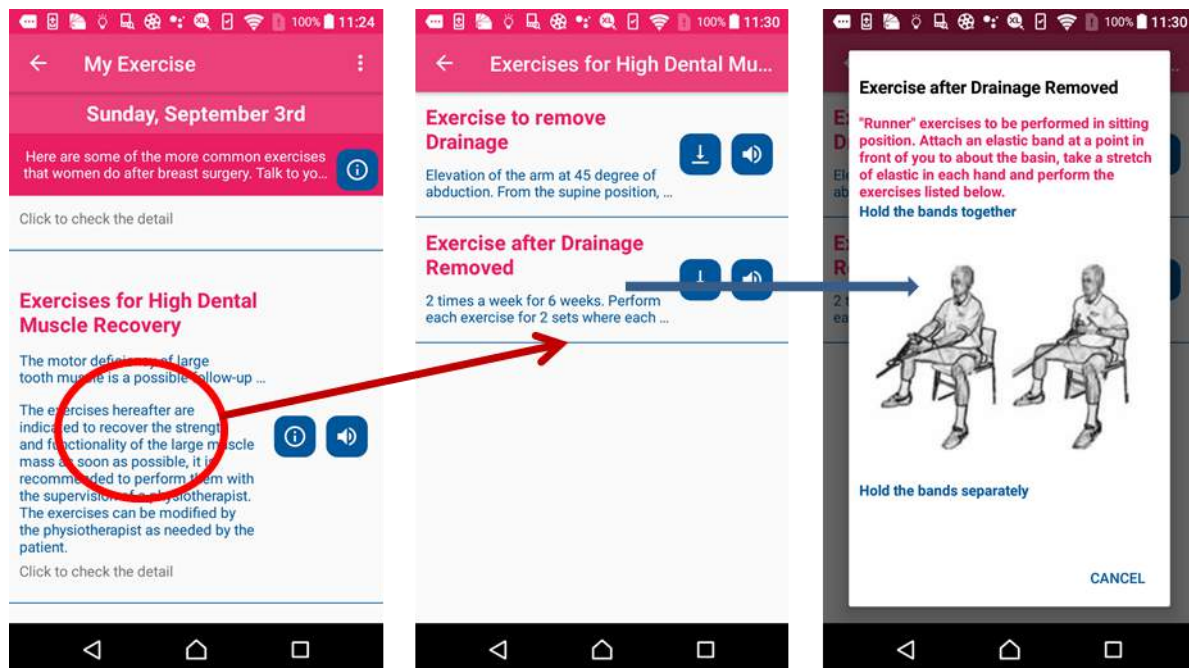


Figure 190: Exercise detail

5.4.5.7 Access Questionnaire Suite

Action 1: Click the Questionnaire Suite icon in Overview to show the Questionnaire Suite page.

Action 2: If android 6.0 or above, allow the audio permission.

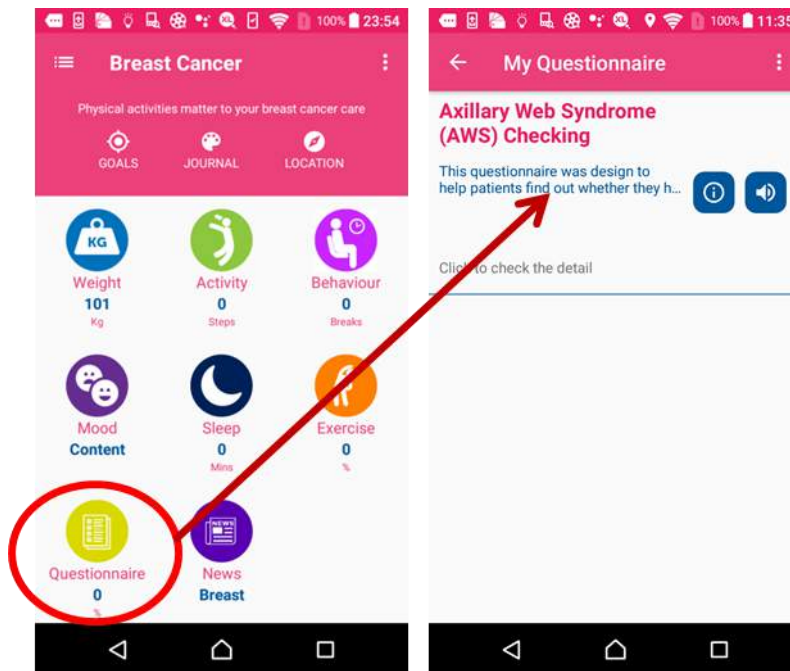


Figure 191: Questionnaire suite overview (1)

Action 3: Click the item in list and check the exercise detail.

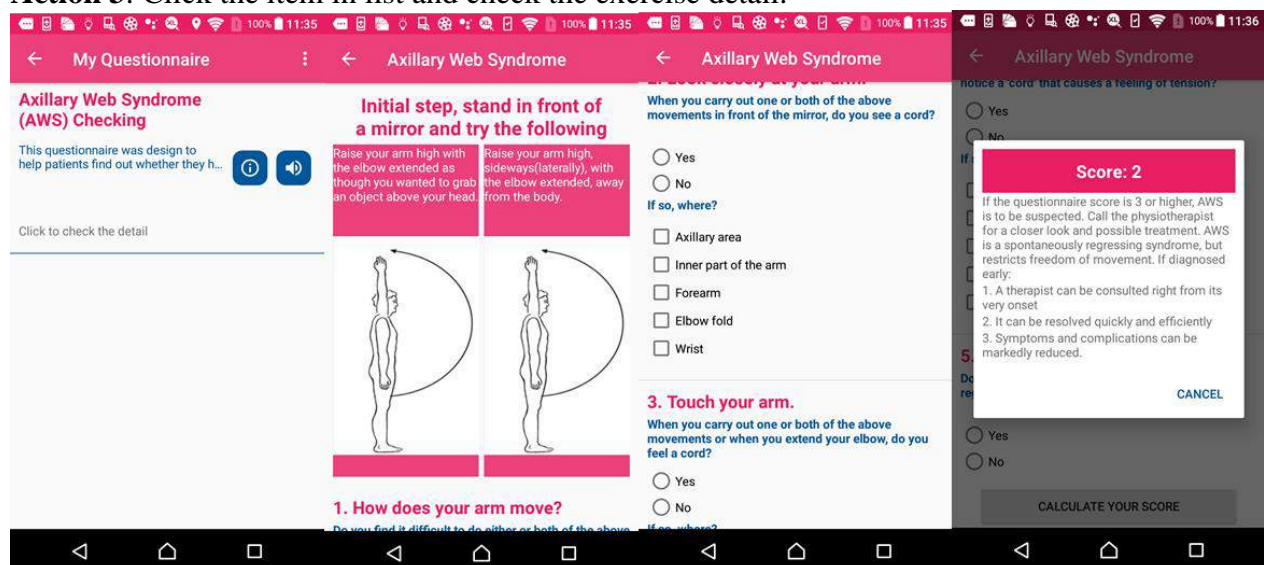


Figure 192: Questionnaire suite overview (2)

4.4.5.9 Access Goals Setting

Action 1: Click the button “Set Goals”

Action 2: Click the item to edit the goals

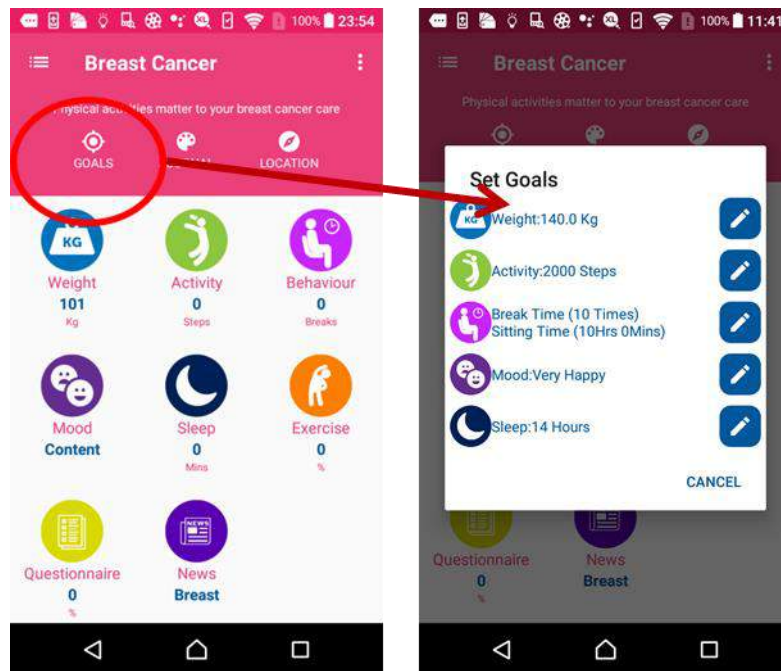


Figure 193: The Goals page overview

Result Goals Setting works fine.

5.4.7. Access Journal Function

Action 1: Click the Journal in the left menu

5.4.6.1 Access Journal Item Edit Panel

Action 1: Click “+” icon at the bottom of Journal modules.

Action 2: Click the “edit” icon in the appearing panel.

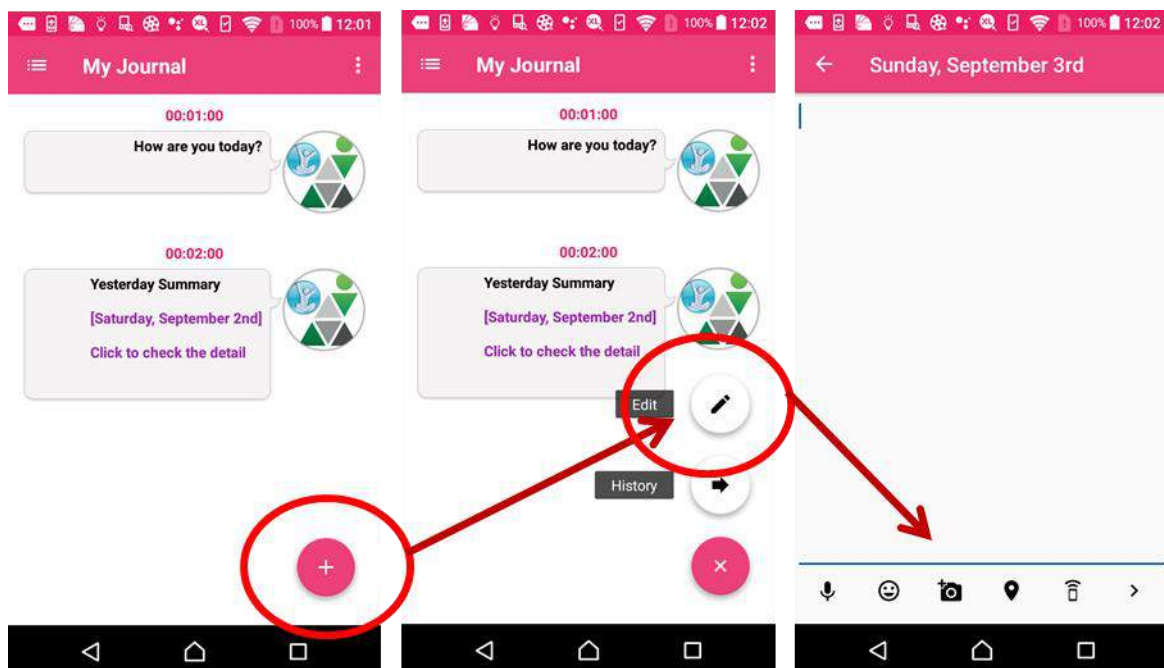


Figure 194: the control panel overview (1)

Action 3: Send the icon to the Journal item edit screen.

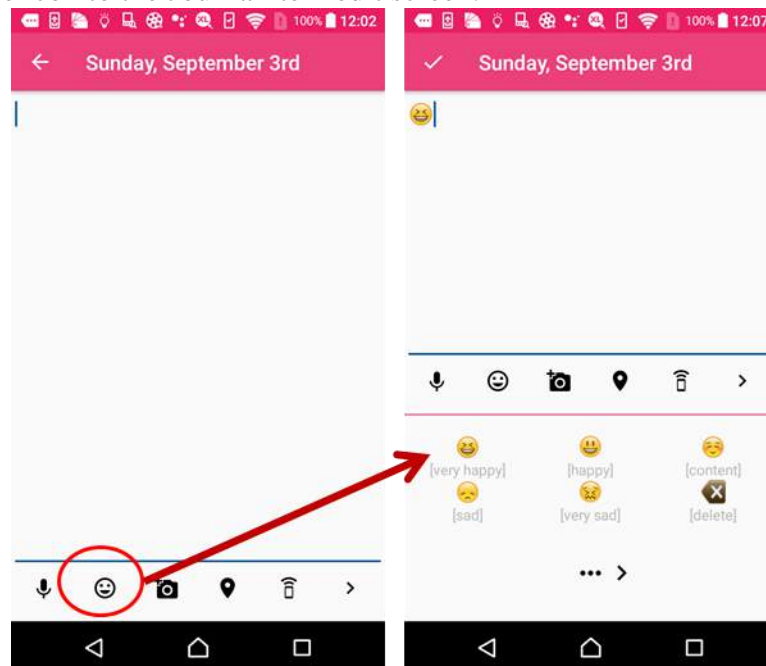


Figure 195: the control panel function (2)

Action 4: Send the message to the Journal item edit screen by using voice input.

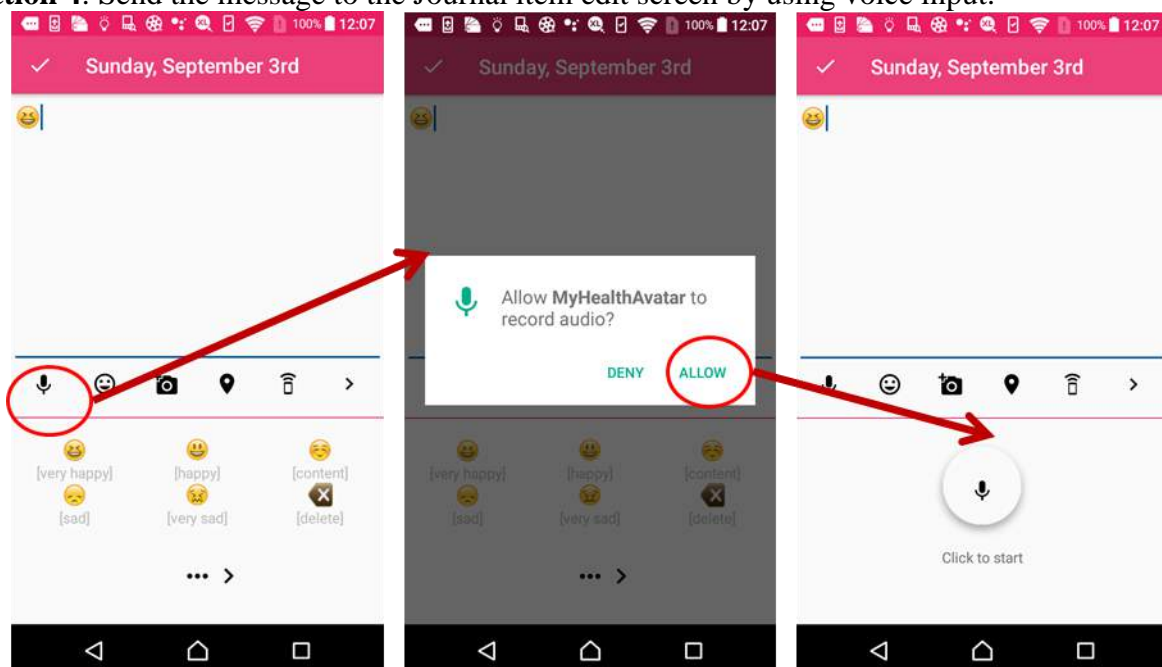


Figure 196: the control panel function (3)

Action 5: Take photo to the Journal item edit screen.

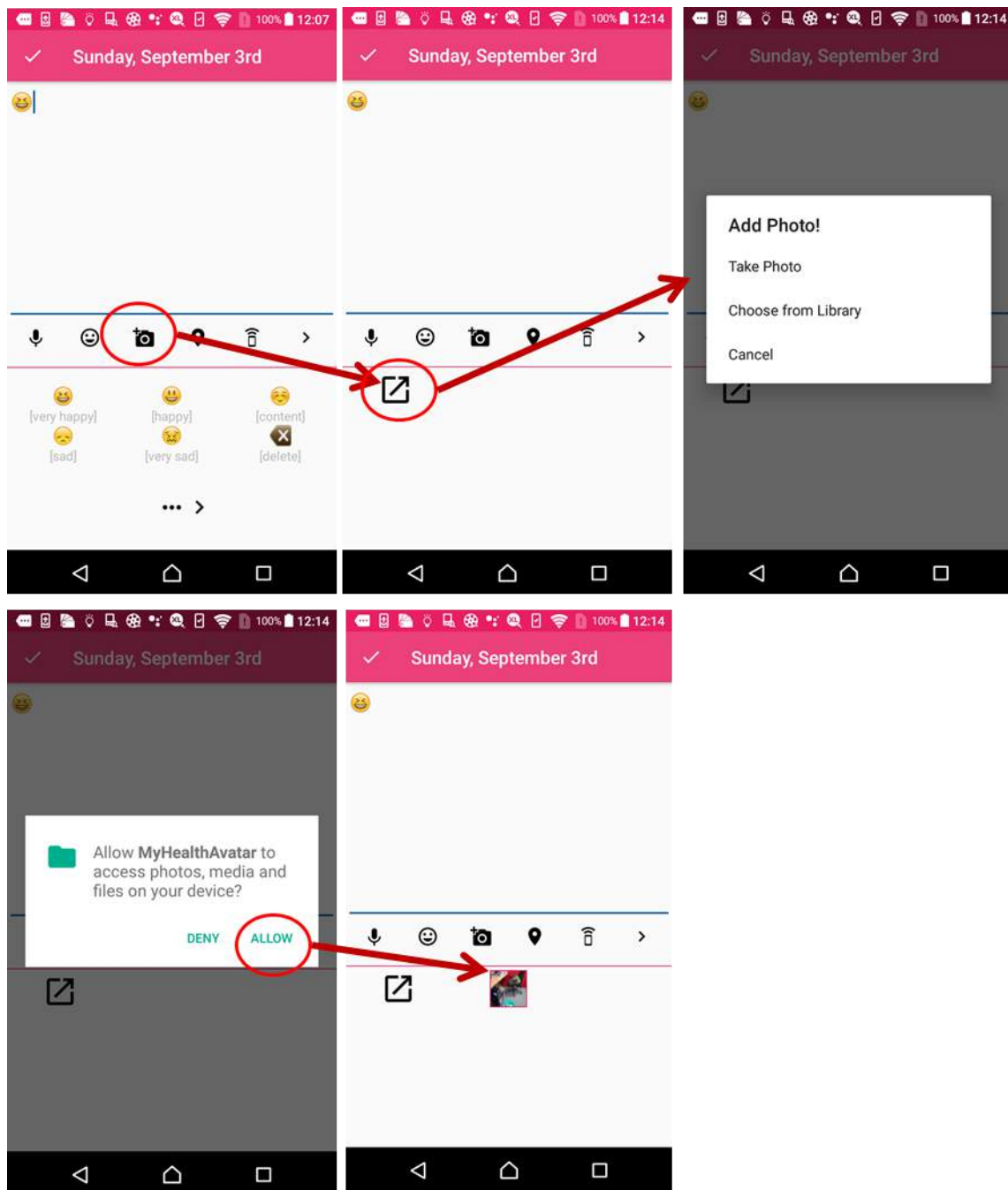


Figure 197: the control panel function (4)

Action 6: Take current location to the Journal item edit screen.

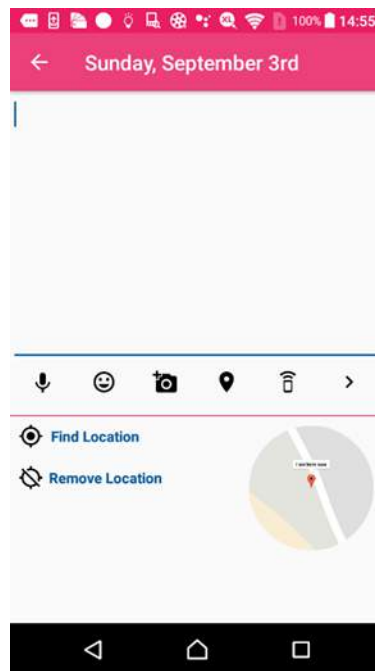


Figure 198: the control panel function (5)

Result icons control panel works fine.

4.4.6.2 Access Mood Monitor

Action 1: As mentioned in previous section, the Mood Questionnaires will appear around at 12:30, 16:30 and 20:30. Once the message appears in Journal page, user can pick the options.

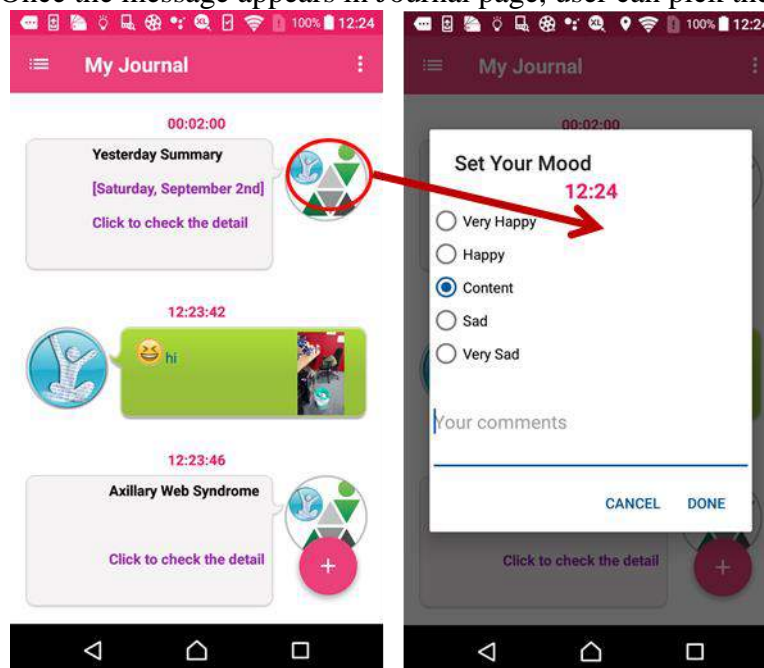


Figure 199: the Mood in Journal

Result: Mood works fine.

4.4.6.3 Access Summary Function

Action 1: As mentioned in previous section, the Summary will appear according to the setting time. Once the message appears in Journal page, user can click the message and check the detail.

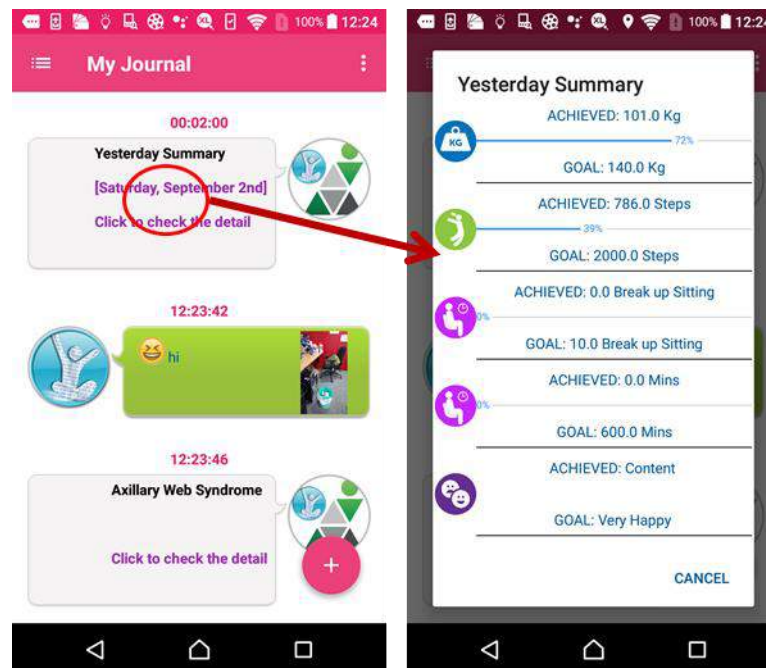


Figure 200: the Summary in Journal

Result: Summary function works fine.

4.4.6.4 Access Breast Cancer Questionnaire

Action 1: As mentioned in previous section, the activity notice will appear according to the user's activity.

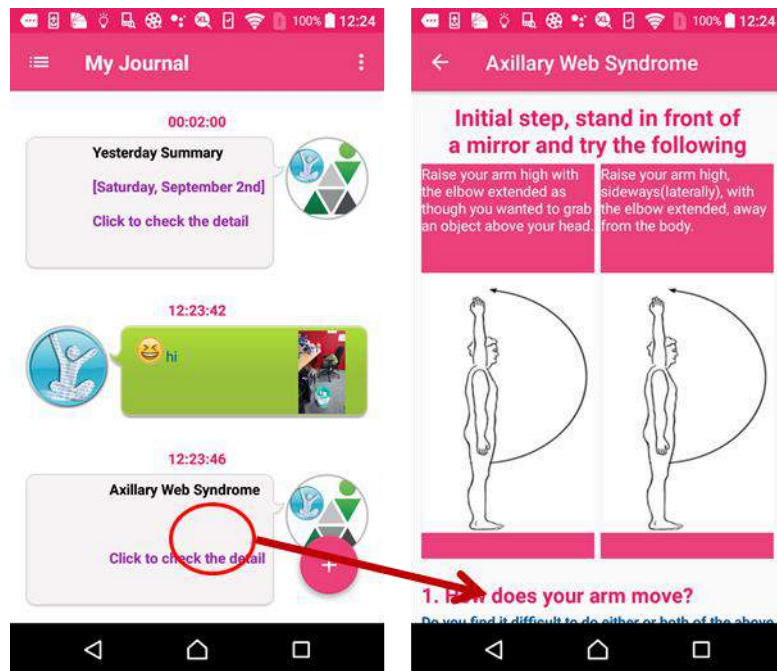


Figure 201: the Questionnaire in Journal

Result: Activity notice works fine.

5.4.8. Access Calendar Function

Action 1: Click the Calendar in the left menu

Action 2: Swap calendar to the left or right to change the month.

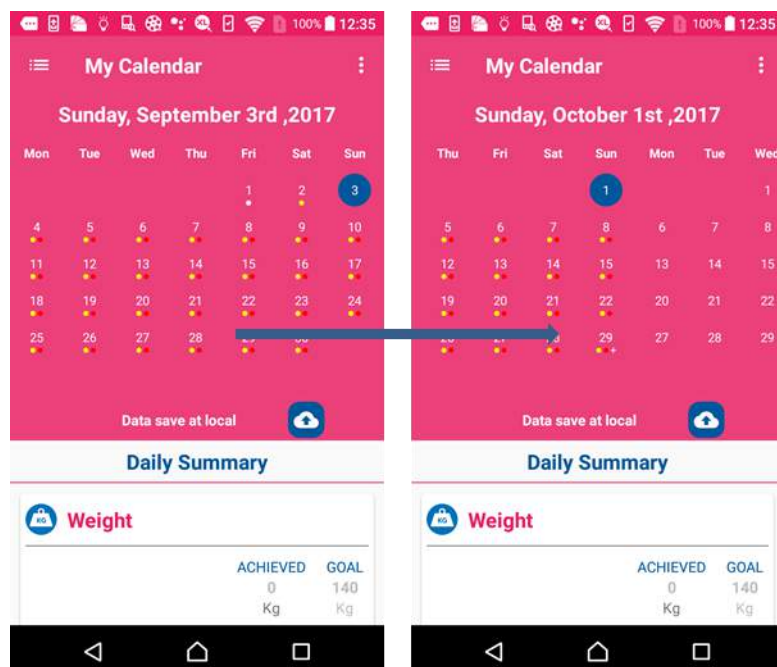


Figure 202: Calendar overview (1)

Action 3: Click a date in the Calendar and check the daily summary

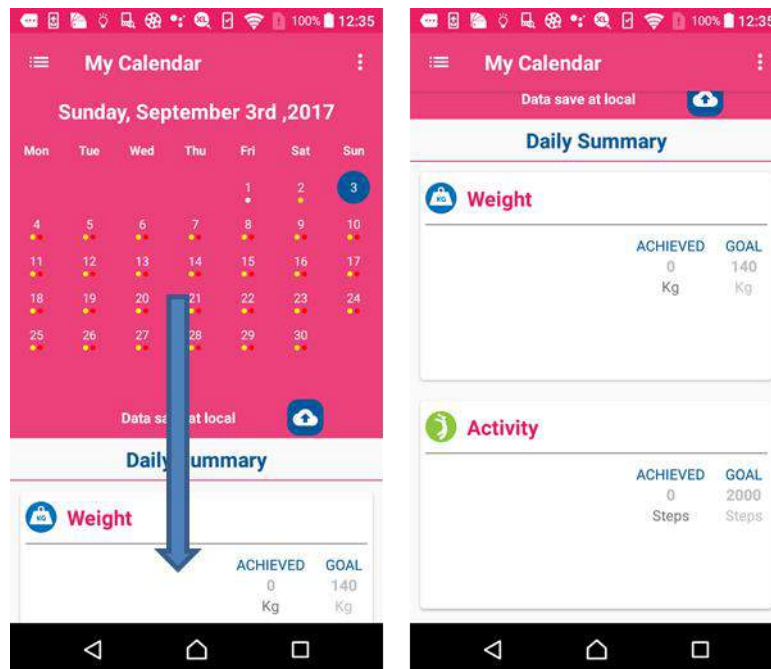


Figure 203: Calendar overview (2)

5.4.9. Access Profile Function

Action 1: Click the Profile in the left menu

Action 2: if any new changes on the Website, the profile will ask user to choose download the changes or use the local one.

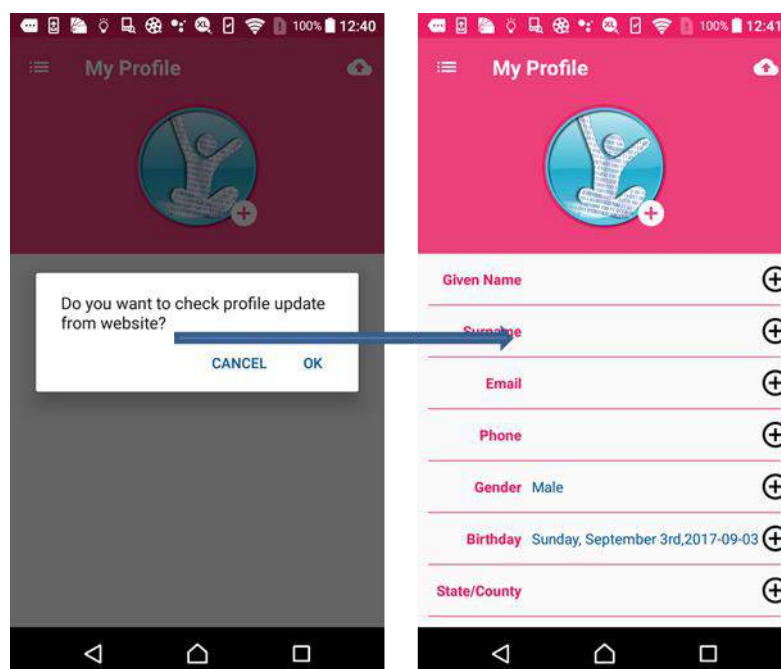


Figure 204: Profile updating

Result: Sync function works fine.

5.4.8.1 Update Profile Picture

Action 1: Click “+” to add the user profile picture.

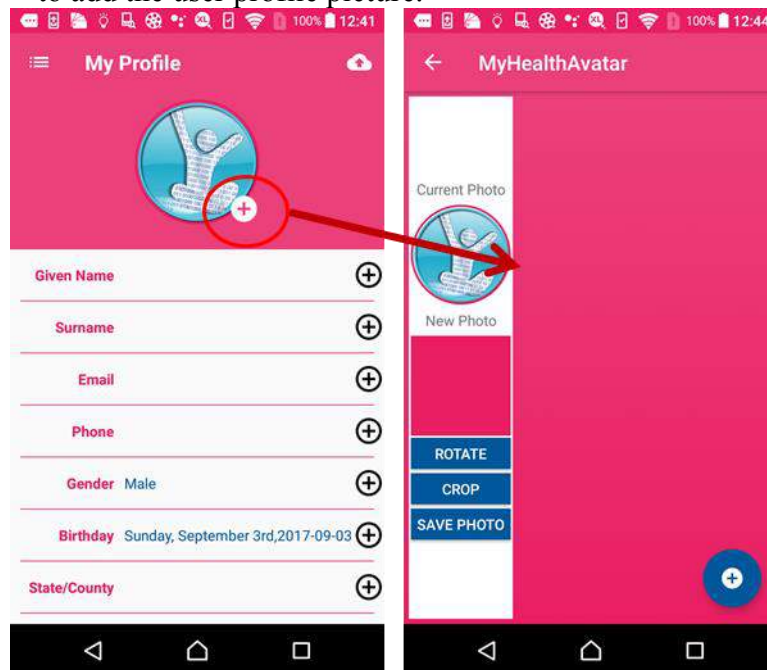


Figure 205: Profile image updating (1)

Action 2: Click “+” button to load a picture and give permission to the App.

