National College of Ireland
BSc in Computing
2016/2017

David Byrne
13109863
x13109863@student.ncirl.ie

IronSight

Technical Report

National College of Ireland

Contents

Executive Summary

The video game industry is saturated with new releases at the moment. Digital distribution giant Steam releases on average 10 new games every day and a lot of these games go by unnoticed and unbought. Consumers must often make compromises when deciding what to buy. Mistakes are often made and impulse buying can often take hold.

To solve this problem, IronSight aims to not only help put the massive amounts of releases of games into a consumer's perspective but also to help the consumer decide on what's best for them. The app will list all major and minor releases to the user, allowing the user to search and filter what they want to buy. These titles can then be added to a personal list where updates and notifications will be sent to the user.

Through this personal list of titles, a suggested list will be generated with other games related to what the user is interested in. The data from other users may also add to this suggested list.

The main list is supported by an online database where all titles will be stored. The personal users list is stored on a local database on the device which will be managed by Firebase.

Suggested titles are generated by randomising the titles in the main list and generating a separate list of diverse titles based on genre. This is to encourage the user to broaden their horizons with regard to games they may not have previously considered.

Notifications of game releases are sent to the user by using the Firebase console to schedule notifications to users about the release of games.

Meet ups are also supported as a feature that enables users to contact other users with the intent of joining them in playing any game that has been added to their own personal list.

Links to digital distribution sites will also be included to maximise user convenience.

1 Introduction

As it stands, on the web or through application providers on both Android and iPhone, there are no services providing comprehensive and concise information on video game release dates. While there are sites which provide extensive scheduling, there are none providing a complete package of filtered release dates, pricing information and current pre-orders. In the current state of affairs, a consumer must navigate through popular websites often cluttered with verbose articles of content to get to the release date of the product they wish to purchase.

The absence of a platform to be able to track and schedule upcoming games can often lead to disappointment from the consumers point of view. Given the amount of games which are released over the course of even a month, there can be difficulties around keeping track of what people want and when they can get it.

Games in the modern market are being priced higher than ever before with new releases usually labelled at €75. Given the current arrangement of available information online, most people would struggle with effectively prioritising games that they can afford over a certain stretch of time. Often a person may see a good review of a game they had not previously been aware of and would buy this game. For many people, this may result in not having the finance for a game they would much rather have purchased, released maybe a week later.

While there are some services providing lists of release dates online, these lists can be cluttered and lack any refined filtering options or else be lacking in some other features. The use of multiple services from different websites is often required to get all information desired. The ability to track release dates on a personal calendar, be aware of the cheapest option available in the market and have a link to where this game can be bought is currently not available.

Digital video game distributor “Steam” are introducing on average 8 games per day to their library. The market is in need of a service that can track these games in a clear and concise manner while offering users the option to save their own personal lists rather than having to traverse massive amounts of data each time they log on. As well as providing further information about pricing and pre-ordering.

Within the gaming community there is a big divide between “indie” gaming and “AAA” gaming. “Indie” games are typically developed by a single person or small team and are centered primarily on an artistic environment with simple gameplay. While “AAA” games are developed by large companies made up of multiple teams of developers. These games emphasise the use of the latest technology available to game developers and so boast the best visuals and complex game mechanics. There are gamers who appreciate both of these styles of game though many people find themselves enjoying either one or the other. No current scheduling service divides these two styles and it is a feature which is sought after.

1.1 Aims

The scope of this project is to develop a mobile application which delivers a platform for users to browse extensive lists of current and upcoming products. These products can then be sent to a personal list which only that specific user may access.

The app will make use of a simple, intuitive UI created through the Android Studio. IronSight will require the use of an online database to store the large quantities of information in relation to game.

There will be a notification feature which will alert the user when a product on their list has released.

Recommendations on similar products will show on personal list UI, generated through randomizing genres.

1.2 Technologies

Technologies in use:

1.2.1 Java with Android Studio

The programming language used to create the project will be Java. The app will be developed through the Android Studio development environment. Android Studio is a very versatile IDE for Android development as it provides an inbuilt emulator which supports nearly all Android devices. It also has intelligent code editing, auto completing code and identifying problems with code already written.

While being an easy to use IDE, it also has an extensive library of UI elements to choose from. For this project, a powerful UI technology called Material Design will be implemented to give the app an attractive modern appearance.

1.2.2 Firebase

Firebase will be used to create and manage the server storage database. Through Java, the database will be set up and managed with Firebase authentication and real time database.

Firebase authentication is a user account management service which securely stores the users email address paired with a password.

Firebase real time database is tasked with handling the application content such as the various lists of games as well as the personalized user profiles.

1.2.3 Material Design

This is used for various design features in the app such as the navigation drawer, game profile screens and login/register pages.

1.2.4 ProGuard Security

This is a security feature available to Android Studio developers. It shrinks the code of the project to remove any unnecessary code or resources in the release build. It also obfuscates the project so as to make it difficult to reverse engineer.

2 System

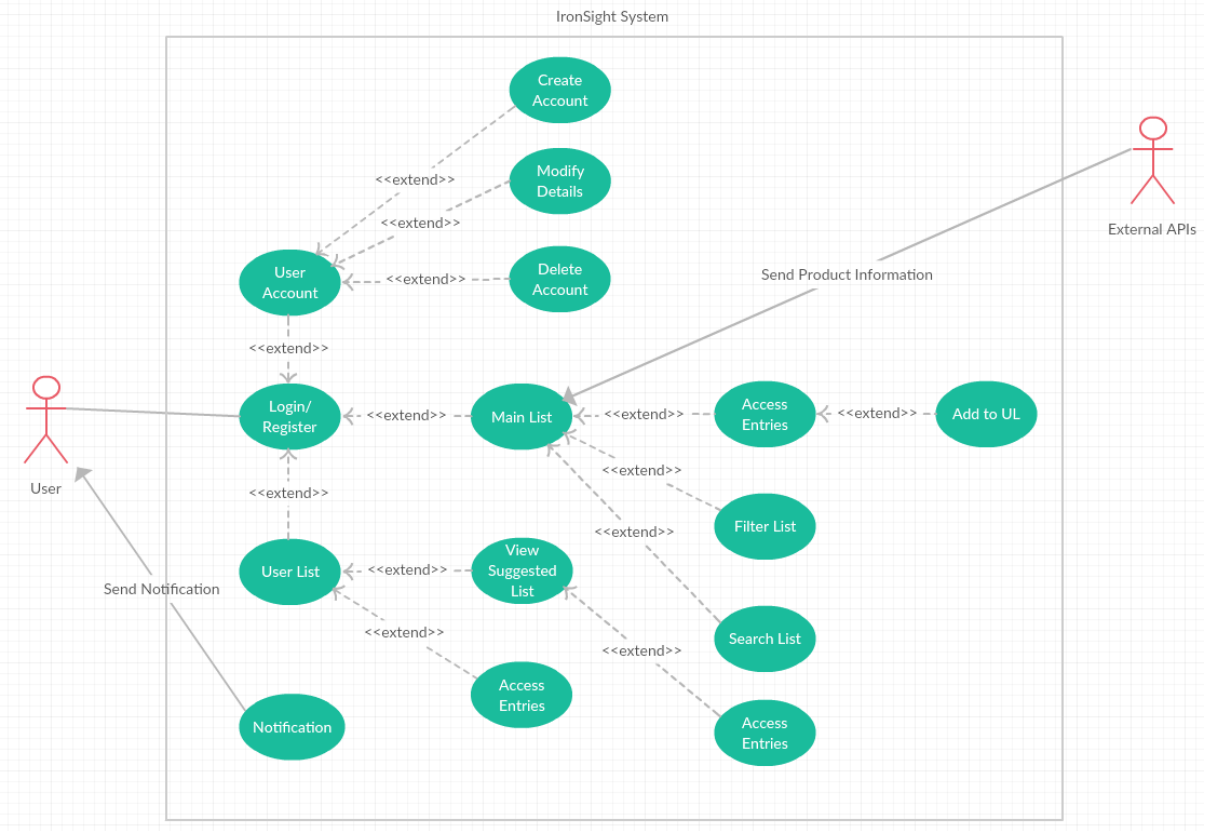
2.1 Functional requirements

Functional requirements by order of importance:

- User accounts – register/log in
- Main list
- Personal user list
- Suggested list
- Product release notifications

These main requirements will be outlined further following this summary. A use case diagram will provide visual representation of the use cases encountered by a user. This will then be followed by a breakdown of all main use cases, describing the process of using each of them. Extended use cases of these main ones will be included in the descriptions also.

2.1.1 Use Case Diagram



2.1.2 Requirement 1: User Accounts

2.1.2.1 Description:

The ability for users to log in to an account is most important as products they wish to track are saved to each account. Without an account there could not be notifications or recommended products.

2.1.2.2 Use Case Description:

Use Case

Login/Register

Scope

The scope of this use case is for the user to either login using their pre-made user account for to create a new account. As well as this, the user should be able to manage their account.

Description

User either logs in and then continues to either Main list or User list or else must create a new account. Users can either create, modify or delete accounts through this option.

Flow Description

Precondition:

The application is started and the first screen presented to the user is the login screen.

Activation

Process begins when user activates the application. The app will then prompt user to either login or register new account.

Main Flow

- User not registered, presses create new account
- The system shows registry page
- User fills out require fields and submits
- System returns user to login page where they login with username and password

Alternate Flow

- User is already registered
- User enters username and password
- User shown main list page

Exceptional Flow

- User enters unregistered username or incorrect password
- System instructs user to register or retry username/password

Post-condition

System enters main list page

2.1.3 Requirement 2: Main List

2.1.3.1 Description:

This page is where all entries of the database are displayed to the user on the UI, limited to a certain amount per page. Here the user can browse the list, search it, filter it or add entries to their own user list.

