



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**

***D7.2
Serious game for adults***

Contractual Due Date: 30th Apr 2017
Actual Submission Date: 24th May 2017

Lead partner for deliverable: University of Bedfordshire

Dissemination Level: Public

Revision: v1.0

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.:	D7.2
Deliverable Name:	Serious Game for Adults
Nature (R, DEM) ¹	DEM
Dissemination Level (PU, CO) ²	PU
Version:	1.0
Actual Submission Date:	
Editor: Institution: E-Mail:	Stephen Wilson University of Bedfordshire Stephen.wilson@beds.ac.uk
Contributors (Institution)	University of Bedfordshire
Reviewers (Institution)	Chiara Renzi (IEO), Richard Stephens (Philips-UK)

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
24.05.17	1.0	First version after internal review

Table of Contents

1. Executive summary	5
2. Introduction	5
2.1 About the project.....	5
2.2 Purpose of the document.....	6
2.3 Structure of this document	6
3. Summary of use scenarios and requirements from D2.2 and D2.3	7
3.1. Serious Game for Adults overview	7
3.1.1. Use case.....	8
3.1.2. Requirements.....	10
4. System concepts	11
4.1.1. iManageCancer	11
5. Technical implementation	11
5.1. Development tools & environment	13
5.2. Interfaces	14
5.3. Databases and structures	14
5.4. Data Protection.....	20
5.5. Serious Game for Adults features and interface	21
5.5.1. Features	21
5.5.2. User interface summary	22
6. Conclusions	29
7. References	30

1. Executive summary

Chronic cancer treatment places new demands on patients and families to manage their own care. The iManageCancer project will support this challenge and provide a cancer disease self-management platform designed according to the specific needs of patient groups and focusing on the wellbeing of the cancer patient with special emphasis on psycho-emotional evaluation and self-motivated goals.

The iManageCancer platform will be designed on clinical evidence and in close collaboration of clinical experts, IT specialists and patients and will be assessed in clinical pilots with adult and paediatric cancer patients.

This document presents the initial implementation the serious game for adults (SGA), pre-evaluation. Within the iManageCancer project there exist two serious game applications, this deliverable focuses on the game targeting adult cancer patients, and tackles the unique issues and challenges of delivering an effective educational tool that is also fun and engaging to use. The deliverable covers the use-cases and subsequent functional requirements generated early on in the design process and how these functional requirements have been realised over the course of the project into a fully 3D lifestyle simulation game.

The Serious Game for Adults aims to help users engage with the wider iManageCancer platform while placing them in a situation requiring critical thinking and strategy. The game also affords users the opportunity to create and deploy a virtual avatar that they can share in a social environment, while also gently exposing the users to educational content and examples of situations that promote healthier lifestyle choices.

The serious game for adults has been designed and implemented to address the core objectives of Work Package 7 as outlined in the Description of Action:

- Encourage healthy habits
- Face coping with the disease in a different perspective
- Promote disease management
- Support patients in reducing stress, anxiety and related negative impacts of the disease on their lives and social relations
- Enhance patient's knowledge through education.

2. Introduction

2.1 *About the project*

Chronic cancer treatment places new demands on patients and families to manage their own care. The iManageCancer project will support this challenge and provide a cancer disease self-management platform designed according to the specific needs of patient groups and focusing on the wellbeing of the cancer patient with special emphasis on psycho-emotional evaluation and self-motivated goals.

The iManageCancer platform will be designed on clinical evidence and in close collaboration with clinical experts, IT specialists and patients, and will be assessed in clinical pilots for adult and paediatric cancer patients.

The main objectives of iManageCancer listed in the Description of Action are summarised as:

1. **Empower patients and their relatives through an information and communication technology based self-management service platform.**
2. Allow patients **to keep track of their health and disease status.**
3. Provide the patients with **personalized, context-sensitive, data driven information services.**
4. Help adult and young cancer patients through **serious games** to manage the impact of the disease on their psychological status.
5. Provide patients with **decision support and guidance through a knowledge base of formal care flow plans.**
6. Support patients and their doctors in **managing medications.**
7. Provide clinicians and patients an **interactive psycho-emotional health assessment instrument.**
8. Increase patients' safety by developing and incorporating **predictive models in the system for the early detection of severe adverse events during chemotherapy.**
9. **Support patients in following a healthy and active lifestyle.**
10. Provide the patient with an easy-to-use **interactive cockpit for disease self-management** on mobile platforms.
11. Incorporate an instrument in the platform for **data driven analysis services** on anonymised clinical information to be used for public health research.
12. Conduct and assess **three pilots**, two for adult cancer patients and one for children.
13. Design an innovative **ecosystem for the empowerment of cancer patients.**

2.2 Purpose of the document

This document presents the initial version of the serious game for adults. The document outlines the tools and technologies leveraged in the creation of the serious game as well as the technical implementation of the application and supporting server. The document also provides a context for the application's deployment within the iManageCancer platform.

We like to mention that the original Description of Action asks for the integration of existing serious games for cancer patients on the market into the iManageCancer platform. However, after a review of the state of the art we created own ideas for a serious game for cancer patients and decided to develop such a sophisticated tool for this project that provides additional value to patients. In consequence, we slightly changed the original name of this deliverable to match it with the content.

2.3 Structure of this document

This document comprises the following three main sections:

- Summary of use scenarios
 - Use case diagrams

- Functional requirements.
- System concepts
- Technical implementation
 - Development tools and environments.
 - Game structure and user interfaces.
 - Databases and data structures
 - Data protection
 - Interface and features
- Conclusion

3. Summary of use scenarios and requirements from D2.2 and D2.3

3.1. *Serious Game for Adults overview*

Within the iManageCancer project there are two serious games, the focus of this deliverable is the game intended for use by adult cancer patients. The serious game for adults augments the capacity of the iManageCancer platform in promoting self-management through a stylised simulation game.

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.

In deliverable D2.2 'Scenarios and use cases' the targets of the serious game are listed as:

- (Concrete) Learn healthier life-styles (balanced diet, appropriate physical exercise) and increase problem solving skills (adjustment-oriented self-management approach)
- (Meta) To create a game that educates patients (and individuals) to healthy behaviors while having fun
- (Meta) Improve self-efficacy in self-management skills through mastery and vicarious experience of the Avatar in the serious game.

The game is careful not to make medical recommendations or specifically advise the user about appropriate lifestyle changes, instead opting to provide examples of good lifestyle management within a controlled context using a fictional character. The game also provides an educational element through a mini-game which leverages a large set of health and lifestyle trivia questions

in a quiz style challenge. The mini-game is repeatable and can be used to dissuade the fictional cancer patient from un-helpful lifestyle choices.

Registered iManageCancer users can also access a feature that utilises their physical activity data, this data is collected through other iManageCancer applications and allows the user to convert it in order to spend it within the game as a form of currency. Currently the feature allows the user to buy a random prize chance box, this box is exchanged for a spin on a prize wheel that yields a small beneficial prize in game. The feature is intended to encourage a user to utilise more of the iManageCancer platform and potentially motivate the user to engage in more physical activity.