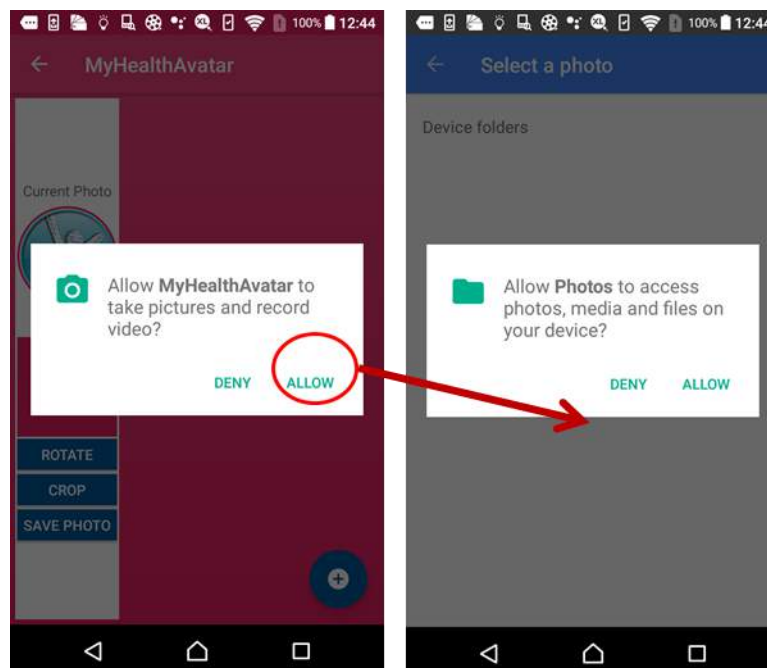


Figure 206: Profile image updating (2)

Action 3: Click the cropped picture and save the photo. Then go back to the main page of Profile.

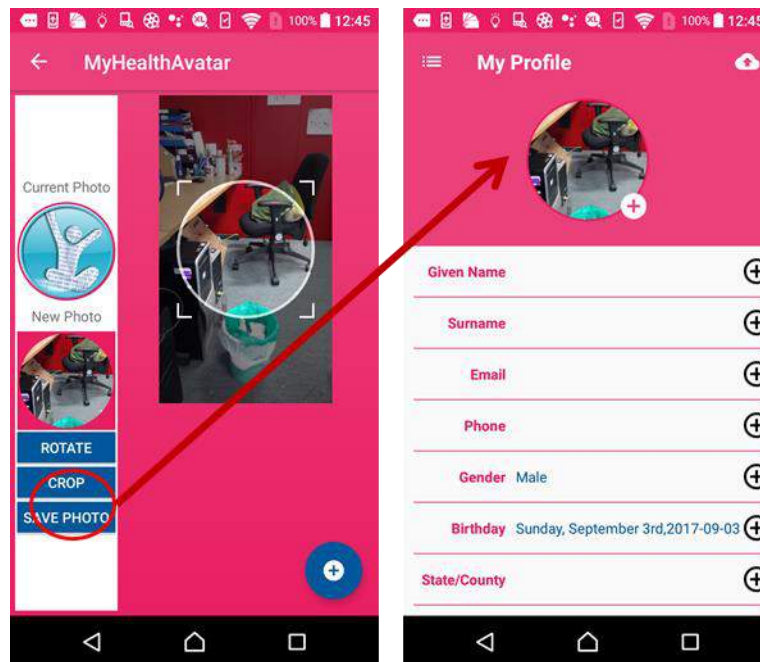


Figure 207: Profile image updating (3)

Result: Profile Picture function works fine.

5.4.8.2 Update User Profile

Action 1: Click “+” to edit profile item.

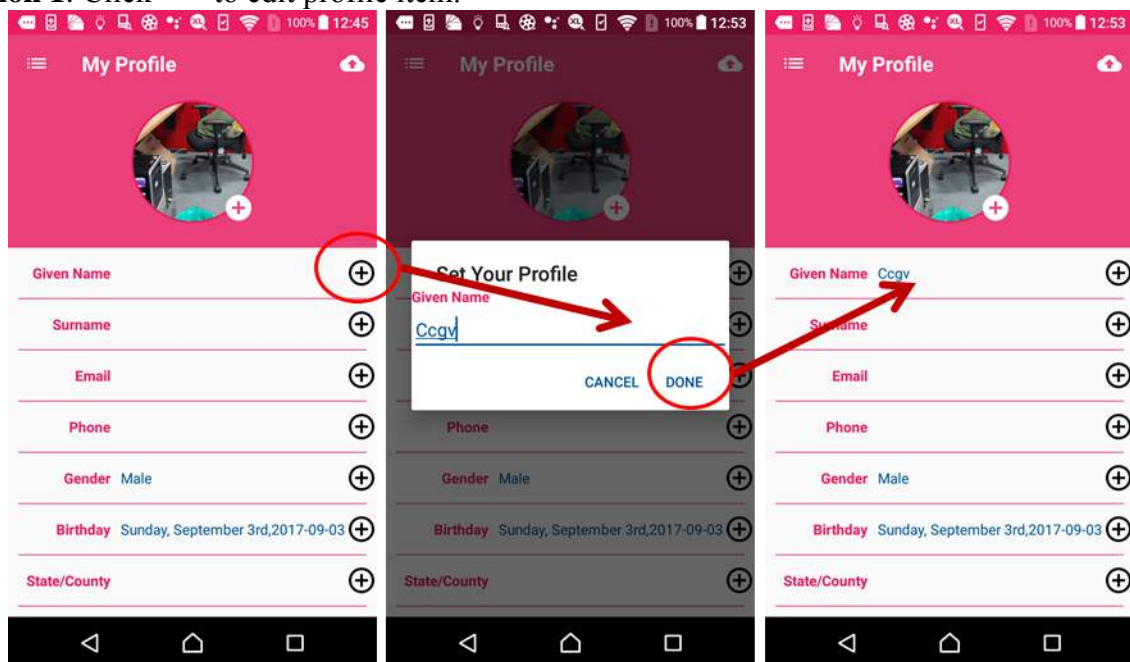


Figure 208: General profile overview

Action 3: Upload the new changes to server by click the “Cloud” Button.

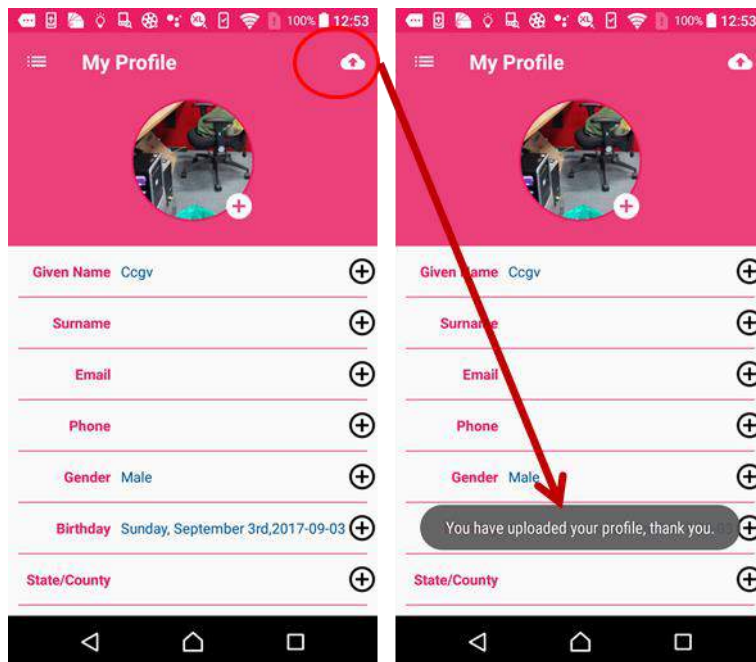


Figure 209: General profile upload

Result: Profile function works fine.

5.4.10. Access Sharing Function

Action 1: Click the Sharing in the left menu

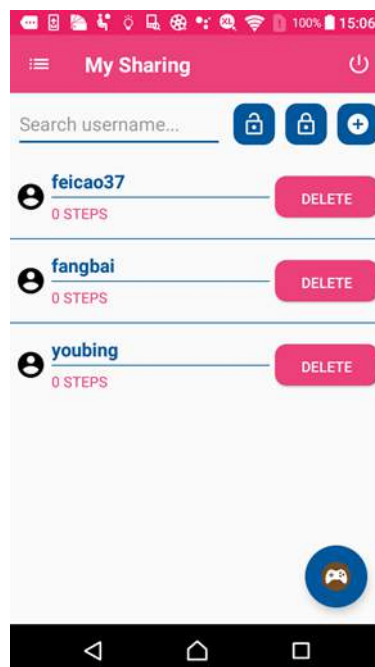


Figure 210: Sharing overview

Action 2: Enter the user name that you want to find and click find button.

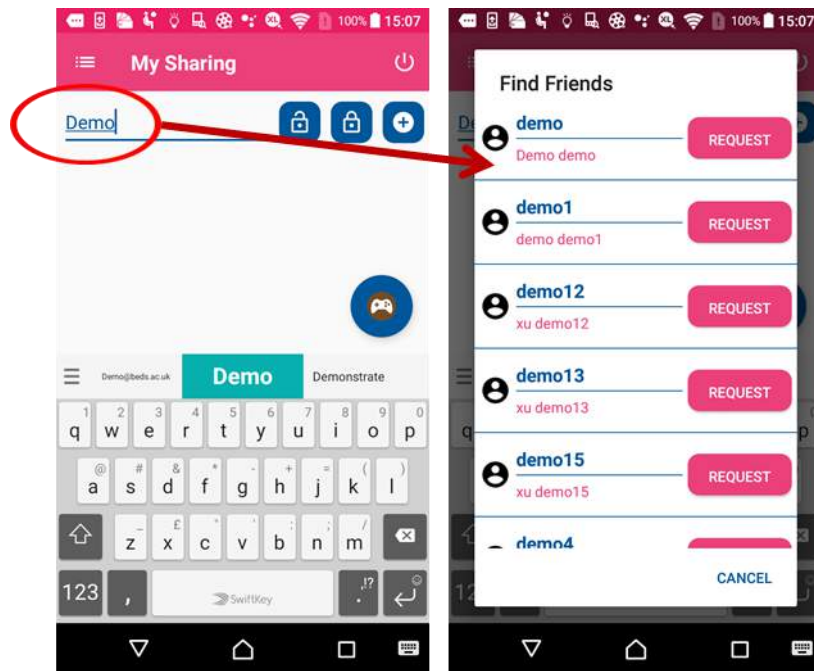


Figure 211: find the friend list

Action 3: Click “Add” button to send request to the user who you want to add as friend. Once the friend user confirm that request, she/he will be appear in the list.

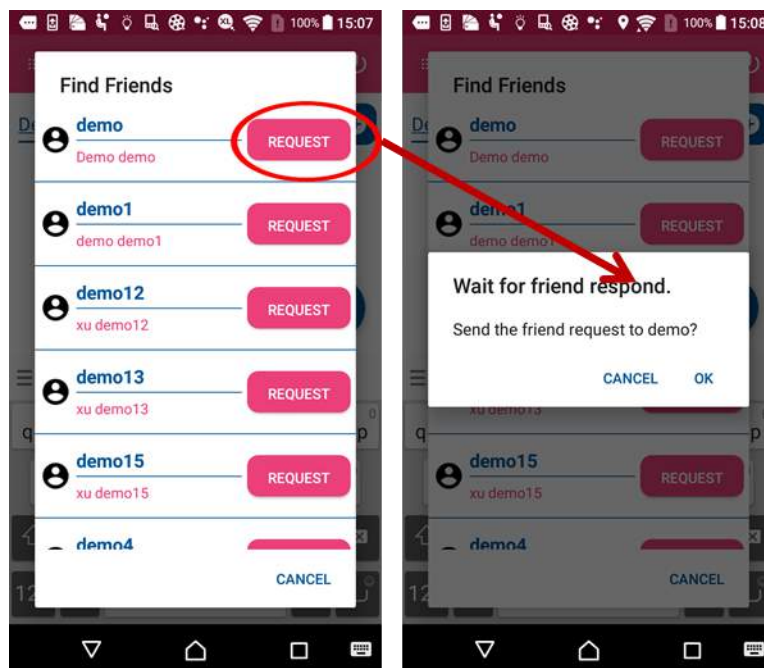


Figure 212: add new friend

Result: Sharing function works fine.

5.5. Installation and configuration guidelines

App correctly running with all features on all Android 5, 6, 7 devices.

General permission——Basic access permission, no need patients give the permission manually. The app can access the device features.

6. Device network access permission
7. Device internet access permission
8. Device receive boot completed access permission
9. Device vibrate access permission
10. Device wake lock access permission

Patient permission——App needs patients to give the permission manually in order to access the device features.

- App can access the device GPS sensor feature
- App can access the device storage feature
- App can access the device camera feature
- App can access the device audio record feature

6. Serious game for adult cancer patients

6.1. *Intended Purpose*

The aim of the serious game is to promote self-efficacy, i.e. the belief of the patients to be able to manage and to face their disease, also to help the patients deal with the psychological dimension of their disease, promote a healthier lifestyle and disease management. The serious game for adults puts the user in the role of an authority figure who manages a small town where they help residents with their cancer related lifestyle problems. The user is forced to think critically and strategically in order to balance their resources and time, while also viewing the issues surrounding their cancer from a different perspective. The game is intended to promote the concept that with good management of a person's cancer disease they can still be happy and achieve a sense of wellbeing.

6.2. *Provided functionality*

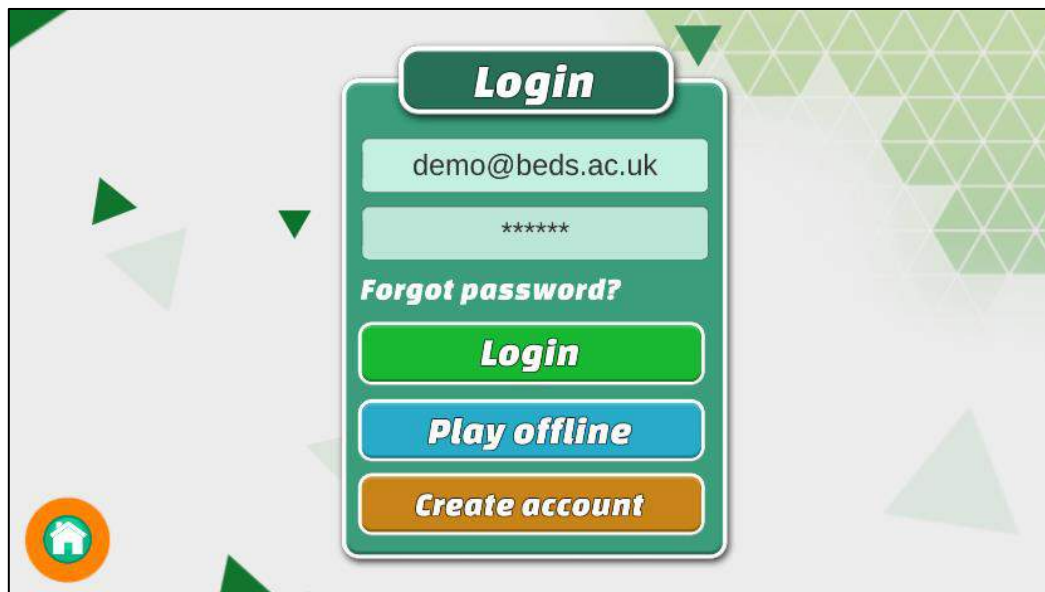
6.2.1. **Multi Language Support**

The game uses an xml language file to populate language strings at runtime, so the game can easily add extra languages by simply providing new entries within the language xml file. The game automatically interrogates this language file to populate the language options at the start of the game.



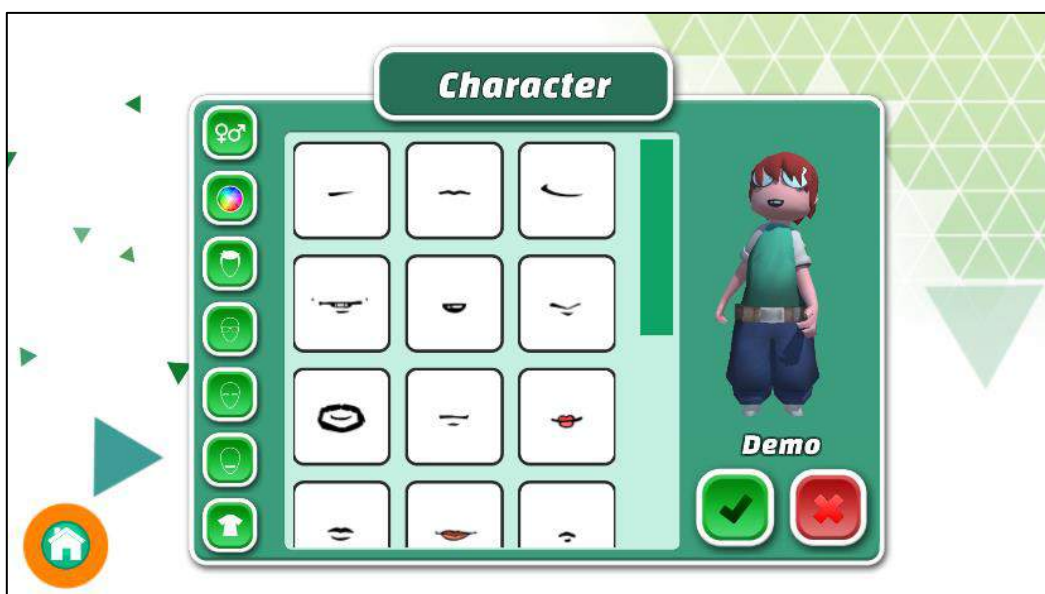
6.2.2. **Account Management**

The game provides online and offline play, to facilitate iMC audit tracking, social features, and account management features such as profile migration to other devices the user must log in. Once a user has connected their iMC account the application will remember them and automate the login process in future sessions.



6.2.3. Customisable Avatars

The early design specifications of the serious game for adults required the ability to make a customisable avatar, the game realises this feature in a full 3D character creation tool. The character creator allows for the basic customisation of character attributes such as hair style, gender, skin colour, eyes, and mouth. The character creator also allows the user to equip items won through play. The characters are very simple in design, this helps improve efficiency on mobile devices as the character is only visible as a small element in the main 3D game.



6.2.4. Social Interactions

Accessible from within the main game mechanics is the social interface, this interface is only functional if a user has signed in with their iMC credentials or created a new iMC account. Social interactions are asynchronous by design, this removed the need for an active connection to a game back-end system which would have added very little in terms of game play. The social systems currently allows a user to search for other platform users and send them a friend request, accept incoming friend requests, and view the avatars of their friends.



6.2.5. iMC Rewards System

The iMC rewards feature is designed to convert any calorie burn a user has recorded in their PHR into points that can be spent in game on a prize wheel. The points are converted into a spin token, this token is consumed when the user spins the wheel, and the user gets a reward when the wheel stops. The feature is intended to incentivise the user to engage with the iMC platform more, as engagement equates to prizes.



6.2.6. 3D Game world

The main game was developed in Unity as a fully realised 3D town. The game assets are simple and colourful. The Unity game engine allows the user to navigate around the town using a simple touch interface.



6.2.7. Main Game Mechanics

The main game mechanics of the serious game for adults are time and resource management within a simulation. The user must help a fictional citizen of their small town attain a state of wellbeing. The user decides the daily activities of their citizen and must balance their life and resources to achieve wellbeing in the fastest possible time. The game has no loose condition, instead simply reducing the completion rating from gold to silver and then bronze. The user is forced to think critically about time management, it is hoped that this critical thinking will be applied to their own life.



6.2.8. Trivia Mini-game

The trivia mini-game is a mechanic added to expose a user to a range of health and lifestyle trivia knowledge. A user can engage with the trivia game when they would like to remove a challenge from the challenge bar without completing the challenge activities, this may be because the challenge is counterproductive to the citizen. The game itself leverages a large database of trivia, and picks three questions at random, the user must answer at least two questions correctly to complete the challenge using their general knowledge.



6.2.9. Feedback System

The game provides and in build feedback questionnaire for all the main game features, the interface is simple to use and provides a free text box in addition to the standard five questions. The feedback forms are all unified in appearance and question content, only made unique by

6.2.10. Game Back-end Systems

The serious game for adults has a dedicated back end server which handles all game related data that is not integral to the users of the iPHR. The server acts as the bridge between the mobile game client and the iPHR system. Authentication, feedback, and audit data is all passed through the game back end.

6.3. Technical implementation

6.3.1. Introduction

The serious game for adults is intended to be played on mobile devices such as smart phones or tablets running the Android operating system. Figure 213 Serious game deployment view diagram

displays the overall configuration, highlighting the games back end server as an intermediary between the game client and the iMC platform.

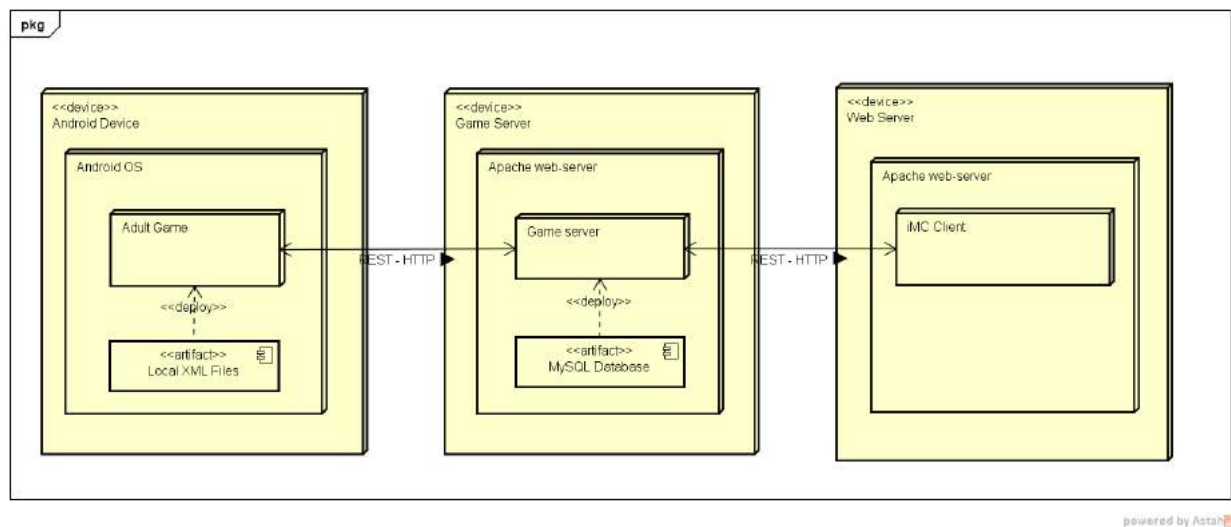


Figure 213 Serious game deployment view diagram

Below in Figure 214 Serious game runtime view diagram, there can be seen a simplified overview of user interaction and its scope within the iMC framework.

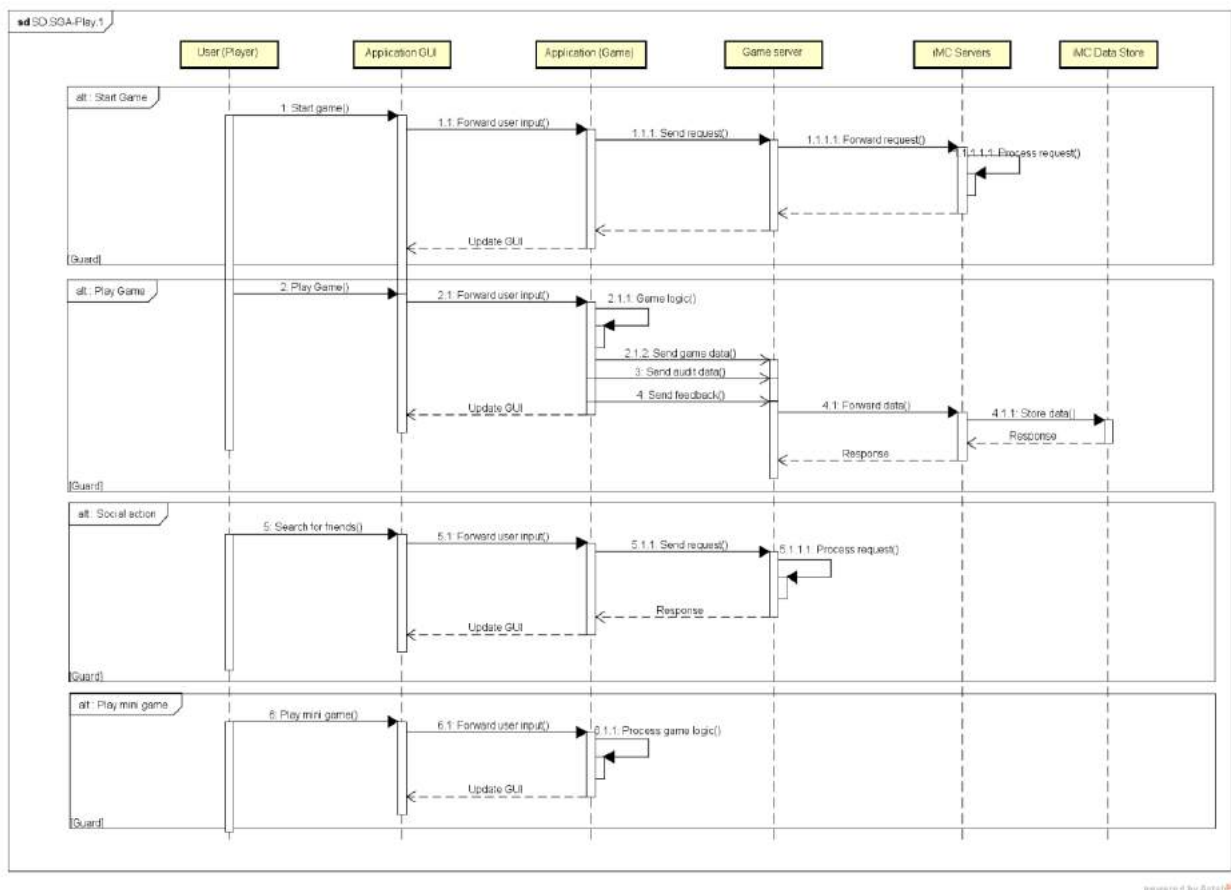


Figure 214 Serious game runtime view diagram

The game back end server is responsible for storing game specific data that is unrelated to a patient's health (Game state and avatar configuration), as well as providing the social matchmaking functionality without exposing sensitive patient data.

The data for the game is stored in MySQL v5.7.x database, audit information is stored in iPHR central store through API calls. There is also local debugging output to file for administration and debug purposes, the debugging data does not contain any user privacy data. See Figure 215 Game server architecture, for information about the structure of the game back end server.

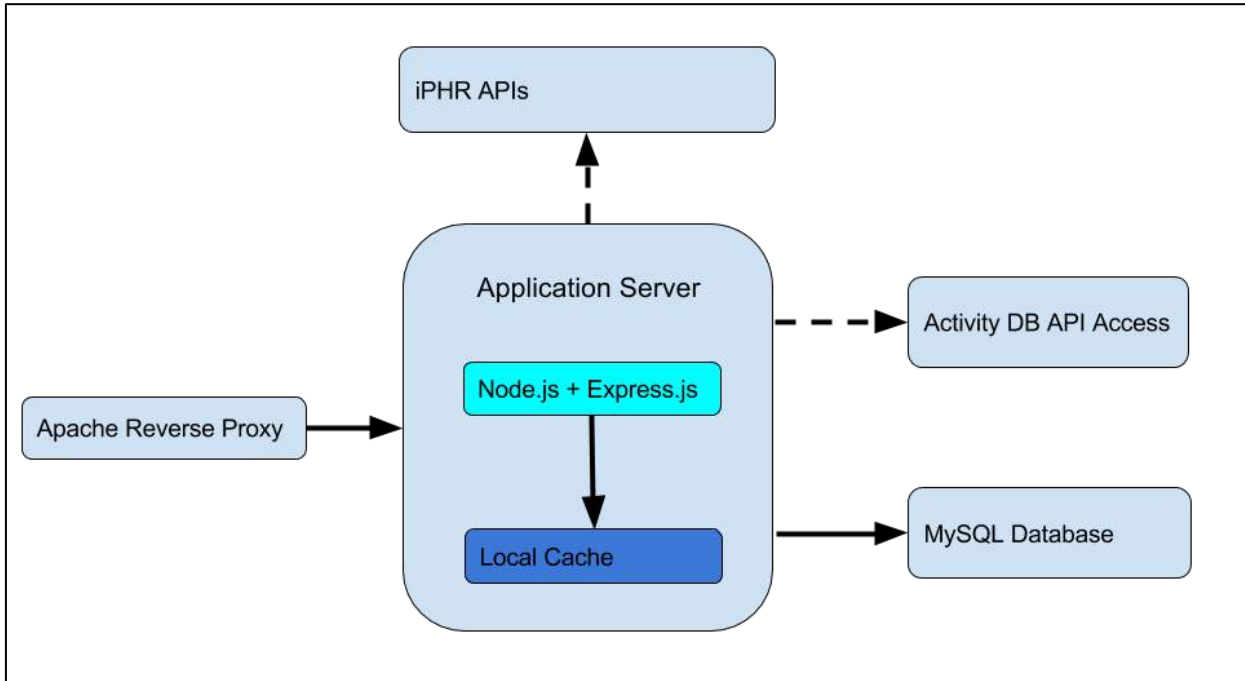


Figure 215 Game server architecture

6.3.2. Interfaces

Below are all the interfaces implemented on the game back end server:

Interface	Address
Log in	GET http://live.ccgv.org.uk/signin
Sign up	GET http://live.ccgv.org.uk/signup
Upload audit data	POST http://live.ccgv.org.uk/api/game/user/audit/
Upload feedback	POST http://live.ccgv.org.uk/api/game/user/questionnaire/
List feedback	POST http://live.ccgv.org.uk/api/game/user/questionnaire/
Download player profile	GET http://live.ccgv.org.uk/api/game/user/data/
Upload player profile	POST http://live.ccgv.org.uk/api/game/user/data/
Update player profile	PUT http://live.ccgv.org.uk/api/game/user/data/
Delete player profile	DELETE http://live.ccgv.org.uk/api/game/user/data/:id/
Friend search	GET http://live.ccgv.org.uk/api/game/user/search/
Get friends list	GET http://live.ccgv.org.uk/api/game/friend/list/
Friend request	POST http://live.ccgv.org.uk/api/game/friend/request/
Friend accept	POST http://live.ccgv.org.uk/api/game/friend/accept/
Friend reject	POST http://live.ccgv.org.uk/api/game/friend/reject/
Friend revoke	POST http://live.ccgv.org.uk/api/game/friend/revoke/

Friend reset	POST http://live.ccgv.org.uk/api/game/friend/reset/
Get activity data	POST http://live.ccgv.org.uk/api/game/user/activity/

6.3.3. Databases and structures

The serious game for adults running on Android within the Unity game engine utilises a system of serialised text assets for game state saving and loading. The decision to use XML serialised game files instead of an SQL database solution was made to reduce the games dependency on external libraries and third-party solutions. The relatively small amount of data simply didn't justify a more complex database solution. The tables within the SQL database on the server are listed below:

Table Name	Example
IMC_GameFriends	<pre>IMC_GameFriends CREATE TABLE `IMC_GameFriends` (`id` int(11) NOT NULL AUTO_INCREMENT, `acceptedAt` datetime DEFAULT NULL, `hasAccepted` tinyint(1) NOT NULL DEFAULT '0', `hasRejected` tinyint(1) NOT NULL DEFAULT '0', `hasRevoked` tinyint(1) NOT NULL DEFAULT '0', `requestMessage` text, `usernameFrom` varchar(255) DEFAULT NULL, `usernameTo` varchar(255) DEFAULT NULL, `createdAt` datetime NOT NULL, `updatedAt` datetime NOT NULL, PRIMARY KEY (`id`)) ENGINE=InnoDB AUTO_INCREMENT=25 DEFAULT CHARSET=latin1</pre>
IMC_GameUserData	<pre>IMC_GameUserData CREATE TABLE `IMC_GameUserData` (`id` bigint(20) NOT NULL AUTO_INCREMENT, `data` longtext, `type` varchar(255) DEFAULT NULL, `username` varchar(255) DEFAULT NULL, `timeUpdated` datetime DEFAULT NULL, `timeCreated` datetime DEFAULT NULL,</pre>

	<pre> `timeModified` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP, PRIMARY KEY (`id`)) ENGINE=InnoDB AUTO_INCREMENT=282 DEFAULT CHARSET=utf8 </pre>
--	--

6.3.4. Data Protection

A summary of key points about data protection implemented in the serious game:

- Anonymous user registration as a privacy preserving mechanism.
- Data storage is physically and technically secure.
- System follows ISO/IEC 27002:2013 in regard to best practice.
- System puts control of access to data in the hands of the users.
- Users own their own data and can close their accounts and request all data be destroyed.
- System uses JSON based identity protocol in its security model.
- HTTPS is the preferred method of communication.

The game client itself running on the Android OS and further secures user's data by storing no critical or sensitive data within the game. The Unity project also stores user data in its application file path, effectively sandboxing the game files in the Android environment from other apps or processes. The data could be further encrypted if necessary, but at this time the non-sensitive nature of this data doesn't warrant the extra security.

6.4. User manual

6.4.1. Application launching

Begin by locating the iMC Serious Game for Adults launch icon on your Android mobile devices desktop, or through the applications menu on your device.

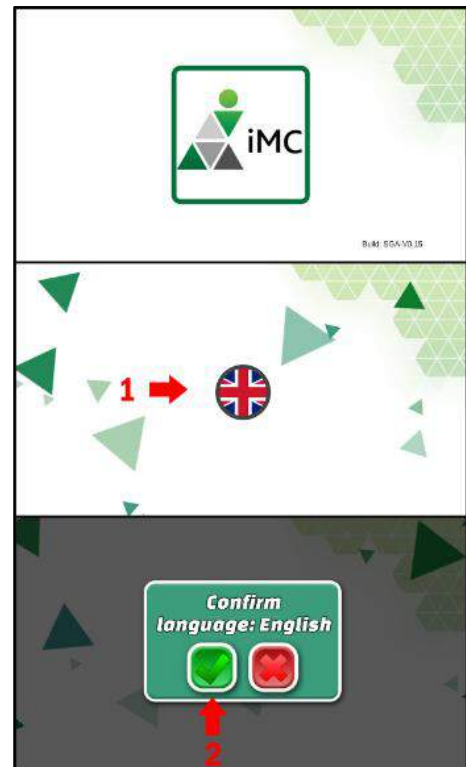


6.4.2. Language Selection

On first run you will see the splash screen followed by the language selection screen, select the icon that represents your chosen language (1).