2.1.3.2 Use Case Description:

Use Case

Main List

Scope

On this screen, users can scroll up and down through entries of the database. Users can access each entry by clicking them. Searching and filtering are also enabled here. Entries displayed at the list menu, prior to accessing full information page, will only show entry title and release date.

Description

This is considered to be the main menu of the app. By using pop up or slide across menus, users can manipulate the list. When an entry is accessed, data on the entry is presented as well as the option to add to personal list.

Flow Description

Precondition

User has successfully logged in and is shown the main list page.

Activation

The main list screen will display after a successful login made by the user.

Main Flow

- User browses through entries of main list by scrolling up or down
- User presses on an entry in order to get information on product
- The user can then save an entry to their personal list

Alternate Flow

- User chooses to search the database for a particular entry using sliding search bar
- Results of search are returned to user
- User continues at second step of Main Flow

Exceptional Flow:

- User chooses to filter database for a group of entries
- User selects parameters from options presented in sliding search bar
- User continues from step 2 of Alternate Flow

Post-condition

User is returned to main list page.

2.1.4 Requirement 3: Personal User List

2.1.4.1 Description:

Having added entries from the main list page to the personal list. Entries will then be displayed on the user list page in the same fashion as on the main list page. It will also be possible to search or filter this list. Recommended products will also be shown through the user list page.

2.1.4.2 Use Case Description

Use Case

User List

Scope

Similar to the main list, users can scroll through entries though the list will be much shorter. Accessing each entry displays full information on it. Searching and filtering is enabled here. Entries displayed at the list menu, prior to accessing full information page, will only show entry title and release date.

Description

This section is where users will track their own products of interest. It is from this list that notifications will be sent to users. Also by reading the tags associated with a user's favourited entries, a recommended products list is generated.

Flow Description

Precondition

User is shown the main list page.

Activation

User list page is activated by the user when switching over to it from main list page.

Main Flow

- User browses through entries of user list by scrolling up or down
- User presses on an entry in order to get information on product

Alternate Flow

- User chooses to search the list for a particular entry using sliding search bar
- Results of search are returned to user
- User continues at step two of Main Flow

Exceptional Flow

- User chooses to filter list for a group of entries
- User selects parameters from options presented in sliding search bar
- User continues at step 2 of Alternate Flow

Post-condition

User is returned to user list page

2.1.5 Requirement 4: Suggested List

2.1.5.1 Description:

The suggested list of products, or recommended list, is a collection of entries that are generated randomly from the main list of games using a a diverse list of genres.

2.1.5.2 Use Case Description

Use Case

View Suggested List

Scope

Similar to the main list, users can scroll through entries though the list will be much shorter. Accessing each entry displays full information on it. Searching and filtering is enabled here. Entries displayed at the list menu, prior to accessing full information page, will only show entry title and release date.

Description

This list will be for users to find similar products to those already in their personal list. Entries will be automatically added here from the application by reading popular tags from the user's already existing personal list. The list will be ordered by frequency of tag found in personal list and then by release date. Entries will only display title and release date in list entry, further information when entry is accessed.

Flow Description

Precondition

User is on user list page

Activation

Accessed through personal list page.

Main Flow

- User browses through entries of user list by scrolling up or down
- User presses on an entry in order to get information on product
- User may add entry to personal list

Alternate Flow

- User chooses to search the list for a particular entry using sliding search bar
- Results of search are returned to user
- User continues at step two of Main Flow

Exceptional Flow

- User chooses to filter list for a group of entries
- User selects parameters from options presented in sliding search bar
- User continues at step 2 of Alternate Flow

Post-condition

User is returned to user list page.

2.1.6 Requirement 5: Product Release Notification

2.1.6.1 Description:

Each user will have their own personal list of products which they are waiting for release. If a product in their user list has not yet been released, the app will send a notification to the user on the morning of the products release.

2.1.6.2 Use Case Description

Use Case

Notification

Scope

The user dates of all games are documented and registered with the Firebase console. The console is used to schedule and then distribute the notifications to the users.

Description

The aim of the application is for users to be able to plan and schedule their purchases. So the implementation of a notification is useful for a user to know the day of release.

Flow Description

Precondition

User is in user account menu where modification or deletion is possible.

Activation

User switches notifications on from user account menu.

Main Flow

- Notifications are activated
- User receives notifications on days of user listed product release dates
- Notification can be dismissed

Alternate Flow

- Notifications are deactivated from user account menu
- User does not receive notifications

Exceptional Flow

- Notifications are active
- User adds product already released
- No notification received

Post-condition

User returned to user account page.

2.2 Non-Functional Requirements

2.2.1 Performance

- The app can be accessed online or offline but only personal lists are accessible from offline mode.
- The app must be booted to main menu within 5 seconds.
- Access to the online main list must be achieved within 3 seconds.

2.2.2 Usability

- App will be completely intuitive to use at any level of computing knowledge.
- Menus to be kept minimal and easy to read.
- Application operations will be explained through use of the app, making all options clear as a user explores the application.

2.2.3 Reliability

- The application should have 99% uptime in relation to online functionality, otherwise the app will be available to a user at any time once it is downloaded.
- After launch, the app will be under constant maintenance to ensure optimal functionality.
- In the event of an application crash, it will attempt to restart itself.
- Any errors or exceptions will be handled through Toast messages to the user.
- The app must ensure the safety of user information.

2.2.4 Supportability

- The app may eventually support pre-ordering of products through digital distributors.
- The system will be maintained by its creator, David Byrne.
- IronSight will be supported by Android devices.

2.2.5 Interface

- The system will interact with the date and time system of the phone it is installed on.
- Data that populated the main list will be imported from an online database.
- If the user has silent mode enabled on their device, the app should consider this when issuing notifications

2.3 Data requirements

Data taken from the IGDB and Steam APIs are necessary for the app to be functioning at full capacity. However, there will be an offline database on the user's device which will still track any personal entries on the user favourite list.

2.4 User requirements

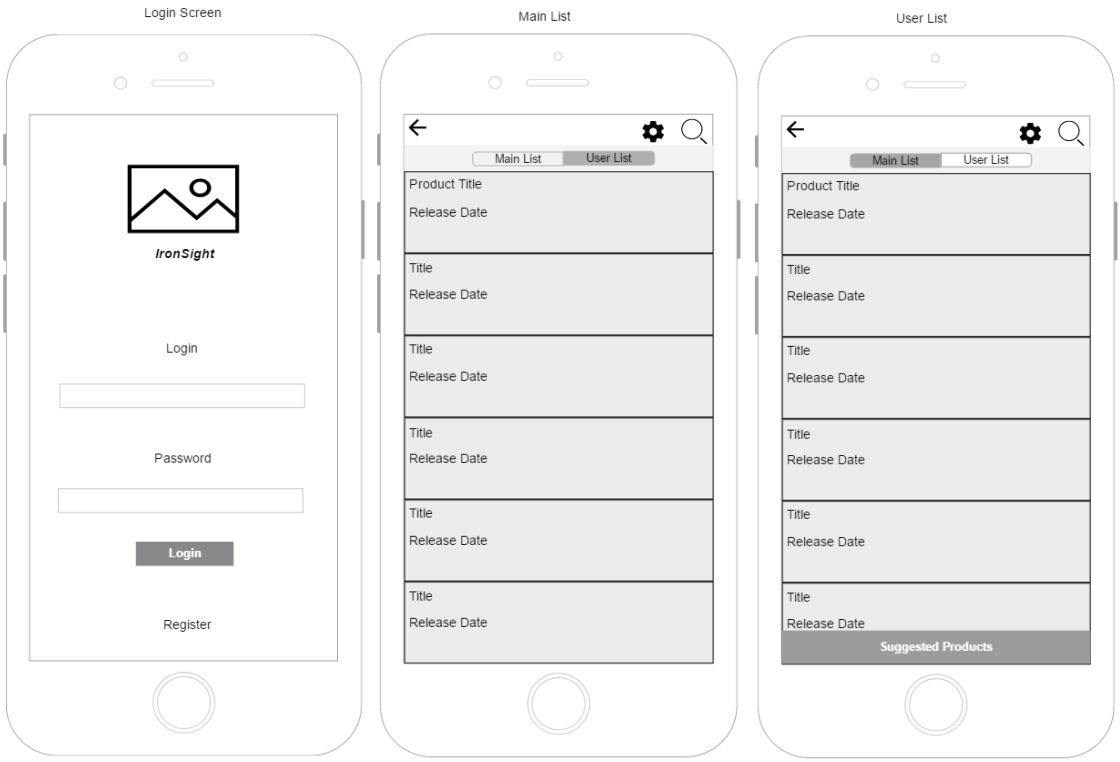
Users must be able to log into their own personal account. The application should display information on upcoming products and supply the user with information about precisely what product they may want to buy. The ML should be easy to filter with options covering all attributes of products listed. So by accessing the ML, users can filter out genres, platforms etc. from returned results. A user must be able to save any entry from the ML to their own UL easily and then be able to view their UL separately from the ML. Each entry must display up to date pricing as well as up to date release dates, taking into account any possible publisher delays. All other information about products must also be accurate and up to date. Notifications must deliver to the user through the application on time on the correct morning of release, displaying the correct product. All prices and release dates must conform to the users registered region of residence. Recommended products must also relate to already listed products in users UL through the use of tags.