A registered user can also save their game data to a game back-end service once they have logged in. This service allows for registered users to provide feedback and audit data which is forwarded through the back-end service to their PHR (Personal Health Record) within the iManageCancer platform. Further to basic game profile uploading and migration a user can also search for other people's profiles and send them friend requests. This social mechanic is an asynchronous form of social play, a friend's profile can be viewed in game and there is a mechanic that allows the friend to occupy the user's town providing a passive effect. The social profiles contain no medical data or information beyond their avatars current appearance and contact e-mail.

3.1.1. Use case

Below Figure 1 displays the relationship between users and the iManageCancer platform, this example has been designed to allow for an un-registered user to still engage with the game in a limited fashion while providing social and activity related features to registered users.

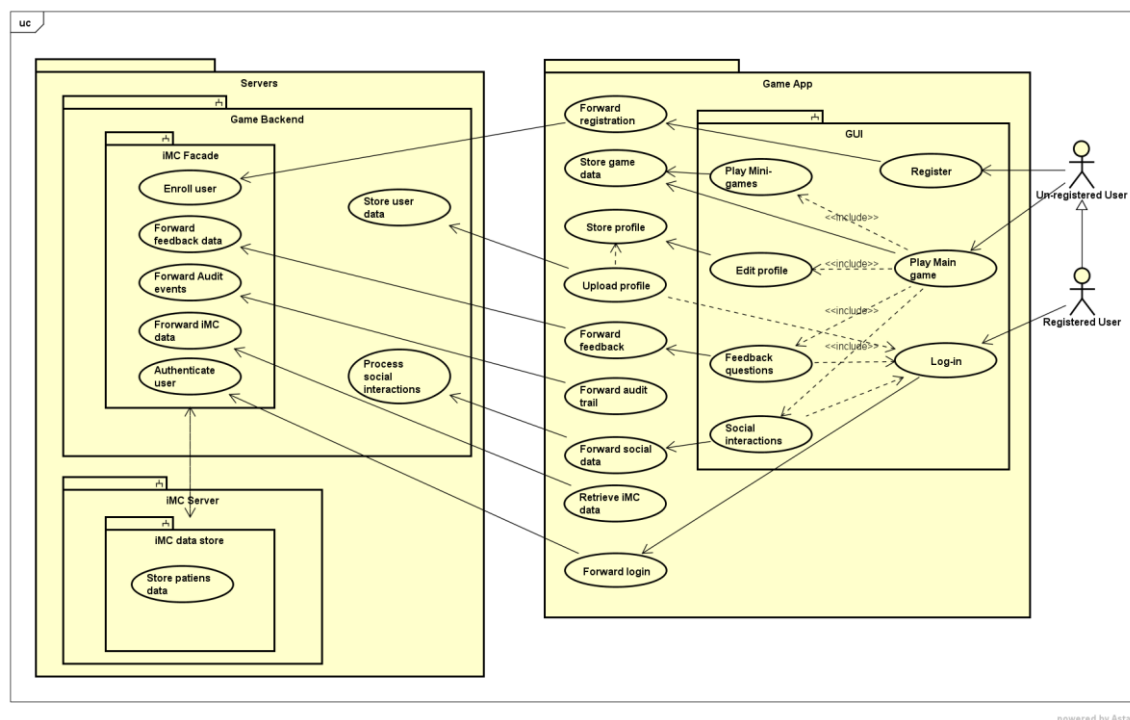


Figure 1: Serious game use case diagram

Use Case ID	UC.SGA.1	Priority	REQUIRED
Use Case name	Serious games application		
Scenarios	GI2		
Actors	New user, registered user.		
Brief description	The user’s interface with the iManageCancer serious game.		
Trigger	<ul style="list-style-type: none">(Actor) On actor’s demand.		
Pre-conditions	<ul style="list-style-type: none">Actor has the app installed on a mobile device.		
Post-conditions	-		
Successful end condition	Actor utilises the app for recreation and simultaneously generates useful data and learns positive behaviours or concepts.		
Fail end condition	Actor does not enjoy their experience and disengages with the application.		
Relationships	Extends		Includes -
Basic flow	Step	Action	
	1	Actor submits registration request.	
	2	Actor submits log-in request.	
	3	Actor interacts with the main game.	
		Actor interacts with sub modules of main game.	
	5	Main game attempts to upload data to game server.	
	6	Social module communicates with game back-end, allows friend search and acceptance, requires connection.	
	7	Profile module allows Actor to edit personal details, and then attempts upload to game server if connected and registered.	
	8	Mini-Games module allows Actor to select from list of included games.	
	9	Game store, allows Actor to add virtual items to their profile.	
Usage frequency	Medium		
User interfaces	Game GUI		
Notes and issues	-		

3.1.2. Requirements

In consequence, Deliverable D2.3 ‘Technical system requirements’ lists 10 important functional system requirements which were derived from this use case. They are briefly listed in the following table:

ID	Name	Description
REQ.SGA.1	Social support mechanism	The game will allow a user’s friends to affect in game mechanics for positive effects.
REQ.SGA.2	Game back end data storage	Server storage to store data generated by users and uploaded to the back-end server.
REQ.SGA.3	Access to iManageCancer data storage	The serious game will need to connect to the iMC data store through the game’s back-end server to acquire user data and upload new data.
REQ.SGA.4	Data analysis services support	The serious game server should support data analysis services. Data should be uploaded to PHR for further processing.
REQ.SGA.5	Data Exchange API	The serious game server should implement an API to allow client apps to interact with data storage such as the iMC data store.
REQ.SGA.6	Game GUI	The client app requires a graphical user interface. Each modular component will require a unique interface. A high level of design and testing will be required as the GUI comprises a large portion of the apps appeal and functionality.
REQ.SGA.7	Provide user management	The app must perform user management through an API provided by the game back-end.
REQ.SGA.8	Integration of mini-games	The overall app is supplemented by a number of more focused mini-games.
REQ.SGA.9	Integration of a social module	The social module requires an internet connection. The user will have access to a social interface allowing searching and adding of friends.
REQ.SGA.10	Integration of a store module	The store module is integrated into the main game. The user will have access to an interface that facilitates item purchases for their avatar.

4. System concepts

4.1.1. iManageCancer

The high-level architecture of the iManageCancer platform is shown in Figure 2.