A dialog will appear prompting you to confirm your selection, click the accept button (2) to proceed.

The screen will darken and a loading screen will appear.



6.4.3. Login & Account Management

After language selection you will see the login page, if you already have login credentials then enter them into the fields provided (1).

If you are using a device which has already been used for this game or you or anyone else have played this game before with this account you will see a dialog informing you of existing game data (2).

If you see this dialog (2) select the Red Cross button to reset the game for a fresh start, or click the green tick to download the existing game profile.



6.4.4. Character Creation

Once you are logged in you will see the main menu, click new game.

You will arrive at the character creator, you will need to enter a character name in the input box (1).

After entering a name select the green tick (accept) button (2).

After accepting the name, you will have access to the character creator, also note that your name is now present under your avatar, you can still click and edit this later (3).



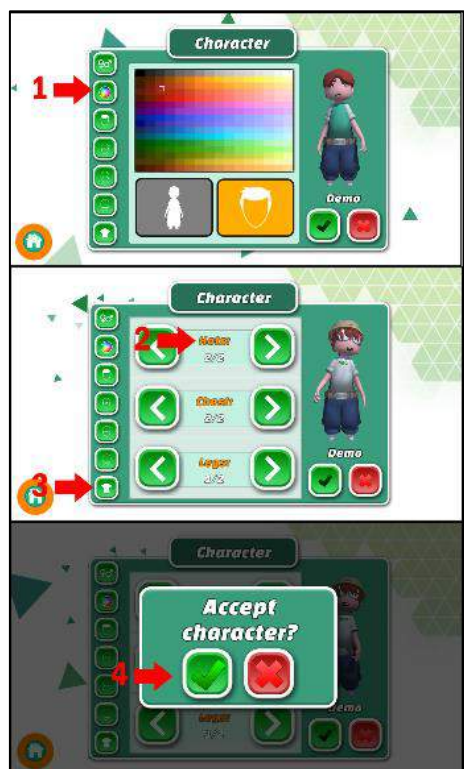
6.4.5. Character Customisation

Now you can experiment with the character customisation features to build a likeness or a fun avatar.

Select the colour wheel tab button (1), here you can colour the characters skin and hair.

Select the clothing tab button (3), here you can cycle through clothing options. This list of clothing items grows as you complete levels and unlock new pieces.

Accept your character by pressing the green accept button, you will then need to confirm this action (4).



6.4.6. Level Selection

On a fresh game start, the only level available to you is level one. Select it now to load it.

Level icons also show a star rating that indicates the gold, silver, or bronze award for completion.

You can replay a level by clicking it again.



6.4.7. Game Introduction

Upon level loading completion, and assuming a fresh new play start (No existing user data) you will see the new simplified tutorial (1).

To go through the tutorial, click the right direction chevron button (3). Some pages in the tutorial create a pulsating green indicator circle around the element they are referring to (2).

Click through and read carefully the instructions. This tutorial has been reduced to be as short as possible while still conveying the main game mechanics.

After the tutorial is complete a dialog appears with the avatar of the current level's citizen patient (4), they will be wearing an item of clothing which will become the level completion reward



6.4.8. Main Game Mechanics

The tutorial has requested that you select any building in the town to see available activities, go ahead and click one. In this example, we have clicked on the fast food restaurant and have selected the activity Salad (1).

Clicking an activity creates a confirmation popup, click accept (2).

Once an activity is selected and accepted, wellbeing is generated by the lifestyle balance meters at the bottom of the screen (3) the amount each generates is visualised and moves from the meter to the wellbeing indicator on the top, middle of the interface.



6.4.9. Progression & Winning

Continue doing activities to keep generating wellbeing, both positive and negative amounts. Since each activity takes time, eventually a game day will have passed.

At the end of each day you earn wages (2) and you are informed how much wellbeing you generated for your citizen during the day (3).

To win you must both get your citizen's wellbeing to 100% and clear all the challenges (1).

Once wellbeing is 100% or more, and all the challenges are completed then you will have finished the level. The level completion dialog appears and informs you how well you did with a gold, silver, and bronze star rating (4).

Click the big green accept button to progress.



6.4.10. Building new Buildings

Some levels (after level 1) will have empty plots of land (1), here you can build new buildings or build copies of existing ones.

Click the empty plot (1), select a building (2), click accept (3).

If you can afford the building the plot will become a building site for the indicated number of in-game hours, after that period of time the building will transform into the completed version of the building you selected (5).



6.4.11. Social Interface

During gameplay of any level click the social interface icon (1).

The social interface launches (2). This interface has an option to view any existing friends (3), view any new friend requests (4) and search for new friends (5).

Enter the name of an account into the search field to the left of the search button (5). Click the search button. If a result is returned you can click the green accept button to the right of their name to send them a friend request.

You will be prompted to confirm this action, click accept (6) to send the request. If the user accepts you they will appear automatically in your existing friends list when you next load this interface.

If you do not have anyone to search for feel free to search for the character "Demo".



6.4.12. iMC Rewards Interface

Click the iMC rewards icon (5) to launch the iMC interface.

The interface loads any calorie burn recorded by other iMC platform tools as activity points, if you do not have any points then it is likely you have not linked any activity trackers or sensors to your iMC account.

If you do have activity data then clicking the prize box button (1) will convert 500 points into a spin on the prize wheel.

Spin the wheel by pressing the spin button (2).

If you win any building credit on a spin, or earn any elsewhere in the game you can use it by pressing the building credit button (3). Any building currently under construction (4) will be sped up, consuming your building credit.



6.4.13. Trivia Mini-game

The trivia mini-game allows you to skip a challenge if it is proving difficult or damaging to your citizen. Click the challenge icon (1) to get more detailed information about the challenge.

To launch the trivia game, press the mini-game button labelled skip (2).

Select answers from the multiple-choice responses to answer the questions (3).

Click submit to get feedback on your selection, then click next to proceed.

If you answer two of the three questions correctly then you will clear the challenge. Click finish to get your reward (4).



6.4.14. Feedback Dialogs

After using key features of the game, you will be prompted to fill in a feedback form. Each form is titled after the module it refers to.

Select a star rating for each question (1).

Select “Send feedback” to submit the form, this form will not appear again for at least one week.

6.5. Installation and configuration guidelines

At time of writing (August 2017) the Game’s Android mobile application is only available through the iMC file repository as a .apk file, and through the iMC auto application download utility.

7. Serious game for paediatric cancer patients

The Serious Game for kids is developed by Serious Games Solutions/ Promotion Software.

The app is developed for mobile platforms (iOS, Android) and includes a shooter game with several social multiplayer features, and also a match three game for other target audiences.

7.1. Intended Purpose

The patients fight as a virtual submarine through several parts of the human body to fight cancer cells. Relatives and friends (supporters) will play for them, so that they get extra power (weapons, shields and more) within the game. The idea behind the game is to give psycho-emotional support to the young patients, so that they feel supported and to give them an active role in fighting the cancer cells. The message in the Serious Game is clearly “There is a weapon against cancer, and I have to focus on using it”.

7.2. Provided functionality

7.2.1. Decision menu

Here the central decision within the game has to be made: If you play the patient or the supporter of a patient.

The main idea behind this is that the game is mainly made for children with cancer (patient). Supporters could be relatives, friends – and other children with cancer.

The menu forces an active decision which role the player chooses within the game.



Figure 216: Game playing by patients or their supporters (friends or family).

7.2.2. Patient shooter game



Figure 217: Patient shooter game in action.

The patient shooter game is the core part of the app. Here the players (mainly the children with cancer) dive with their vehicle through several parts of the body and shoot cancer cells. There are obstacles and missiles from the enemies, which try to stop the players from their aim. Finally each level ends with a giant boss cancer cells, which offers a special hard attack for the players.

The players can choose from their extras to face the increasing challenges within the game. The extras come from their supporters. The more the friends and relatives play for the children, the stronger they are in the game.

7.2.3. Supporter shooter game

The supporter mode of the shooter game is similar to the patient mode, except the aspect that the players don't have any extras there for themselves, they collect them for their patients. The players cannot play this mode without choosing another player.

There are two supporter modes, one for family members and one for friends. The main difference between these two mode that the tutorial is different between both modes.

7.2.4. Game backend

The game backend is the link between supporter and patient mode, and it's also the link between the app and the iPHR. The game backend contains player data. The app sends all supporter game

results to the backend and from there the results are sent to the target patients. Also the game backend interacts with the iPHR servers for authentication, audit trail and configuration data.

7.3. Technical implementation

7.3.1. System concept and internal architecture

The “serious game for kids” is played on mobile devices as smartphones and tables. Supported are newer devices, which allow 3d graphics.

The game for kids also is connected to a game backend in the internet. This server is responsible for all the data transfer between explorers and supporters, and all account interaction. The game backend does not contain the account database on its own, it negotiates with the iMC platform.

All account information is stored in the PHR DB. From there you can choose several options, which are made for parents to control the app (like “enabling/disabling supporters” and more).

The game backend transfers all relevant ingame events to the audit trail.

A static connection to the internet is not necessary at any time of the use, only for registration and the first login it is mandatory. Of course the use of the social features needs internet connection too. But the use case is defined in a way that the child takes the smartphone to the hospital and has only limited access to WIFI or Internet. To make this possible, the social interaction is buffered whenever a device cannot connect to the internet. When a connection can be established, data is uploaded and downloaded.

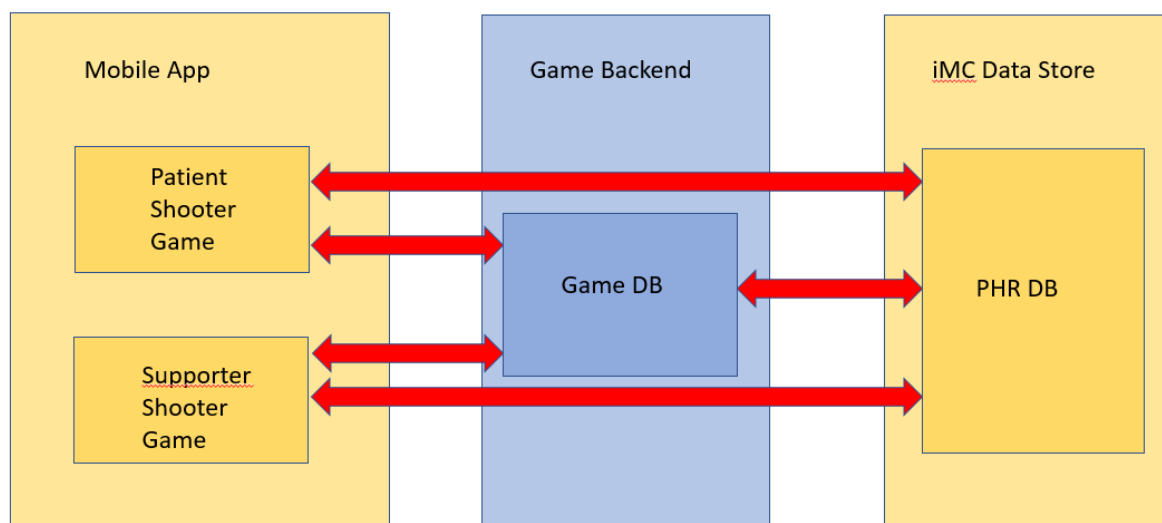


Figure 218: Serious Game for Kids – internal architecture.

7.3.2. Interfaces

Communication App - Game Backend

```

internal static void GetAccessToken(LoginInputModel loginModel,
Action<PlayerData> onSuccessCallback, Action<int, string> onFailureCallback)

private static void GetAccessToken(NameValueCollection postData,
Action<PlayerData> onSuccessCallback, Action<int, string> onFailureCallback)

private static void GetAccessToken(NameValueCollection postData,
Action<PlayerData> onSuccessCallback, Action<int, string> onFailureCallback)
  
```

```

private static void CreateAccount(NameValueCollection registerModel,
Action<PlayerData> onSuccessCallback, Action<int, string> onFailureCallback)

internal static void SavePlayerData(PlayerData playerData, Action<PlayerData>
onSuccessCallback, Action<int, string> onFailureCallback)

private static void SavePlayerData(NameValueCollection playerData,
Action<PlayerData> onSuccessCallback, Action<int, string> onFailureCallback)

private static void SavePlayerData(NameValueCollection playerData,
Action<PlayerData> onSuccessCallback, Action<int, string> onFailureCallback)

internal static void GetAllPlayers(Action<List<PlayerData>>
onSuccessCallback, Action<string> onFailureCallback)

internal static void SupportPlayer(string playerId, Action<PlayerData>
onSuccessCallback, Action<int, string> onFailureCallback)

private static void SupportPlayer(NameValueCollection patientId,
Action<PlayerData> onSuccessCallback, Action<int, string> onFailureCallback)

internal static void RemoveSupporter(string supporterId, Action<PlayerData>
onSuccessCallback, Action<int, string> onFailureCallback)

private static void RemoveSupporter(NameValueCollection supporterId,
Action<PlayerData> onSuccessCallback, Action<int, string> onFailureCallback)

internal static void GetInventoryItems(Action<List<InventoryItem>>
onSuccessCallback, Action<string> onFailureCallback)

internal static void AddInventoryItems(string itemTypeId, int count,
Action<List<InventoryItem>> onSuccessCallback, Action<int, string>
onFailureCallback)

private static void AddInventoryItems(NameValueCollection itemData,
Action<List<InventoryItem>> onSuccessCallback, Action<int, string>
onFailureCallback)

internal static void RemoveInventoryItems(string itemTypeId, int count,
Action<List<InventoryItem>> onSuccessCallback, Action<int, string>
onFailureCallback)

private static void RemoveInventoryItems(NameValueCollection itemData,
Action<List<InventoryItem>> onSuccessCallback, Action<int, string>
onFailureCallback)

internal static void SendPresents(string itemTypeId, string userId, int
count, Action onSuccessCallback, Action<int, string> onFailureCallback)

private static void SendPresents(NameValueCollection itemData, Action
onSuccessCallback, Action<int, string> onFailureCallback)

internal static void GetActivityForPlayer(Action<List<Activity>>
onSuccessCallback, Action<string> onFailureCallback)

internal static void CreateActivity(string userId, int activityType, string
serializedObject, Action onSuccessCallback, Action<int, string>
onFailureCallback)

private static void CreateActivity(NameValueCollection activityData, Action
onSuccessCallback, Action<int, string> onFailureCallback)

internal static void SaveActivity(string activityId, string serializedObject,
Action onSuccessCallback, Action<int, string> onFailureCallback)

private static void SaveActivity(NameValueCollection activityData, Action
onSuccessCallback, Action<int, string> onFailureCallback)

internal static void SaveActivityAsSeen(string activityId, Action
onSuccessCallback, Action<int, string> onFailureCallback)

private static void SaveActivityAsSeen(NameValueCollection activityId, Action
onSuccessCallback, Action<int, string> onFailureCallback)

```

```

internal static void SaveQuestionnaireData(string data, Action
onSuccessCallback, Action<int, string> onFailureCallback)

internal static void repostRequests(InvokeProcess req, RequestData reqData =
null)

private static SSG.Process getandStartNextRequestprocess()

internal static void RepostRequestsWithToken(Action<string> onTokenExists,
Action<bool> onFinished, KeyValuePair<string,string> login)

internal static void onRequestSentFailure(int code, string message,
InvokeProcess inv)

private static JSONNode HandleResponse(string responseFromServer)

private static T XmlDeserializeFromString<T>(string objectData)

```

Communication Game Backend – iPHR Server

```

bool ImcForthHelper::getAccessToken()

bool ImcForthHelper::accountExists(const std::string& email)

bool ImcForthHelper::createAccount(const std::string& email, const
std::string& full_name)

std::string ImcForthHelper::getAccount(const std::string& email)

std::string ImcForthHelper::getAccount(const std::string& email)

bool ImcForthHelper::userExists(const std::string& username)

std::string ImcForthHelper::getEmailFromUser(const std::string& username)

std::string ImcForthHelper::getAccountState(const std::string& email)

bool ImcForthHelper::createUser(const std::string& email, const std::string&
username, const std::string& password)

bool ImcForthHelper::isPasswordCorrect(const std::string& email, const
std::string& password)

bool ImcForthHelper::createRecord(const std::string& email)bool
ImcForthHelper::sendAnsweredQuestions(const std::string& patientId, const
std::string& questionnaireId, const std::string& dateString, const
std::string& answers)

```

7.3.3. Database

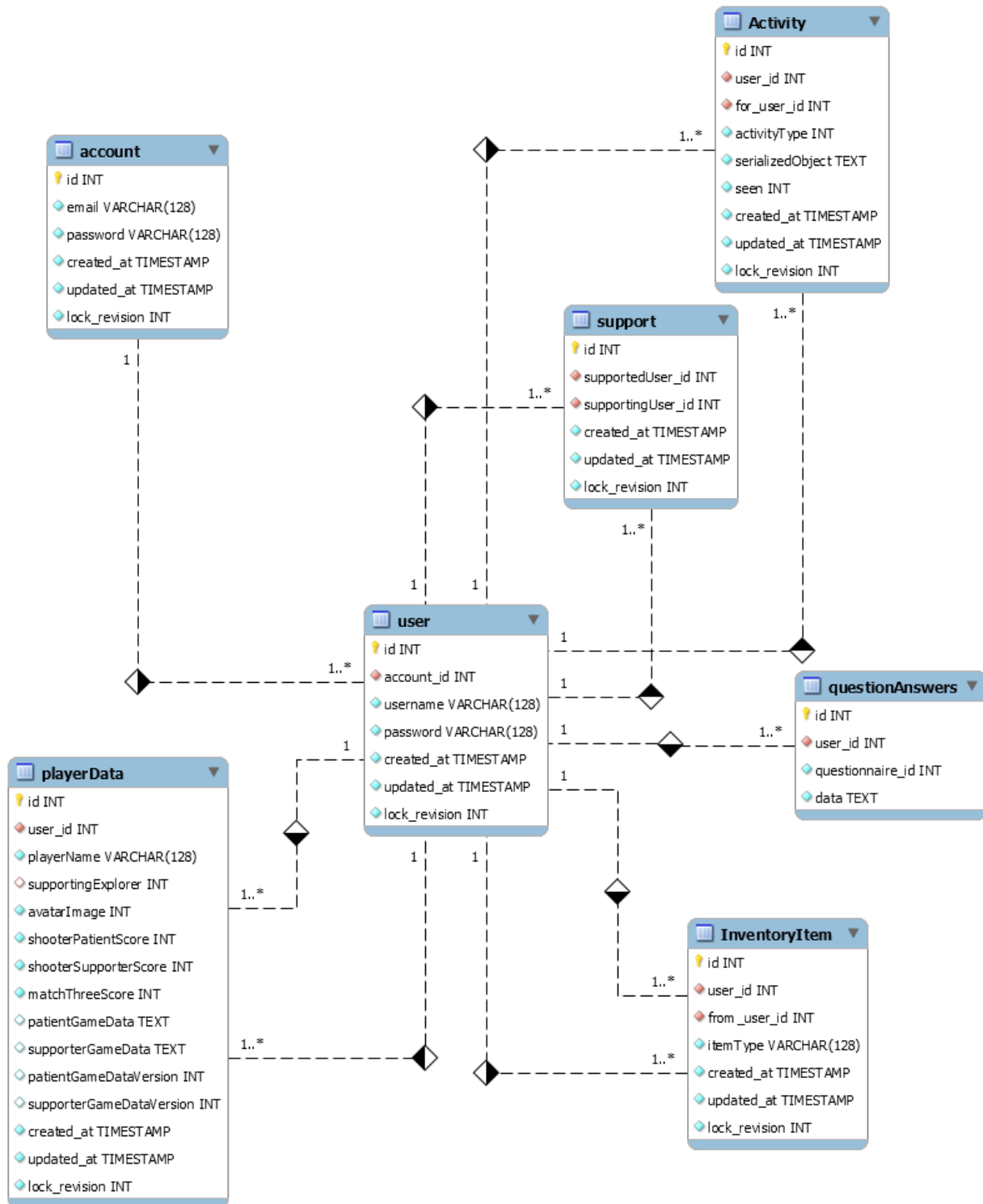


Figure 219: Database schema of game backend.

7.3.4. Data protection

Every player has its own account registered at the iPHR platform.

The data is protected by the system memory protection of the mobile operating systems. It is not possible to access it from other apps or processes.

The communication between app and backend, also the communication between backend and iPHR server is encrypted (SSL).

The app only handles game results. The most critical information is the email address of the user.

7.4. User manual

- The serious game for kids should be tested with Android mobile devices with an operating system greater than V4.0.3.
- The testing should be carried out on an internet connected device.
- To ensure a consistent testing environment it is recommended that the tester clear any application data generated by the IMC_Serious_Game application in the settings, this will reset any user changes and cause an “out of box” start-up.
- Format: only landscape
- This test scenario is designed for testing the online mode.

7.4.1. Starting the Serious Game for Kids (First Time)

Description: Test if game starts after installation.

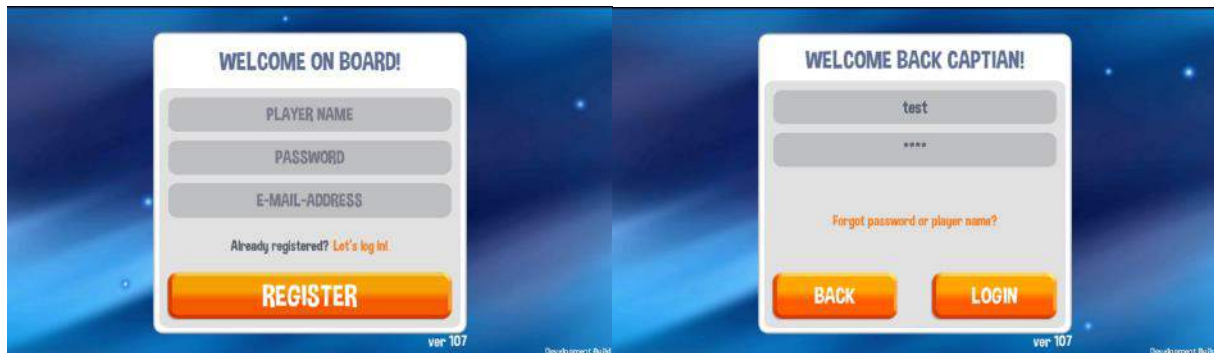
Action 1: Find the IMC_Serious_Game icon and single tap it to launch the application.

Result: Game starts /First screen - unity screen /than login-screen.

Notice: At the bottom of the screen, you will find the “fps and Version” of the current development build.

7.4.2. Login Screen Test

Description: Test if offline login is possible.



Action 1: Press “Back”

Result: You will be directed to the registration screen (via iMC Platform – currently deactivated).

Action 2: Press Register

Result: You will be directed to the Registration Menu

Action 3: Type Player name (min 5; max 15 characters)

Result: you will be directed to the login-screen

Action 4: type in invalid Password, E-Mail, and Name (both min 5; max 15 characters)

Result: defect note

Action 5: type in valid Password, E-Mail, and Name (both min 5; max 15 characters)

Result: you will be directed to the dashboard-screen

7.4.3. Choose Game – Mode



Description: Choose a Game Mode

Action 1: Choose “I’m a Friend of a patient” (“I belong to the family of a patient” is a similar mode with different tutorial text)

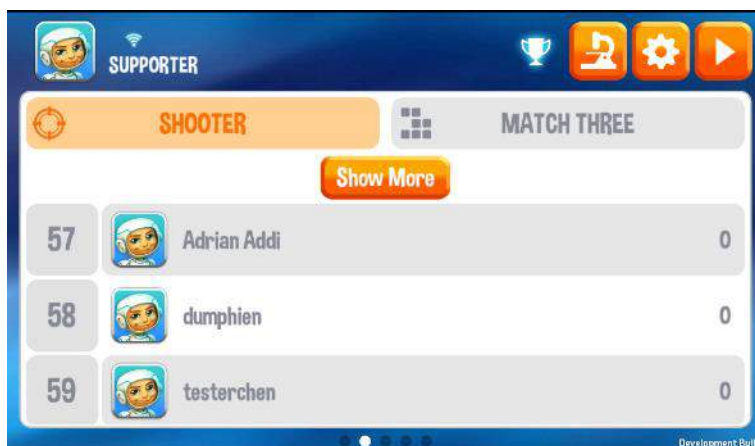
Result: You will be directed to the Supporter-Dashboard “News”

Action 2: Choose “I’m a patient”

Result: You will be directed to the Patient-Dashboard “News”

Notice: You can change the Game-Mode in the Option-Menu

7.4.4. Dashboard Test



Description: Test, if main menu and buttons (called dashboard) is fully functional

Notice: There 5 types of Dashboards:

- News
- Highscore
- Information
- Supporter (Patient)
- or supported Patients (Supporter)
- activated enemies (cancer cells)

There are 3 types of buttons:

- Catwalk (shows all cancer cells)
- Options
- Play Button
- Avatar Icon (press on it and you can change your player name)

Action 1: swipe all Dashboards

Result: Dashboards smoothly move. Highlighted Icons change (left near by orange buttons)

Action 2: use the buttons in the different dashboards

Notice: There are 3 Types of Buttons (orange, top, right, corner): Catwalk, Option, Play

Result: buttons function

7.4.5. Main Buttons – Dashboard: Option Test



Description: Test: is changing the options fully functional?

Notice: Release finger from screen will slow down application

Action 1: Press option button in the dashboard menu

Result: options will be opened

Action 2: choose a language

Result: language changes

Action 3: choose Control

(Notice: changed control-systems are used in the shooter game)

Notice:

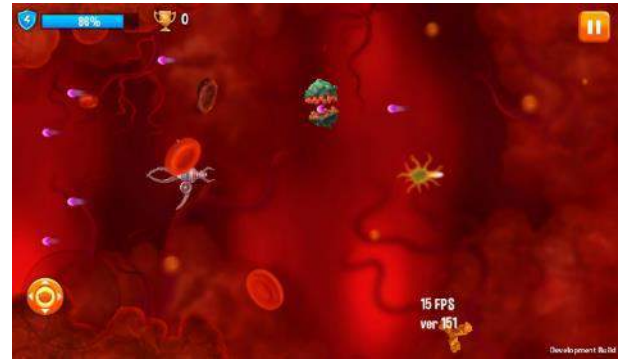
- Dpad means – control the submarine via press on screen
- Touch Direct – control the submarine via press on move finger

Result:

- controls have changed in the shooter game



Shooter Game – Touch direct



Shooter Game – manual via joystick

Action 4: change “play as”

Result: fire mode changes in the shooter game

- Automatically – submarine fires itself
- Manuel – use the shoot button in the shooter game and tap on it



Shooter Game –automatic fire mode



Shooter Game – manual fire mode

Action 5: press “change mode”

Result: Your role will be changed

Action 6: press “save”

Result: Current settings will be saved

Action 7: Logout button (top, left, corner)

Result: you will be directed to the login menu

Action 8: press the X in the right top corner

Result: you will be directed to the dashboard menu

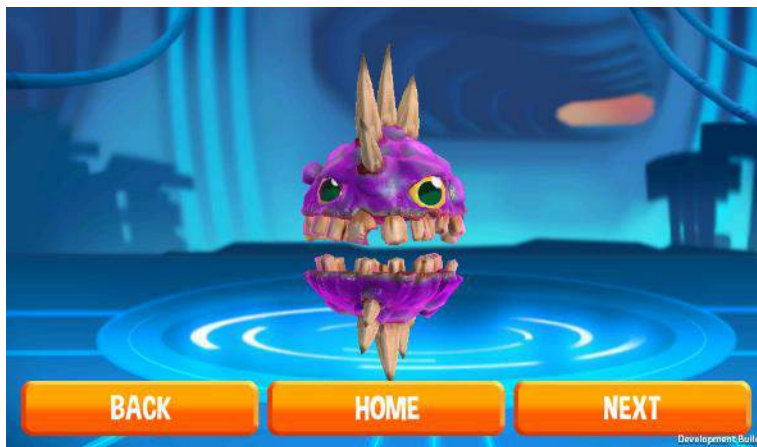
7.4.6. Main buttons – Dashboard: Catwalk Test

Description 1: Test: is catwalk is fully functional?

Description 2: The Catwalk shows every cancer cell in the game. In the later development process, there will be explanations about their function in the game and in the human body

Action 1: press the catwalk button

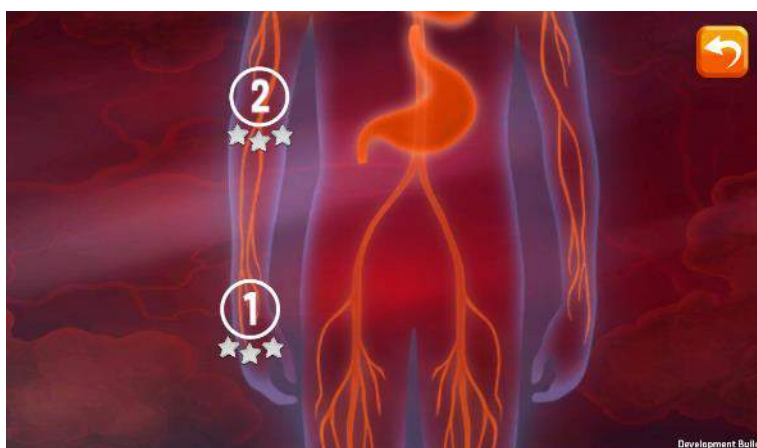
- Result:** you will be directed to the catwalk
- Action 2:** watch the cancer cell animation
- Result:** cancer cell moves around. There are no glitches, performance problems. If the animation stops, then it's a bug.
- Action 3:** press back and next button
- Result:** different cancer cell types will be shown
- Action 4:** use two fingers to zoom in and out
- Result:** Cancer Cell type gets bigger or smaller
- Action 5:** press the home button
- Result:** you will be directed to the dashboard menu



The Catwalk

7.4.7. Level Selection Map Test

- Description 1:** Test: is the level selection map fully functional?
- Description 2:** It's a picture of the human body with different levels connected to body parts, where typical cancer can exist and design sets. Available in the current Version: Level 1 and 2.



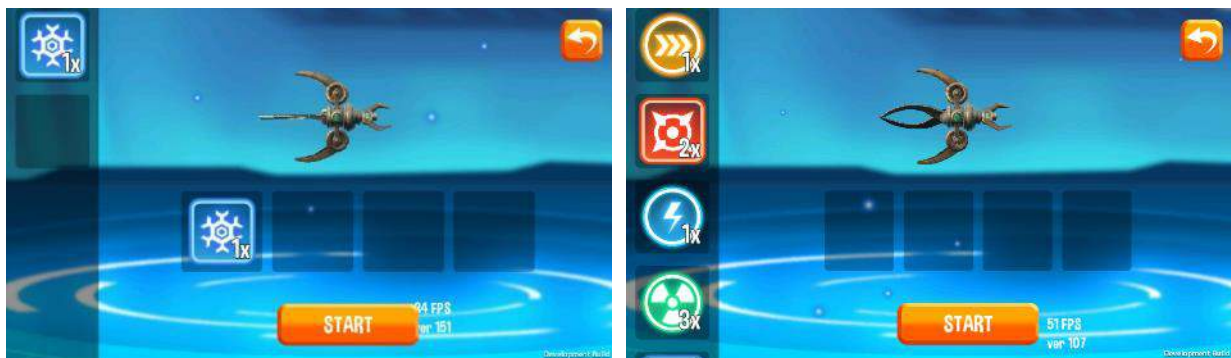
The Levelmap

- Action 1:** use one finger to navigate through the level map
- Result:** you can move over the level map layout

- Action 2:** use 2 fingers to zoom in and out
- Result:** zoom in, zoom out
- Action 3:** press the back button (right, top corner)
- Result:** you will be directed to the dashboard
- Action 4:** Select level 1
- Result:** you will be directed to the special weapon menu
- Action 5:** press the home button
- Result:** you will be directed to the dashboard menu

7.4.8. Special Weapon Menu Test

- Description 1:** Test: is the weapon menu fully functional?
- Description 2:** The player can choose different types of special weapons for the following levels.
- Notice:** In Evaluation-Version available: Cryo Gun



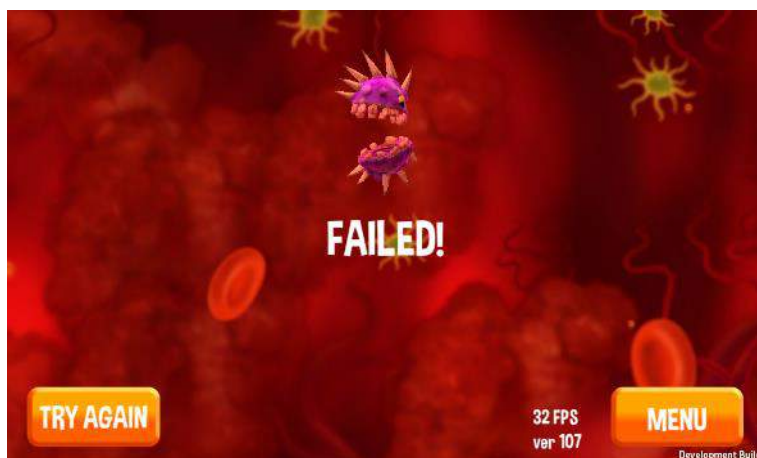
Evaluation Version – only cryo gun

final Version – all items

- Action 1:** press the play button in the dashboard
- Result:** you will be directed to the special weapon menu
- Action 2:** drag and drop a special weapon from the select menu the “select for ship menu” (bottom)
- Result:** drag and drop function
- Action 3:** drag and drop a special weapon from the “select for ship menu” (bottom) to the select menu (left)
- Result:** drag and drop function
- Action 4:** press back button (right, top corner)
- Result:** you will be directed to the levelmap
- Action 5:** press the start button
- Result:** you will be directed to the shooter game

7.4.9. Shooter Game

- Description 1:** Test: is the shooter game fully functional? Are all animations shown, that have to be shown in this stage of development?
- Preparation:** please turn speakers on. Shooter game includes sounds (early stage of development sound).
- Notice:** In Evaluations-Version: only Level 1 and 2 is available
- Action 1:** just watch the design and animations
- Result:** submarine has a flying animation, cancer cells a predefined movement pattern and shoot- /move animations, three layers (fore-, mid- and background) show different types of obstacles, props and designs.
- Action 2:** use one finger to move the submarine
- Result:** submarine moves via touch gesture
- Action 3:** shoot. Depending on your option entry, the submarine fires automatically or you have to tab the shoot button (right bottom corner)
- Result:** submarine shoots.
- Action 4:** shoot a cancer cell - kill
- Result:** cancer cell gets hit (animation) cancer cell “explodes” after some hits by the shots of the submarine (how many times you must hit a cancer cell, depends on the type of cancer cell)
- Action 5:** shoot a cancer cell – high score points
- Result:** if you shoot a cancer cell you get high score points (bar – left upper corner).
- Action 6:** get hit by a cancer cell - energy
- Result:** cancer cell shoots. If you get hit by the shot, you will lose energy (bar – left upper corner). There will be a hit by shot animation for the submarine.
- Action 7:** get hit by a cancer cell – dead
- Result:** after getting hit by the cancer cells so many times, that your energy bar goes done to zero, your submarine will explode in an animation. The lose dialog is shown.