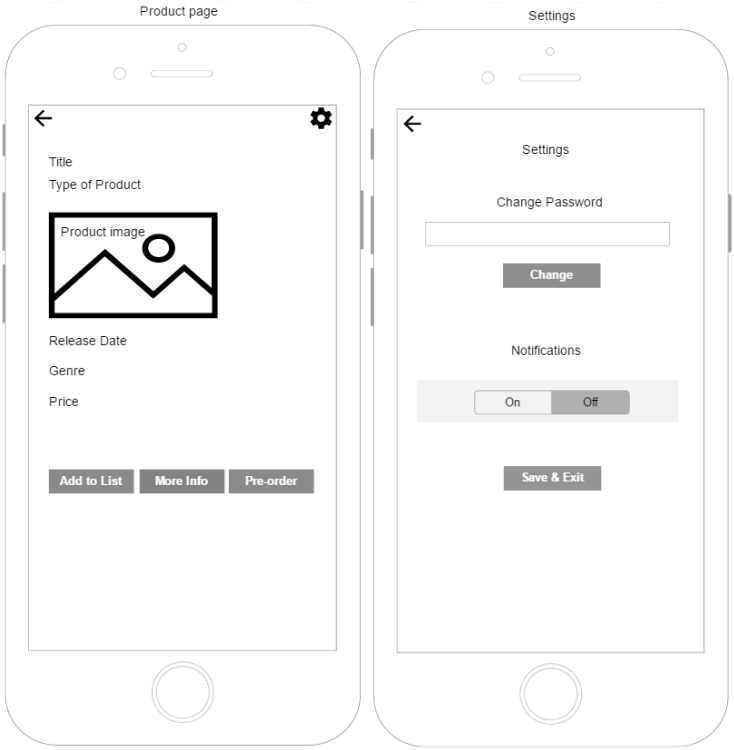
2.5 Interface Requirements

The applications database will interface with multiple APIs in order to populate the product entries with information such as title, release date, genre etc. The app itself will interface with the online database which provides the data with which to populate the main list.

Users will interface with a GUI, which will enable the user to provide inputs to the application as well as supply the user with output.

2.5.1 GUI



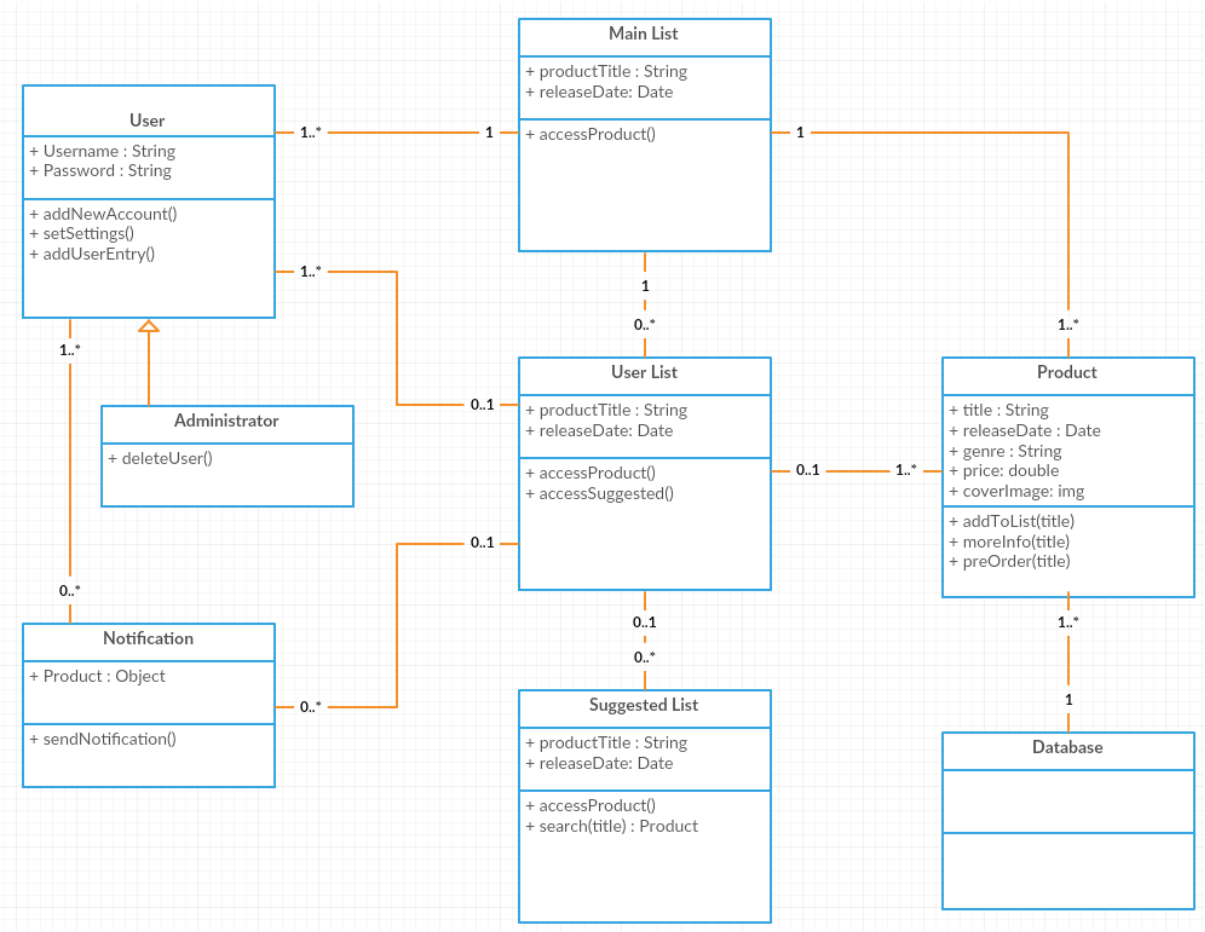


2.5.1.1 GUI Structure

- Login screen will enable the user to either log in to their account or register a new account
- Once successfully logged in through pressing the login button, the user is brought to the main list page.
- On main list page there will be a navigation button which will display a side navigation drawer when pressed.
- The user list shows a user's saved products, ordered by release date by default.
- On both main and user list pages, users can access options through the options icon, search to either filter or search by text input the list and a back to previously screen button.
- The product page outlines some concise information on the product entered from either the main or list pages.
- It includes further options to add to personal list if browsing through main list, more info which opens a web browser to information on a different site about the product and pre-order option (possible future feature) for buying products through digital distributors.

- Settings page can be reached from any other page (apart from the login/register page) using the settings cog in the top right corner. Here a user can change their password or set notifications to be on or off.

2.6 Design and Architecture



Class diagram of IronSight

The layout of the app will be broken into activities. The standard format of Android Studio development is through separate activities which represent user interfaces. The primary activities will be for the login screen, the main list of games, the users list of games, the suggested list of games as well as a settings menu.

The login will integrate with the server database where user accounts are checked and returned or created. After having logged in, the user may change password or alter some settings and these settings will be saved to their profile on the database.

The main list page is integrated with the online database due to the massive load of entries populating the page. The page will be limited to a certain number of entries so as to improve user experience. Given the amount of data involved in populating this list, it is required that it be stored online. From this page, entries can be accessed for further information and from there, be added to the user's personal list.

Personal lists are generated through the selection of products on the main list. The personal lists are also stored on the server database and are written to by the user through use of the app.

The notification feature will be implemented through the Firebase console. On the day or day before the release of a listed game, a notification will be sent to the user alerting them.

2.7 Implementation

2.7.1 Firebase Authentication

Firebase is a mobile and web application development platform. It offers multiple features which developers can use to enhance the functionality of their applications. IronSight will make use of some of these features, one of which is Firebase Authentication.

Firebase Authentication allows the app to support user accounts. By simply instantiating FirebaseAuth in the code, the application is capable of interacting with the Firebase user database. The database of application users can be accessed and managed through the Firebase console. The Firebase console is where the app, which is developed through Android Studio, is linked to the Firebase tools and where user management can be performed.

New user accounts are created through the SignUp activity in the IronSight application.

2.7.2 SignUp Activity

An activity represents an interface with the user. The sign up activity is where the application takes user inputted information and uses it to create a new user. Each activity, which is where the Java code exists, is most often linked to an XML layout file which represents the page displayed to users. So SignupActivity has a corresponding layout file called activity_signup.xml.

The sign up activity instantiates all views in the layout file, such as text entry fields, text labels or buttons. In this activity there are inputs for email and password as well as a submit button. If either field is left empty or do not fulfil necessary requirements, the app will return an error to the user and prompt them to retry. Otherwise, upon submission of

valid email and password inputs, the instance of FirebaseAuth will create a new user with email and password and pass the user on to the CreateProfileActivity.

If Firebase is unable to create a new user, an exception will be thrown and the user will be informed of the failure.

```
auth.createUserWithEmailAndPassword(email, password)
        .addOnCompleteListener(SignupActivity.this, new OnCompleteListener<AuthResult>() {
            @Override
            public void onComplete(@NonNull Task<AuthResult> task) {
                Toast.makeText(SignupActivity.this, "createUserWithEmail:onComplete:" + task.isSuccessful(), Toast.LENGTH_SHORT).show();
                progressBar.setVisibility(View.GONE);
                // If sign in fails, display a message to the user. If sign in succeeds
                // the auth state listener will be notified and logic to handle the
                // signed in user can be handled in the listener.
                if (!task.isSuccessful()) {
                    Toast.makeText(SignupActivity.this, "Authentication failed." + task.getException(),
                            Toast.LENGTH_SHORT).show();
                } else {
                    startActivity(new Intent(SignupActivity.this, CreateProfileActivity.class)); //change back to MainActivity instead of
                    finish();
                }
            }
        });
```

Figure 1 – SignUp activity

2.7.3 Create Profile

The requirement of each user to have a profile is essential for the application as each user will have their own personal list of games in which they are interested in. These games will be saved to the user's profile which is located on the Firebase database which is separate to Firebase authentication.

The CreateProfileActivity is accessed only after passing the requirements of the Signup activity, which are submitting an email address and password, and is completely inaccessible from any other part of the application so as to avoid users creating more than one profile.

First the activity will instantiate the current user, which has just been created, from Firebase authentication. Firebase authentication handles some user details but this application will be using a separate "profile" for each user which will be saved to the Firebase database as the primary way of handling users. By instantiating the FirebaseUser, the app will take the user ID from authentication and use it to create a new profile associated with the user.

The layout associated with CreateProfileActivity prompts the user to enter a user name. This username can be anything chosen by the user. The user then clicks the submit button to create their new profile. The submit button takes the user inputted name, the user ID string from their account and the email address also from their account and creates a new profile to the database.