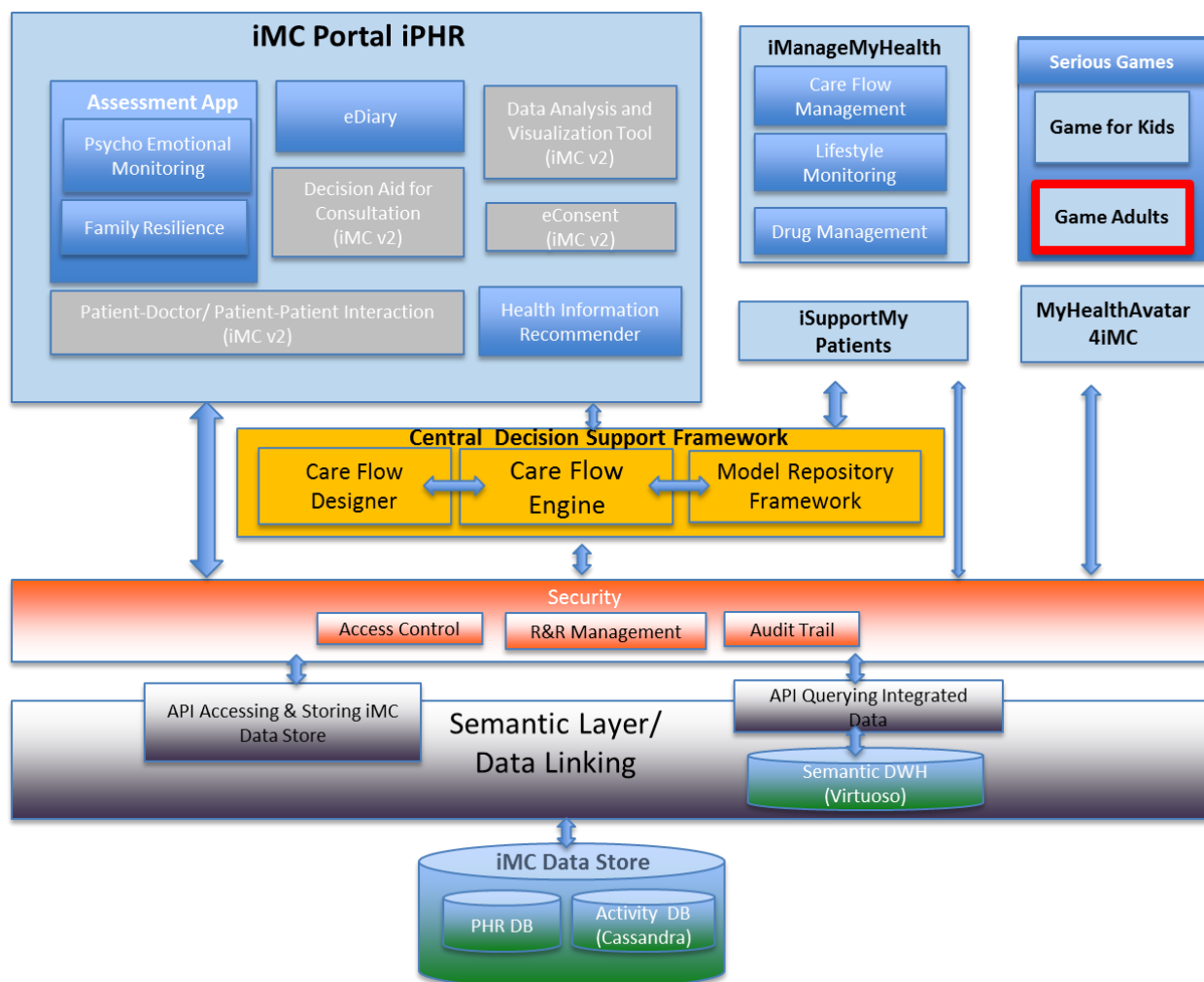
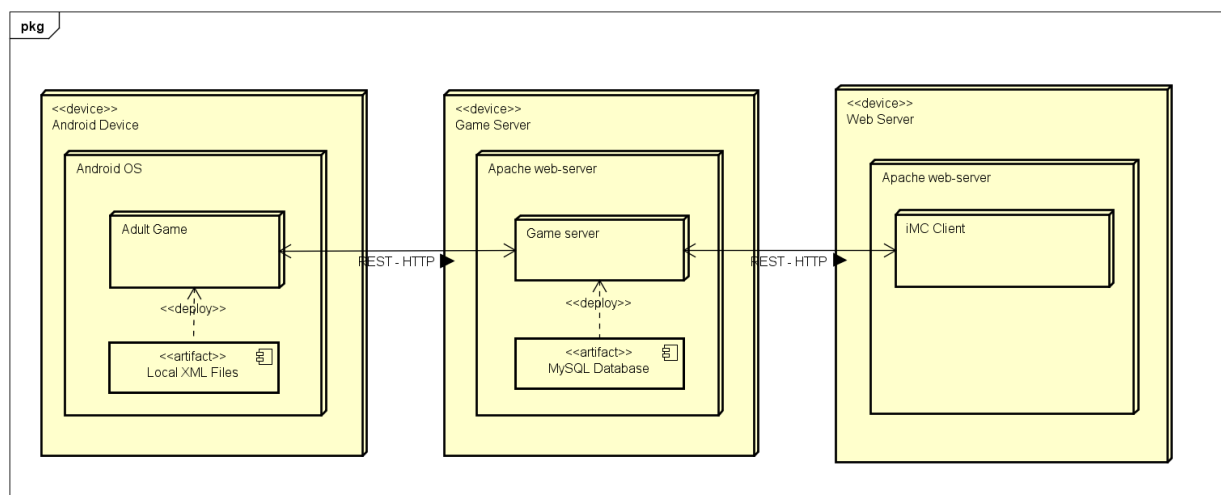


Figure 2: High level iManageCancer platform architecture

5. Technical implementation

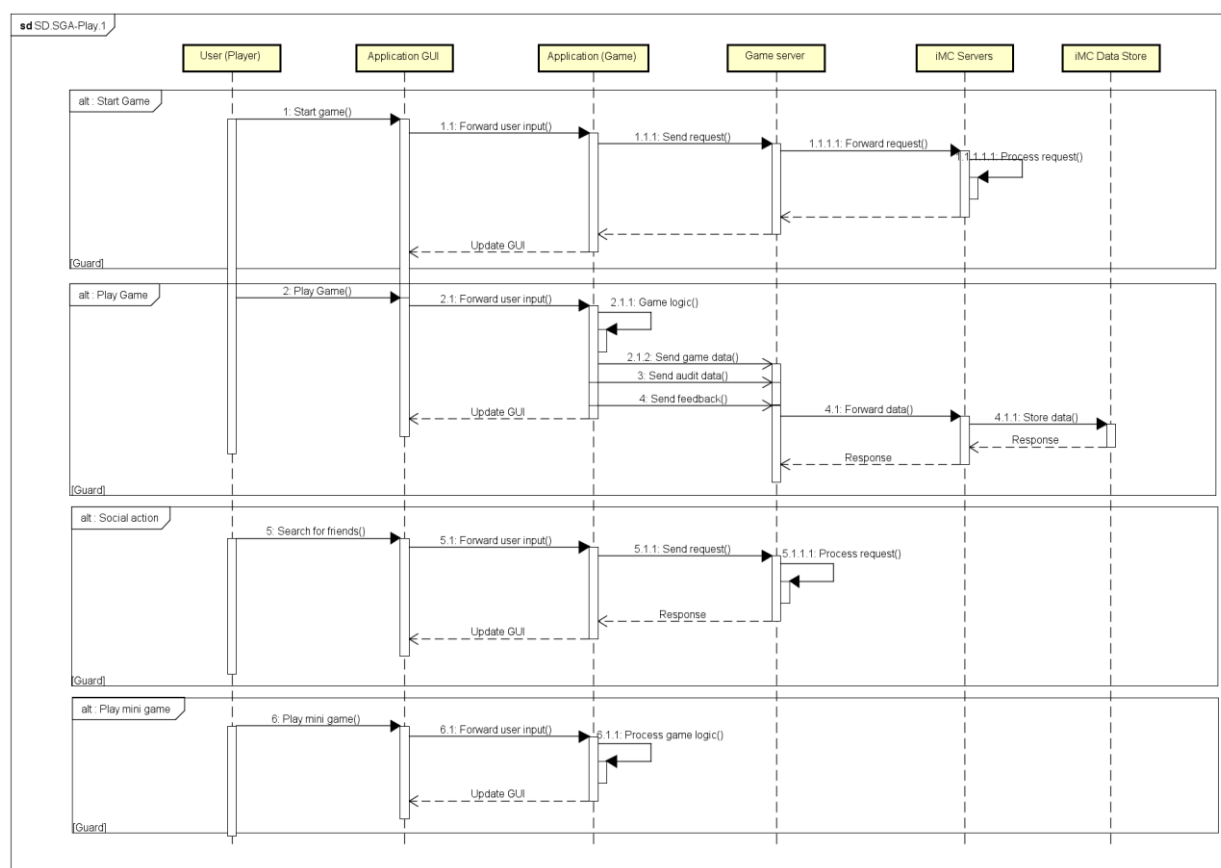
The serious game for adults is intended to be played on mobile devices such as smart phones or tablets running the Android operating system. Figure 3 displays the overall configuration, highlighting the games back end server as an intermediary between the game client and the iManageCancer platform.