- Action 8:** lose, choose the menu
- Result:** you will be directed to the dashboard menu

Action 9: lose, choose the try again button

Result: level starts from beginning

Action 10: reach the level end and

Result: the boss cancer cell appears



Action 11: win against the boss cancer cell



Result: after winning against the boss cancer cell the win dialog opens. You will get stars, depending on your high score points, information about levels that you have unlocked.

Action 12: win, choose the next button

Result: you will be directed to the level selection menu. Your stars for level 1 are shown.

Action 13: choose level 2

Result: do the same testing, like in level 1

Action 14: win, choose the home button

Result: you will be directed to the dashboard menu

Action 15: pick up an energy power up

Result: if a cancer cell drops an energy power up, you can collect it. After collect it, your energy bar refills

Action 16: pick up an antibody (yellow x)

Result: submarine slows down

Action 17: collide with a dead red blood cell

Result: blocks your shots and the submarine. Same result for the cancer cells. You can shoot this obstacle.

7.4.10. **Present Transfer between Supporter and Patient**

Description 1: Test if you can transfer a present as a supporter

Preparation 1: Register a Supporter and a Patient

Preparation 2: Login in with the Supporter Account

Preparation 3: Tape “Play Button” in Dashboard Menu (top-right corner).

Action 1: Information: You need to select a patient you would like to support before you play the game!

Result: Dashboard with all Patient will be opened.

Action 2: Choose a Patient

Result: Patient will be highlighted and chosen

Action 3: automatically

Result: Levelmap will be shown

Action 4: Choose a Level

Result: You will be directed to a level

Action 5: Play the Level and collect power up’s, special weapons and so on.
After you have won the level go back to the dashboard “news”

Result: In the News is an entry: “you supported Patient XY”

Action 6: Log out and Log in with your patient account

Result: After login you will be directed to the news dashboard. There you will find the entry: “you were supported Supporter XY”

Action 7: Tape on a present

Result: Overlay with present will be opened. The present will extract itself.

Notice: Booster or Cryo Gun will appear after playing level 2

Action 8: Close the present overlay

Result: You will be directed to the news dashboard. Now there is an entry that you with the received type of present

7.5. ***Installation and configuration guidelines***

When the app is downloaded from the app store, it is installed automatically. Technical configurations of the app are not included, you only can change settings within the app. See 5.4. for details.

When downloaded as an apk/ipa, the app needs to be installed manually. In the system options you have to enable installations from “unknown sources”. Once the app is installed, you can deactivate the installations from “unknown sources” again.

8. iManageMyHealth

8.1. *Intended Purpose*

iManageMyHealth is a multipurpose informative app intended for citizens and patients for general health management. It supports users i) in managing their drugs and drug intakes and provides information on drugs and their interactions, ii) in managing and understanding their paper based health documents, and iii) in recording and overviewing specific vital signs and laboratory parameters.

Further to this, for demonstration purposes, the app also provides specific management services in the context of cancer, as for the monitoring of pain and fatigue that include recommendations and linkage to public information resources.

The app as a whole is not intended for diagnosis and therapy but just as a supportive electronic solution for information provision and management of medical information.

8.2. *Provided functionality*

8.2.1. **Drug management**

The following functionality is offered under Drug Management:

- Compose your medication plan: add a new drug to the medication plan. When the drug name is entered, an autocomplete feature proposes drugs with the help of external drug information services or an integrated list of drugs. The user is encouraged to take pictures of the drug package and the drug itself. Furthermore, she/he can specify drug intake times. The system reminds her/him with a message to take a drug. She/he can also add a comment to the drug. Information on a new drug in the medication plan is also sent to iPHR.
- The new drug is checked for interactions with other drugs in the medication plan if the drugs are known to the external drug information service. Warning symbols are shown in the medication plan and drug-drug-interaction information is presented when the user touches the warning symbol.
- Delete a drug from the medication plan: all information related to the drug is deleted in the system. Patient is asked about the effectiveness of this drug. Information is also sent to iPHR.
- More information about a drug is shown by forwarding the user to a web page containing structured information about the drug in the language configured on the app. For German users she/he is forwarded to <http://www.beipackzettel.de/>, for Italian users to www.torinomedica.it for and for all other languages in the smartphone - to <https://www.drugs.com/>.

This component leverages two external internet based drug database services and an integrated list of Italian drugs to select drugs from a list, to present information on drugs and to check drug-drug interactions. Patients are warned for drug-drug-interactions by leveraging two external internet based drug repository services, the Canadian DrugBank of OMx Personal Health Analytics⁴ Inc. and the German SCHOLZ Datenbank of ePrax AG⁵. The Canadian DrugBank does not only contain approved drugs for the US and Canadian market but also includes drugs registered at EMA for the European market. However, the brand names of the different drugs are those of the UK market. The German Scholz Datenbank lists those drugs that are approved in

⁴ <https://omx.io/>

⁵ <http://scholz-datenbank.de/>

Germany. In consequence, the app iManageMyHealth uses the online services of the Scholz Datenbank for German users. For Italian users we included in our app the official list of drugs published by the Italian Medicines Agency under <http://www.aifa.gov.it/content/dati-sulle-liste-dei-farmaci-open-data>. For smartphones with other language preferences, the online services of the Canadian DrugBank are used.

It shall be noted that the Italian drug data that are used by iManageMyHealth does not include structured information to check interactions. In consequence, this feature is disabled in the Italian version of the app. Further to this, the external drug databases have disclaimers and restrictions concerning the use of their database due to the fact, that the drug information might be wrong, obsolete or incomplete. Similar restrictions in use and disclaimers are presented to the user in our app. Patients must be made aware that drug information might be wrong, obsolete or incomplete and that they shall always discuss eventual interactions of their drugs with their treating physician.

The implemented use cases and requirements are listed in deliverable D5.3.

8.2.2. Managing health documents

The app offers the user to scan paper based health documents and to store them in the Android device and to annotate them. Scanning is done with the camera and a document is assembled as a set of images. The user can go through a document page by page and can zoom in and out in a page. She/he can mark a section with a keyword of interest and search with this marked text in Wikipedia or in the Personal Health Information Recommender. Results are presented to the user.

This functionality is derived from scenario SC1 in deliverable D2.2 and use case UC.PHR.1 of D2.3.

8.2.3. Vital sign monitoring

A patient can perform measurements using a specific set of 3 devices for the measurement of his weight, his blood pressure and body temperature and record the measurement results. The patient can perform measurements with these compatible devices on demand or following corresponding notifications received from the Care Flow Engine from executed Care Flows for such patient.

There are two possibilities for entering measurement values in the iManageMyHealth app:

- Entering measurement values automatically: this option is possible if one of the above-described medical devices is connected to the smartphone via Bluetooth Low Energy (LE) protocol.
- Entering measurement values manually: this is an alternative possibility if the patient uses other medical devices than the above mentioned devices.

Measurement results are presented in charts.

The following devices are supported by the app iManageMyHealth under Android 5 devices only:

- Scale A&D Medical UC-352BLE⁶
- Blood pressure device A&D Medical UA-651BLE⁷

The following device is supported by the app iManageMyHealth under Android 5 and 6 devices only:

- Ear thermometer FORA IR21b⁸

⁶ https://www.andonline.com/medical/products/details.php?catname=Scales&product_num=UC-352BLE

⁷ https://www.andonline.com/medical/products/details.php?catname=Blood_Pressure&product_num=UA-651BLE

⁸ <http://www.foracare.ch/Meter-IR21.html>

These devices follow the Bluetooth LE communication protocol. A mobile device on which the iManageMyHealth app is installed, has to be configured respectively to use one of these devices.

This functionality is derived from scenario SC11 in deliverable D2.2 and use case UC.LSM.1 of D2.3.

This app offers management services provided by the so-called Care Flow Engine, which is the central component of the decision support system of the iManageCancer platform. This version of the iManageCancer platform provides provisional services for the self-management of pain and fatigue.

8.3. Technical implementation

The internal structure of the iManageMyHealth app shown on the picture below contains components for the building the user interface, for the management of the app modules (e.g. medication plan, health documents, patient's profile), REST services for communication with external services, for management of the patient's data in the local database and for notification functionalities for drug intakes as well as to execute the care flow tasks.

Figure 220 : Internal architecture of app iManageMyHealth.

The medication plan management module of the app allows the patients to define intake times for drugs in the medication plan. According to this definition, the patient will receive scheduled notifications about the drug intakes. This functionality can be activated (default setting) and deactivated on the GUI for app settings.

The notifications functionality essentially uses the Alarm system service provided by Android to set alarms and notification system service for notifications. In order to avoid affecting user interface responsiveness, we have implemented this functionality as a background service by using the *AlarmManager* class to perform time-based operations outside the app, e.g. scheduling to run any function in the future like dynamically creating repeated notification alarms for the drugs in the medication plan. The service uses the *IntentService* class, which provides straightforward structure for running an operation on a single background thread. This approach minimizes the app's resource requirements because we can schedule drug intake notifications without relying on timers or continuously running background services. Registered alarm also works if device is asleep but it does not work if the mobile device is turned off. *AlarmManager* fires these events at set times and/or intervals. We use this class in conjunction with broadcast receiver to start services and perform showing notifications for drug intakes to the patients. The alarms operate outside of the app, so they trigger notification events even when the app is not running, and even if the device itself is asleep.

The following classes have been implemented:

1. *Activity* class, which uses the *AlarmManager* to set the alarm according to the drug intake times in the medication plan and send notification on alarm trigger. *AlarmManager* holds a wait lock on CPU as long as receiver's *onReceive()* method is executing. This feature is because *AlarmManager* has to ensure that phone will not sleep till it is broadcasting.
2. *AlarmReceiver*, which is a *WakefulBroadcastReceiver* that receives the alarm trigger on set time. From here, we create notifications when the alarm is triggered. We initiate three type of notifications:
 - a. show a message to user
 - b. play the alarm ringtone
 - c. vibrate the mobile device

The receiver will start the following *IntentService* to send a standard notification to the user.

3. Class for restarting the notification functionality: by default, all alarms are cancelled when a device shuts down. To prevent this from happening, we have designed the app to automatically restart a repeating alarm if the patient reboots the device. This ensures that the *AlarmManager* will continue doing its task without the patient needing to manually restart the alarm.

For the client functionality to use the services provided by the Care Flow Engine the app can:

- Get all available Care Flows / management services for patients, subscribe to them or unsubscribe from a Care Flow.
- Get all executed Care Flows for a specific patient and the pending tasks.
- Execute tasks: the tasks received from the Care Flow Engine and displayed to the user should be executed and the result values are sent back to the Care Flow Engine in order to complete the task and allow the Care Flow to proceed to the next task
 - for health enquiries: the questions should be answered by the user
 - for measurement tasks: the measurements should be performed
 - for information tasks: reading of any text message should be confirmed.

The app also responds to messages about open tasks received by Google Cloud Messaging Service. The details are presented in deliverable D5.3.

8.3.2. Interfaces

There are several implemented REST services for the communication with different external services:

- For communication with the iManageCancer platform, e.g. for synchronization of medication plan, measurement values and health documents as well as retrieving explanation from the Personal Health Information Recommender for selected text in the health documents, sending questionnaires about usability of the app modules and audit data about user interactions to the iManageCancer platform.
- For communication with the external drug databases for retrieving information (incl. drug-drug interactions) about prescribed drugs.
- For communication with the Care Flow Engine for management and execution of the care flow tasks.

For communication with the Google Cloud Messaging services for receiving notifications about care flow tasks. Rest services for communication with the external drug databases

1. German SCHOLZ Datenbank

This is confidential information of the company eprax GmbH.

2. Canadian DrugBank of OMx Personal Health Analytics

This is confidential information of the company OMx GmbH. The services are described in the internal deliverable D3.2.

The REST services to communicate with iPHR, Careflow Engine and Google Cloud Messaging services are described in detail in deliverable D5.3.

Furthermore, the app communicates with the Google Firebase platform for sending crash reports.

Details on the interface to 3 specific medical devices are presented in deliverable D6.3.

8.3.3. Database

SQLite is used as database system.

The following database tables are created to store the data retrieved from the drug databases.

i. Table ‘medicationplan’ for storing the patient’s medication plan:

Spalten (17)			
Column ID	Name	Type	Not Null
0	id	INTEGER	0
1	drug_id	INTEGER	0
2	start	INTEGER	0
3	end	INTEGER	0
4	dosagemax	NUMERIC	0
5	reason	VARCHAR	0
6	physician	VARCHAR	0
7	prescriptiondate	INTEGER	0
8	addingdate	INTEGER	0
9	lasteditdate	INTEGER	0
10	deletingdate	INTEGER	0
11	sendstatus	VARCHAR	0
12	senddate	INTEGER	0
13	imcdocumentid	VARCHAR	0
14	checkinteraction	VARCHAR	0
15	checkinteractiondate	INTEGER	0
16	comment	VARCHAR	0

This table contains a relation to the table ‘drug’ with general information about the drug and the identification id of the drug in the medication plan ‘imcdocumentid’ on the iManageCancer platform.

ii. Table ‘drug’

Spalten (9)			
Column ID	Name	Type	Not Null
0	id	INTEGER	0
1	name	VARCHAR	0
2	pharmform	VARCHAR	0
3	takingkind	VARCHAR	0
4	comment	VARCHAR	0
5	identcode	VARCHAR	0
6	idenkind	VARCHAR	0
7	drugpicture	VARCHAR	0
8	packagepicture	VARCHAR	0

iii. Table ‘dosage’

Spalten (12)			
Column ID	Name	Type	Not Null
0	id	INTEGER	0
1	drug_id	INTEGER	0
2	min	NUMERIC	0
3	max	NUMERIC	0
4	unit	VARCHAR	0
5	minage	INTEGER	0
6	maxage	INTEGER	0
7	ageunit	VARCHAR	0
8	minweight	NUMERIC	0
9	maxweight	NUMERIC	0
10	weightunit	VARCHAR	0
11	gender	VARCHAR	0

This table for storing details about the drug dosage will be used in the version 2 of the app.

iv. Table ‘ingredients’

Spalten (6)			
Column ID	Name	Type	Not N
0	id	INTEGER	0
1	drug_id	INTEGER	0
2	name	VARCHAR	0
3	strength_number	VARCHAR	0
4	strength_unit	VARCHAR	0
5	code	VARCHAR	0

This table stores information about drug ingredients.

v. Table ‘intaketime’

Spalten (8)			
Column ID	Name	Type	Not Null
0	id	INTEGER	0
1	medicationplan_id	INTEGER	0
2	daydefinition	VARCHAR	0
3	timedefinition	VARCHAR	0
4	tonotify	INTEGER	0
5	dosage	NUMERIC	0
6	notificationtime	VARCHAR	0
7	comment	VARCHAR	0

This table stores information about defined drug intake times, which are used for generating notifications about drug intakes for the patient.

vi. Table ‘interaction’

Column ID	Name	Type	Not Null
0	id	INTEGER	0
1	drug1_id	INTEGER	0
2	drug2_id	INTEGER	0
3	interactiontext	VARCHAR	0
4	comment	VARCHAR	0

This table contains information about interaction warnings for drug pairs.

vii. Table ‘ndccode’

Spalten (3)		
Column ID	Name	Type
0	id	INTEGER
1	drug_id	INTEGER
2	code	VARCHAR

This table contains drug codes used in the external drug databases.

viii. Table ‘outcome’

Spalten (5)			
Column ID	Name	Type	Not Null
0	id	INTEGER	0
1	medicationplan_id	INTEGER	0
2	score	NUMERIC	0
3	comment	VARCHAR	0
4	sideeffectssuffer	INTEGER	0

This table contains a small survey information about patient’s experience with the drug, which is sent to the iManageCancer platform when a drug is deleted in the medication plan.

The following database tables are created for storing the measurement data:

ix. Table “measurement”

This table is for storing general information about a measurement. The field ‘*task_id*’ is filled only if the measurement has been performed on request of the Care Flow Engine (CFE)

measurement	
id	INTEGER
timepoint	INTEGER
sendstatus	VARCHAR
senddate	INTEGER
taskid	VARCHAR
name	VARCHAR

The possible content is shown in the example below

Tabelle: measurement						
	id	timepoint	sendstatus	senddate	taskid	name
	Filter	Filter	Filter	Filter	Filter	Filter
1	1	1456828250116	0	0		blood pressure
2	2	1456828250216	0	0		temperature
3	3	1456828250316	0	0		weight
4	4	1456839050117	0	0		blood pressure
5	5	1456839050217	0	0		temperature

x. Table “measurementkind”

This table is for storing information about a measurement component and their measurement units (e.g. storing of systole, diastole and pulse with units for the blood pressure measurement). The field ‘instructionFlag’ stores information if the patient does not want to view instructions for performing measurements with medical devices supporting Bluetooth LE protocol for automatically entering measurement values in the app.

measurementkind	
id	INTEGER
kind	VARCHAR
unit	VARCHAR
instructionFlag	INTEGER

The possible content is shown in the example below.

Tabelle: measurementkind				
	id	kind	unit	instructionFlag
	Filter	Filter	Filter	Filter
1	1	pulse	bpm	1
2	2	systolic	mmHg	1
3	3	diastolic	mmHg	1
4	4	temperature	°C	1
5	5	weight	kg	1

xi. Table “measurementvalue”

This table is for storing measurement values.

measurementvalue	
id	INTEGER
value	NUMERIC
measurementnumber_id	INTEGER
measurementkind_id	INTEGER

xii. Table “measurementitem”

This table is for storing different components of a measurement and their values which should be sent to the CFE. Important is the field “itemid” that is equal to the question id used on the CFE.

measurementitem	
id	INTEGER
task_id	INTEGER
itemid	VARCHAR
type	VARCHAR
datatype	VARCHAR
value	VARCHAR
unit	VARCHAR

xiii. Storing care flows

For each patient a template (one row in the table) for every available care flow is created. If a care flow is subscribed for the patient, the columns ‘*patientprocessid*’ and ‘*created*’ will be filled in. If the care flow is unsubscribed, the column ‘*deleted*’ will be filled in and a new template for the patient will be created to allow starting the care flow again in future.

cfeworkflow	
id	INTEGER
workflowkey	VARCHAR
workflowname	VARCHAR
description	VARCHAR
patprocessid	VARCHAR
patient_id	INTEGER
subscribed	VARCHAR
created	LONG
deleted	LONG
physicianstarter	VARCHAR

xiv. Storing care flow tasks

In this table, a general information about all possible tasks (health enquiries, measurement tasks and information tasks) is stored. If a task is executed, the columns ‘*completed*’ and ‘*completingdate*’ are filled in and the task is no longer shown to the user. It is possible for the next versions to implement displaying already executed tasks to the user.

task	
id	INTEGER
taskid	VARCHAR
creatingdate	LONG
category	VARCHAR
processid	VARCHAR
name	VARCHAR
description	VARCHAR
labels	VARCHAR
completed	VARCHAR
completingdate	LONG

In the following tables ‘question’, ‘answer’ and ‘info’ additional specific data for health enquiries and information tasks are stored. The measurement tasks for patients are not stored in the specific tables: all measurements can be always viewed in the ‘Measurement’ module of the iManageMyHealth app, which is achievable from the main menu of the app.

xv. Storing health enquiry questions

A value of the column ‘questionid’ is important for sending results back to the Care Flow Engine.

question	
id	INTEGER
task_id	INTEGER
questionid	VARCHAR
name	VARCHAR
labels	VARCHAR
datatype	VARCHAR
answer	VARCHAR

xvi. Storing answers in the selectable answer possibilities

If a question on the enquiry task has a data type ‘enum’ for multiple values shown for user selection, the possible answers will be stored in the ‘answer’ table.

answer	
id	INTEGER
valueid	VARCHAR
valuenam	VARCHAR
labels	VARCHAR
question_id	INTEGER

xvii. Storing information messages

In this table are stored specific data for an information task like an identifier for a “yes” answer, information text and helpful web links.

info	
id	INTEGER
task_id	INTEGER
confirmquestionid	VARCHAR
infotext	VARCHAR
yesanswerid	VARCHAR

xviii. Storing health documents

In the 'file' table are stored locations of the pictures taken from paper based health documents, links to the file id on the iPHR retrieved after synchronisation as well as status and time point of sending the files to the iPHR.

Spalten (8)				
Column ID	Name	Type	Not Null	Default Value
0	id	INTEGER	0	null
1	filemetadata_id	INTEGER	0	null
2	picturepath	VARCHAR	0	null
3	thumbnailpath	VARCHAR	0	null
4	parentid	INTEGER	0	null
5	sendstatus	VARCHAR	0	null
6	senddate	INTEGER	0	null
7	imcdocumentid	VARCHAR	0	null

In the 'file_metadata' are stored details about the health document.

Spalten (7)				
Column ID	Name	Type	Not Null	Default Value
0	id	INTEGER	0	null
1	title	VARCHAR	0	null
2	type	VARCHAR	0	null
3	comment	VARCHAR	0	null
4	date	INTEGER	0	null
5	sendstatus	VARCHAR	0	null
6	senddate	INTEGER	0	null

8.3.4. Data protection

With the help of the app iManageMyHealth, the patient collects and manages sensible health data on his smartphone. This data are stored unencrypted on the device in a local database but are also sent to iPHR for storing them in his personal health record. The app itself is not protected by a password. The user is advised to make precautions in the settings of his Android device that it is protected by a PIN code, screen lock is activated and optionally that his phone is encrypted.

In order to get the full benefit of the iManageCancer online health services and to ensure that the data is also stored on the iManageCancer server, the patient has to create an account in the iManageCancer portal before she/he can enter the generated and approved credentials into the iManageMyHealth app. These account credentials (user name, password and e-mail) are further used for authentication to synchronize the data collected on the mobile device with his personal

health record on iPHR and to exchange data with the Care Flow Engine following the OAuth 2.0 protocol for secure authorization. The https protocol is used for transport layer and guarantees secure communication of the app with the iManageCancer portal and with the Care Flow Engine.

8.4. User manual

This manual is for testers of the app.

The following functionality is included to the **iManageMyHealth** app:

- Accessing the main menu page from the start page.
- Accessing modules in the main menu
 1. My Drugs – a module for the management of prescribed drugs
 2. Measurements – a module for the management of measurements
 3. Health docs – a module for the management of medical documents
 4. Health Manager – a module for the management of Care Flows (communication with the Care Flow Engine) – **needs internet connection!**
 5. Diagnoses – a module for patient's diagnoses. It also includes additional data like smoking and pregnancy. Functionality of this module is not yet fully implemented.
 6. Settings – a module for managing the patient's profile on the iManageCancer platform, drug intake notifications, and for the reporting of problems.

There is a possibility to rate the first four modules of the app and to answer some questions as well as to make suggestions about the functionality of each module. the functionality of the app is described in the following sections.

8.4.1. Starting the iManageMyHealth app

Action: Find the launcher logo for the **iManageMyHealth** app in the **Apps** of your mobile device and press on it.

Result: the start page of the app is displayed

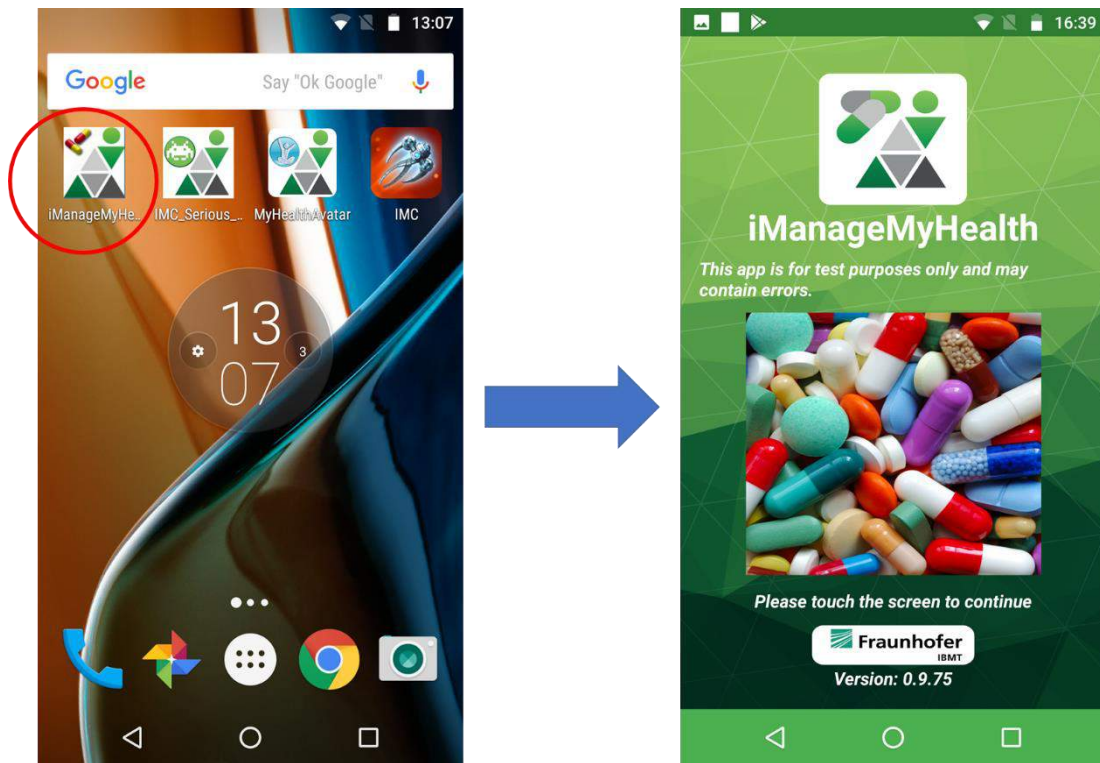


Figure 221: Starting the iManageMyHealth app.

Action: Press on the start page of the app for forwarding to the next page, which is the main menu page.

Result: when starting the app first time, the patient will be forwarded to a page where she/he will have the following three possibilities:

1. Entering user credentials an existing iManageCancer account (default setting)
2. Creating a new account on the iManageCancer platform – she/he will be forwarded to the iManageCancer portal page
3. Using the app without the iManageCancer platform

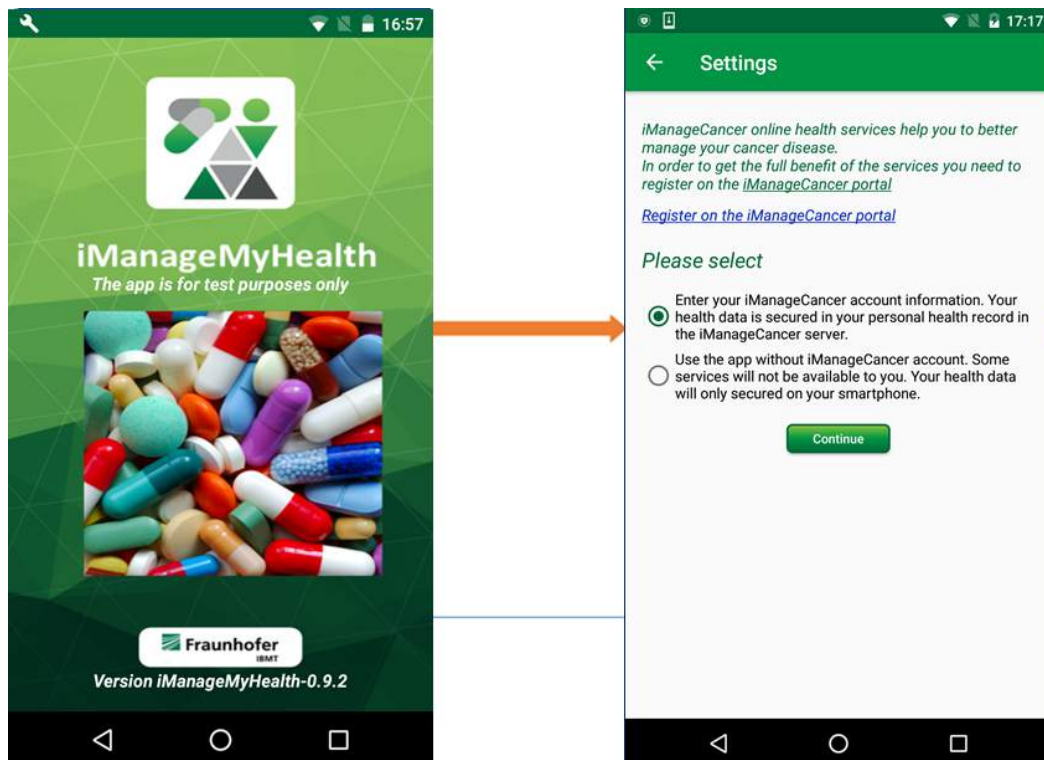


Figure 222: First start of the app.

8.4.2. Using the app without iManageCancer iPHR

If the patient wants to use the app without functionality provided by the iManageCancer platform, she/he has to select this option and click on the *Continue* button. As a result, she/he will be forwarded to the page with the main menu:

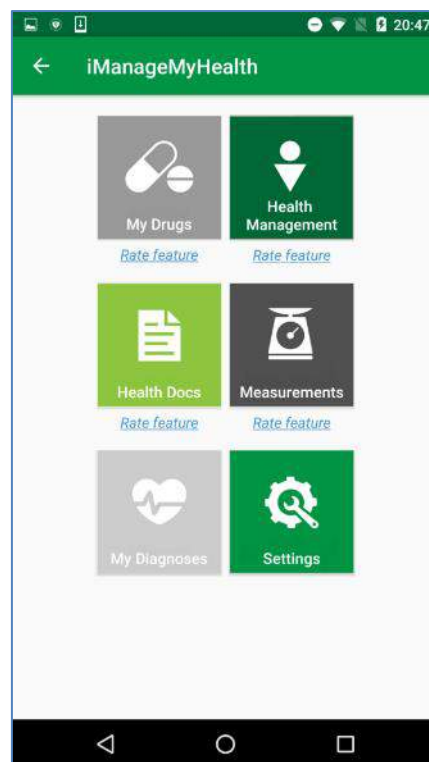


Figure 223: Main Menu page.

The patient can select different modules on this page, e.g. *Measurements*, *Health Docs*, *Health Management* or *My Drugs*. In addition, she/he can rate each of these four modules.

8.4.3. Entering an existing iManageCancer account information

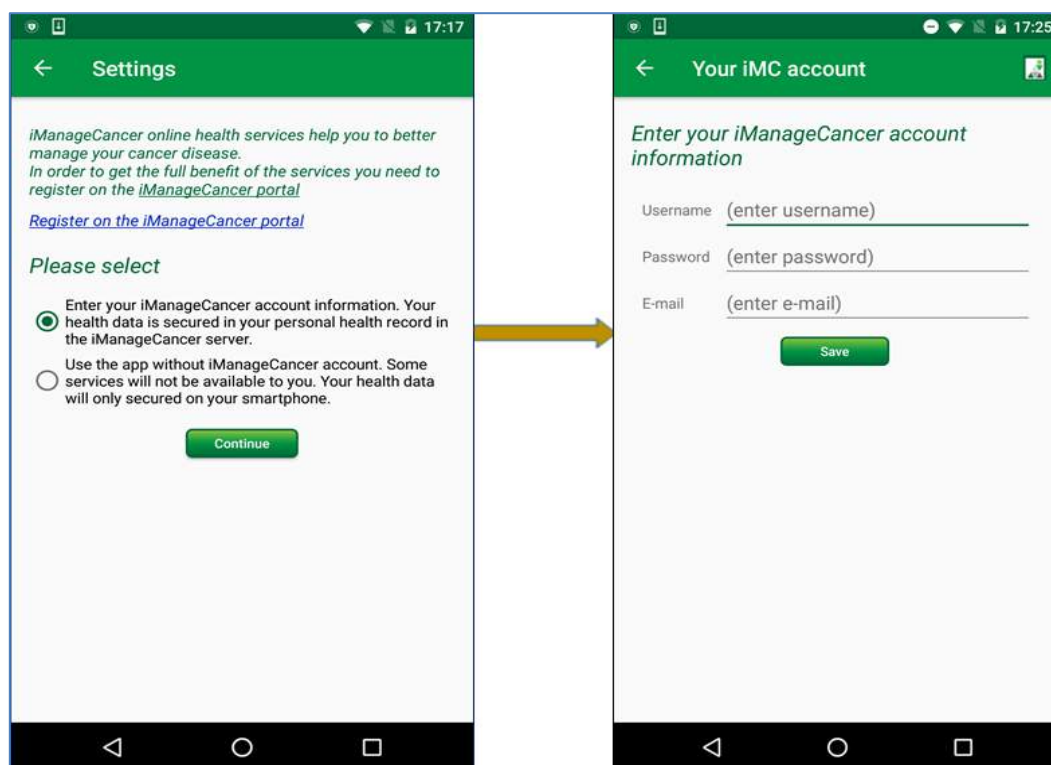


Figure 224: Entering existing iManageCancer account information.