This write is performed on the button click by calling the writeNewUser method from the activity. This method takes the user ID string, the user name and email as parameters and uses them to create a new user using a User model. This model is then written to the

database as a new user profile. The user model also supports an array of game objects but this is assigned as null for now as the user is only newly created. Following the creation of a new user to the database, the user is directed to the main list activity of the app.

```
submit.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        String userName = userNameET.getText().toString();
        writeNewUser(user.getUid(), userName, user.getEmail());
        startActivity(new Intent(CreateProfileActivity.this, MainActivity.class));
    }
});
```

Figure 2 – Submit new profile

```
private void writeNewUser(String userId, String userName, String email) {
    User user = new User(email, userName, null);

    databaseUsers.child(userId).setValue(user);
}
```

Figure 3 – Write new user profile method

2.7.4 Firebase Database of Users

To access and manipulate the firebase database, it must first be referenced in code. A particular node can be referenced by inputting the address within the code. So in the case of the users, this line of code is used: DatabaseReference databaseusers = FireBaseDatabase.getInstance().getReference("users");. The database is structured using JSON so it is not strict as a relational database would be.

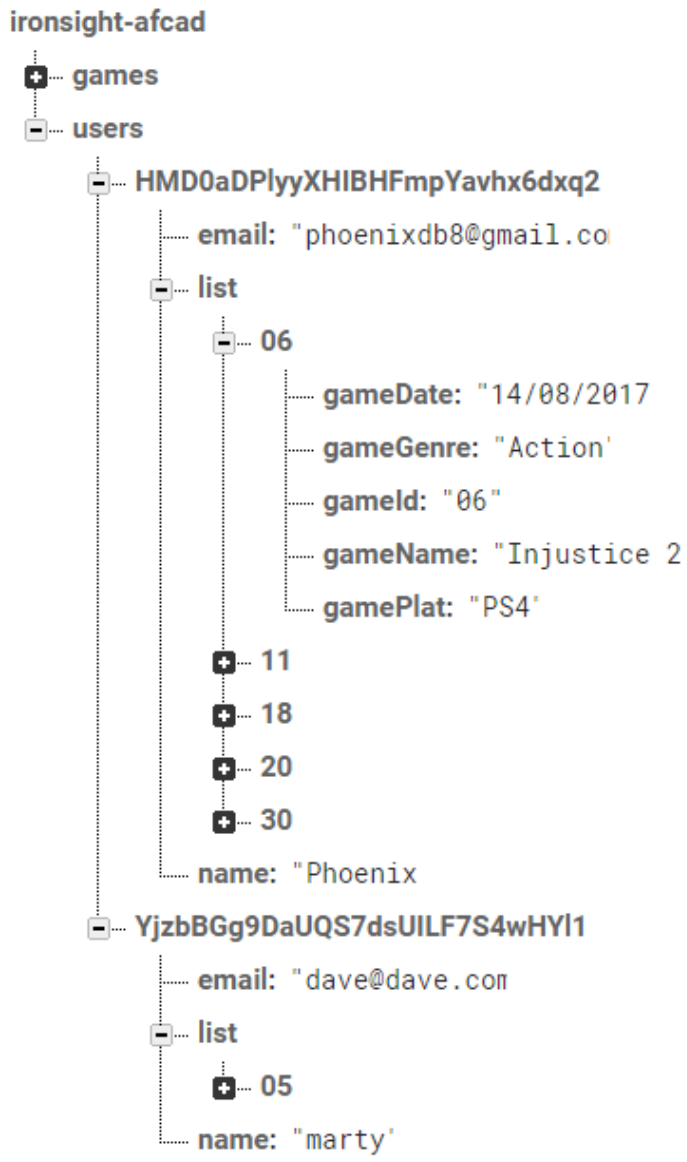


Figure 4 – Structure of JSON database for users

Shown above is the structure of the users in the database. At the top is the root node and below it are the child “users” and “games” nodes. In the users node the name of the node is taken from the user’s firebase authentication user ID which is a long string of randomised characters. Within this node are all the profile details for the user such as email address, user name and a list of personally chosen games. The writeNewUser method is what adds a new user to this “users” node.

2.7.5 Login Activity

This activity is very similar to the Signup Activity. The user is presented with email and password entry fields and a submit button. Upon clicking the submit button, the activity

will check the inputted details to see if they are a valid user. If the user is not valid, the user will be informed or if the fields are left the empty the user will be informed. If valid, the user is directed to the main list activity.

```
auth.signInWithEmailAndPassword(email, password)
        .addOnCompleteListener(LoginActivity.this, new OnCompleteListener<AuthResult>() {
            @Override
            public void onComplete(@NonNull Task<AuthResult> task) {
                // If sign in fails, display a message to the user. If sign in succeeds
                // the auth state listener will be notified and logic to handle the
                // signed in user can be handled in the listener.
                progressBar.setVisibility(View.GONE);
                if (!task.isSuccessful()) {
                    // there was an error
                    if (password.length() < 6) {
                        inputPassword.setError("Password too short, min. 6 chars");
                    } else {
                        Toast.makeText(LoginActivity.this, "Authentication failed, check yo
                    }
                } else {
                    Intent intent = new Intent(LoginActivity.this, MainActivity.class);
                    startActivity(intent);
                    finish();
                }
            }
        });
```

Figure 5 – Handling of attempted logins

2.7.6 Activities and Fragments

After a user has either registered or logged in, they will be directed to the main activity. This page will display a list of all games coming out soon by release date. However this list is in fact not a true activity. Activities may be divided into fragments. Fragments behave the same way as activities do though many can be loaded at the same time. So in the case of this application, there are three fragments attached to the main activity. Each are lists of games being read by from the database. So navigating between the three pages does not require the app to read from the database each time as all three fragments are loaded and remain active when the main activity is active.

```
private void displayView(int position) { //the stuff f
    Fragment fragment = null;
    String title = "IronSight";
    switch (position) {
        case 0:
            fragment = new MainListFragment();
            title = "Everything";
            break;
        case 1:
            fragment = new UserListFragment();
            title = "My Games";
            break;
        case 2:
            fragment = new SuggestedListFragment();
            title = "Check These Out";
            break;
        default:
            break;
    }
```

Figure 6 – Three fragments within the Main Activity

2.7.7 Main List

The main list fragment is where all games from the database are displayed in a list format. Each item in the list displays the name of the game, the genre and the release date. The entire list is ordered by game ID number in the database which is set by release date.

The data is pulled from the database and displayed on the page by iterating through all the entries in the database and passing them to an array and then this array is passed through an adapter. This adapter takes the values from the array of game objects and displays them as a list view, see figure 8.

```
databaseGames.addValueEventListener(new ValueEventListener() {
    @Override
    public void onDataChange(DataSnapshot dataSnapshot) { //any time database is

        gameList.clear(); //need to clear if it contains any game previously as

        for(DataSnapshot gameSnapshot: dataSnapshot.getChildren()){ //iterate th

            Game game = gameSnapshot.getValue(Game.class);

            gameList.add(game);

        }

        //getActivity() in a Fragment returns the Activity the Fragment is curre
        GameList adapter = new GameList(getActivity(), gameList);
        listViewGames.setAdapter(adapter);


    }
```

Figure 7 – Iterating through the games in database and setting to adapter

2.7.8 User List

The user list is identical in appearance to the main list. This is where a user's added games are. How the function of adding a game from the main list to the user list works will be covered in the Game profile page description. Games here are also listed in order of release date. The user list is read from the list node of the current users profile entry in the database as shown in figure 4.

Figure 8 – User list

2.7.9 "Check these out" list

The "check these out" list uses a method that generates a random list of main list entries to the suggested list to encourage users to branch out to genres that they would not normally be exposed to. The method "randomGenre" assigns each valid genre as an integer type. Each genre integer is then assigned a random value between 0 and 100. The method then determines the genre with the highest numerical value and assigns that genre as a return string value.

The main body of the class iterates through every entry in the games database and compares each instance of the games genres to the return value from "randomGenre". So for each time the class iterates over a game instance, the random method is run on it returning a different genre each time. If this random genre matches the genre of the current game, it is added to the list.

```
int strategy = ThreadLocalRandom.current().nextInt(0, 100 + 1);
int rpg = ThreadLocalRandom.current().nextInt(0, 100 + 1);
int fps = ThreadLocalRandom.current().nextInt(0, 100 + 1);
int sport = ThreadLocalRandom.current().nextInt(0, 100 + 1);
int action = ThreadLocalRandom.current().nextInt(0, 100 + 1);
int horror = ThreadLocalRandom.current().nextInt(0, 100 + 1);
int adventure = ThreadLocalRandom.current().nextInt(0, 100 + 1);
int platformer = ThreadLocalRandom.current().nextInt(0, 100 + 1);
int fighting = ThreadLocalRandom.current().nextInt(0, 100 + 1);
```

Figure 9 – Randomised value for integer representing each genre

```
if(strategy > rpg && strategy > fps && strategy > sport && strategy > action && strategy > horror && strategy > adventure && strategy > platformer && strategy > fighting){
    favouriteGenreString = "Strategy";
}
else if(rpg > strategy && rpg > fps && rpg > sport && rpg > action && rpg > horror && rpg > adventure && rpg > platformer && rpg > fighting){
    favouriteGenreString = "RPG";
}
else if(fps > strategy && fps > rpg && fps > sport && fps > action && fps > horror && fps > adventure && fps > platformer && fps > fighting){
    favouriteGenreString = "FPS";
}
else if(sport > strategy && sport > fps && sport > rpg && sport > action && sport > horror && sport > adventure && sport > platformer && sport > fighting){
    favouriteGenreString = "Sport";
}
else if(action > strategy && action > fps && action > sport && action > rpg && action > horror && action > adventure && action > platformer && action > fighting){
    favouriteGenreString = "Action";
}
else if(adventure > strategy && adventure > fps && adventure > sport && adventure > action && adventure > horror && adventure > rpg && adventure > platformer && adventure > fighting){
    favouriteGenreString = "Adventure";
}
else if(horror > strategy && horror > fps && horror > sport && horror > action && horror > rpg && horror > adventure && horror > platformer && horror > fighting){
    favouriteGenreString = "Horror";
}
else if(platformer > strategy && platformer > fps && platformer > sport && platformer > action && platformer > horror && platformer > adventure && platformer > rpg && platformer > fighting){
    favouriteGenreString = "Platformer";
}
else if(fighting > strategy && fighting > fps && fighting > sport && fighting > action && fighting > horror && fighting > adventure && fighting > platformer && fighting > rpg){
    favouriteGenreString = "Fighting";
}


return favouriteGenreString;
```

Figure 10 – Checks for highest value genre integer

2.7.10 Game profile and functionality

The game profile screen is accessed through clicking on a list entry on any of the three previously discussed lists. This screen details the game entries and provides users functionality around interacting with the games depending on where the profile screen has been accessed from.