powered by Astah

Figure 3: Serious game deployment view diagram

Below in Figure 4, there can be seen a simplified overview of user interaction and its scope within the iManageCancer framework.



powered by Astah

Figure 4: 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 5: Game server architecture, for information about the structure of the game back end server.

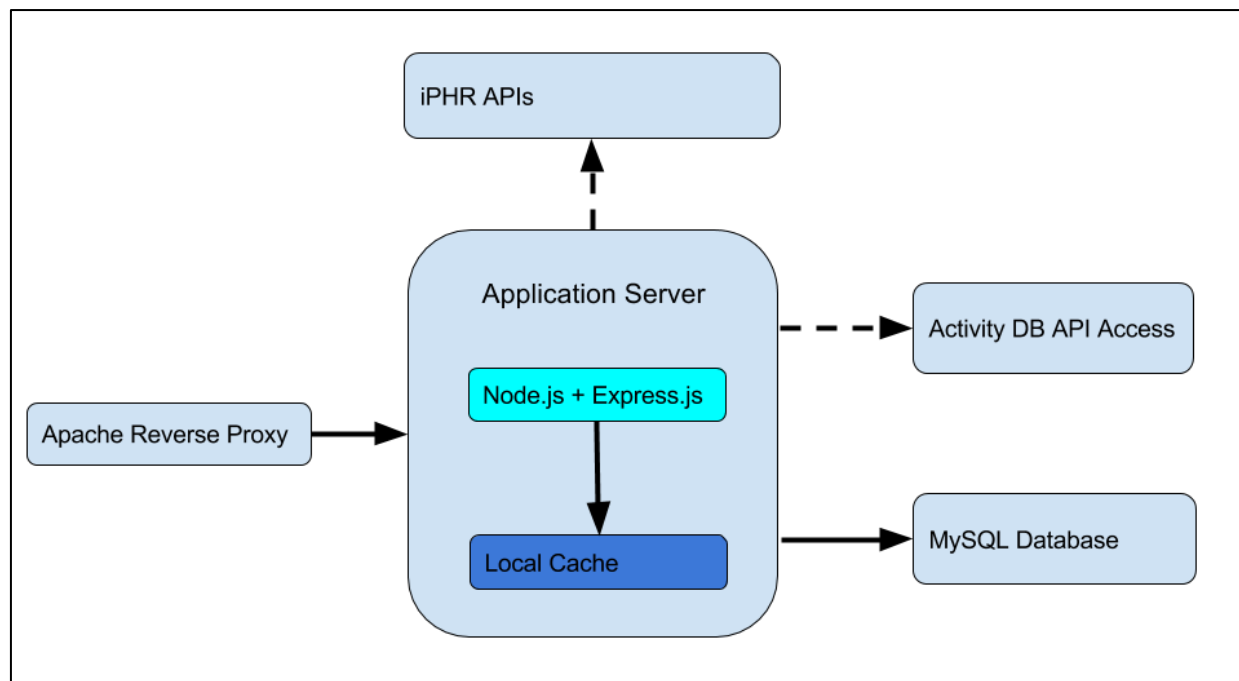


Figure 5: Game server architecture

5.1. Development tools & environment

The target development environment for the core game functionality was Unity [1]. This cross-platform game engine was chosen due to its capability to publish to multiple platforms quickly and reliably. The original version of this tool when development began was 5.2.2, later development had to switch to version 5.3.2p2 in order to cope with an established UI bug causing performance issues on our target test devices. The unity Android publishing systems allows for the targeting of devices running Android API level 9 and up. According to Android developer resources [2] this means that Unity can theoretically be deployed on effectively all currently active Android devices.



Code for the serious game was developed using the IDE Visual Studio Community 2015 with Unity plugins, this allows Unity to utilise Visual Studio's debugging and IntelliSense functionality.



The target operating system for the serious game is Android, at time of writing the Android OS has reached version 7.1 (Nougat) but in early development meetings a target of version 5.0 (Lollipop) was settled on for deployment and testing. The Android operating system is owned and operated by Google and is built upon a Linux kernel and an open source license is still supported by its developers, although in practice many hardware suppliers ship their devices with a combination of the open source Android OS combined with proprietary software.



The server side development was carried out using the following tools:

- Node.js v6.10.2 LTS (MIT) - the web service runtime for the game backend
- Express.js v4.14.0 (MIT) - the web framework which provide the REST API for game client.
- Apache HTTP Server v2.4.7 (Apache v2) - handles request to domain name, and reverse proxy to Node.js for processing, also handles HTTPS requests.
- MySQL v5.7.17 (GPL v2) - data store for the game data, game user profile.