This functionality requires an internet connection.

After entering the iManageCancer account information and clicking on the *Save* button, the account information is verified on the iManageCancer platform and some administrative user data is received and shown to the patient:

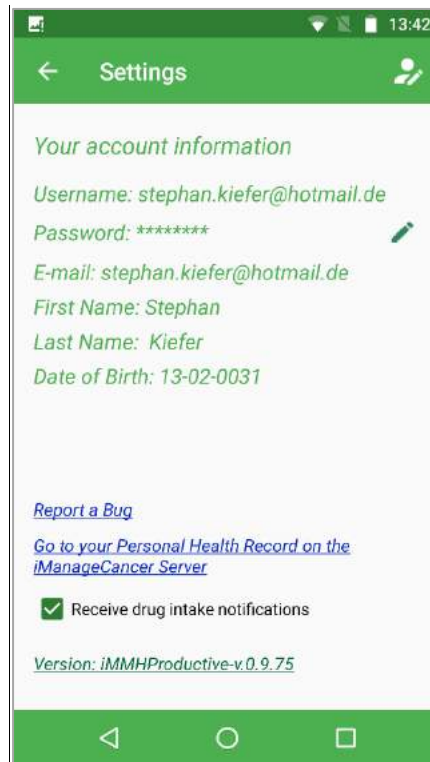


Figure 225: Retrieving the iManageCancer account information.

The patient is forwarded to the main menu page shown in Figure 223 when using the back button on the bottom of the mobile device or the arrow sign.

8.4.4. Creating a new iManageCancer account

For creating a new iManageCancer account, the patient has to click on the web link [Register on the iManageCancer portal](#) shown on the Figure 222. The patient will be forwarded to the login page of the iManageCancer portal. The patient has to enter a valid e-mail address and define her/his username and password. For using the created credentials (username and password) in the iManageMyHealth app, the patient has to confirm an activation e-mail. After successfully activating the account, the patient can enter his credentials in the app (see the Figure 224).

8.4.5. Module *My Drugs*

Action 1: Select 'My Drugs' option in the main menu

Result: A list of drugs entered in the patient's medication plan is opened.

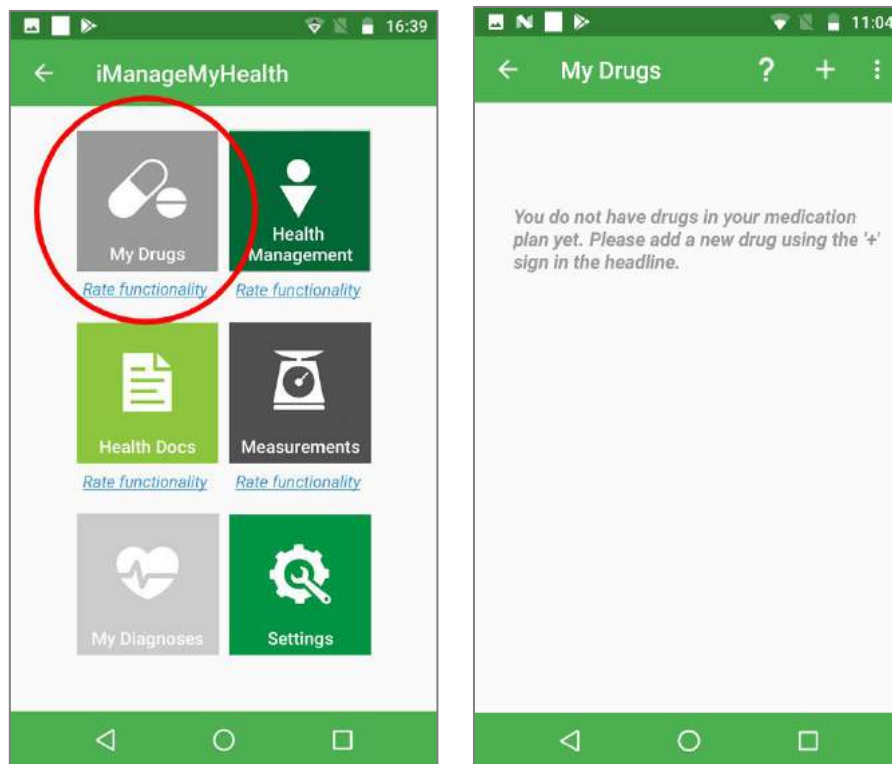


Figure 226: Empty medication plan after the first start of the app.

Adding a new drug

The patient can add a new drug to the medication plan using the '+' sign on the title row of the app. The page for entering drug information will be shown.



Figure 227: Entering a new drug data.

The patient can take pictures of his drug package and for the drug itself when clicking on the icons in the grey rectangles. **Attention:** for using this functionality, the patient has to set permissions for using the camera and device memory by the app. During the installation, the app asks for these permissions.

After entering four characters into the field for a drug name, an autocompleting text with a list of drugs starting with the entered characters is shown. This list is received from an external internet based drug data repository or in case of Italian language from integrated drug information. The patient can scroll through the list and find his drug in the list. The smartphone must be connected to the internet to use services from the external drug data repositories.

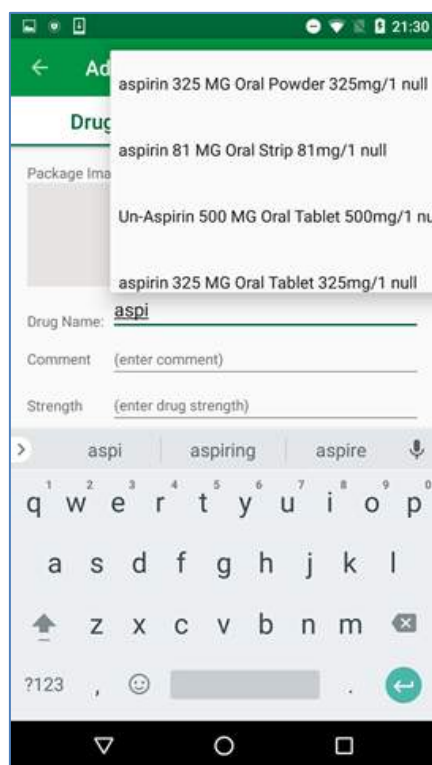


Figure 228: Autocompleting list for selecting a drug name.

The drug data is automatically entered into the corresponding fields. The patient can add some comments to the drug, e.g. about intake conditions.

When selecting the tab *Drug Intake Times*, the patient can use the '+' sign on the title bar of the app to create drug intake times for receiving reminder notifications.

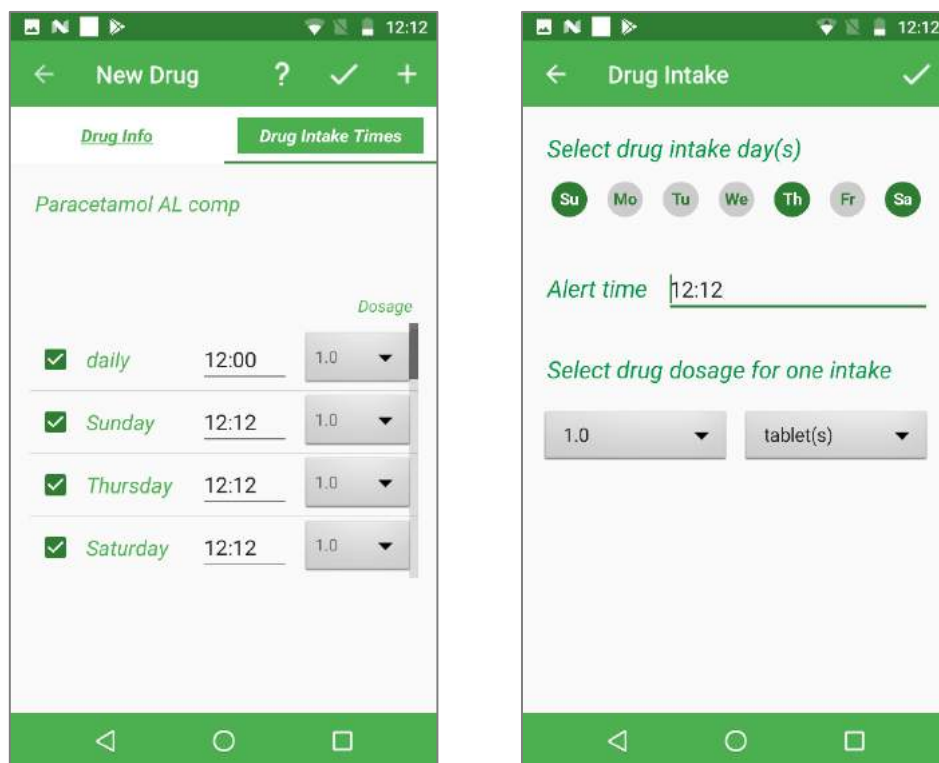


Figure 229: Defining drug intake times.

The new intake times will be added to the list of the drug intake times where the intake times can be further edited

The patient can complete adding of the new drug by using the ‘save’ – sign on the title bar of the app or he can leave the page using the back buttons. The drug is then shown in the medication plan.

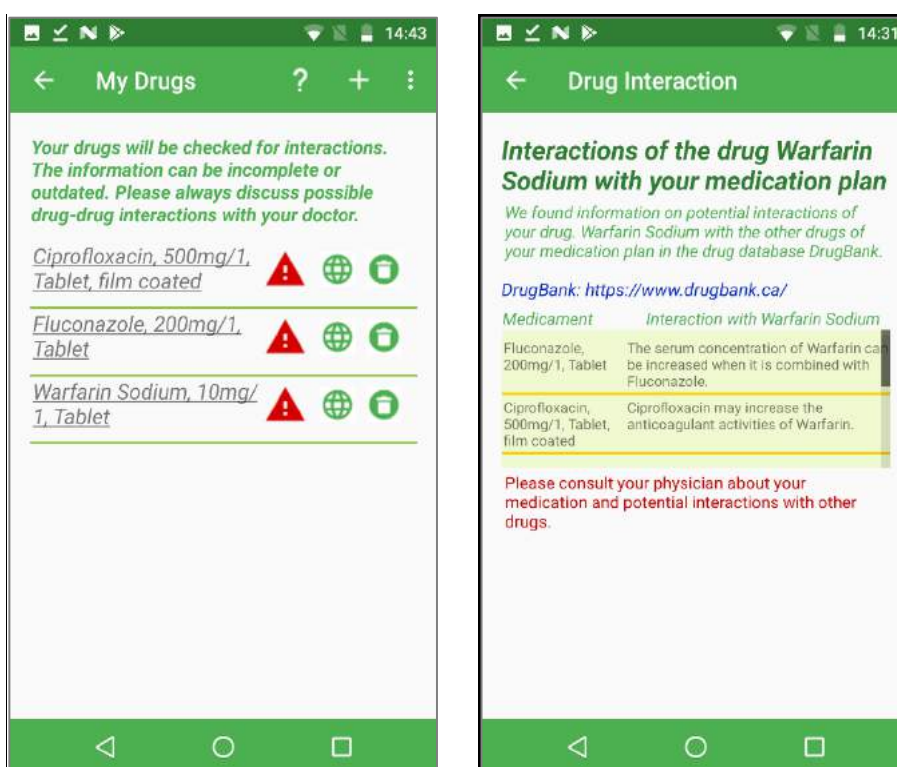


Figure 230: Drugs in the medication plan that can interact with each other. A warning symbol is presented.

The drugs are checked for interactions with each other. The following signs are used for showing the result of the drug-drug interaction checking:



- The system could not find interactions of this drug with other drugs of the medication plan.



- The drug could not be checked for interactions, e.g. because of the missing internet connection or accessing the external drug database was not possible.



- Drug not found in the drug database. In consequence, the system cannot search for interactions with the other drugs in the medication plan.



- A warning sign: The system found interactions of this drug with other drugs of the medication plan.

The details about the interactions can be viewed by clicking on the corresponding interaction sign.

If the patient has an iManageCancer account, information about the new added drug will be sent to the patient's personal health record on the iManageCancer platform (iPHR).

Editing drug

By touching the drug name in the medication plan, the patient can edit the drug information. Currently, one can edit the comment field, drug pictures and drug intake times.

If the patient has an iManageCancer account, updated information about the drug will also be sent to the patient's personal health record on the iManageCancer platform (iPHR).

Deleting drug

The patient can delete a drug from the medication plan with the waste bin sign. The drug is not be really deleted in the system's database, but it is labelled as deleted.

If the patient has an iManageCancer account, information about the deleted drug will be sent to the patient's personal health record on the iManageCancer platform (iPHR). Additionally, a small questionnaire (survey) about the effectiveness of the drug is presented. The patient is prompted to rate the drug and add a comment, if wished. Finally, the patient must confirm deleting the drug.

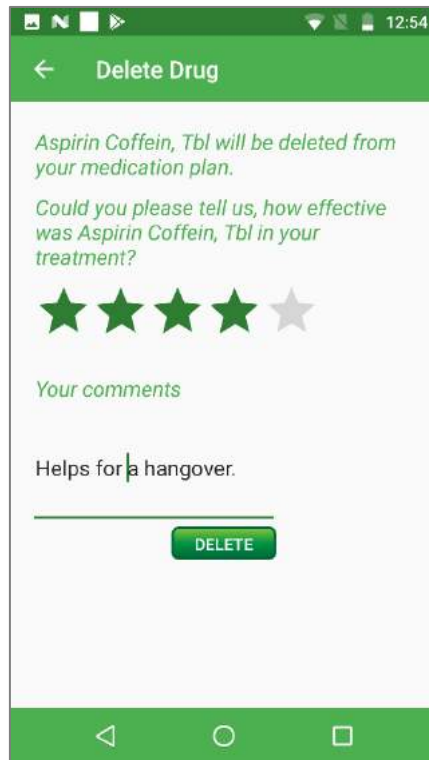


Figure 231: A survey about drug effectiveness when deleting a drug.

Viewing information about a drug

After selecting a web sign next to the drug name in the list, the patient is forwarded to a web site containing structured information about the drug in the language configured on the app. Such information is provided by www.beipackzettel.de for German users, www.torrinomedica.it for Italian users and www.drugs.com for all other users.

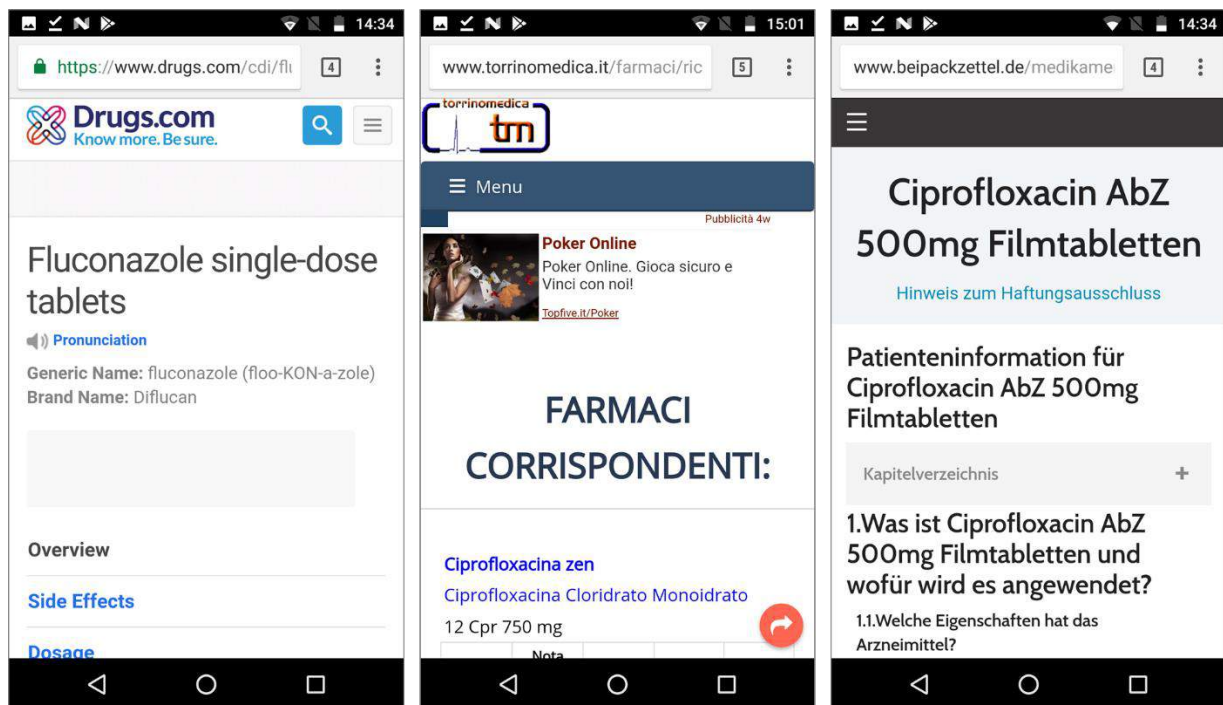



Figure 232: English, Italian and German website that is called to provide further drug information.

Drug intake notifications

The app sends reminders to the patient according to the defined drug intake times. They are shown on the top of the device screen with the icon . The full notification text is shown when swiping the icon down. When clicking on the notification, the patient will be forwarded to the page with the drug intakes defined for the current drug.

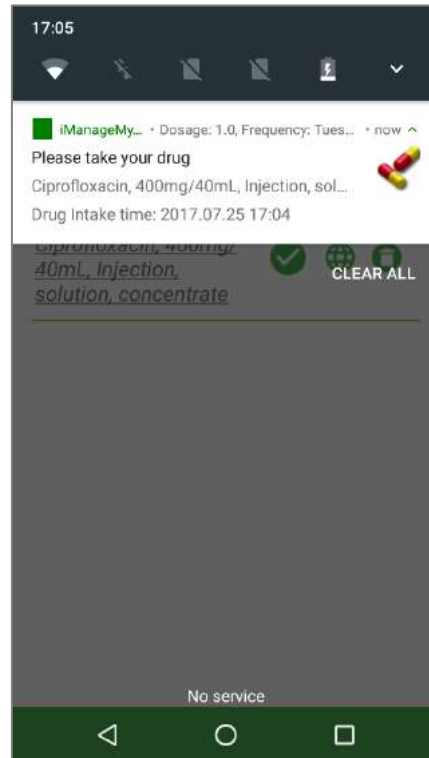


Figure 233: Drug intake reminder.

Setting permissions for the app

For setting permissions for using the camera and device memory, the patient can allow it during the first start of the app or he has to allow shown permissions when selecting *Settings* → *Apps* → *iManageMyHealth* app → *Permissions*

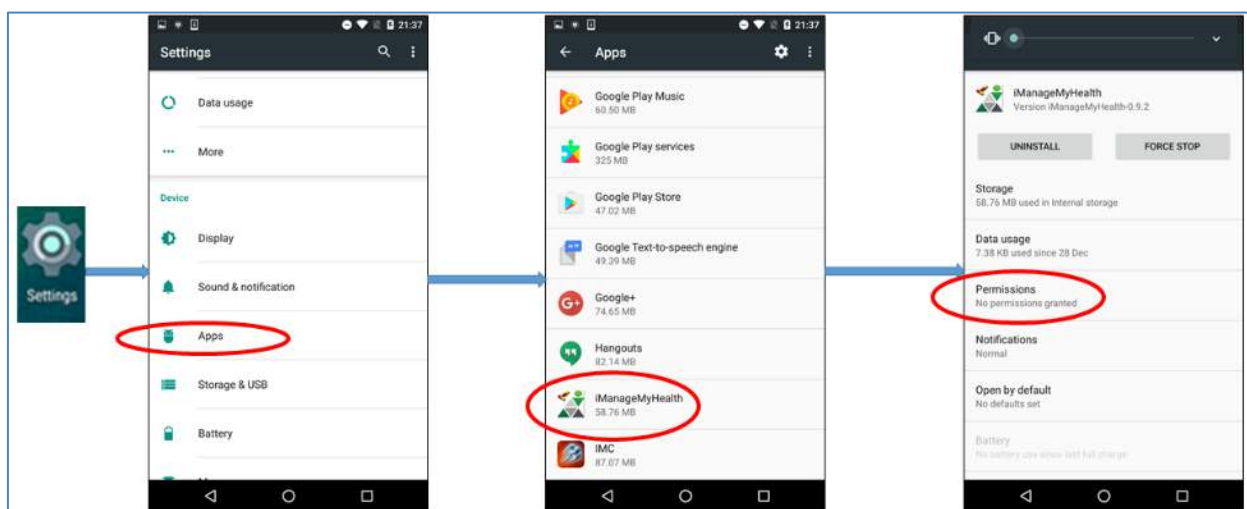


Figure 234: Finding permissions for the app.

The patient has to switch the shown permissions to 'on':

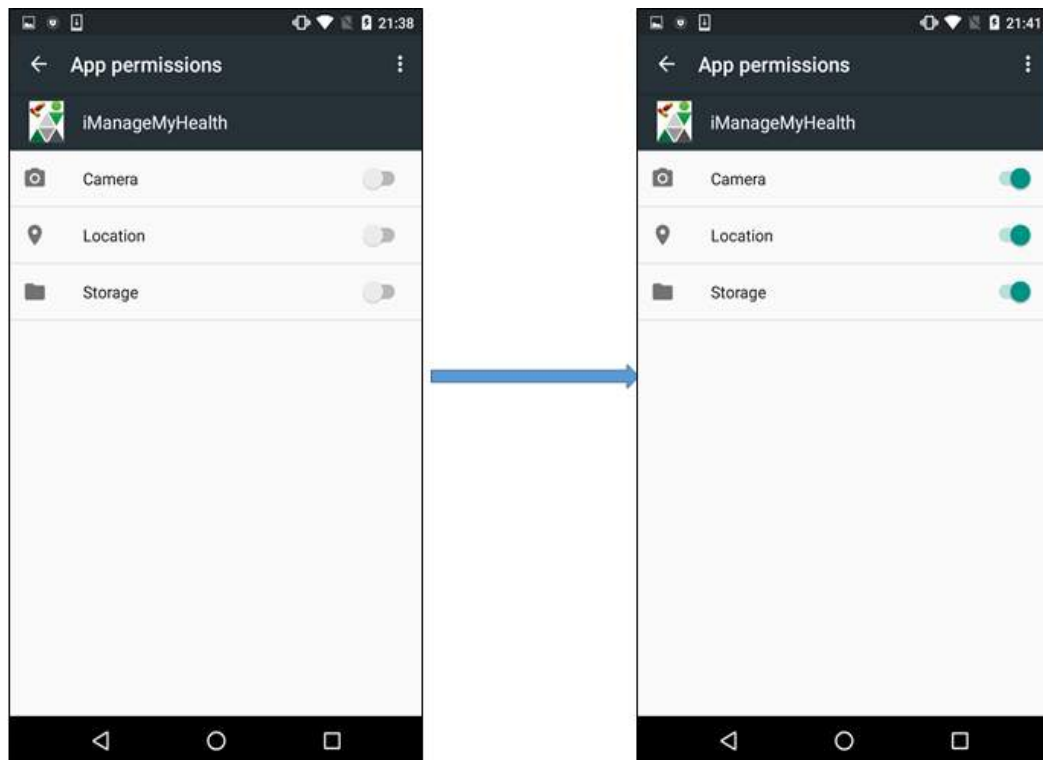


Figure 235: Setting permissions for the app.

Rating of the app modules

The patient can send reports about his user experience with the four different modules Measurements, Health Docs, My Drugs and Health Management. To do so, he has to select the *Rate Feature* link under the corresponding module on the main menu. A screen to rate the feature will open. The patient can rate the selected module and answer some questions as well as make helpful suggestions for future versions of the module. The results are stored locally and are also sent to the iManageCancer platform. When the patient wants to rate a specific module again, his last rating is shown and he can modify it.

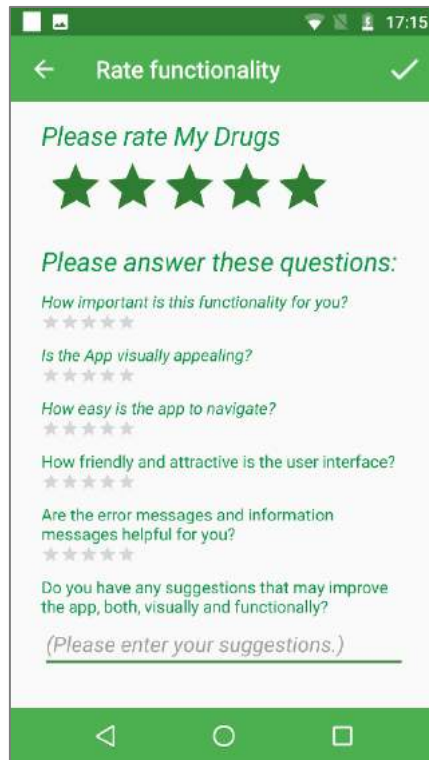


Figure 236: Rating of the module My Drugs.

Getting help

For many screens a help feature is available. By touching the question mark in the menu bar, a screen pops up with information how to use the corresponding screen

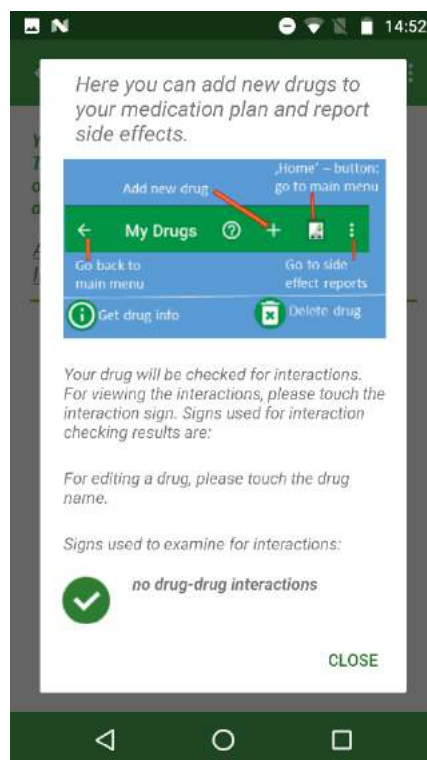


Figure 237: Help information in the screen with the medication plan.

8.4.6. Module Measurements

Action 1: select 'Measurements' module in the main menu page (see Figure 223).

Result: A page opens for accessing different measurements.

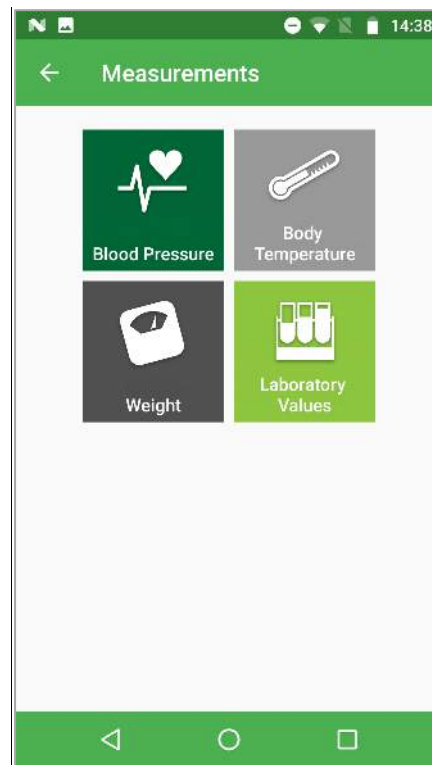


Figure 238: Measurements module of the app.

Action 2: Select an available option, e.g. *Blood Pressure*

Result: A view opens with measurements displayed as graph.

Switching between 'Day', 'Week' and 'Month' views as well as zooming and swiping are possible.

Click the button "<" and receive the data over the past day, week or month and click the button ">" and receive the data of the next day, week or month depending on the chosen period.

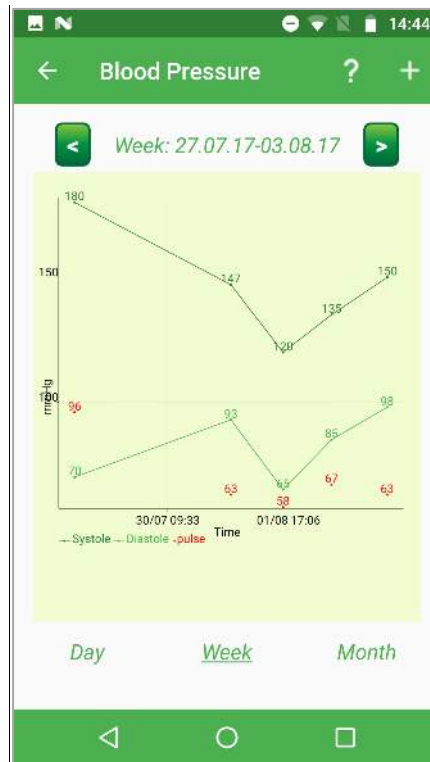


Figure 239: View measurement graphs.

Action 3: For adding a new measurement press the '+' sign in the title row of the view.
Result: A view opens for entering blood pressure values manually.

Figure 240 shows a form titled "Blood Pressure" for entering measurement results. The form includes input fields for Date, Time, Systole (top value), Diastole (bottom value), and Pulse. The values entered are: Date: 30.07.17, Time: 00:44, Systole: 148, Diastole: 87, and Pulse: 53.

Field	Value
Date	30.07.17
Time	00:44
Systole (top value)	148
Diastole (bottom value)	87
Pulse	53

Figure 240: Entering the result of a blood pressure measurement.

Action 4: For adding a new measuring value touch the input field
Result: Keyboard opens to enter the value,

For changing date and time click on date or time field.

After editing the data, the 'save' sign in the title row of the view should be used for storing the data in the local database and also will be transferred to the iManageCancer platform. After saving, the new measurement values are shown on the view with the measurement graph.

Optional: The measurement values are entered into the corresponding text fields automatically if the corresponding supported measurement device is connected via Bluetooth..

8.4.7. Module Health documents

Important: For Android version 6 (Marshmallow) and above, users have to set certain permissions for the Health Docs module.

Please also ensure that internet connection is enabled.

Action 1: Select the *Health Docs* module in the main menu page.

Result: The document list opens. The list shows all available documents (or an empty screen if there aren't any scanned documents yet).

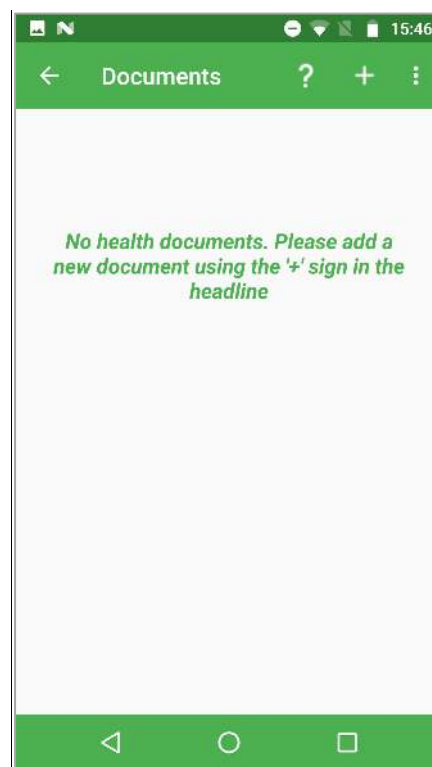


Figure 241: Empty list of scanned health documents.

Action 2: Scan a new document.

Action 2.1: In order to scan a new document press the '+' sign in the title row of the view.

Result: The camera of the mobile device is activated and the user can take a picture. If she/he confirms the taken picture, the image will be saved as document.

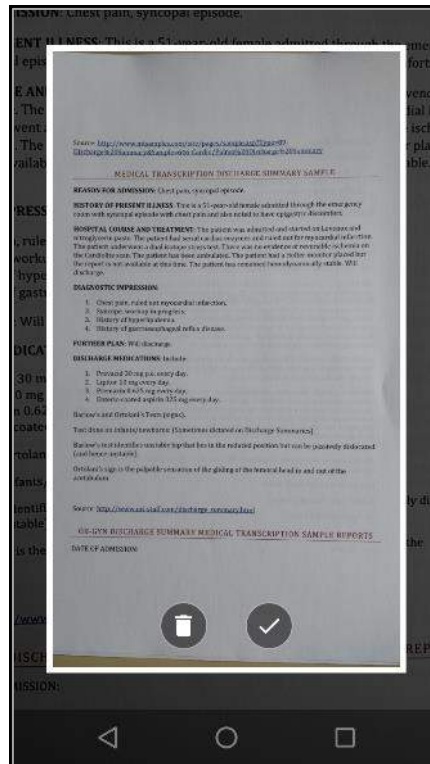


Figure 242: Confirmation for a taken picture.

Action 2.2: Add a second picture for the second page of a health document.
 After the user confirmed the first image, she/he is asked if she/he want to add an additional picture to the set. If she/he selects “yes”, the Action 2.1 has to be repeated.
Result: An additional picture is added to the picture set.