Accessing a game list item from the main list will bring the user to the profile page with game information and three buttons. There is a button to buy the game which directs the user to a retailer web page for the game in question. There is an info button which brings the user to a web site with further information on the game and there is an add button. The add button is what adds the game to the users list.

Clicking the add button will execute the writeNewGame method. This method uses a database reference based on the current user, which accesses the list node of the current user's profile i.e. "users/"+ user.getUid() + "/list". This address is used to add the current game object to the user's profile.

```
private void writeNewGame(Game game) {
    databaseUserList.child(game.getGameId()).setValue(game);
}
```

Figure 11 – writes the current game to user list on the database

```
addBtn.setOnClickListener((view) -> {
        writeNewGame(game);
        Toast.makeText(getApplicationContext(), title + " added to your list.", Toast.LENGTH_SHORT).show();
        Intent intent = new Intent(getApplicationContext(), MainActivity.class);
        intent.putExtra("gameName", title);
        startActivity(intent);
});
```

Figure 12 – method running write new game and directing user back to main activity

Accessing the game profile page from the user list is slightly different. The info and buy button remain the same though as the game is already added to the user list here, instead there is a remove button. The remove button behaves very similarly to the add button though in reverse.

```
private void removeGame(Game game) {
    String key = game.getGameId();
    databaseUserList.child(key).removeValue();
}
```

Figure 13 – remove current game in user list from the list

In addition to the three buttons located on the main list game profile screen there is one more called “meet up”.

2.7.11 Meet up

The meet up function incorporates a social element to the app. With meet up, the user can browse a list of active users subscribed to the app and invite them to play a specific game. Meet up can only be accessed through the user list on the game profile page, so a user must have added the game to their list before being able to utilise meet up.

When clicking on the meet up button, a list of users is presented to the active user. Clicking then on a user will activate the devices email application. The email will be populated automatically with the intended recipients email address, a relevant subject title relating to joining the user in a specific game and an appropriate body explaining in detail that the current user wishes to play a specific game with the user.

To accomplish this, upon clicking the meet up button, the app loads a new activity whilst passing the name of the game currently selected on to the following activity. In the MeetUsersActivity, the activity executes a similar method used by the main game list activity which populates the list, see figure 7. This populates the page with a list of all active users. By using the name of the game passed from the previous activity, the activity will store this name in a local variable. Clicking on a user name in the list will execute an email intent which opens the users email application. The email template is then fully populated with all fields automatically by using the target users email address and username as well the name of the game from the previous activity.

```
meetBtn.setOnClickListener((view) → {
        Intent intent = new Intent(getApplicationContext(), MeetUsersActivity.class);
        intent.putExtra("gameName", title);
        startActivity(intent);
});
```

Figure 14 – Meet up button opening meet up list and passing on name of current game through an intent

```
listViewMeet.setOnItemClickListener(new AdapterView.OnItemClickListener() {
    @Override
    public void onItemClick(AdapterView<?> parent, View view, int position, long id) {

        User userInstance = userList.get(position);
        String userEmailAddress = userInstance.getEmail();
        String userName = userInstance.getName();

        Intent emailIntent = new Intent(android.content.Intent.ACTION_SEND);
        emailIntent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
        emailIntent.setType("plain/text");
        emailIntent.setClassName("com.google.android.gm", "com.google.android.gm.ComposeActivityGmail");
        emailIntent.putExtra(android.content.Intent.EXTRA_EMAIL, new String[]{userEmailAddress});
        emailIntent.putExtra(android.content.Intent.EXTRA_SUBJECT, userName+" has invited you to play!");
        emailIntent.putExtra(android.content.Intent.EXTRA_TEXT, userName+" wants to play "+ gameName+" with you!");
        startActivity(emailIntent);

    }
});
```

Figure 15 – On click method setting up email for game invitation

2.7.12 Notifications

Notifications are performed through the firebase console. They are executed manually by referencing the release dates of games by date and releasing pop up notifications to users. The notifications include the name of the game, release date and other information about the game such as where they can be purchased and further information. All game release notifications are populated in the firebase console and are scheduled to deliver on the appropriate date.

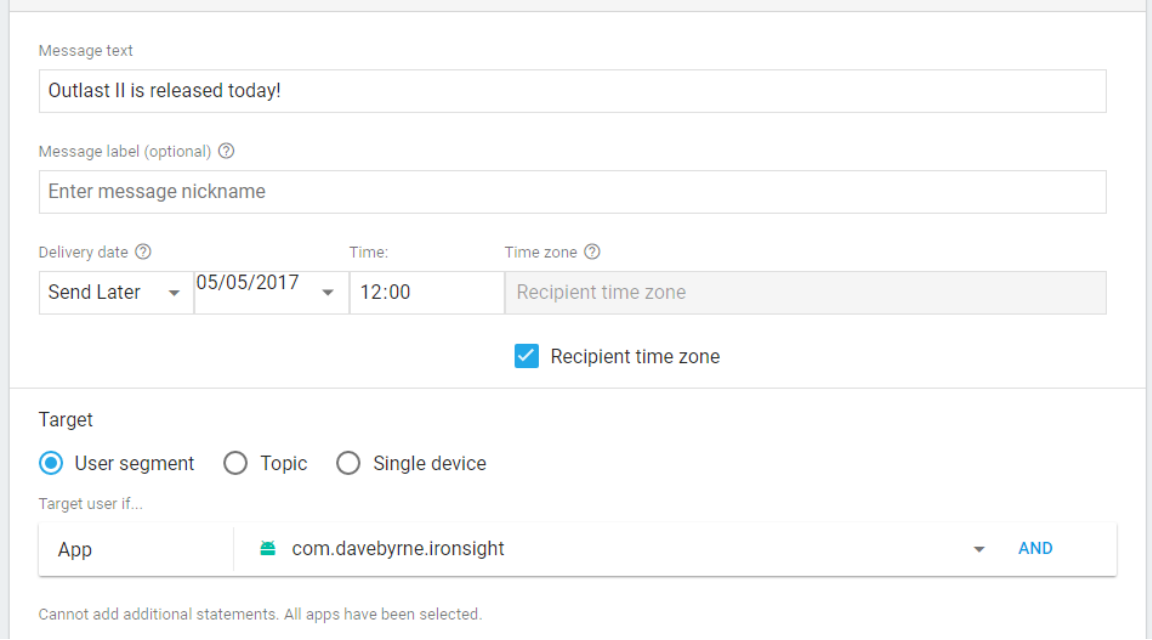
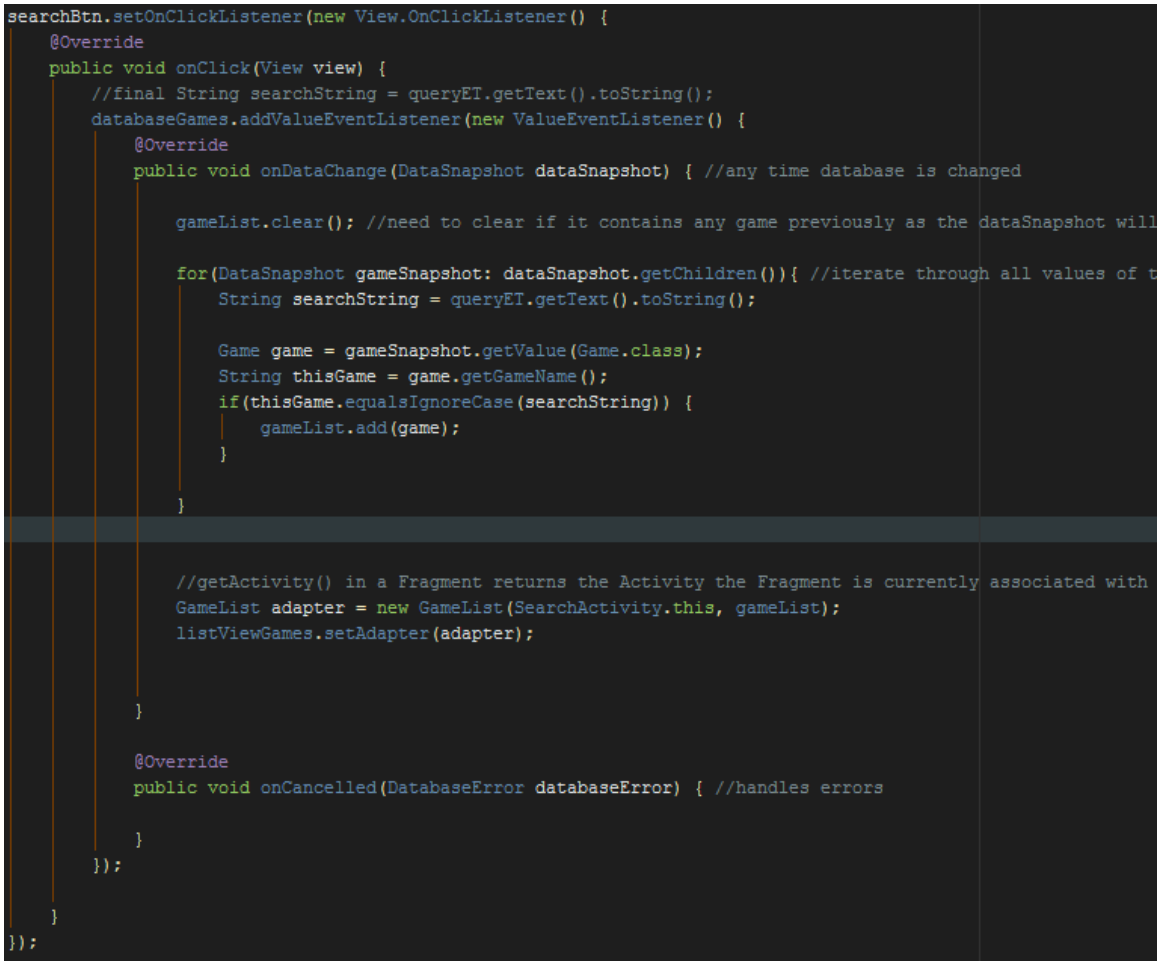


Figure 16 – Firebase interface for notification scheduling

2.7.13 Search

The search function is accessed through the magnifying glass at the top of the screen. When the user clicks the icon the app takes them to the search screen. The search screen consists of a text entry field and search button. The user enters the name of the game they wish to search for and clicks the search button to request the results of the search.