5.2. Interfaces

The interfaces implemented on the game back end server:

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

5.3. 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 in order 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. What follows is a table of existing data structures within the game with abbreviated examples of their use:

File name	Description	Example
Lang.xml	(Read only) Contains all in game language strings in language categories.	<pre> <languages> <English> <string name="language_thumbnail">lang_english</string> <string name="language_hint">English</string> <string name="app_name">Serious Game for Adults</string> </English> </languages> </pre>
Trivia.xml	(Read only) Array of trivia question items.	<pre> <Trivia> <Trivia_collection> <question> <questionid>321199</questionid> <zone_map>General Knowledge: Health And Wellness</zone_map> <diffrank>3</diffrank> <question>Glucose, fructose and galactose are called what?</question> <wrong_answer_1>Disaccharides</wrong_answer_1> <wrong_answer_2>Complex Carbohydrates</wrong_answer_2> <wrong_answer_3>Complex Sugars</wrong_answer_3> <CORRECT>Monosaccharides</CORRECT> </question> </Trivia_collection> </Trivia> </pre>
StoreInventory.xml	(Read only) Array of clothing and character assets.	<pre> <ItemCollection> <Items> <Item> <category>hair</category> <itemName>Hair Default</itemName> <gender>none</gender> <description>item_description_default</description> <mesh>hair_00</mesh> <texture>none</texture> <value>0</value> <quality>1</quality> <preview_texture>hair_preview</preview_texture> <previewIndex>0</previewIndex> <modTarget>none</modTarget> <modVal>0.0</modVal> <activityCategory>none</activityCategory> <activityChanceBonus>0.0</activityChanceBonus> </Item> </Items> </ItemCollection> </pre>

Levels.xml	(Read only) Array of level starting states and towns.	<pre> <LevelCollection> <Levels> <Level> <level_id>level_0001</level_id> <level_name_id_string>level_1_name</level_name_id_string> <level_description_string>level_1_description</level_description_string> <town>town_config_001</town> <condition>condition_001</condition> <tier_1_reward>Hat2</tier_1_reward> <level_number>1</level_number> <tire_1_target>5</tire_1_target>; <tire_2_target>10</tire_2_target>; <tire_3_target>15</tire_3_target>; <starting_money>300</starting_money> </Level> </Levels> </LevelCollection> </pre>
Challenges.xml	(Read only) Array of challenges and completion conditions.	<pre> <ChallengeCollection> <Challenges> <Challenge> <challenge_id>challenge_doctor_bad_news</challenge_id> <challenge_name_sting_id>challenge_title</challenge_name_sting_id> <challenge_description_string_id>challenge</challenge_description_string_id> <challenge_type>category</challenge_type> <challenge_owner>doctor</challenge_owner> <target_activity_id>none</target_activity_id> <target_activity_quantity>0</target_activity_quantity> <target_activity_time_constraint>0</target_activity_time_constraint> <target_activity_category>social</target_activity_category> <target_category_quantity>5</target_category_quantity> <target_category_time_constraint>0</target_category_time_constraint> <target_building_id>none</target_building_id> <target_building_level>0</target_building_level> <target_attribute>none</target_attribute> <target_attribute_value>0</target_attribute_value> <target_attribute_deviation>0</target_attribute_deviation> <target_attribute_time_constraint>0</target_attribute_time_constraint> </Challenge> </Challenges> </ChallengeCollection> </pre>
Conditions.xml	(Read only) Array of conditions and their attached challenges.	<pre> <ConditionCollection> <Conditions> <Condition> <condition_id>condition_001</condition_id> <condition_name_string_id>condition_001_title</condition_name_string_id> <condition_description_string_id>condition</condition_description_string_id> <weight_optimal_value>10</weight_optimal_value> <social_optimal_value>20</social_optimal_value> <mental_optimal_value>6</mental_optimal_value> <physical_optimal_value>15</physical_optimal_value> <weight_degeneration_rate>1</weight_degeneration_rate> <social_degeneration_rate>2</social_degeneration_rate> <mental_degeneration_rate>3</mental_degeneration_rate> <physical_degeneration_rate>1</physical_degeneration_rate> <starting_action_points>24</starting_action_points> <daily_wages>50</daily_wages> <starting_weight_value>5</starting_weight_value> <starting_social_value>5</starting_social_value> <starting_mental_value>5</starting_mental_value> <starting_physical_value>5</starting_physical_value> <patient_avatar/> <Challenges/> </Condition> </Conditions> </ConditionCollection> </pre>

TownConfigurations.xml	(Read only) Array of complete towns composed of buildings and their current states.	<pre> <TownCollection> <TownConfigurations> <Configuration> <configuration_id>town_config_001</configuration_id> <landscape_id>town_001</landscape_id> <Buildings> <Building> <building_id>factory_001</building_id> <building_level>2</building_level> <building_x_pos>30</building_x_pos> <building_y_pos>0</building_y_pos> <building_z_pos>-30</building_z_pos> <building_rotation>180</building_rotation> <active_tasks /> </Building> </Buildings> </Configuration> </TownConfigurations> </TownCollection> </pre>
Buildings.xml	(Read only) Array of buildings and their corresponding attributes.	<pre> <BuildingCollection> <Buildings> <Building> <building_base_id>empty_plot_001</building_base_id> <building_name_string_id>building_empty</building_name_string_id> <building_description_string_id>building</building_description_string_id> <building_level1_mesh_id>empty</building_level1_mesh_id> <building_level2_mesh_id>empty_plot_001</building_level2_mesh_id> <building_level3_mesh_id>empty_plot_001</building_level3_mesh_id> <building_preview_material>buildings</building_preview_material> <building_preview_index>0</building_preview_index> <building_level1_cost>0</building_level1_cost> <building_level2_cost>0</building_level2_cost> <building_level3_cost>0</building_level3_cost> <building_time>0</building_time> <Activity_categories> <Category /> </Activity_categories> </Building> </Buildings> </BuildingCollection> </pre>
Activities.xml	(Read only) Array of available activities with their corresponding attributes and approved spawn locations.	<pre> <ActivityCollection> <Activities> <Activity> <activity_id>Play a board game</activity_id> <activity_name_string_id>activity_001_title</activity_name_string_id> <activity_description_string_id>description</activity_description_string_id> <weight_modifier>0</weight_modifier> <social_modifier>3</social_modifier> <mental_modifier>1</mental_modifier> <physical_modifier>-1</physical_modifier> <ap_cost>3</ap_cost> <money_cost>0</money_cost> <Categories> <Category /> </Categories> <ApprovedBuildings> <Buildings/> </ApprovedBuildings> </Activity> </Activities> </ActivityCollection> </pre>

CampaignState.xml	Contains the current state of play as well as an array of challenge related events, and array of active challenges and an array of friends present within the town.	<pre> <CampaignState> <current_level_id>level_0002</current_level_id> <play_days>0</play_days> <time_passed>0</time_passed> <mental_value>5</mental_value> <social_value>5</social_value> <physical_value>5</physical_value> <weight_value>5</weight_value> <money_value>140</money_value> <totalAction_points>24</totalAction_points> <avaialableAction_points>24</avaialableAction_points> <wellbeing_value>0</wellbeing_value> <active_friends>0</active_friends> <level_first_run>false</level_first_run> <currenttownConfiguration> </currenttownConfiguration> <Challenges/> <Challenge_logs /> <Friends_in_town /> </CampaignState> </pre>
UserData.xml	Contains persistent user data that can be migrated from device to device. Persistent data includes friends, items, and completed levels (Progress)	<pre> <UserData> <socialVisibility>true</socialVisibility> <levelActive>false</levelActive> <creationDate>1488986057133</creationDate> <lastModified>1492614171020</lastModified> <rewardLastRequest>2017-04-18</rewardLastRequest> <rewardPoints>10600</rewardPoints> <rewardBoxes>6</rewardBoxes> <character> Data... </character> <Account_item_inventory> Data... </Account_item_inventory> <Completed_levels> Data... </Completed_levels> <Active_friends> Data... </Active_friends> </UserData> </pre>
Inventories.xml	Contains array of items that are actively used by the character.	<pre> <ActiveItemCollections> <CharItems> <ItemRef><itemName>Hat6</itemName><quality>1</quality></ItemRef> </CharItems> </ActiveItemCollections> </pre>