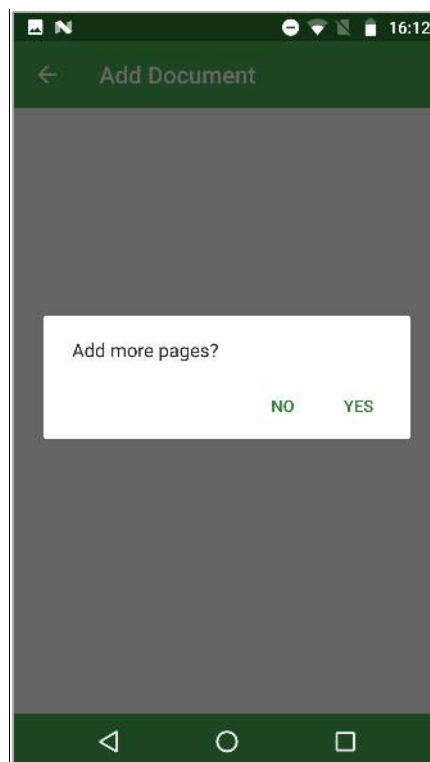


Figure 243: Dialog for adding a new picture to a set.

Action 2.3: Save Metadata of the scanned document.

Result: If the user selects “no”, a page with the document and its metadata opens. Here, she/he can type in the title, date and type of the document (laboratory values, findings and others).

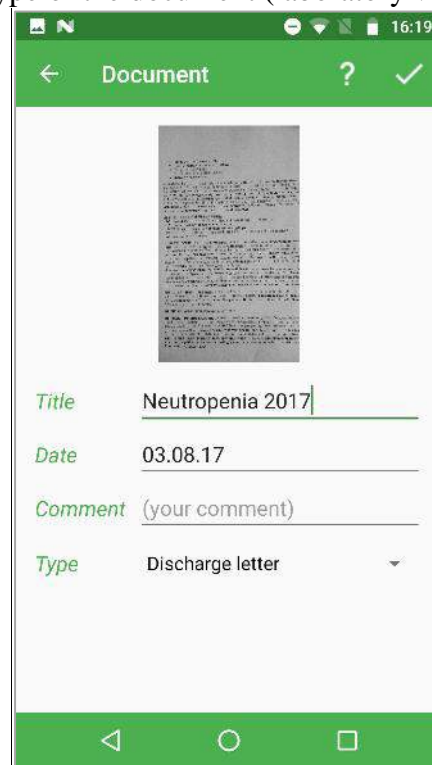


Figure 244: Entering details for a scanned health document.

After pressing the ‘save’ sign in the headline, the document is stored and will be available in the list of documents.

Action 3: Select a scanned health document from the list of documents.

Result: The first page of the documents opens and you can navigate through the document with the “forward” and “back” icons.

Action 3.1: Zoom in and zoom out. You can zoom in and out with two fingers.

Result: The displayed image is zoom able and moveable. The patient can also edit with the pencil icon the description of the document or delete it with the waste basket on the title row of the view.

For editing: a page for editing the file name, the date of document and the category will open. Available categories are: laboratory report, discharge letter, finding and other. Finally, the ‘save’ icon in the title row should be pressed for storing the changes in the local database as well as on the iManageCancer platform.

For deleting: a confirmation dialog for deleting the file will be opened. After pressing the ‘yes’ button, the file is deleted and the user is forwarded to the main view for management of documents. After pressing the ‘no’ button, the file is not deleted and the user is forwarded to the current document view.

Action 3.2: Request specific information

The app allows marking a text in the document and request specific information for a marked text from the Personal Health Information Recommender tool (PHIR) of the iManageCancer platform or from Wikipedia.

Action 3.2.1: Mark a text in the picture of the selected document

The user has to tick the box in order to activate marking. Now he can draw a half-rectangle (down and from left to right) around the selected word (or a sequence of words) with **one finger**.

Result: A specific section of the image is marked by a red rectangle. A dialog box with the selected text is displayed.

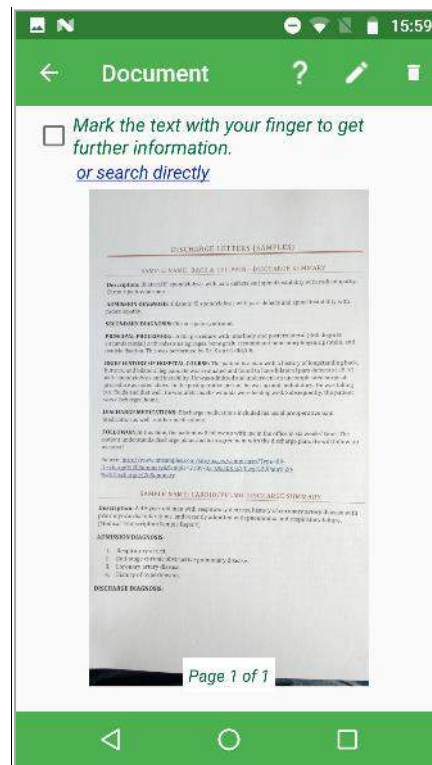


Figure 245: Selecting text in a picture of a document after the box has been ticked.

Action 3.3.2: Request information from the internet

It is possible to edit the shown text in the opened dialog. The patient decides if she/he wants to get information from the PHIR or from Wikipedia.

Result: Dialog Box is closed, a request for explanation for the specific information is sent to PHIR or to Wikipedia.

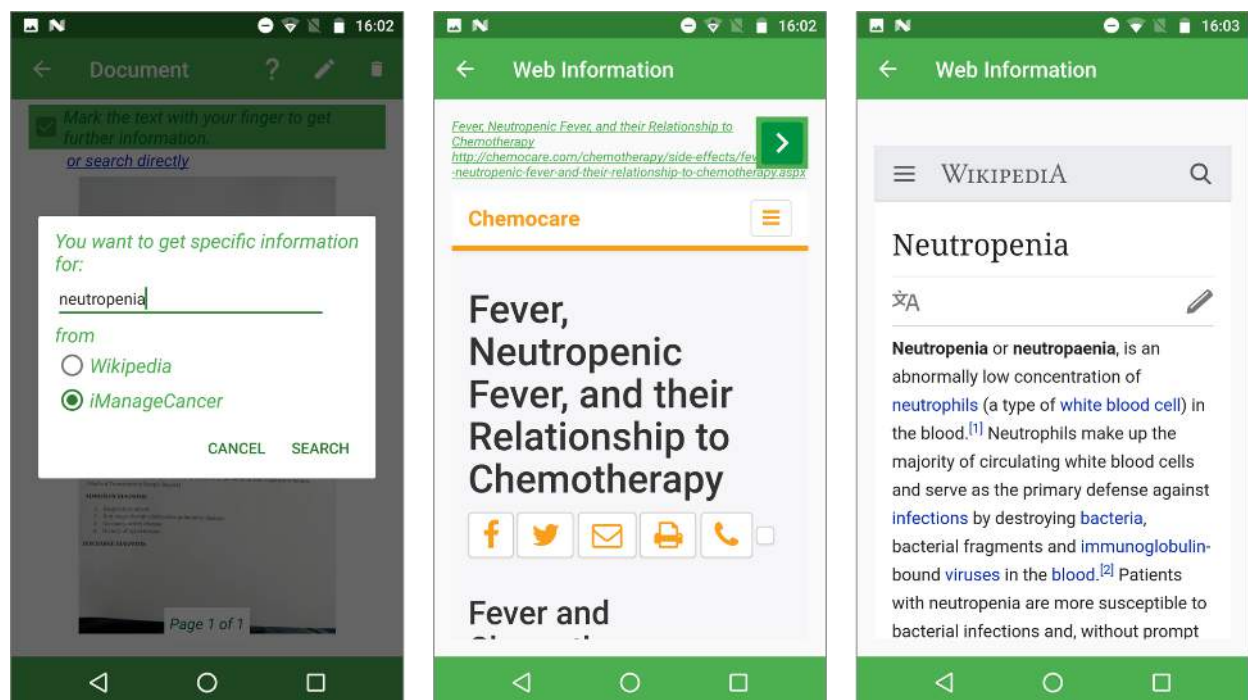


Figure 246: Requesting explanation for a selected text either from Personal Health Information Recommender of the iManageCancer platform or from Wikipedia.

Result: The first result from the received list of URLs from PHIR is shown in the screen. The user can go to the next URL with the button ►. For Wikipedia, the corresponding Wikipedia page is shown.

8.4.8. Module ‘Health Management’ (in combination with app iSupportMyPatients)

Preconditions:

- This functionality can only be tested in combination with the iSupportMyPatients App and with **available internet connection!**
- The patient’s iManageCancer account data should be already stored in the app (see the section 8.4.3).

Action 1: select ‘Health Management’ module in the main menu page (Figure 223)

Result: A page with available health management services of the Care Flow Engine for patients opens.

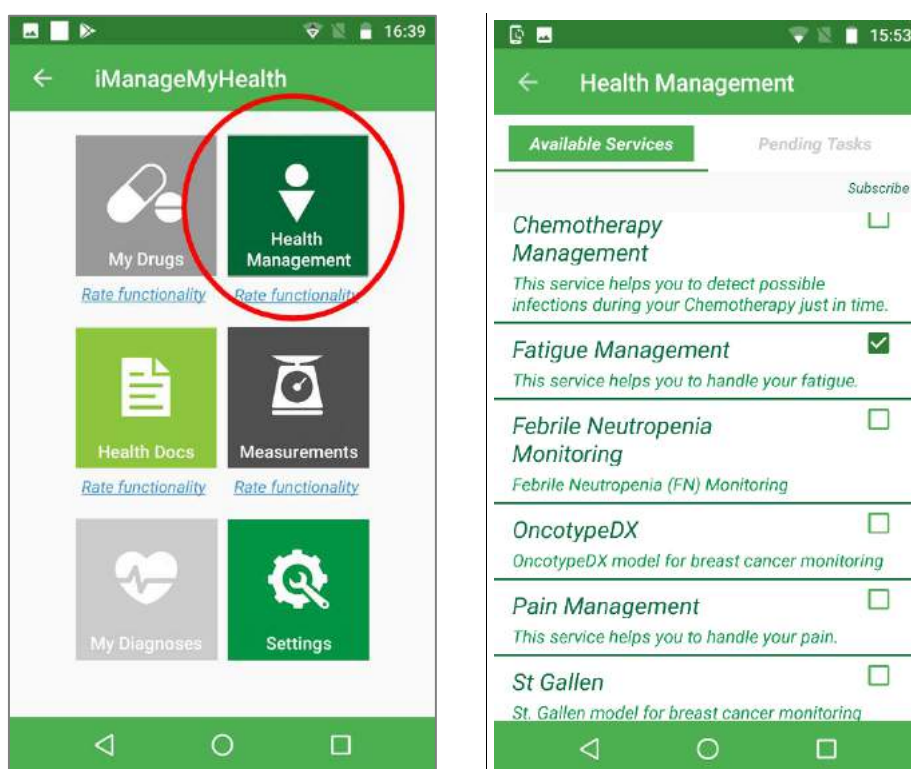


Figure 247: Care Flow Diagrams in the iManageMyHealth app. The patient can subscribe to these services or can unsubscribe from a Care Flow. .



Figure 248: Pending tasks of the patient for the subscribed services Pain Management and Fatigue Management.

Action 2: Subscribe to the *Febrile Neutropenia Monitoring* service in the list of the available services (Figure 247). This service involves also the patient’s physician.

Result: A confirmation dialog is shown for a few seconds.

In background: A health enquiry with questions for the MASCC model is generated for the patient’s physician. The MASCC model calculates a risk score for complications of a patient due to his febrile neutropenia (see deliverables 5.1 and 5.3). A corresponding notification is sent to the iSupportMyPatients app of the physician. A prerequisite for this is that the patient has enabled sharing of his health data with such physician in his iPHR.

The iSupportMyPatients app for health professionals will start automatically when it receives notifications. A notification is shown on the top in the box if the app is opened or as a small icon if the app is closed.

Action: Doctor touches on the notification.

Result: The questionnaire of the MASCC model is shown and the doctor can answer the questions. After pressing the ‘Submit’ button, the answers are sent back to the Care Flow Engine.

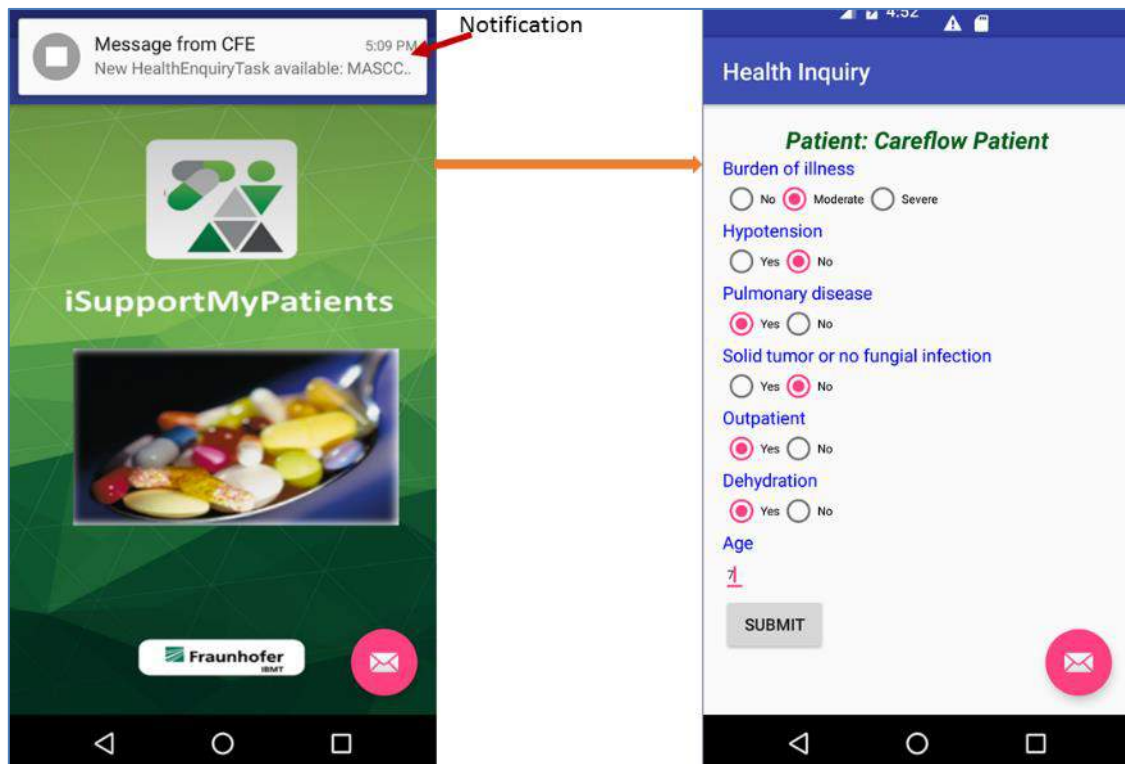


Figure 249: Doctor receives a notification about a health enquiry and opens the enquiry

In background: The answers are used by the Care Flow Engine to call the corresponding model in the model repository. The patient is either classified as low risk or as high risk patient. The Care Flow Engine received this result of the calculation and generates a corresponding information tasks for the doctor and for the patient. The doctor and the patient will receive notifications about the new available information.

Below is the notification for the doctor. After pressing on the notification the page with the information about the classification of the patient to the low or high risk is shown.

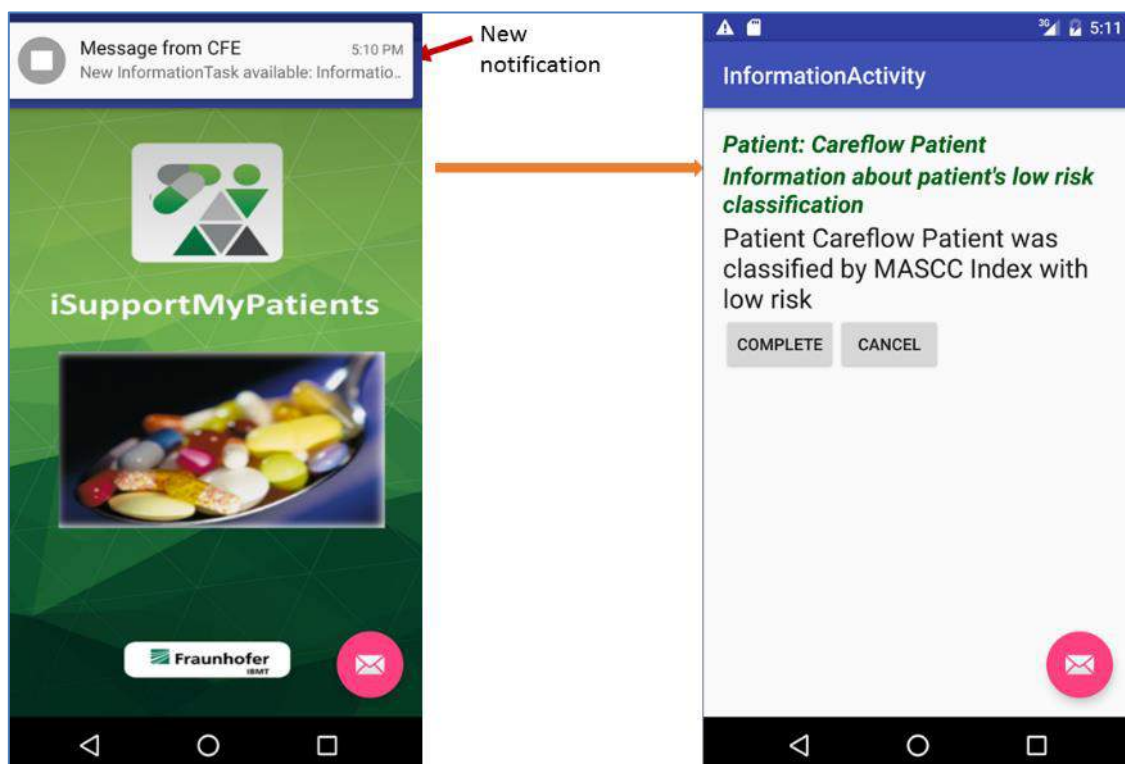


Figure 250: Doctor receives a notification about information about his patient and opens the message
Below is the notification on the iManageMyHealth app about the information task for the patient.

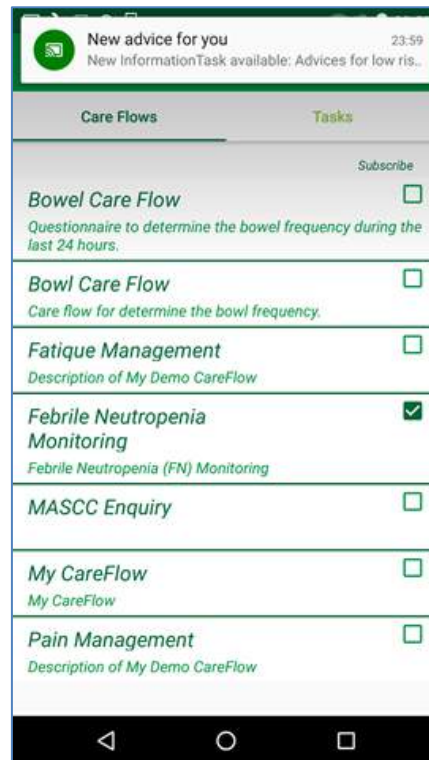


Figure 251: Patient receives notification about helpful information

Action: The user (patient or physician) can touch the notification or he can press directly on the ‘Advices for low risk patient’ task on the main view of the Health Manager.

Result: The advices are shown to the patient. For completing the task, she/he has to press the ‘Close’ button or leave the page using the back buttons.

Explanation: Web links with recommendations for the patient in such messages are shown as links and the patient can select them. In this case she/he will be forwarded to the corresponding web page. Another possibility is to forward the patient directly to the first web link. The patient has the possibility to go back to the page with advices for completing the task and/or to select the next web link. This message also appears in the list of the patient’s tasks (Figure 248) and opens if the task is selected.

Additional information: The web links with recommendations for the patient have been retrieved by the Care Flow Engine from the Personal Health Information Recommender (PHIR).

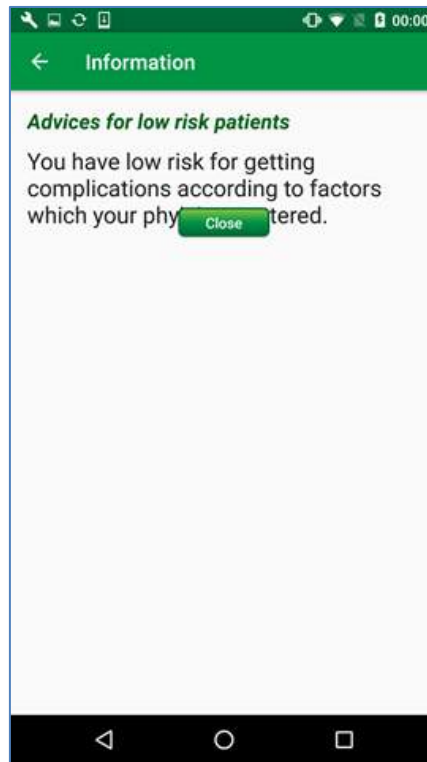


Figure 252: Patient opens a message from the Care Flow Engine.

After pressing the *Close* button or leaving the page using the back buttons, the task is completed and will be removed from the list of the patient's pending tasks (Figure 248).

In background: The Care Flow Engine will generate a new task for the patient. She/he is asked to perform a temperature measurement. The patient will get a notification for this task.

Action: The patient presses on the notification for the measurement task.

Result: She/he is forwarded to the page for entering a measurement value (Figure 240). After pressing the 'Save' sign in the title row of the view, the measurement values will be stored locally and sent to the Care Flow Engine for completing the task. The Care Flow Engine stores the measured values on the iManageCancer platform. The currently entered values can be viewed in the iManageCancer iPHR.

Additional information: In this workflow, the patient will receive a new information task with an advice to visit his doctor if his body temperature is higher than 39.5. His doctor will also receive the notification about his patient's health status and about the recommendation to visit the doctor.

Remark: The available health services are further described in the public deliverable D5.3. Pain Management and Fatigue Management are deployed in the system for the clinical pilots. The other services involve models in the model repository and physicians. They are for demonstration purposes only.

8.5. Installation and configuration guidelines

This app runs on Android 5, Android 6 and Android 7 devices. It has been tested with Motorola Moto G3, Moto G4, Google Nexus 5x, Huawei P8 Lite and Samsung Galaxy S6. We use our own deployment infrastructure for the app and its updates. The app can be downloaded and installed from iPHR with the menu item 'Other apps' after the user has registered on iPHR. It has a built-

in auto-update feature to download new versions when they become available. New versions are downloaded from the protected BSCW server of the project.

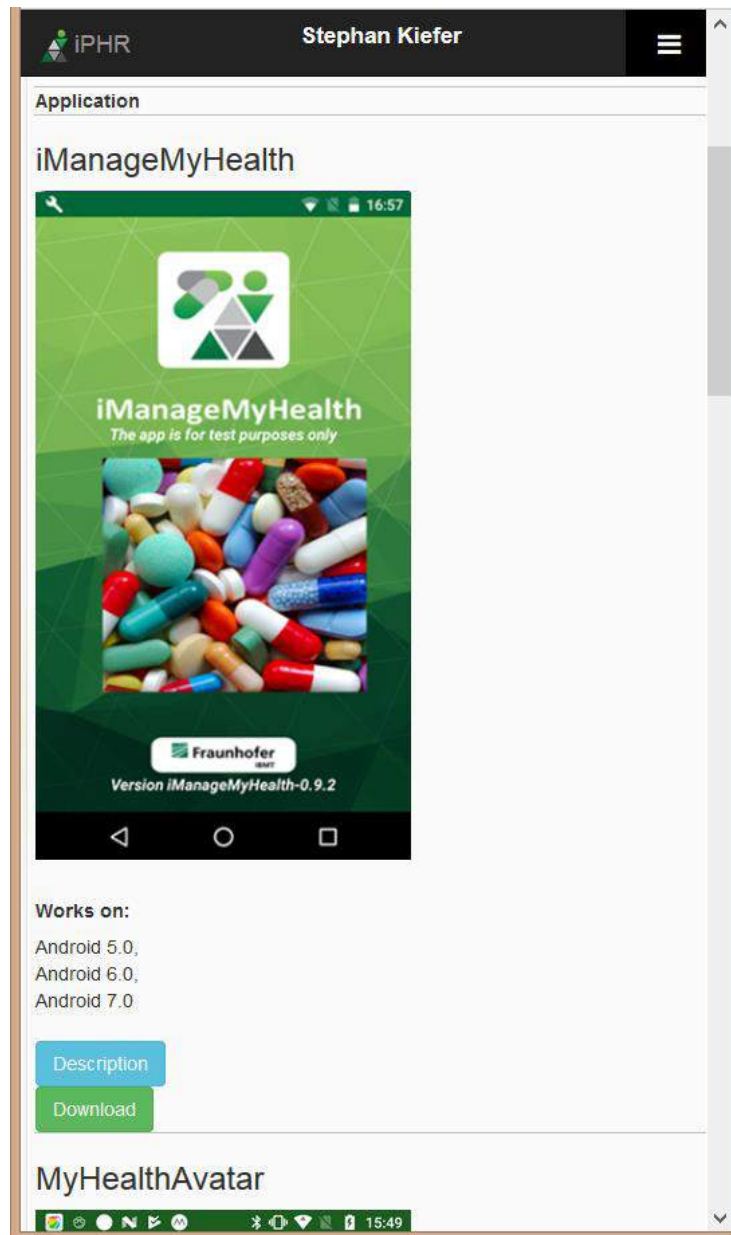


Figure 253: Download option of iManageMyHealth app in iPHR.

The app supports also crash reporting leveraging the Firebase Crash Reporting tool. Errors in the iManageMyHealth app are automatically reported to the Firebase console and the app developers are informed about the errors via e-mail.

▼ Exception java.lang.NullPointerException: Attempt to invoke virtual method 'java.lang.String de.ibmt.his.imc.immd.careflows.model.Task.getId()' on a null object reference

```

de.ibmt.his.imc.immd.careflows.TasksFragment.storeTasksInDb (TasksFragment.java:213)
de.ibmt.his.imc.immd.careflows.TasksFragment.access$700 (TasksFragment.java:56)
de.ibmt.his.imc.immd.careflows.TasksFragment$TasksReceiver.onPostExecute (TasksFragment.java:150)
de.ibmt.his.imc.immd.careflows.TasksFragment$TasksReceiver.onPostExecute (TasksFragment.java:104)
android.os.AsyncTask.finish (AsyncTask.java:660)
android.os.AsyncTask.-wrap1 (AsyncTask.java)
android.os.AsyncTask$InternalHandler.handleMessage (AsyncTask.java:677)
android.os.Handler.dispatchMessage (Handler.java:102)
android.os.Looper.loop (Looper.java:154)
android.app.ActivityThread.main (ActivityThread.java:6123)
java.lang.reflect.Method.invoke (Method.java)
com.android.internal.os.ZygoteInit$MethodAndArgsCaller.run (ZygoteInit.java:867)
com.android.internal.os.ZygoteInit.main (ZygoteInit.java:757)

```

User's approval is required before any app can access other data or apps on an Android device. The iManageMyHealth app requires access to camera and to memory for storing pictures, measurements, and health enquiries:

9. iSupportMyPatients

9.1. *Intended Purpose*

iSupportMyPatients is an informative app intended for health professionals of cancer patients who share their health data with such health professionals in iPHR. This app allows health professionals to start disease management services offered by the Care Flow Engine and to be involved in such management services in order to provide health information about their patients and to obtain information and advice.

The app is not intended for diagnosis and therapy but only for demonstration and test purposes.

9.2. *Provided functionality*

This app allows health professionals to start disease management services offered by the Care Flow Engine and to be involved in such management services in order to provide health information about their patients and to obtain information and advice.

The management services comprise in particular the management of febrile neutropenia during chemotherapy. During the execution of this service, the health professional has to answer a specific questionnaire for his patient who suffers from febrile neutropenia. The service informs him whether the patient has a high risk or a low risk to experience severe complication due to febrile neutropenia.

9.3. *Technical implementation*

9.3.1. *System concept and internal architecture*

This app follows the system concept of the app iManageMyHealth regarding the interaction with the Care Flow Engine. In addition, it retrieves a list of patients from iPHR who wished to share their health data with this health professional.

9.3.2. *Interfaces*

The applied REST services of iPHR and Care Flow Engine are described in deliverable D5.2.

9.3.3. *Database*

The data base tables correspond to tables of iManageMyHealth app for Care Flows.

9.3.4. *Data protection*

The app for health professional manages health enquiries and information messages regarding his patients and is integrated in the security infrastructure of the iManageCancer platform including usage of the OAuth 2.0 protocol for authorization. The app can be used only if a user has created an account as a physician on the iManageCancer portal. However, she/he will only have access to health information of a patient if such patient has invited her/him to share his health information with him in iPHR. In this case, the patient is shown in this app but without additional health information.

The generated and approved iManageCancer account credentials (user name, password and e-mail) should be entered in the iSupportMyPatients app. The app doesn't collect any clinical or personal data of the physician and his patients. The app stores only the unique identifiers of the patients in order to distinguish them from each other.

The physician's account credentials will be further used to exchange data with the Care Flow Engine. The https protocol is used for the transport layer and guarantees secure communication of the app with the iManageCancer portal for collecting unique identifiers of the own patients and with the Care Flow Engine.

9.4. User manual

This manual is for testers of the app.

The app can only work if the internet connection is available.

The following functionality is included to the **iSupportMyPatients** app:

- Entering of the own iManageCancer account
- Viewing own patients
- Receiving of notifications and handling of health enquiry tasks and information tasks from the Care Flow Engine for physicians.

This app is for health professionals. It serves demonstration purposes only. For using the app a health professional should have an iManageCancer account, which can be created on the iManageCancer portal page <https://iphcr.ics.forth.gr/login>. This app requires that patients registered on the iManageCancer Portal enabled sharing of their healthcare data with this specific health professional (at least one patient should add the doctor to a health professional's 'carenet' in their iPHR).

Starting the app

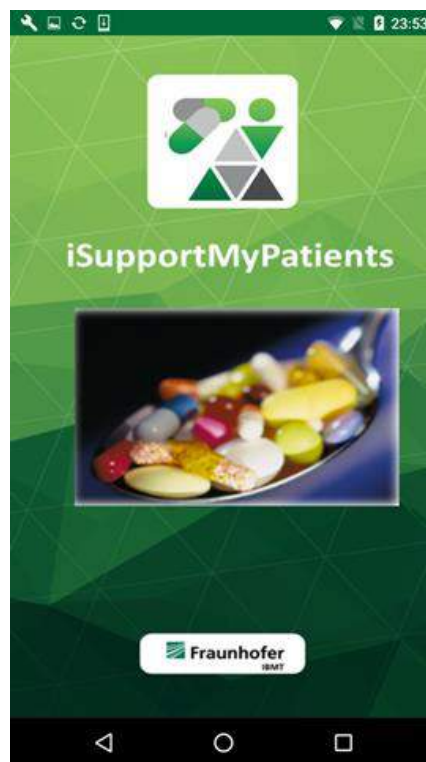


Figure 255: Start page of the iSupportMyPatients app.

After starting the app for the first time, the physician is forwarded to a page where she/he can enter the existing iManageCancer account information.

After storing the account data in the app, a list of his patients is shown to the health professional who wished to share their data with her/him. She/he can select one of the patients and see all available health services for the selected patient to which the doctor can subscribe or unsubscribe as already shown in the user manual of the app iManageMyHealth.

9.5. *Installation and configuration guidelines*

Please refer the installation guidelines for iManageMyHealth app.

10. CareFlowEngine

10.1. *Intended Purpose*

The Care Flow Engine is a prototypic decision support and guidance service engine for patients and health professionals who use the apps iManageMyHealth and iSupportMyPatients. It is a task oriented business process management system, adapted to the eHealth domain that allows domain experts to design and to deploy small-scale disease management programs called Care Flow Diagrams that offer new patient management and patient self-management services to the users. The system is intended to test and demonstration services only and not for usage in daily routine. Usage in daily routine needs to be decided individually for each Care Flow Diagram deployed in the system based on a risk analysis. We propose to release the system only with Care Flow Diagrams that do not underlie the European Medical Device Regulations.

10.2. *Provided functionality*

The Care Flow Engine is a web application. Its user interface, which is shown in the following figures, contains three components:

- a designer to construct Care Flow Diagrams in a customized BPMN format as well as an editor to create enquiries
- a configurator to configure the Care Flow Engine
- a runtime dashboard to manage the deployed Care Flows Diagrams and to monitor their execution
- exemplarily two provisional services for the management of pain and for the monitoring of infections
- exemplarily one provisional service for the management of febrile neutropenia in collaboration with the model repository

The provided functionality implements the use cases and requirements as listed in deliverable 5.2 (Pages 10-12).

10.3. *Technical implementation*

10.3.1. System concept and internal architecture

Together with the Model Repository Framework, the Care Flow Engine forms the Central Decision Support System (CDS) of the iManageCancer Platform. It executes personalised and workflow oriented Care Flow plans that can also leverage the results of predictive models at decision points. The Care Flow Designer shall allow clinical experts to design the corresponding Care Flow diagrams. Two prerequisites for useful Care Flow diagrams from which patients can benefit are a high degree of automation in the iManageCancer platform that allows incorporating the services of the different backbone components in such Care Flow diagrams. Secondly, apps must exist that can leverage the Care Flow Engine with its available Care Flow diagrams and can incorporate this in its user interface concept. With iManageMyHealth and iSupportMyPatients two apps exist that incorporate a client for the services of the Care Flow Engine. An important aspect of Care Flows is to educate patients in the management of specific aspects of the disease. This is supported by offering the patients access to corresponding high quality web content to read or watch and is provided by the Personal Health Information Recommender. This tool can be incorporated in Care Flow diagrams.