When the search button is pressed, the app will take the value in the search input field as a string and save it as a query variable. The app then iterates through the main list of games and checks if the query matches the name variable of each game. Once a match is made, the game object is added to the list view adapter and is shown as a list on the same page.



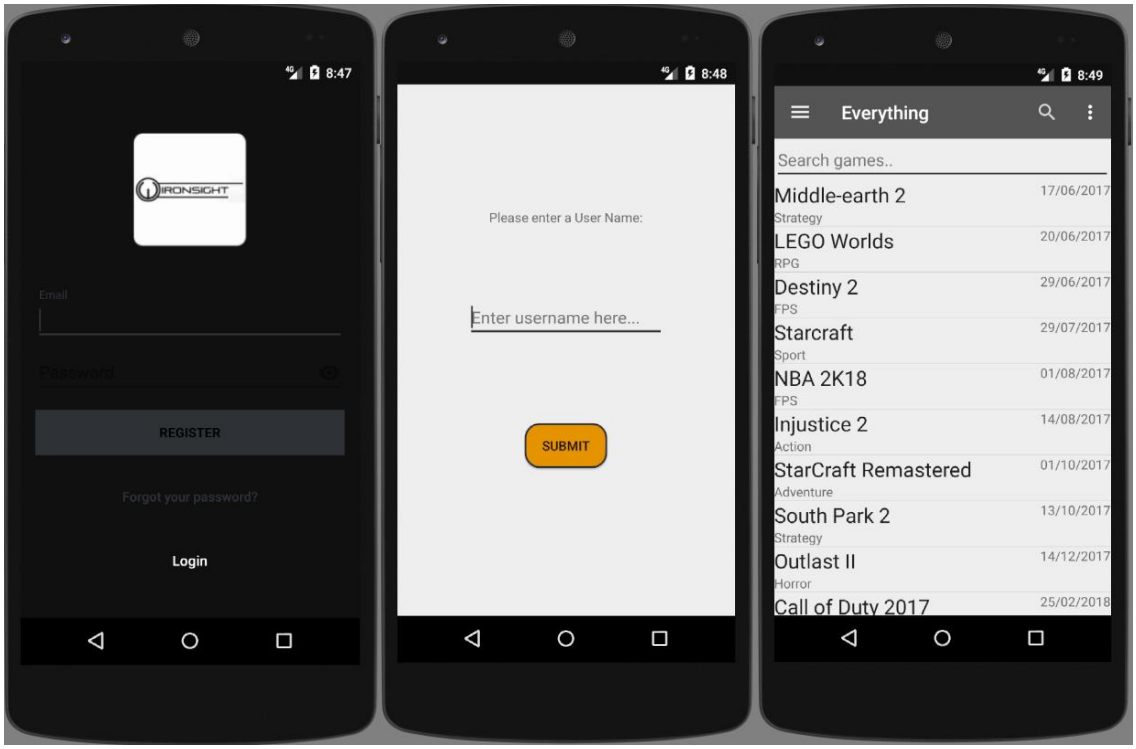
```
searchBtn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        //final String searchString = queryET.getText().toString();
        databaseGames.addValueEventListener(new ValueEventListener() {
            @Override
            public void onDataChange(DataSnapshot dataSnapshot) { //any time database is changed

                gameList.clear(); //need to clear if it contains any game previously as the dataSnapshot will

                for(DataSnapshot gameSnapshot: dataSnapshot.getChildren()){ //iterate through all values of t
                    String searchString = queryET.getText().toString();

                    Game game = gameSnapshot.getValue(Game.class);
                    String thisGame = game.getGameName();
                    if(thisGame.equalsIgnoreCase(searchString)) {
                        gameList.add(game);
                    }

                }



                //getActivity() in a Fragment returns the Activity the Fragment is currently associated with
                GameList adapter = new GameList(SearchActivity.this, gameList);
                listViewGames.setAdapter(adapter);


            }

            @Override
            public void onCancelled(DatabaseError databaseError) { //handles errors

            }
        });

    }
});
```

Figure 17 – Search iterating through all games and comparing to user input string for game name

2.8 Testing

Black Box Test 1			
AUT Name	Register user	**Version**	1.0
Iteration ID	1.0	**Date of Test**	20/04/17

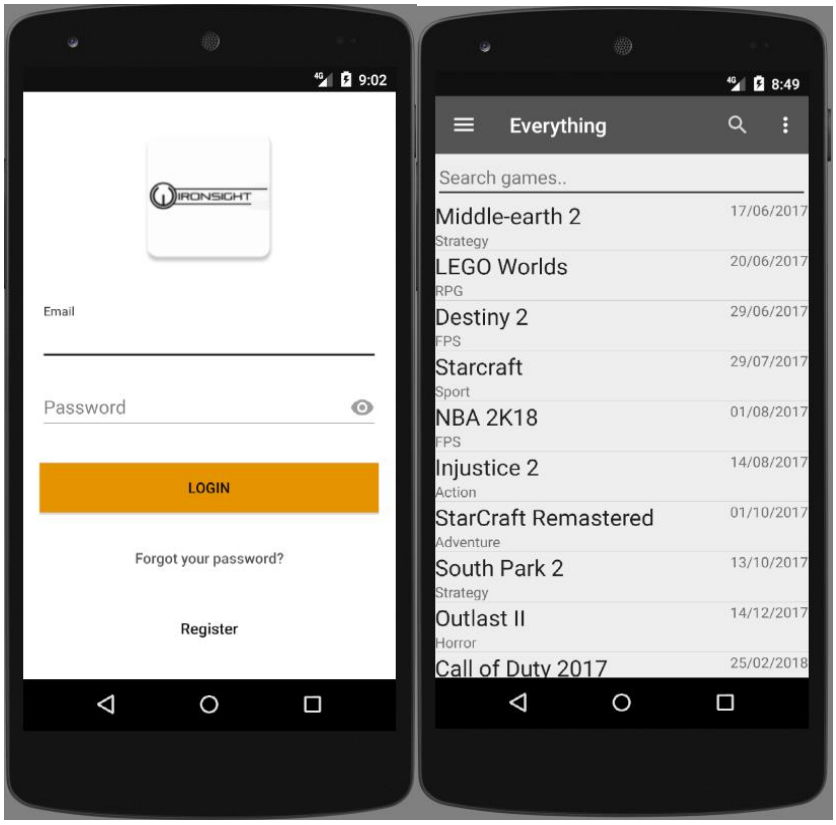
Test ID	BB1
Purpose of Test	The purpose of the test is to see if a user can register as a user using The form provided.
Test Environment	Hardware: Custom built PC running Windows 10 with active internet Connection.
Test Steps	Enter email address and password into the provided fields from the Registration page. The user then clicks the submit button to pass the Details on to the application. The user must then enter a user name And submit this through the use of a submit button.
Expected Result	The user is presented with a pop up message in the form of a toast Which confirms the creation of their new account and profile. The App then forwards the user to the main list page.
Actual result	The result complied with the expected result. The user was brought to the user name page after entering email and password and then To the main page after entering a user name. Each step accompanied By messages to confirm the actions were successful.
Suggested action	Test successful, no actions required.
Resolution	Test successful, no actions required.



Captions left to right: register page, user name page, and main page.