ConfigData.xml	Contains all the user's preferences regarding the game as well as a list of feedback already submitted.	<pre> <Configuration> <language>English</language> <authenticatedStatus>true</authenticatedStatus> <enforcedOffline>false</enforcedOffline> <audioLevel>1</audioLevel> <audioMute>false</audioMute> <currentScene>Simulation</currentScene> <lastScene>Simulation</lastScene> <firstTimeRun>false</firstTimeRun> <rememberLogin>true</rememberLogin> <autoLogin>true</autoLogin> <characterAttached>true</characterAttached> <username>demo@beds.ac.uk</username> <UserUUID /><password>#####</password> <eMail /> <refreshToken /> <apiKey /> <access_token /> <mha_token></mha_token> <expires_in>186967800</expires_in> <last_request>1492613775776</last_request> <internet_connection>true</internet_connection> <feedback_form>true</feedback_form> <FeedbackModuleList> <FeedbackModule> <module>feedback_module_outfitting</module> <date>1490881720424</date> </FeedbackModule> </FeedbackModuleList> </Configuration> </pre>
----------------	---------------------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

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

5.4. Data Protection

Server side considerations for data protection are covered extensively in Deliverable D3.2 'Initial iManageCancer platform prototype' under heading 4.3.4 Data protection. A brief summary of key points addressed in that deliverable are listed below:

- 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 further secures users data by virtue of 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.

5.5. Serious Game for Adults features and interface

5.5.1. Features

Feature	Description
Splash screen	Simple iMC branded and timed screen that transitions during the file loading task.
Language selection	Dynamically generated from the contents of the language file, allows a user to select a language and confirm that selection.
Login page	Provides a simple username and password input screen with additional options to play offline or create a new account.
Account creation page	Simple edit text form with submit or cancel buttons.
Main menu	Contextual menu driven by the existence of game data, new layers get offered the option to create their avatar here. Existing players will be able to continue the game from this menu.
Settings	Language and character settings are available here.
Common home menu	Animated common home button provides basic high level navigation and settings functionality.
Profile management	Application logic giving the user the option to download or overwrite their games saved data from the game back end server.
Character creator	An interface to build the main characteristics of the user's avatar, this feature allows for naming and settings up the appearance of the avatar, as well as gender.
Level selector	Dynamically generated list of available levels with progression and completion status indicators.
3D town	The main playable area of the game is set in a full 3D stylised town featuring 3D navigation and zooming.
Game GUI	The main game GUI provides a number of status meters and resource indicators as well as intractable buttons providing access to other features and help information.
iManageCancer conversion spinner	This random chance mini-game converts physical effort recorded by the iManageCancer platform into spins on a prize wheel.
Social interface	This interface allows a user to inspect existing friends and search for new friends, while also providing a feature to receive invites from other users.
Character item store	The character outfitting screen allows a user to dress their avatar in clothing that effects the main game, the interface allows the user to preview their items before purchase.

Trivia mini-game	This mini-game is accessed through the town's citizen goal interface located in the wellbeing bar, it provides the option to dissuade a citizen from a personal goal that may not be good for them. The game requires a user to score two correct answers out of three random questions in order to persuade the citizen.
Challenge dialogs	The levels in the game come with three types of challenge, mandatory Doctor's orders challenges, Citizen's personal challenges and Mayoral Duties challenges. The interface details the challenge and the actions required to complete it.
Activity interface	Every active building in the town provides activities, this interface displays the available activities with their cost, category and effects.
Building interface	Empty plots in the town can be built upon, this interface shows the user what buildings are available and their build costs. This interface also highlights in green buildings that have already been added to the town.
Feedback mechanism	Every feature can invoke a feedback form, these forms provide their context in their title and have 5 questions and one open ended edit text field for user feedback.
Audit functionality	Automatic audit functionality sends an audit to the iManageCancer platform after any significant user action.

5.5.2. User interface summary

Splash screen

The splash screen is displayed while the games data files are read from device storage and de-serialised into the main game controller.



Figure 6: Splash screen

Language selection

Every language group in the XML resource Lang.xml generates an icon button.

This scene also has an animated background, this is not essential but helps to give this scene a little more interest.



Figure 7: Language screen

Login screen

Two editable text fields allow a user to enter his/her log-in credentials, returning users will have these fields automatically filled in. Play offline will make the application ignore the login process.



Figure 8: Login screen

Character creation screen

The character creation screen allows the user to edit 6 characteristics of their avatar as well as name it.

1. Edit text field for name.
2. Interactive avatar preview.
3. Gender tab.
4. Colour tab.
5. Hair style tab.
6. Eye accessories tab.
7. Eyes tab.
8. Mouth tab.
9. Accept character?
10. Reject character?

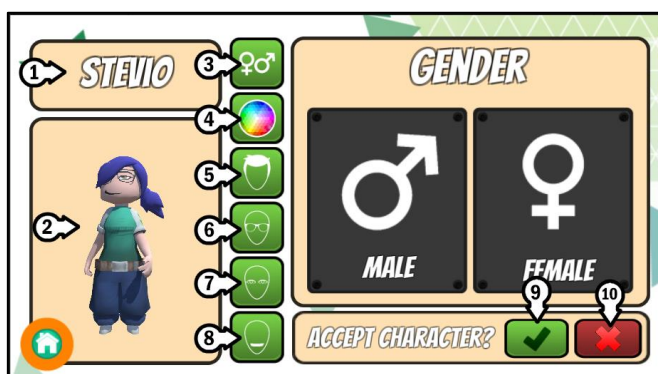


Figure 9: Character screen

Main menu

This menu is contextual in nature, the continue button only appears when a returning user is playing, new users will see a “New Game” button. The login-logout button displays the appropriate action depending on authentication status.

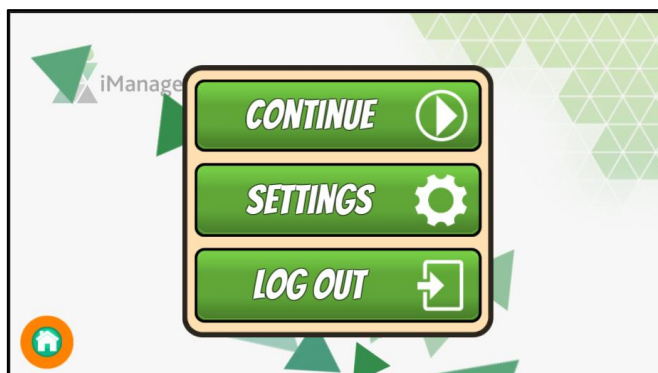


Figure 10: Main menu

Settings menu

Static menu with core settings options, language, character and coming in the next update general settings that effect audit and feedback settings.



Figure 11: Settings menu

Level selection

The level selector has a hard limit of 15 levels in its current state, but can easily have a paged solution to extend this.