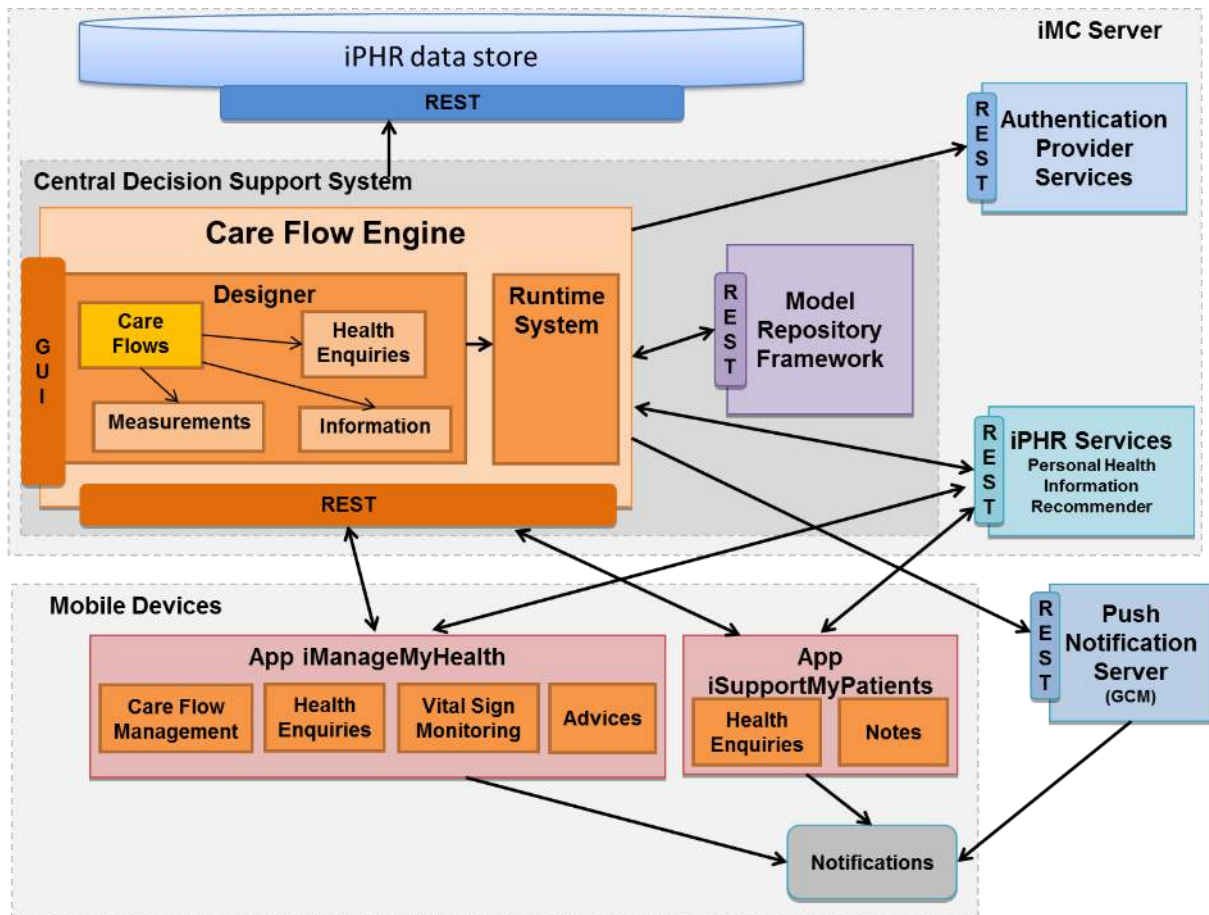


Figure 256: Care flow driven central decision support system of iManageCancer as the control instance of the applications of the users.

Figure 256 presents an architectural view from the perspective of the central decision support system with its Care Flow Engine and the model repository. Communication with other components of the platform is based on REST interfaces.

The Care Flow Engine represents a workflow-driven and business process management based approach to decision support that guides patients and doctors with their end devices and corresponding applications through the care pathway and supports them in decision-making and management with corresponding precise information provision. The details are presented in D5.1 and D5.2.

The main idea behind this approach is that domain experts design formal definitions of a care flow based on clinical guidelines, knowledge on care pathways and an organisational model for integrated care with the patient as the co-manager of his health in the centre of it. These Care Flow diagrams are personalised for a specific patient in individual Care Flow plans and executed by the Central Decision Support Unit of the iManageCancer Platform, the so-called Care Flow Engine. Based on these Care Flow plans, the Care Flow Engine guides the patient, but optionally also the healthcare team, through the management of his disease and related co-morbidities by issuing tasks and recommendations to the patient and his healthcare team and by controlling the execution of the Care Flow plan based on the results of tasks and monitored health status of the patient. In the design phase of the Care Flow diagram, further knowledge is modelled as a set of clinical rules that control execution of the plan.

The iManageMyHealth app installed on the patient's mobile device, downloads and processes these tasks for the patient, issued by the Care Flow Engine. Tasks for patients are typically health

enquiries and measurement tasks, but also information tasks with the recommendations obtained from the Care Flow Engine. Results are sent back to the Care Flow Engine for further assessment and control. Similarly, iSupportMyPatients, the app for physicians, downloads and processes tasks for the patient’s physician. The apps receive also notifications from the Care Flow Engine about tasks assigned to health professionals or patients.

An initial set of care flows was modelled in collaboration with oncologists, involved in the iManageCancer project, for different aspects of the management of cancer with a focus on those aspects of the disease that can be managed by the patient him/herself. Proper integration with the e-diary and PHR of patients was achieved by implementing client and server functionality in these systems in order to access the iManageCancer data store or to download tasks, to process them and to send results of their execution back. A view on the high level architecture of Central Decision Support Unit of the iManageCancer platform and its interfaces to the rest of the platform is presented in Figure 256.

Health enquiries represent a fundamental type of tasks in a Care Flow diagram. As measurement tasks, they represent a possibility to collect actual information on the patient’s current condition by the patient himself or his doctors.

When a new Care Flow diagram is deployed in the system, it is converted in the mentioned Activiti BPMN object model for its execution by the Care Flow Engine. During the execution of the care flow for an individual patient, client apps are able to query for user tasks. The different types of tasks are rendered and presented to the user in the app.

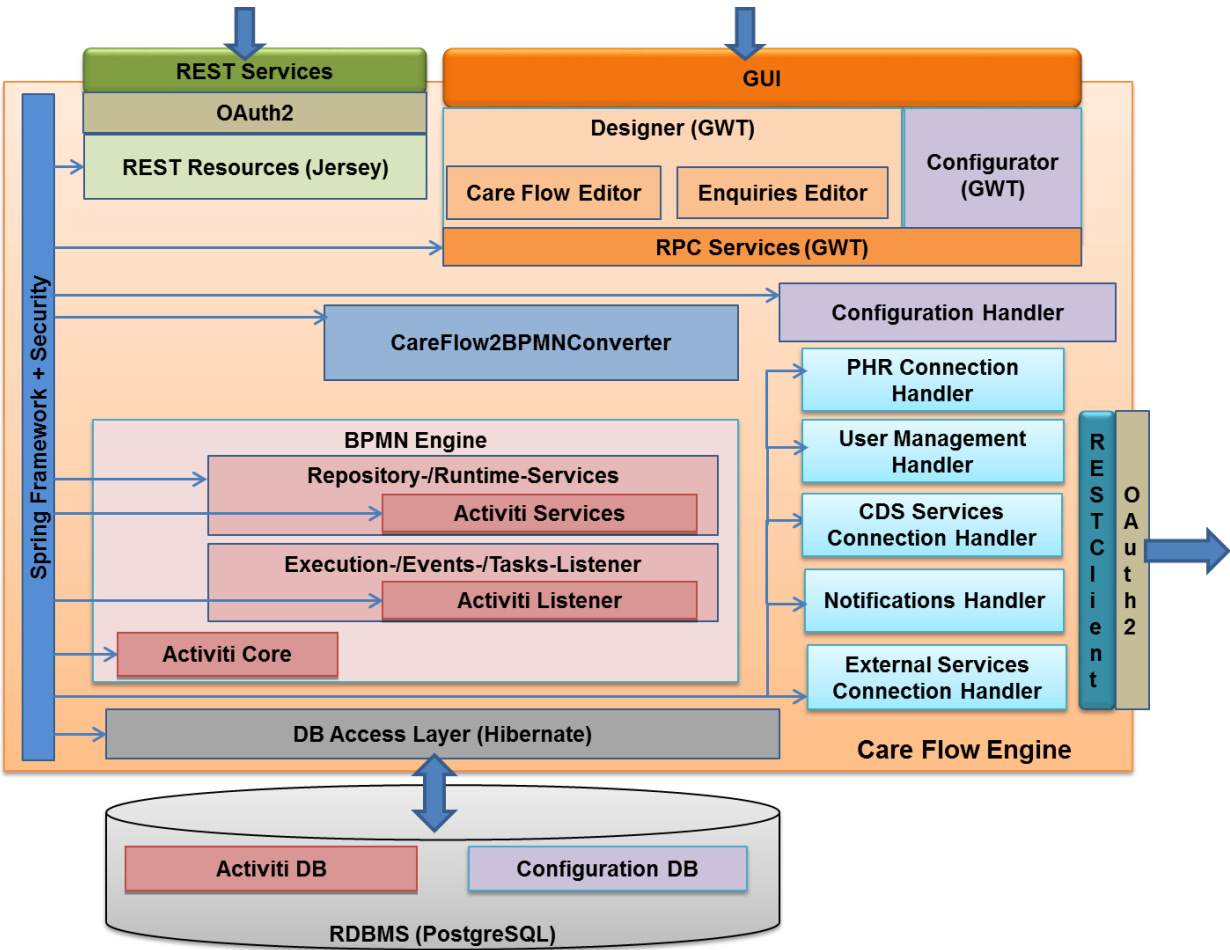


Figure 257: Internal architecture of the Care Flow Engine.

The internal architecture is depicted in the picture above. The kernel is provided by the open source BPMN Engine Activiti⁹.

The system provides the following functionalities:

- Designing basic Care Flow diagrams
- Deploying and running Activiti BPMN workflows
- Providing repository services
- Providing runtime services
- Querying tasks (with properties)
- Integration of external components via listener (events, gateways and tasks)
- Storage of process definitions and processes

10.3.2. Interfaces

Internal and external interfaces are described in detail in deliverable 5.2

10.3.3. Database

The Care Flow Engine is built on an Activiti¹⁰ kernel. Activiti is an open-source workflow engine that can execute business processes described in BPMN 2.0. The Activiti DB is explained in <http://activiti.org/userguide/index.html#database.tables.explained>.

10.3.4. Data protection

The Care Flow Engine does not store personalised information of a patient in its database outside an executed Care Flow for this patient. As soon as the Care Flow is terminated, the data collected in this Care Flow is deleted except of data that is sent to and stored in iPHR.

The OAuth2 authentication and authorization of the iManageCancer platform protects the user interface of the CFE and it's REST Services from unauthorised access. HTTPS is used to secure the data transmission.

10.4. User manual

This user manual is for system administrators and clinical domain experts.

The Care Flow Engine is a web application. Its user interface, which is shown in the following figures, contains three components:

- a designer to construct Care Flow diagrams in a customized BPMN format as well as an editor to create health enquiries
- a configurator to configure the Care Flow Engine
- a runtime dashboard to manage the deployed Care Flows diagrams and to monitor their execution.

⁹ <https://www.activiti.org/>

¹⁰ <https://www.activiti.org/>

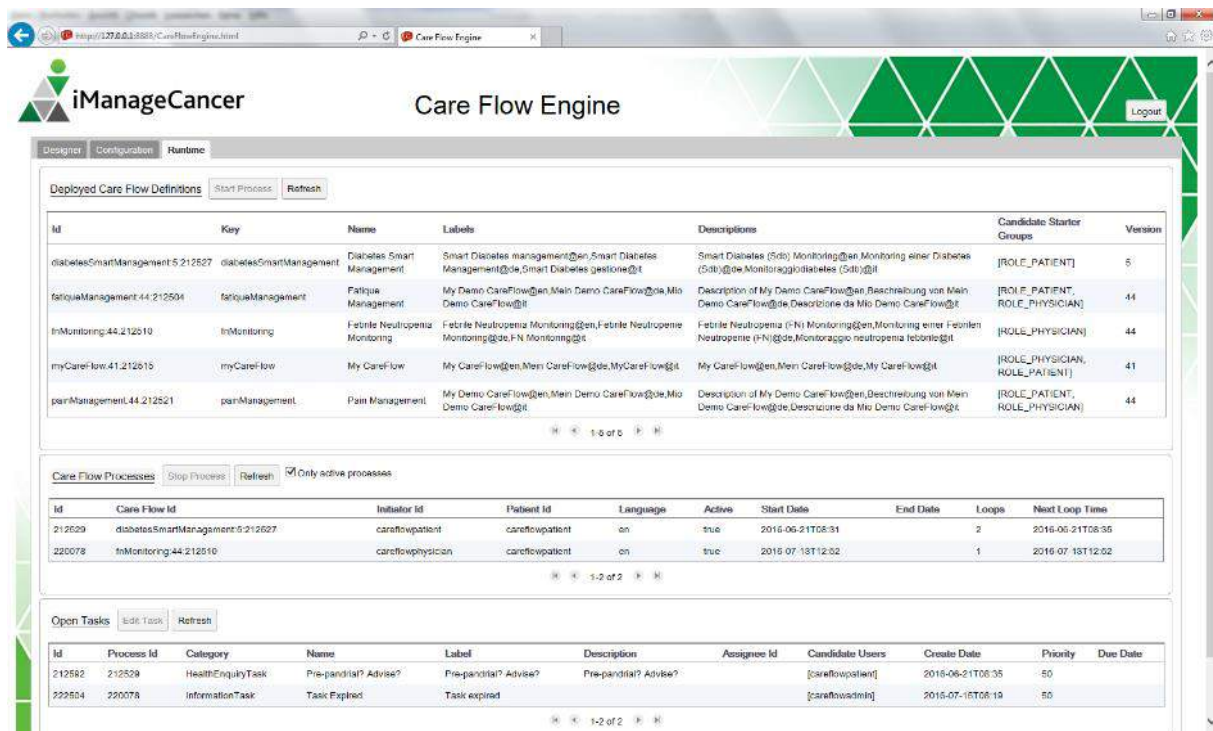


Figure 258: Main user interface of Care Flow Engine with three tabs 'Designer', 'Configuration' and 'Runtime'. This screenshot shows the 'Runtime'.

The GUI of the Runtime provides the following functionality:

- List of Care Flow diagrams deployed in the system (section 'Care Flow Definitions')
- Monitoring of the processes of the executed Care Flow processes for individual patients
- Monitoring of the open tasks of these processes
- Possibility to simulate executing processes and tasks in the runtime environment.

In the following screenshot, a Care Flow process is started for a specific patient. This feature is made available for test purposes. Usually, Care Flow plans are started by the users themselves with their apps by calling the corresponding REST service of the Care Flow Engine.

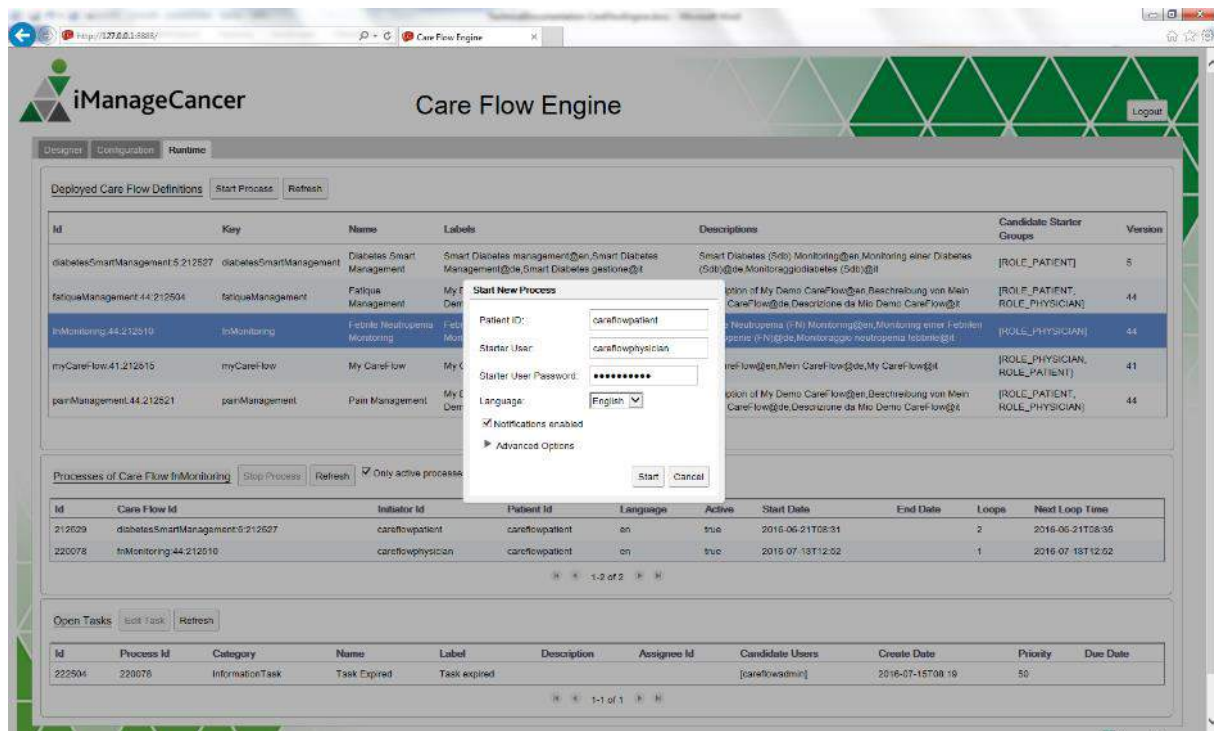


Figure 259: Dialogue 'Start New Process' in the Runtime dashboard.

In the following screenshot, an open enquiry task for a specific user was edited. Answers can be given in this dialogue for testing purposes. Usually, open tasks are presented to the corresponding user in his app. The app of the user gathers the answer from him and sends it back to the Care Flow Engine through the corresponding REST service of the Care Flow Engine.

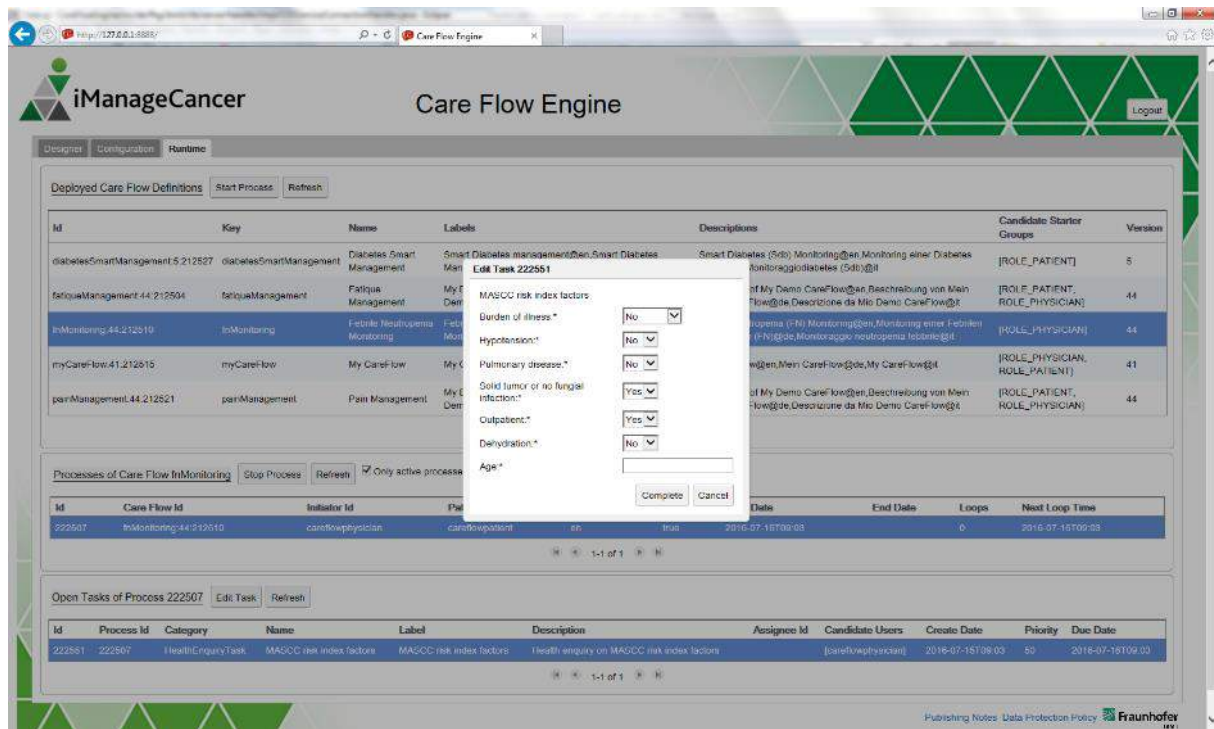


Figure 260: Dialogue 'Edit task' in the Runtime dashboard.

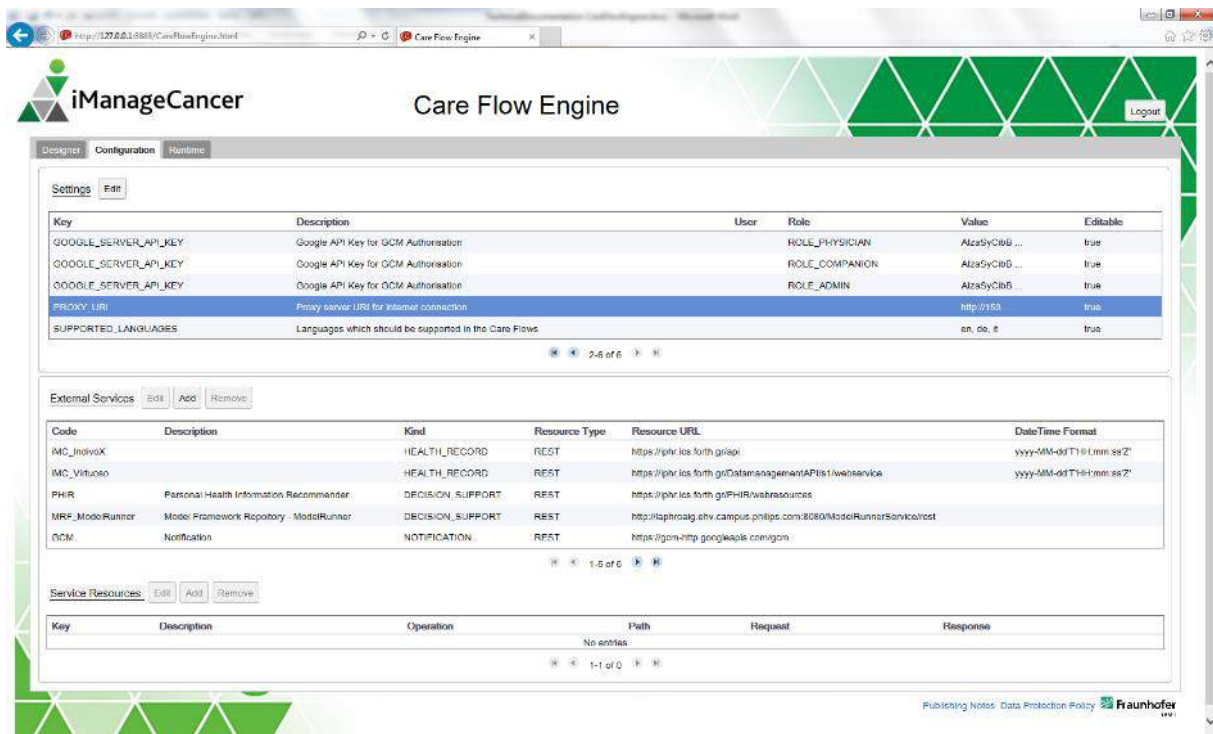


Figure 261: Main user interface of Care Flow Engine with three tabs 'Designer', 'Configuration' and 'Runtime'. This screenshot shows the 'Configuration'.

Figure 261 shows the configurator. The GUI offers

1. to specify external services with their keys as resources that can be used in care flows. The following services are actually available to the Care Flow Engine:
 - iManageCancer Health Record
 - iManageCancer Model Repository
 - Notification service using Google Cloud Messaging (GCM) services
 - Central iManageCancer User Management
2. to configure
 - Languages which should be supported in the care flows
 - Proxy server URI for internet connection
 - Google API Key for GCM authorisation
 - GCM tokens for addressing users in the push notifications. They are entered dynamically by apps of users through the corresponding REST service of Care Flow Engine.

Creating Care Flows with Care Flow Designer

The graphical user interface of the designer for care flows is presented in the following figure with an example of a Care Flow diagram composed of different types of tasks.

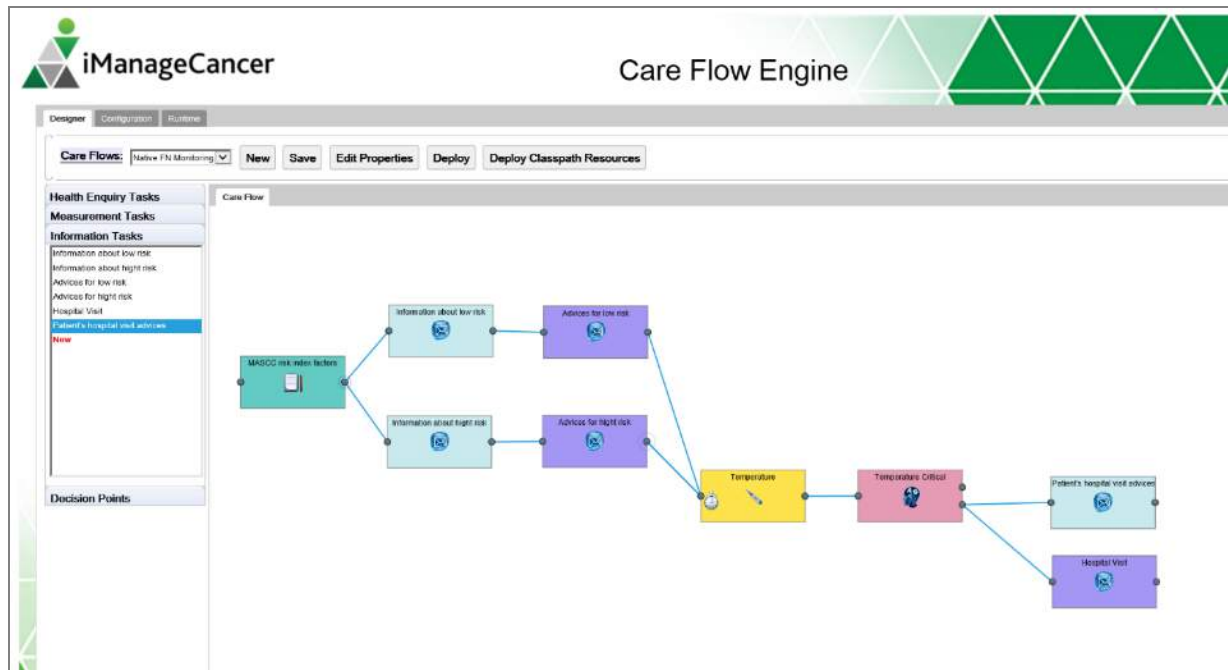


Figure 262: Canvas of the designer with an exemplary Care Flow Diagram.

The GUI of the Designer has the following functional areas:

1. Upper toolbar for the management of Care Flow Diagrams with the following functions:
 - **Select menu 'Care Flows':** Serves to select a care flow from the internal database.
 - **New:** Create a new Care Flow diagram.
 - **Save:** Store a care flow in the internal database.
 - **Edit Properties:** Opens a dialog for editing the properties of the care flow.
 - **Deploy:** Deploys a care flow from the database in the runtime environment of the Care Flow Engine.
 - **Delete:** Delete a workflow from the database and the BPMN repository
 - **Deploy Classpath Resources:** Deploys Care Flow diagrams stored in XML notation in the runtime environment of the Care Flow Engine. This is an alternative method for software developers to deploy Care Flow diagrams that have been developed with external tools with a software update of the Care Flow Engine.

Care Flow Properties

Please edit the care flow properties.

Name Registry Careflow

Used Roles

☒ Role Physician ☐ Role Patient

Start Times 8:00,20:00

Day Cycle 3

Description

Careflow for cancer registry

Ok Cancel

Figure 263: Editor for properties of the care flow.

2. Left toolbar serves to select an element to edit it and add it to a Care Flow diagram in the canvas. The following type of elements can be chosen:

- health enquiry task
- measurement tasks
- information tasks
- decision points (under preparation)

A specific editor is available for health enquiries that allows the user to design questionnaires with the following type of answers

- Text
- Date
- Yes/No
- Enumerations
- Numbers, optionally with a unit
- Score

The individual values can be linked with an external service. This mechanism allows, for example, storing the answers in the PHR. E.g. questionnaires (health enquiries) are stored in JSON format in the iPHR database.

The tool also contains an editor for information tasks that allows the user to design simple information notifications.

Health Enquiry Tasks

Local enquiry
Cancer Registry
New

Measurement Tasks
Information Tasks
Decision Points

Care Flow Enquiry

Name: Cancer Registry

Due Date: PT1400M

Candidate Users: Physician

Description:
Cancer registry by physician

Questions

Label	Datatype	Unit	Values	Keys	Up	Down	Delete
Diagnosis	Text			add delete	up down	delete	
Tumor Localization	Text			add delete	up down	delete	
Date of Diagnosis	Date			add delete	up down	delete	
Participation in study	Yes/no			add delete	up down	delete	
Diagnosis done by	Enumeration		Symptoms	add delete	up down	delete	
Tumor Size	Number	mm3		add delete	up down	delete	
Grading	Score		1	edit delete	up down	delete	

New

Save Cancel

Figure 264: Designing health enquiries. A cancer enquiry is shown in this screenshot that utilises all supported types of question items.

3. A canvas to design a Care Flow diagram by placing graphical symptoms of different tasks on the dashboard and connecting them in the desired order. The designed Care Flow diagram is stored in JSON format in the database too. It also contains references to the definitions of tasks in the database that are used in the Care Flow diagram.

The user can attach one or several conditions to the connection of two subsequent tasks. The expression, which combines these conditions, must be fulfilled to allow the Care Flow plan to proceed to the next task. For this purpose, the user clicks on the connection and an editor will open to enter conditions. For each condition, a pre-existing variable of this Care Flow plan will be compared with another variable or with a value. The logical operators '=', '<', '>', '<=' and '>=' can be used for this. The different conditions can be connected with the Boolean operators 'and', 'or', 'nor' and 'nand'.

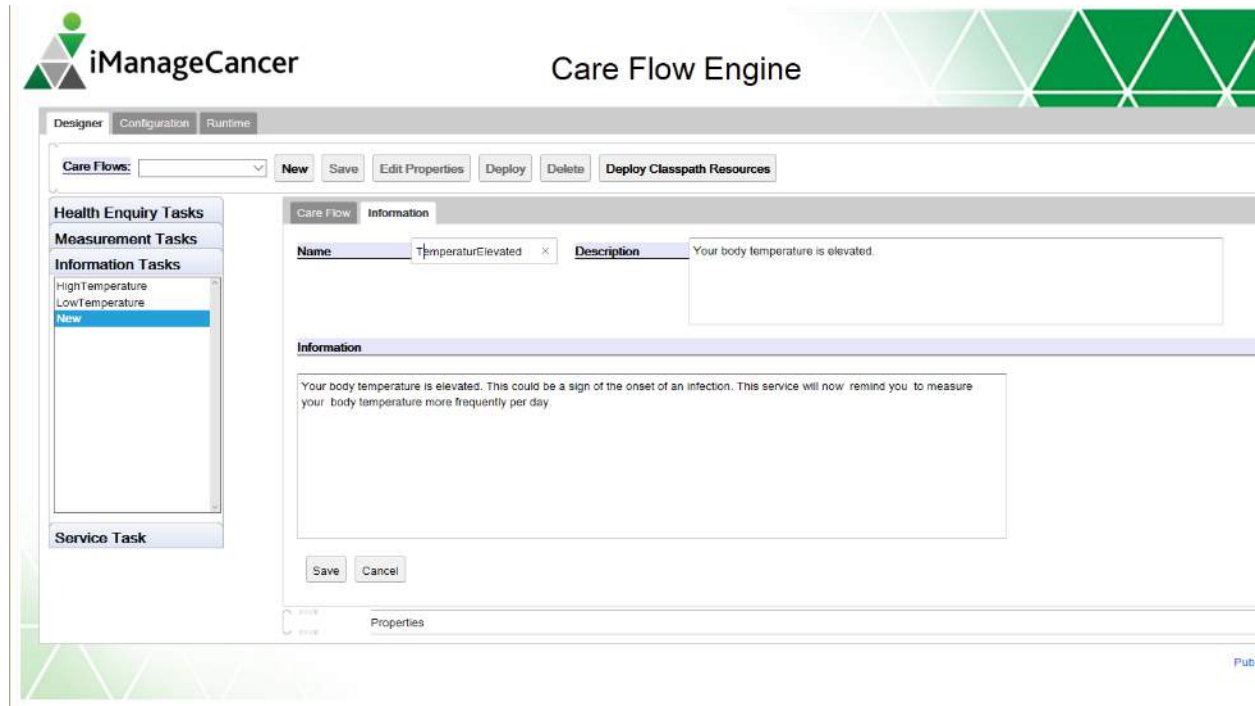


Figure 265: Designing information tasks.