Black Box Test 2			
AUT Name	Login user	**Version**	1.0
Iteration ID	1.0	**Date of Test**	02/05/17

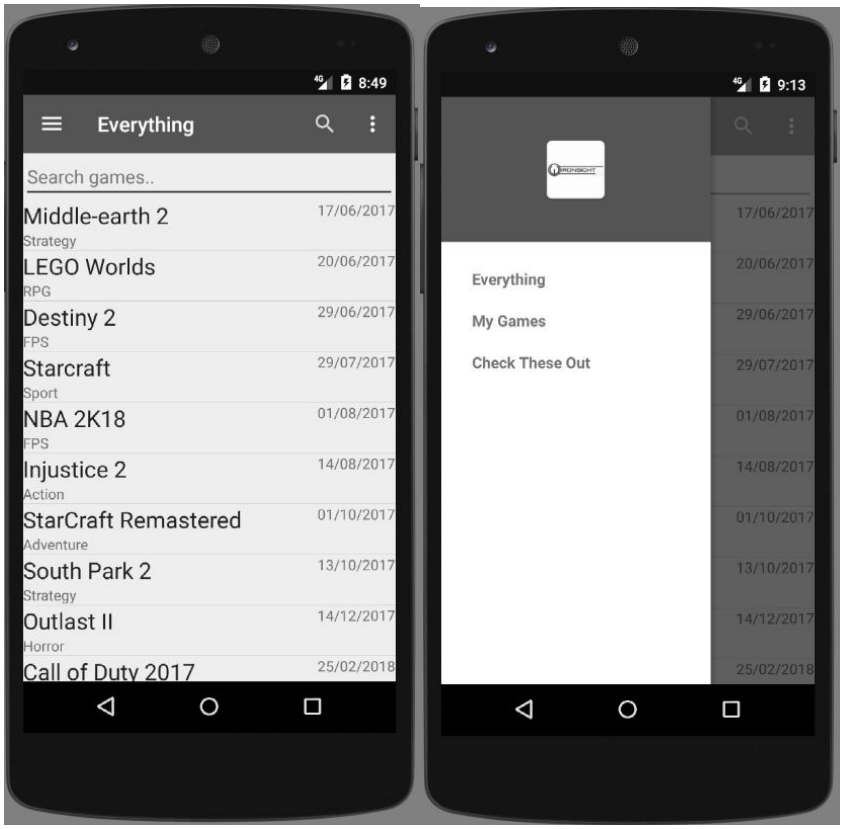
Test ID	BB2
Purpose of Test	To test if a user can log in to their account and so gain access to the Application.
Test Environment	Hardware: Custom built PC running Windows 10 with active internet Connection.
Test Steps	Enter email address and password into the provided fields from the Log in page. The user then clicks the submit button to pass the Details on to the application.
Expected Result	The user is presented with a pop up message in the form of a toast Which confirms the validity of their account details. The App then forwards the user to the main list page.
Actual result	The result complied with the expected result. The user was brought To the user name page after entering email and password. Each step Accompanied by messages to confirm the actions were successful.
Suggested action	Test successful, no actions required.
Resolution	Test successful, no actions required.



Captions left to right: login page, main page.

Black Box Test 3			
AUT Name	Access drawer menu	**Version**	1.0
Iteration ID	1.0	**Date of Test**	02/05/17

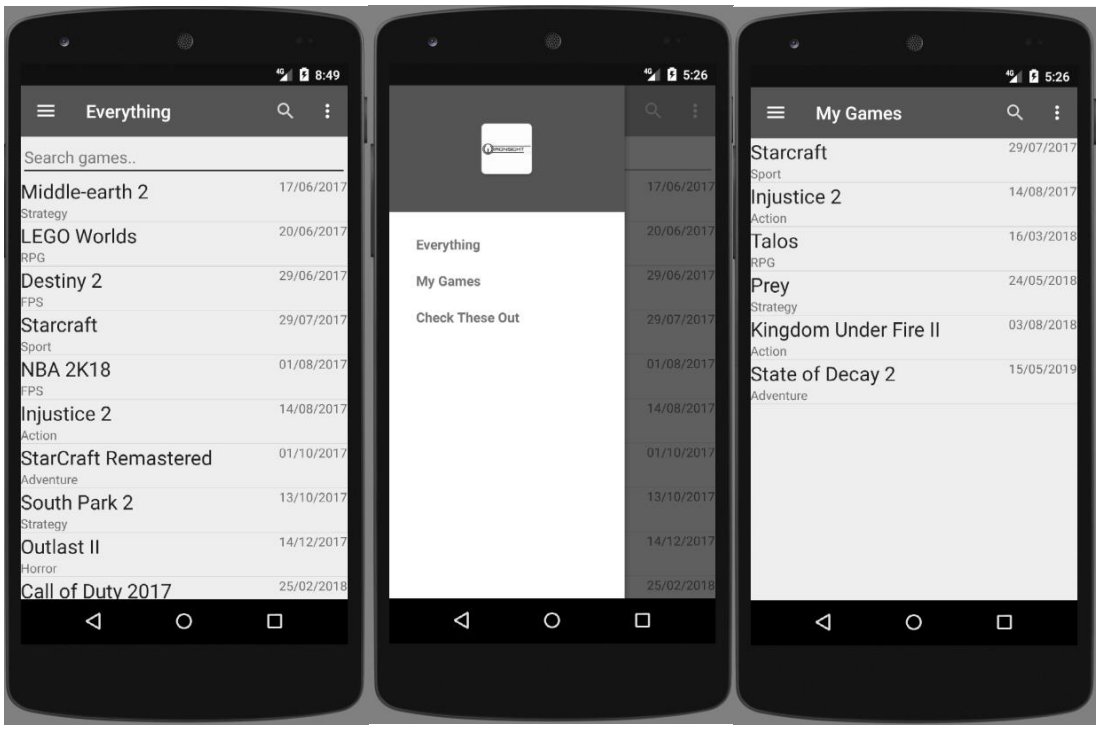
Test ID	BB3
Purpose of Test	To test if a user can open the drawer menu on the left of the main List, user list and suggested list pages from the top menu icon.
Test Environment	Hardware: Custom built PC running Windows 10 with active internet Connection.
Test Steps	The user must be on one of the three aforementioned pages. From Here the user presses the three parallel lines which represents the navigation menu.
Expected Result	Upon pressing the navigation button, the side menu drawer list slides across from the left side and displays the three navigation options.
Actual result	The result complied with the expected result. The user clicks the navigation button and the navigation menu slides across from the left.
Suggested action	Test successful, no actions required.
Resolution	Test successful, no actions required.



Captions left to right: main page, navigation menu.

Black Box Test 4			
AUT Name	Access My Games	**Version**	1.0
Iteration ID	1.0	**Date of Test**	02/05/17

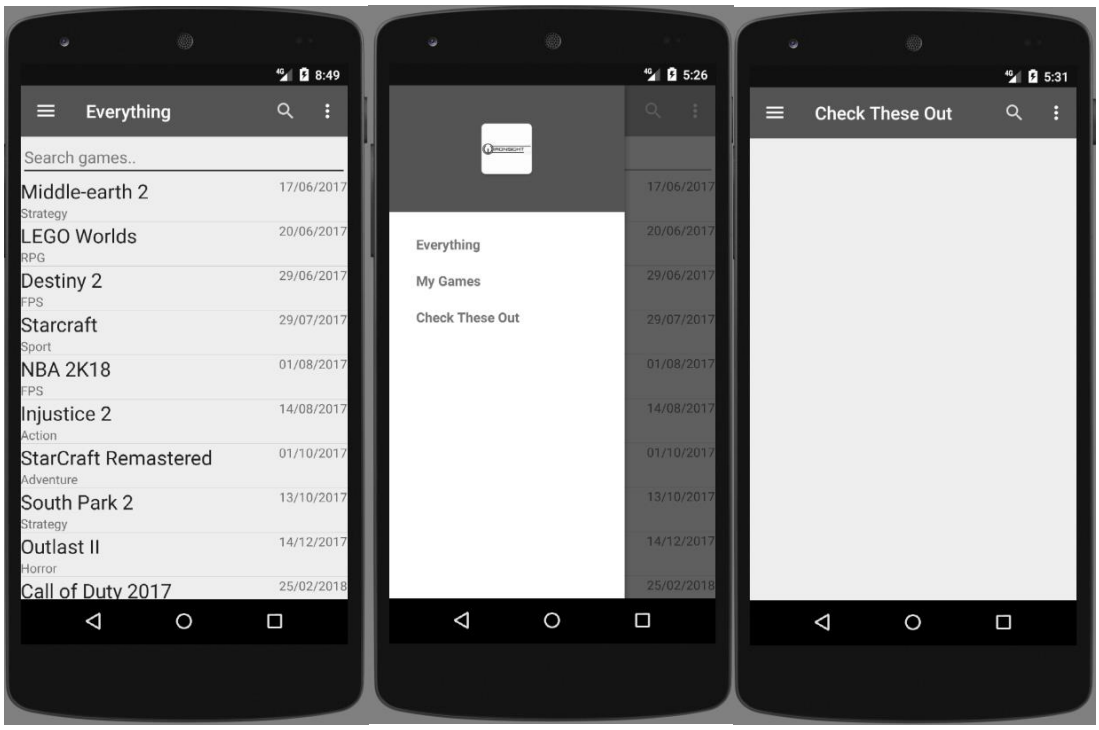
Test ID	BB4
Purpose of Test	To see if a user can access the My Games option from the navigation menu on the left of the screen.
Test Environment	Hardware: Custom built PC running Windows 10 with active internet connection.
Test Steps	From the main list menu, open the navigation menu from the button in the top left. From here click the My Games option.
Expected Result	Upon pressing the My Games option the user should be directed to a page listing their saved games.
Actual result	The result complied with the expected result. The user clicks the My Games option and is taken to their personal games list.
Suggested action	Test successful, no actions required.
Resolution	Test successful, no actions required.



Captions left to right: main page, navigation menu, My Games.

Black Box Test 5			
AUT Name	Access Suggested List	**Version**	1.0
Iteration ID	1.0	**Date of Test**	02/05/17

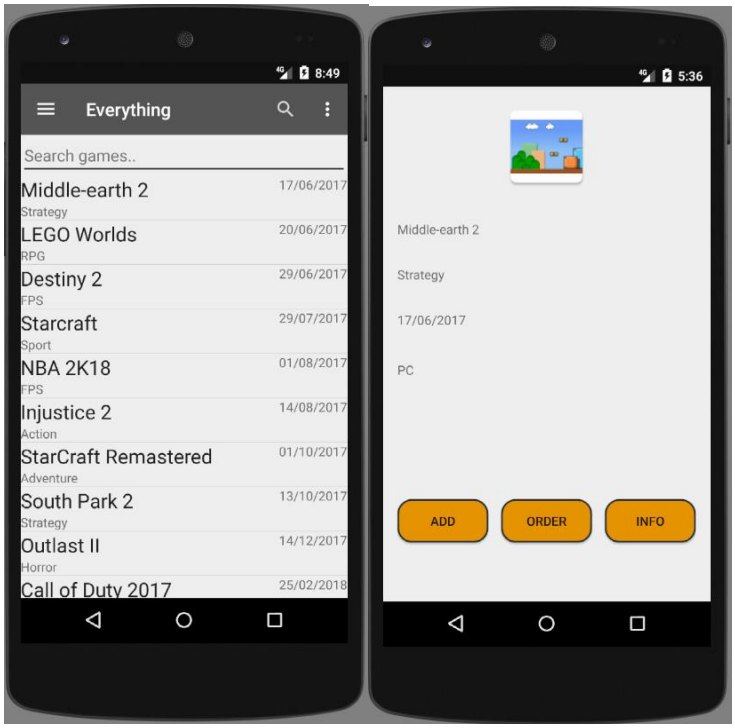
Test ID	BB5
Purpose of Test	To see if a user can access the Suggested Games option from the navigation menu on the left of the screen.
Test Environment	Hardware: Custom built PC running Windows 10 with active internet connection.
Test Steps	From the main list menu, open the navigation menu from the button in the top left. From here click the Suggested Games option.
Expected Result	Upon pressing the Suggested option the user should be directed to a page listing their saved games.
Actual result	The result complied with the expected result. The user clicks the Suggested Games option and is taken to their personal games list.
Suggested action	Test successful, no actions required.
Resolution	Test successful, no actions required.