1. Level button, pressing it will immediately launch the selected level if it is available.
2. Locked level status indicator, this indicated that this level is unavailable.
3. Completion achievement, this simple three-star rating system indicates to what level of success the player completed this level.
4. Current status icon, this example displays a play icon, this means this level is available to play. A pause icon would be displayed if the level had already been started.

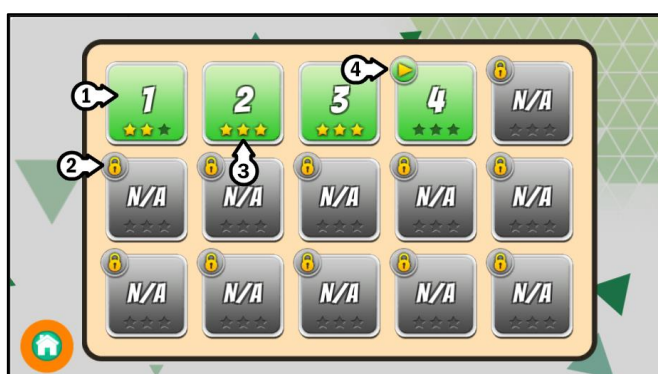


Figure 12: Level selection screen

Main game user interface

The main game UI is always visible during play within the town.

1. Money status, currently doubles as a trigger button for the iMC rewards feature. Displays the players balance to the right-hand side of the icon.
2. Social status button, triggers the social interface, also displays the number of available friend slots in the town and how many are filled with friends.
3. In-game outfitting/ store page. Launches the character clothing and accessory interface.
4. Wellbeing meter, light grey bar indicated the predicted amount of wellbeing gain, a green bar shows achieved wellbeing. The level is only complete when this bar is filled.
5. Doctors challenge button, this button occupied the wellbeing bar as an obstacle. Clicking this button will display the challenge and its completion conditions.
6. Mayoral duty challenge, this challenge is an obstacle to wellbeing, clicking the button reveals the completion conditions.
7. Patient challenge button, this challenge is an obstacle to wellbeing. Clicking this button reveals the completion conditions and exposes the dissuasion trivia mini-game button.
8. Action point meter, this label displays the amount of Action Points available this game day.
9. AP meter, a visual meter of remaining AP in the day.
10. Day counter, shows how many days have passed so far, currently also doubles as a button to trigger an end of day summary.
11. Weight meter, button triggers a tutorial help dialog. The meter shows the current weight of the town's cancer patient (citizen), balancing the meter as close to the white marker generates the most wellbeing. The meter represents the scale from too thin to overweight.
12. Social meter, similar to the weight meter but generally has no negative extreme.
13. Mental meter, similar to the weight meter, but shows the range from bored to stressed out.
14. Physical meter, similar to weight meter but shows the range from idle to exhausted.
15. Home animated navigation button.



Figure 13: Game GUI overview

Activity interface

The activity interface is launched when a user clicks on a completed building within their town. The user will get a number of activities that depends on the level of the building, level one will generate one activity etc.

1. Categories, some activities can be considered to belong to multiple categories.
2. Effects, the amount each of the four status meters will be effected by this activity.
3. The cost, this button will trigger a confirmation dialog, and is labelled with the monetary cost (if applicable) and the AP cost of the action.
4. Upgrade slot button, if a building is lower than level three there will be an option to upgrade the building to add new activity slots.

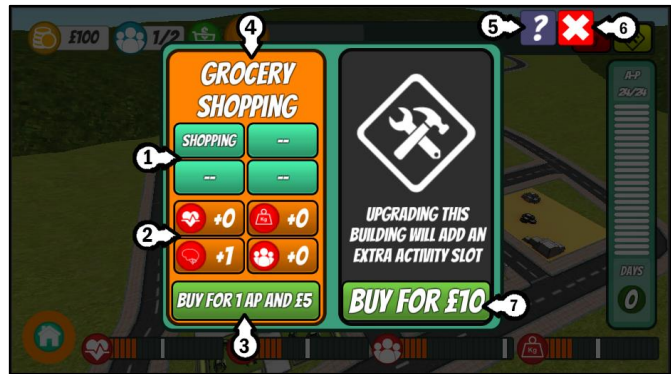


Figure 14: Activity interface

Building selector

Within the town there are empty plots of land which can become new buildings. Clicking these plots launches this building selection dialog.

1. Building button, if it is grey then there is not an instance of this building type in the town.
2. Help button, launches a tutorial dialog.
3. Close button, closes the dialog with no further action.

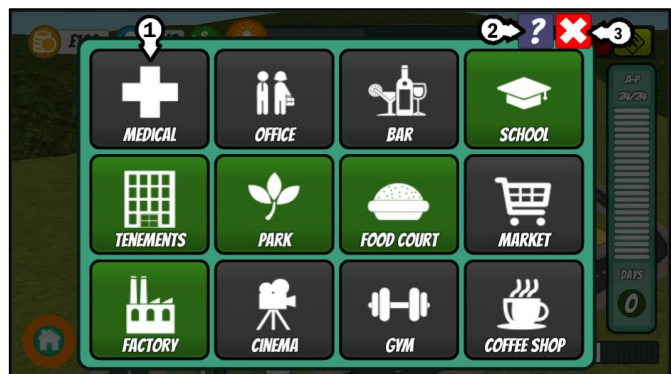


Figure 15: Building selector interface

Building inspector interface

After selecting a building to build in the previous dialog (Figure 15: Building selector interface) This inspection dialog appears displaying the costs and opportunities this building brings.

1. Building icon, this is the icon that is related to the buildings type.
2. Categories, this list shows the kind of categories of activity that will appear here.
3. Examples list, a random selection of activities able to spawn in this building.



Figure 16: Building inspector interface

4. Building name.
5. Existing status, if no example of this building exists then the user will be informed that it is a new type of building.
6. The monetary cost of the building.
7. Build time in AP, the amount of AP that have to be expended before the building is completed, this is not an amount that is deducted from the AP meter, but a timer.
8. Reject building, this leads back to the building selector dialog.
9. Build? The user will be prompted to accept this action, upon accepting this interface closes and the selected plot becomes a building site.

Trivia mini game interface

The user can attempt to dissuade a patient from desiring potentially unhelpful activities by showing their knowledge and convincing the patient to reconsider. The user must answer at least two out of three questions correctly to prevail.

1. User's section, text is highlighted in yellow.
2. Progress, correct answers get green ticks and wrong answers get red crosses.
3. The question body.
4. Submit the user's selection, upon clicking the answer will be processed and the correct answers highlighted, the button will then change to "next".
5. Preview of the patient asking the questions. This will be a unique character for each level.