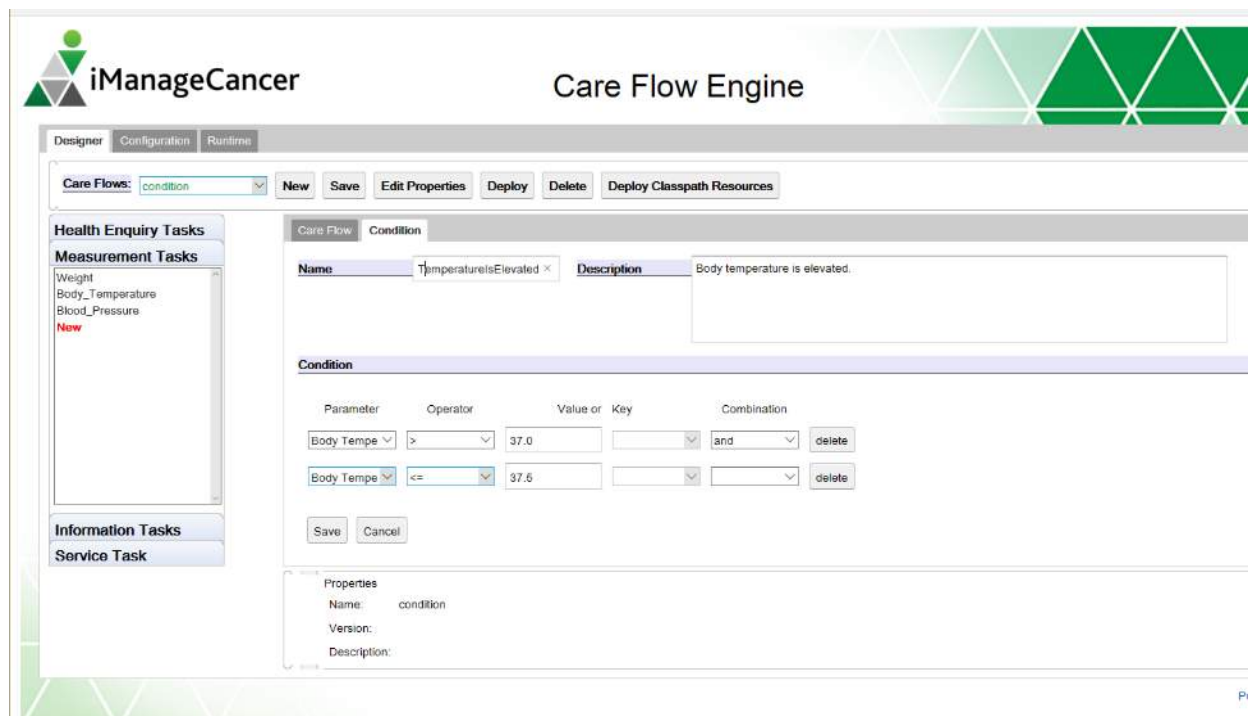


Figure 266: Defining conditions.

Care Flow diagrams can also be composed with the Activiti BPMN Designer that needs to be installed as a plug-in in the integrated development environment Eclipse used for the development of Care Flow Engine. The created Care Flows are stored as BPMN files (in XML notation) in the project of the Care Flow Engine (folder `src/resources/bpmn`). With an update of the Care Flow Engine, the new Care Flow will be transferred to the war folder of the web application server. After a restart of the application server, the new Care Flow can be added to the BPMN repository of the Care Flow Engine by pushing the button 'Deploy Classpath Resources'. It can now be executed. Alternatively, the BPMN file can be stored in the web space of the web server.

In this way, more types of Care Flow Diagrams can be designed and data objects from iPHR can be included in conditions at decision points. The latter is not possible yet in the current version of the Care Flow Designer.

10.5. *Installation and configuration guidelines*

Technical requirements

- Cores 2:
- RAM: 8 GBytes
- HD/SSD: 16 GBytes

Software requirements

- SUSE Linux Enterprise Server 12 SP1
- Java 8
- Tomcat 8.5
- PostgreSQL 9.5

Installation details

- The GUI and the Rest-Service run on port 443
- The PostgreSQL server runs on port 5432
- The CFE use two database 'activiti' and 'cfe' that you must create before your start the application.
- The schemata and the data for 'activiti' are created by the application itself.
- For the schemata and the data for 'cfe' you must start the class `de.fhg.ibmt.cfe.server.utils.DBUtil.java` as Java application.

11. Model Repository

11.1. Intended Purpose

The Model Repository objective is to store predictive models, it is part of the CDS (Clinical Decision Support) along with the Care Flow Engine.

Models added are the validated knowledge ones from the research, and they can also be used by the CDS framework for their continuous validation on new data.

The model repository with its models is intended for research and demonstration purposes only. It shall not be used in clinical practice and patient management.

11.2. Provided functionality

The main use of the model repository is to call the predictive models through web services.

The available models are the following:

- **St. Gallen**, applies St. Gallen for early breast cancer. 2013 Consensus: http://www.oncoconferences.ch/mm/Consensus_SG-2013.pdf
- **Body Mass Index**, calculates the body mass index. Weight is in kg, height is in cm.
- **MASCC (Multinational Association of Supportive Care in Cancer)**, MASCC model for Febrile Neutropenia
- **ALL-BFM 2000**, the objective of the model is to build a predictive model for the Acute Lymphoblastic Leukemia (ALL), the most common malignant disease in childhood. The treatment protocol includes three basic treatment arms according to patients with standard risk (SR), medium risk (MR) and highrisk (HR) of relapse. For the most effective treatment with less side effects, it is essential to know the risk group as early as possible. The value of minimal residual disease (MRD) at treatment day 78 has been found out to be a strong predictive value for the outcome. The objective of the predictive model is to calculate a risk for high MRD values – and therefore high risk of relapse – based only variables which are available very early. <https://www.ncbi.nlm.nih.gov/pubmed/24110412>
Example: BLAST_15 = [58.8, 78.4] AND KOINI >= 2.2 AND LAENGINI = [51.4, 89.8] => RULE_1_MATCH = TRUE
- **Wilms Tumour**, predicts the phenotype that can determine if a patient is characterized, based on his/her miRNA expression data, to a Wilms tumour patient or to a healthy person.
- **BRC (Brest cancer): Intermittent Bevacizumab Treatment Prediction**, predicts the effect of a user-specified intermittent bevacizumab monotherapy scheme to a specific breast tumour.
- **BRC (Brest cancer): Bevacizumab Comparison**, compares the treatment outcomes while applying fractionated versions (total amount of drug spread out over total treatment period) of an original bevacizumab monotherapy scheme to a specific breast tumour.
- **Vincristine-Actinomycin**, predicts the effect of a user-specified combination treatment scheme consisted of Actinomycin and Vincristine on a specific Wilm's tumour.

11.3. Technical implementation

11.3.1. System concept and internal architecture

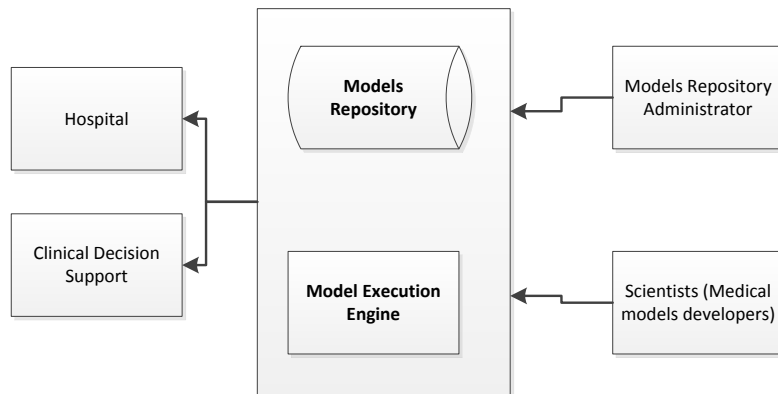


Figure 267 : Model repository and model execution engine

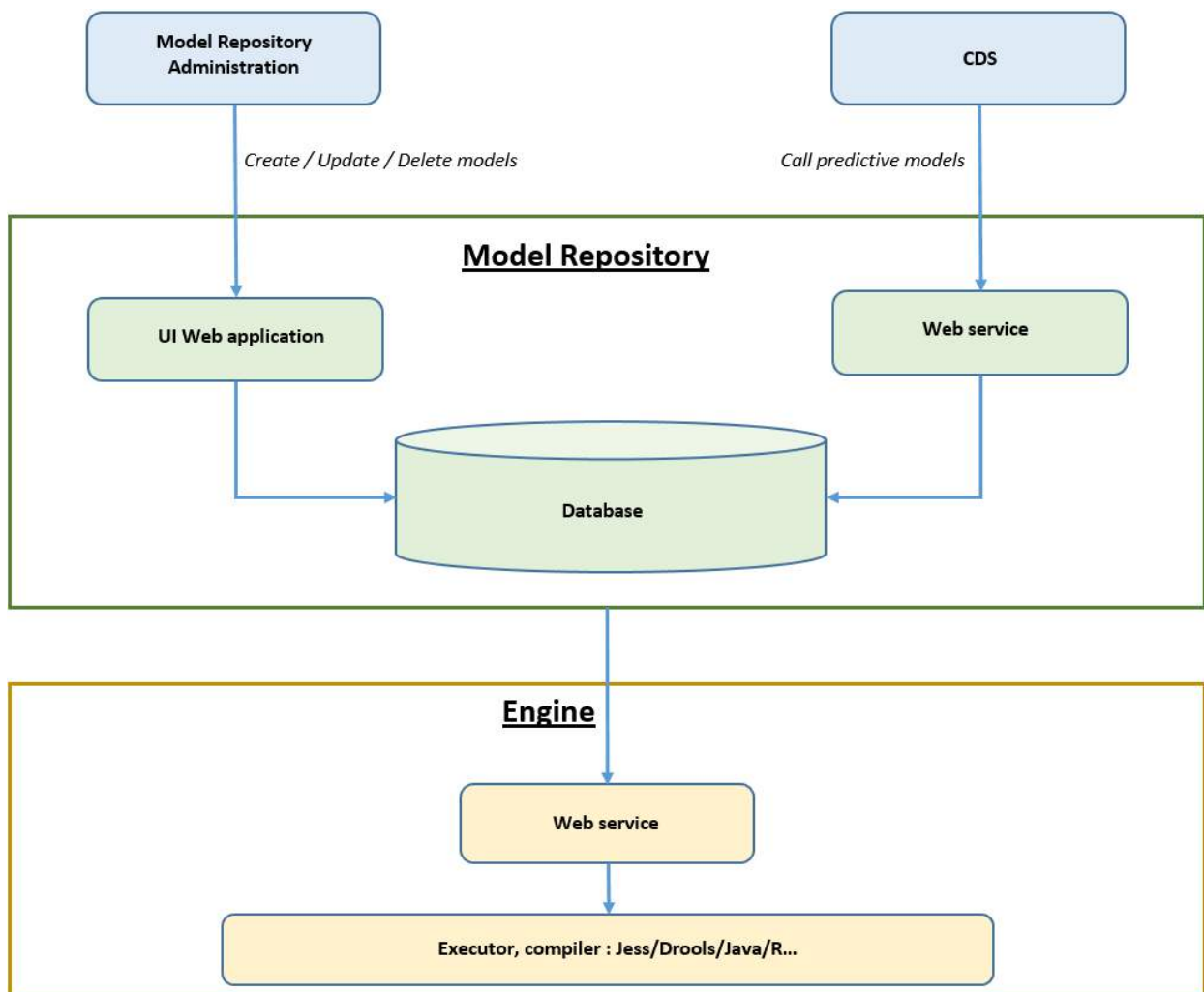


Figure 268 : Models repository and engine architecture.

11.3.2. Interfaces

The Model Repository is accessed through web service as described below.

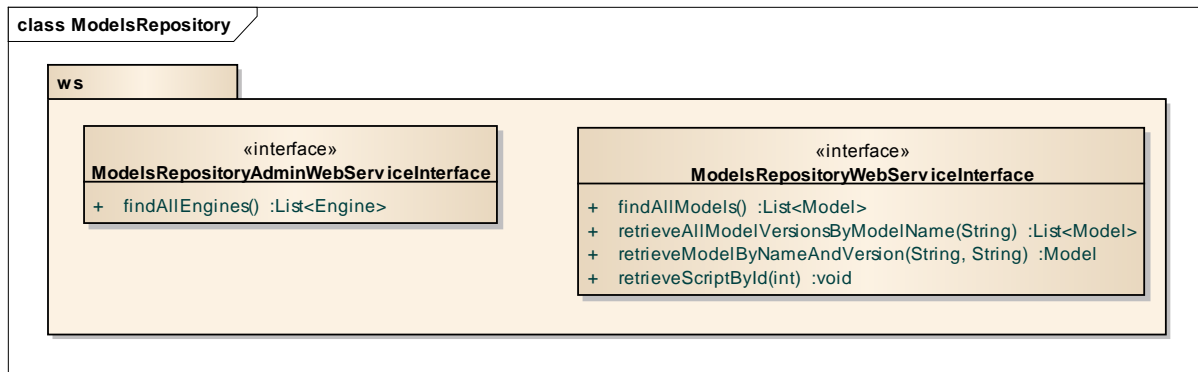


Figure 269 : Models repository interface.

ModelsRepositoryAdminWebServiceInterface

- findAllEngines() – returns a list with all engines register to the Models Repository

ModelsRepositoryWebServiceInterface

- findAllModels – returns a list with all models stored by the Models Repository
- retrieveAllModelVersionsByModelName – returns a list with all versions of a model specified by name
- retrieveModelByNameAndVersion – returns a model with a given name and version.
- retrieveScriptById – returns the script (blob) of the model.

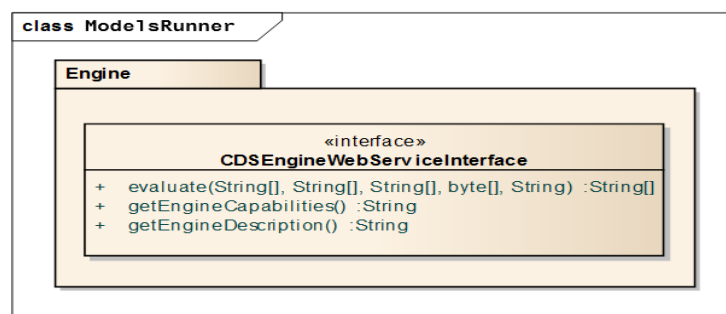


Figure 270 : ModelsRunner interface.

ModelsRunnerServiceInterface

- evaluate – evaluates a given model and returns the output parameters.
 - o *inputParameterNames*: String[] – the name of the input parameters
 - o *inputParameterValues*:String[] – the values of the input parameters (same order as their names)
 - o *outputParameterNames*: String[] – the name of the expected output parameters
 - o *script*: byte[] – the Jess script that will be sent to Jess
 - o *type/action*:String - type or purpose of the script
 - o returns the output parameter values (in the order the names are given as input)
- getEngineCapabilities – returns the engine capabilities (e.g. “jess” which means the engine can run Jess code)
- getEngineDescription – returns a short description of the engine.

11.3.3. Database

The central element in the data model is the Model element. It has a list of Parameter (which could be input and output) as well as a list of Script elements. Each script is bound to one or more Engine elements, depending on the execution requirements of the script (Jess, Drools, Java, R...).

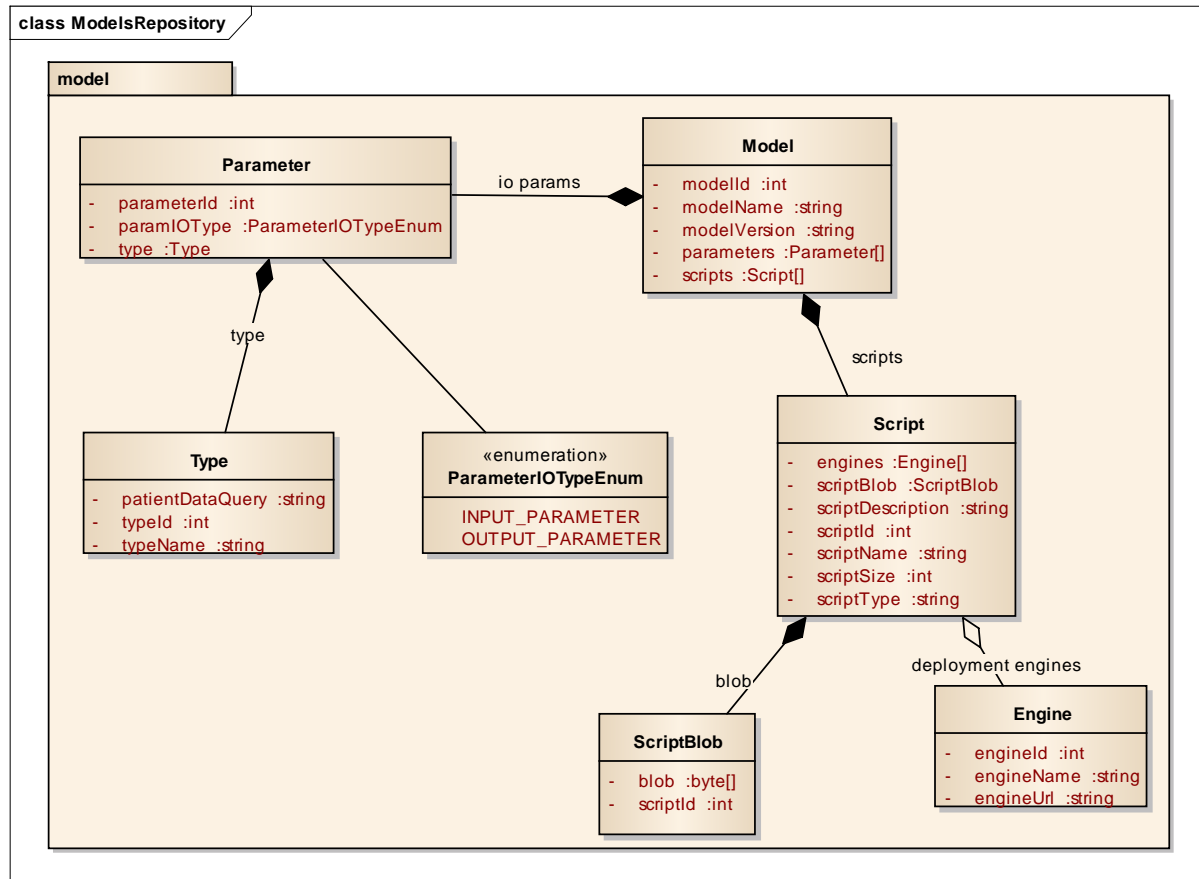


Figure 271 : Models repository class diagram.

11.3.4. Data protection

OAuth2 is used to protect the access to the service; the patient data themselves are not saved in the model repository, they are only provided as input for the models.

The OAuth2 authentication and authorization protects the service to be called by unwanted users.

11.4. User manual

The models are queried and executed through web services; the following details are related to the execution of MASCC model but the same protocol can be used for any other model.

11.4.1. Available models

The available predictive models are the following.

- **St. Gallen**, applies St. Gallen for early breast cancer. 2013 Consensus: http://www.oncoconferences.ch/mm/Consensus_SG-2013.pdf
- **Body Mass Index**, calculates the body mass index. Weight is in kg, height is in cm.
- **MASCC (Multinational Association of Supportive Care in Cancer)**, MASCC model for Febrile Neutropenia
- **ALL-BFM 2000**, ALL-BFM 2000 model implemented in Jess.

*Example: BLAST_15 = [58.8, 78.4] AND KOINI >= 2.2 AND LAENGINI = [51.4, 89.8]
=> RULE_1_MATCH = TRUE*

- **Wilms Tumour**, predicts the phenotype that can determine if a patient is characterized, based on his/her miRNA expression data, to a Wilms tumour patient or to a healthy person.
- **BRC (Breast cancer): Intermittent Bevacizumab Prediction**, predicts the effect of a user-specified intermittent bevacizumab monotherapy scheme to a specific breast tumour.
- **BRC (Breast cancer): Bevacizumab Comparison**, compares the treatment outcomes while applying fractionated versions (total amount of drug spread out over total treatment period) of an original bevacizumab monotherapy scheme to a specific breast tumour.
- **Vincristine-Actinomycin**, predicts the effect of a user-specified combination treatment scheme consisted of Actinomycin and Vincristine on a specific Wilm's tumour.

11.4.2. Rest API

The use of the Model Repository in the CDS (Clinical Decision Support) is through the execution of a model for a specific patient.

URL	<i>http://laphroaig.ehv.campus.philips.com:8080/ModelRunnerService/rest/executeModel</i>
Method	<i>POST</i>

Description	<i>Execute a model given input parameters and model repository url.</i>
Request	
Authentication	<i>OAuth 2</i>
Content Type	<i>application/json</i>
Request header	<pre>{ "Accept": "application/json", "IMC_TOKEN": "" }</pre>
Query Parameters	<i>modelId,</i> <i>modelRepositoryUrl,</i> <i>inputValues</i> <i>Values possible for input parameters</i> - <i>BURDEN_OF_ILLNESS (id = 64) : [NO, MODERATE, SEVERE]</i> - <i>HYPOTENSION (id = 66) : [NO, YES]</i>

	<ul style="list-style-type: none"> - <i>PULMONARY_DISEASE</i> (id = 74) : [NO, YES] - <i>SOLID_TUMOR_OR_NO_FUNGAL_INFECTION</i> (id = 72) : [YES, NO] - <i>OUTPATIENT</i> (id = 76) : [YES, NO] - <i>DEHYDRATION</i> (id = 68) : [NO, YES] - <i>AGE</i> (id = 70) : years (integer)
JSON Object	<pre>{ "modelId": 0, "modelRepositoryUrl": "string", "inputValues": [{ "parameterId": 0, "modelInputValue": "string", "modelInputDisplayTypeId": 0 }] }</pre>
Example	<pre>curl -X POST --header "Content-Type: application/json" --header "Accept: application/json" --header "IMC_TOKEN: 0bcb3d6e72a820a301e9" -d "{ \"modelId\": \"63\", \"modelRepositoryUrl\": \"http://laphroaig.ehv.campus.philips.com:8080/ModelsRepositoryApp/ModelsRepositoryWebService?wsdl\", \"inputValues\": [{ \"modelInputDisplayTypeId\": \"0\", \"modelInputValue\": \"59.99\", \"parameterId\": \"70\" }, { \"modelInputDisplayTypeId\": \"0\", \"modelInputValue\": \"NO\", \"parameterId\": \"64\" }, { \"modelInputDisplayTypeId\": \"0\", \"modelInputValue\": \"NO\", \"parameterId\": \"68\" }, { \"modelInputDisplayTypeId\": \"0\", \"modelInputValue\": \"NO\", \"parameterId\": \"66\" }, { \"modelInputDisplayTypeId\": \"0\", \"modelInputValue\": \"YES\", \"parameterId\": \"76\" }, { \"modelInputDisplayTypeId\": \"0\", \"modelInputValue\": \"YES\", \"parameterId\": \"74\" }, { \"modelInputDisplayTypeId\": \"0\", \"modelInputValue\": \"YES\", \"parameterId\": \"72\" }] } "</pre> <p>http://laphroaig.ehv.campus.philips.com:8080/ModelRunnerService/rest/executeModel</p>
Response	
Status Codes	<ul style="list-style-type: none"> • 200 OK • 401 Unauthorized • 403 Forbidden • 500 Internal Server Error
Content Type	application/json
Response header	<pre>{ "date": "Fri, 22 Apr 2016 07:52:44 GMT", "server": "WildFly/10", "connection": "keep-alive", "x-powered-by": "Undertow/1",</pre>

	<pre>"transfer-encoding": "chunked", "content-type": "application/json" }</pre>
JSON Object	<pre>[{ "parameterId": "int", "modelInputValue": "string", "modelInputDisplayTypeId": "int" }]</pre>
Example	<pre>[{ "parameterId": 77, "modelInputValue": "FEBRILE_NEUTROPENIA_NO_HIGH_RISH", "modelInputDisplayTypeId": 352 }]</pre>
	OR
	<pre>[{ "parameterId": 77, "modelInputValue": "FEBRILE_NEUTROPENIA_HIGH_RISH", "modelInputDisplayTypeId": 352 }]</pre>
	OR
	<pre>[{ "code": 401, "message": "Unauthorized", "description": "The token is no longer valid" }]</pre>
	OR

	<pre>[{ "code": 403, "message": "Forbidden", "description": "The token does not exist" }]</pre>
	OR
	<pre>[{ "code": 500, "message": "Internal server error", "description": "Unable to validate the token, problem with the OAuth server" }]</pre>

11.5. Installation and configuration guidelines

11.5.1. Setting up Wildfly 10.0.0

To start with, you need to download the Wildfly-10.0.0.Final from the following link:

<http://wildfly.org/downloads/>

You then need to carry out the following steps:

Copy zip file

Copy zip file on '/home/laphroaig'

Unzip file

Unzip file from '/home/laphroaig' to '/usr/share'

```
sudo unzip wildfly-10.0.0.Final.zip -d /usr/share/
```

Remove old symbolic link

Remove the previous symbolic link (not mandatory)

```
sudo unlink jboss-as
```

```
sudo unlink wildfly
```

Remove jboss-as service (Fedora)

```
sudo chkconfig jboss-as off
```

```
sudo chkconfig jboss-as --del
```

Add symbolic link

```
sudo ln -s /usr/share/wildfly-10.0.0.Final /usr/share/wildfly
```

Checking running services

Check the running jboss-as/wildfly services to be sure that all have been stopped.

```
service --status-all
```

Stop the running services

Stop the running jboss-as/wildfly services by using the following

```
sudo /sbin/service jboss-as start
```

```
sudo /sbin/service jboss-as restart
```

```
sudo /sbin/service jboss-as stop
```

The jboss-as server can also be stopped using CLI command line.

```
./jboss-cli.sh --connect command=:shutdown
```

OR

```
./jboss-cli.sh -c
```

```
connect
```

```
:shutdown
```

Jre installation

Check java consistency with the needed version of Wildfly

The following command will give the folder where java has been installed

which java

OR

```
$(dirname $(dirname $(readlink -f $(which javac)))) => /usr/lib/jvm/java-1.7.1-ibm-1.7.1.3.30.x86_64
```

OR

```
java -showversion -verbose 2>&1 | head -1
```

OR

```
strace -e open java -showversion 2>&1 | grep -m1 /jre/
```

This hyperlink provides details about the installation of java alternatives.

<http://tecadmin.net/install-java-8-on-centos-rhel-and-fedora/#>

After the download of the jdk version you want to install follow the steps below.

```
sudo tar xzf jdk-8u74-linux-x64.tar.gz -c /usr/lib/jvm/
```

OR

```
(In the folder /usr/lib/jvm/) => sudo tar xzf /home/laphroaig/jdk-8u74-linux-x64.tar.gz
```

```
cd /usr/lib/jvm/jdk1.8.0_74/
```

```
sudo alternatives --install /usr/bin/java java /usr/lib/jvm/jdk1.8.0_74/java 2
```

```
sudo alternatives --config java
```

```
sudo alternatives --install /usr/bin/jar jar /usr/lib/jvm/jdk1.8.0_74/bin/jar 2
```

```
sudo alternatives --install /usr/bin/javac javac /usr/lib/jvm/jdk1.8.0_74/bin/javac 2
```

```
sudo alternatives --set jar /usr/lib/jvm/jdk1.8.0_74/bin/jar
```

```
sudo alternatives --set javac /usr/lib/jvm/jdk1.8.0_74/bin/javac
```

Add the java path to ‘.bash_profile’ file

```
PATH=$PATH:$HOME/bin:/usr/lib/jvm/jdk1.8.0_74/bin:/usr/lib/jvm/jdk1.8.0_74/jre/bin
```

Reload .bash_profile from the command line

```
source ~/.bash_profile
```

OR

```
. ~/.bash_profile
```

User wildfly

Create a user account on the system for wildfly

```
sudo adduser wildfly
```

Remove password for user wildfly

```
sudo passwd -d wildfly
```

Ownership of folder and symbolic link

Change ownership of directory and symbolic link to wildfly

```
cd /usr/share/
```

```
/usr/bin/sudo chown -R wildfly:wildfly wildfly
```

```
/usr/bin/sudo chown -R wildfly:wildfly wildfly-10.0.0.Final
```

Folder init.d

Copy init.d folder

```
sudo cp -r /usr/share/wildfly/docs/contrib/scripts/init.d /usr/share/wildfly/bin/init.d
```

Use of scripts

Edit and make use of scripts

```
cd /usr/share/wildfly/bin/init.d
```

Add wildfly 'user/group' to folder

```
/usr/bin/sudo chown -R wildfly:wildfly /usr/share/wildfly/bin/init.d
```

Add write mode to file

```
sudo chmod u+w wildfly.conf
```

Edit the file that contains the settings/options for the setup script and uncomment 'JBOSS_USER=wildfly'

```
sudo vi wildfly.conf
```

Link the file in Etc

```
cd /etc
```

```
sudo mkdir wildfly
```

```
cd /etc/wildfly
```

```
sudo ln -s /usr/share/wildfly/bin/init.d/wildfly.conf wildfly.conf
```

```
cd /etc/init.d
```

```
sudo ln -s /usr/share/wildfly/bin/init.d/wildfly-init-redhat.sh wildfly
```

Set proper permissions and ownership of 'init.d' script:

```
sudo chown root:root /etc/init.d/wildfly
sudo chmod +X /etc/init.d/wildfly
```

Set some rights

Set wildfly rights on some files

Create directory for logs

```
sudo mkdir -p /var/log/wildfly
```

Change the owner of WildFly directories:

```
sudo chown -R wildfly:wildfly /usr/share/wildfly-10.0.0.Final/
sudo chown -R wildfly:wildfly /usr/share/wildfly
sudo chown -R wildfly:wildfly /var/log/wildfly
```

add the line "export JBOSS_USER"

```
sudo vi wildfly.conf
```

adapt and uncomment JBOSS_HOME (="/usr/share/wildfly") in file wildfly.conf, and other paths in order to match the installation

Add the following in file 'wildfly-init-redhat.sh'

```
sudo vi /usr/share/wildfly/bin/init.d/wildfly-init-redhat.sh
if [ -z "$JBOSS_USER" ]; then
    JBOSS_USER=wildfly
fi
```

```
cd /usr/lib/jvm
```

```
sudo ln -s /usr/lib/jvm/jdk1.8.0_74/jre jre
```

In file /etc/java/java.conf, uncomment the following line

```
JAVA_HOME=$JVM_ROOT/jre
sudo vi /etc/java/java.conf
```

Bind address

Bind address, changing the address to 0.0.0.0


```
sudo vi /usr/share/wildfly-10.0.0.Final/standalone/configuration/standalone.xml
```

```
<interfaces>
    <interface name="management">
        <inet-address value="{jboss.bind.address.management:127.0.0.1}"/>
    </interface>
    <interface name="public">
        <inet-address value="{jboss.bind.address:127.0.0.1}"/>
    </interface>
</interfaces>
```

Start Wildfly

```
sudo service wildfly start
```

Add application server users

```
su wildfly -c ./add-user.sh
```

User/password : admin/admin added to RealManagement

To represent the user add the following to the server-identities definition <secret value="YWRtaW4=" />

User/password : cds4dp/cds\$dp added to RealManagement

To represent the user add the following to the server-identities definition <secret value="Y2RzJGRw" />

Install MySql Jdbc driver

There are 2 ways to install MySql Jdbc driver either by web console or by CLI.

Copy 'module.xml' and 'mysql-connector.jar' in the folder '/usr/share/wildfly-10.0.0.Final/modules/system/layers/base/com/mysql/main'

```
sudo cp -r /home/laphroaig/mysql /usr/share/wildfly-10.0.0.Final/modules/system/layers/base/com/mysql/
```

```
./jboss-cli.sh -c
```

```
/subsystem=datasources/jdbc-driver=mysql:add(driver-name=mysql,driver-module-name=com.mysql.driver,driver-class-name=com.mysql.jdbc.Driver)
```

For ardborg server it is enough to use CLI but for laphroaig it dit not work directly so there is a need to deploy the mysql jar file through the web console.

Thus for arberg, during the creation of the datasource, one will have to select manually the deployed MySql driver.

Create data sources in Wildfly

The application needs the following datasources.

- ClinicalMetadataDB, url = jdbc:mysql://localhost:3306/clinicalmetadatadb.
- ProcedureDB, url = jdbc:mysql://localhost:3306/proceduredb.
- PatientsDB, url = jdbc:mysql://localhost:3306/patientsdb.
- ModelsRepositoryDB, url = jdbc:mysql://localhost:3306/modelsrepositorydb.

On can use the Wildfly web console to create the datasource or do it using CLI.

Please read the Jboss documentation for more details about MySql driver and datasources installation.

<https://docs.jboss.org/author/display/WFLY8/DataSource+configuration>

Delpoy war files

```
./jboss-cli.sh -c
```

User : admin

Password : admin

```
deploy /home/laphroaig/wars/20160216/StratumQueryEngine.war
```

```
deploy /home/laphroaig/wars/20160216/CdsGroovyEngine.war
```

```
deploy /home/laphroaig/wars/20160216/ProcedureService.war
```

```
deploy /home/laphroaig/wars/20160216/ClinicalDataService.war
```

```
deploy /home/laphroaig/wars/20160216/DroolsCDSEngine.war
```

```
deploy /home/laphroaig/wars/20160216/PatientManagementService.war
```

```
deploy /home/laphroaig/wars/20160216/ModelRunnerService.war
```

```
deploy /home/laphroaig/wars/20160216/ModelsRepositoryApp.war
```

```
deploy /home/laphroaig/wars/20160216/JessCDSEngine.war
```

11.6. Database

List of databases

These are the databases used for the project.

- modelsrepositorydb

Initialize data

Use mysql client to connect a database to launch a script.

```
mysql -u user -p data_base_name_here < db.sql
```

12. Conclusions

The iManageCancer platform has been designed on clinical evidence, in close collaboration of clinical experts, IT specialists and patients. The main apps developed to this purpose are the iMC iPHR portal, the iManageMyHealth and iSupportMyPatients, the serious games for kids and adults and the MyHealthAvatar for breast and prostate cancer patients.

With this deliverable milestone 3 “Extended integrated prototype of iManageCancer platform” has been accomplished. The software is released for clinical validation and all related deliverables have been submitted. Pilots have already started and will assess the added value on health and quality of life of the decision support and analysis tools and the platform as a whole.