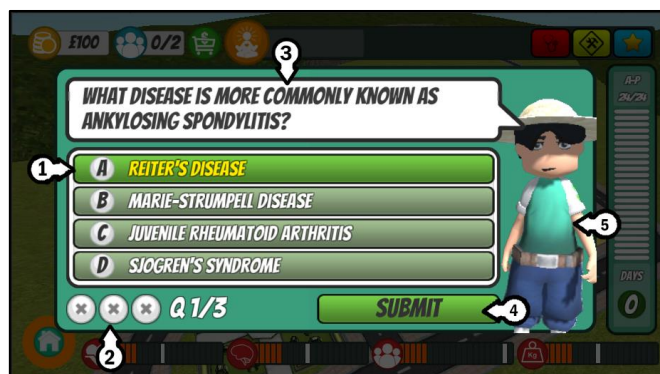


Figure 17: Trivia mini-game

iMC conversion interface

The iMC conversion interface pulls calorie burn data from the patients recorded activity. The number of available calorie points can be spent to get a spin of the chance wheel.

1. Available calorie points. These are calculated from the last time the app read the value.
2. The conversion button, spends the points to add a spin token.
3. The conversion cost, this value is currently hardcoded but may eventually be set by difficulty factors.
4. Spin button and counter, this button sets the wheel spinning and has a counter displaying the number of spins remaining.

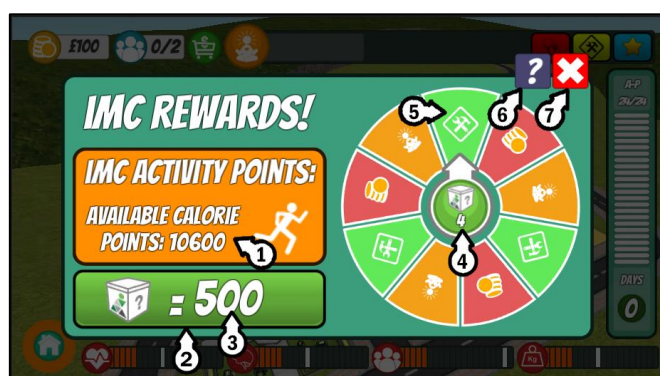


Figure 18: iMC conversion wheel

5. The reward currently selected, the green construction rewards add to the completion status currently unfinished building sites, the orange wellbeing rewards add a small amount of wellbeing to the bar up to the limit imposed by challenges, and the red money rewards simply add an arbitrary amount of money to the user's balance.
6. Help button, launches a tutorial help dialog.
7. Close, closes the dialog.

Social interface

The social interface displays a user's friends when first opened, the user can click on (2) to view this list again. This interface launches a web request upon opening, updating all social information before loading.



Figure 19: Social interface

1. Friend preview view. Displays any currently highlighted friend's avatar.
2. Active friends list, shows all accepted friends.
3. Messages, shows incoming friend requests.
4. Friend search field, allows the user to enter a search string to be added to a friend search query.
5. Friend card, these cards display the friends character name and act as a button to preview their avatar on the preview panel.
6. Help button.
7. Close button.
8. Revoke friendship button, prompt the user to confirm this action.
9. Results pager, allows the user to page through groups of results from any of the categories, search results, active friends or friend requests.

Store interface

The item store allows a user to equip clothing items to their avatar, these items can apply effects to the game and can be used tactically by the user to help address an imbalance in their statistics.



Figure 20: Store interface

1. Avatar preview.
2. Monetary cost of item, this may be zero if the item is already owned or purchased.
3. The effect of the item on a user's statistics, this panel shows the amount and an icon of the category it effects.

4. Purchase or equip button, this buttons description shows the total cost of all currently selected clothing items.
5. Hat category.
6. Chest category.
7. Legs category.
8. Item name and pager, clicking the chevrons either side of this name cycles the equipped items.
9. Help button.
10. Close button.
11. Currently active category label.

Feedback interface

The feedback interface allows a user to provide feedback on a module or feature they have just used. This dialog appears only if a user has not given feedback before or within a one week period.

1. Do not show again check box, if a user selects this and then closes the dialog there will be a record stored for this module and the dialog will not appear again even if left blank.
2. Module title and feedback request. The request for feedback contains the module title that invoked it so users know what they are providing feedback for.
3. Rating guide.
4. Interactive star rating, stars turn gold when selected.
5. Open ended user input text field for more freeform feedback.
6. Submit feedback, initiates an asynchronous task sending the report with no expectation of a response.
7. Cancel this feedback.

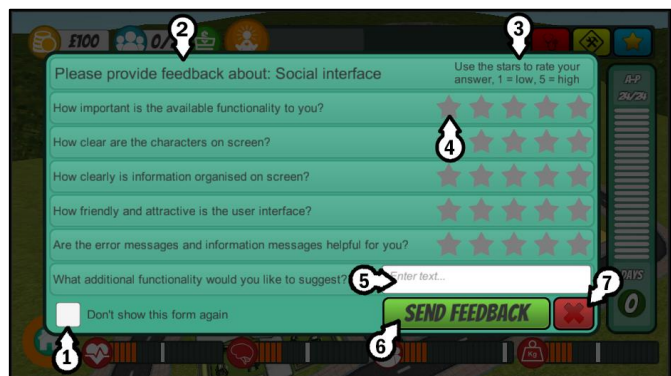


Figure 21: Feedback interface

6. Conclusions

The serious game for adults offers a colourful and engaging game play mechanic which keeps lifestyle and positive action at its core throughout. The design decisions made from the beginning have taken into account the need for positive feedback and opportunities to learn through exploration and vicarious experiences. To this end the game has avoided penalising users for ineffective play, instead it aims to incentivise the development of critical problem solving skills through mastery of the game mechanics and in-game item rewards. The game does not provide explicit medical recommendations and has intentionally focused on a different fictional cancer patient per level. The intent is that users will be able to draw parallels and gain some useful knowledge that they can apply to their own situation. The primary concept of the design is to

reinforce the idea that engaging in healthy lifestyle management can help an individual overcome the challenges in their life caused by cancer and achieve a sense of wellbeing, a principle reinforced by the explicit mechanics of clearing challenges from their wellbeing meter in game.

Early feedback provided by the partner eCancer has led to many design improvements to not only the interface, but also to the core design. By changing the role of the user to that of a supporter of a fictional cancer patient we have removed the confusion of users as to their role and the nature of the actions we expect them to undertake. These changes have also helped remove the issue of users trying to relate too closely to the challenges in the game, in many situations these challenges were contrary to their real-life situations resulting from cancer.

Due to the serious game being in development at time of writing, there is currently a lack of testing and evaluation data to include in this deliverable, fortunately there are workshops planned in the near future and opportunities for more in depth testing and analysis. An addendum to this deliverable is planned to cover this lack of data.

The serious game currently aims to:

- Allow a user to see their cancer fight from the perspective of a supporter.
- Promote critical analysis of lifestyle and work life balance.
- Reward interaction and utilisation of other iManageCancer platform tools.
- Provide a different perspective to elaborate cancer related problems and promote self-efficacy and self-management skills through mastery and vicarious experience of the avatar.
- Provide an opportunity for self-education and information provision through a challenging and repeatable mini-game feature that exposes users to a broad and topical database of medical and lifestyle trivia.

7. References

- [1] Unity product website, available at: <https://unity3d.com> and last accessed 25/05/2017.
- [2] Android API level Dashboard, available at: <https://developer.android.com/about/dashboards/index.html> and last accessed 25/04/2017