Captions left to right: main page, navigation menu, Suggested Games.

Black Box Test 6			
AUT Name	Access Game Profile from Main List	**Version**	1.0
Iteration ID	1.0	**Date of Test**	02/05/17

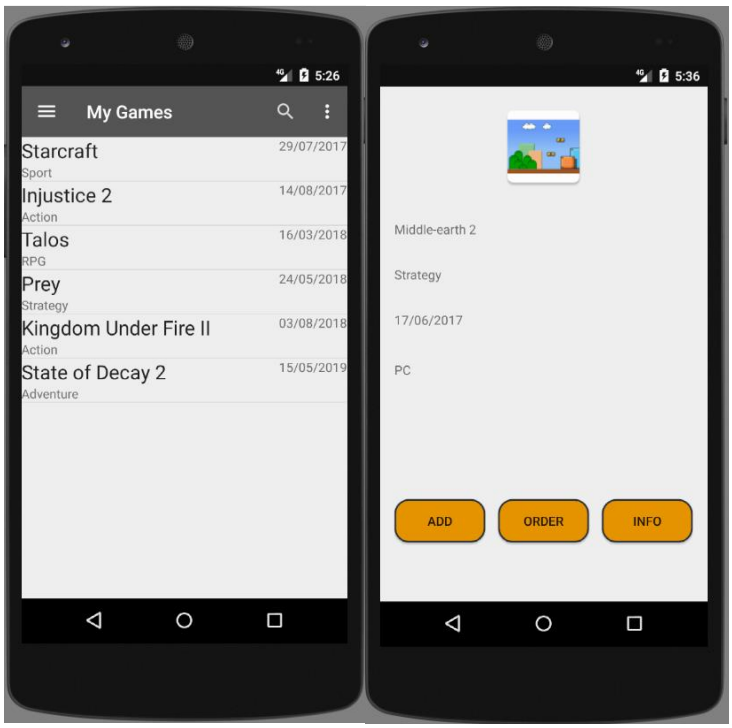
Test ID	BB6
Purpose of Test	To see if a user can access the Game profile page from main list screen.
Test Environment	Hardware: Custom built PC running Windows 10 with active internet connection.
Test Steps	From the main list screen, the user clicks on a game entry.
Expected Result	Upon pressing the game entry in the list on main list page, the user is directed to that particular games profile page.
Actual result	The result complied with the expected result. The user clicks the game from the list and is brought to that games profile page.
Suggested action	Test successful, no actions required.
Resolution	Test successful, no actions required.



Captions left to right: main page, game profile page.

Black Box Test 7			
AUT Name	Access Game Profile from User List	**Version**	1.0
Iteration ID	1.0	**Date of Test**	02/05/17

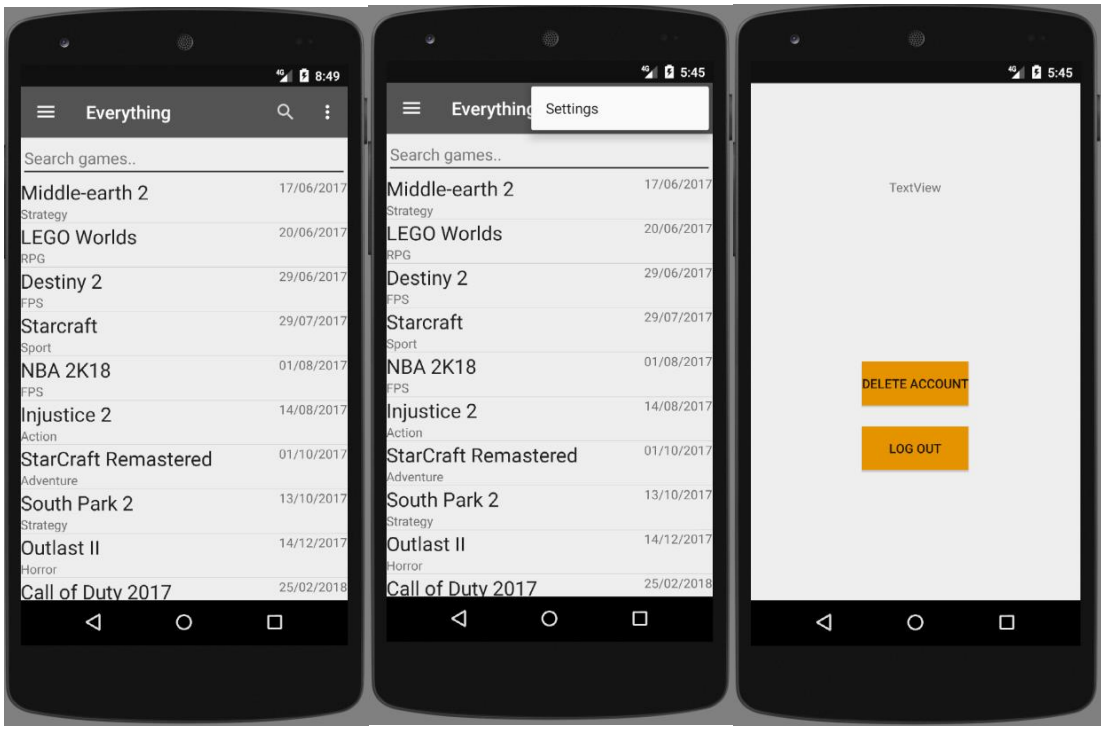
Test ID	BB7
Purpose of Test	To see if a user can access the Game profile page from user list screen.
Test Environment	Hardware: Custom built PC running Windows 10 with active internet connection.
Test Steps	From the user list screen, the user clicks on a game entry.
Expected Result	Upon pressing the game entry in the list on user list page, the user is directed to that particular games profile page.
Actual result	The result complied with the expected result. The user clicks the game from the list and is brought to that games profile page.
Suggested action	Test successful, no actions required.
Resolution	Test successful, no actions required.



Captions left to right: user page, game profile page.

Black Box Test 8			
AUT Name	Access User settings from main list	**Version**	1.0
Iteration ID	1.0	**Date of Test**	02/05/17

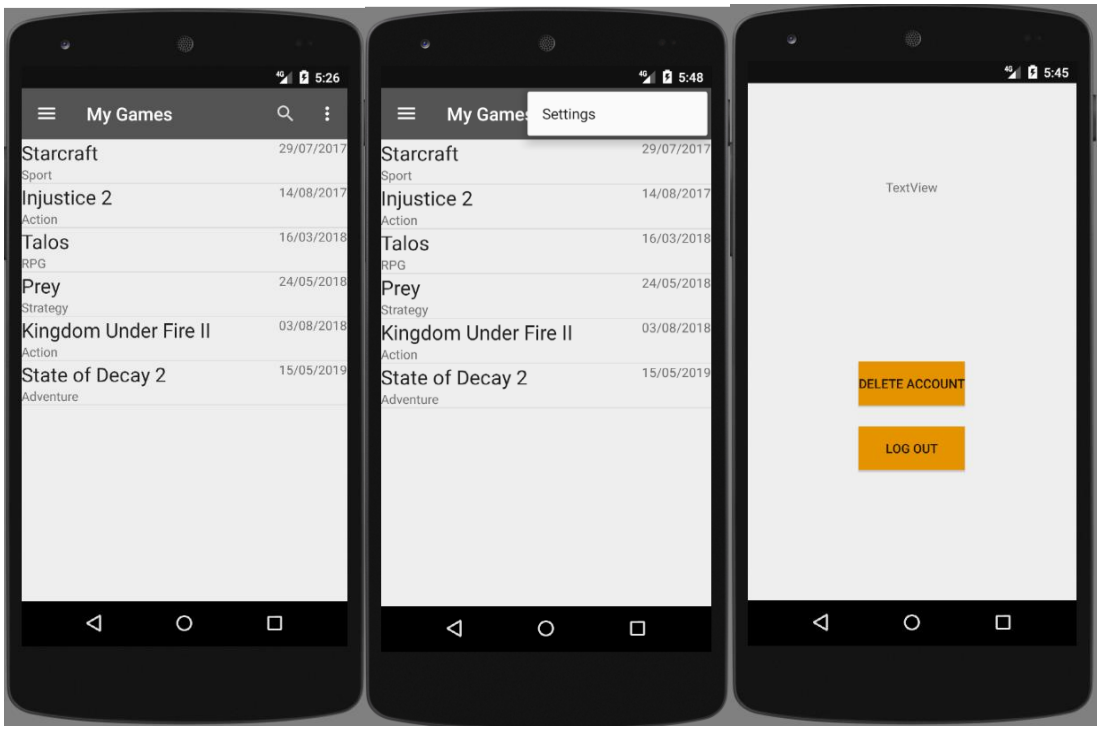
Test ID	BB8
Purpose of Test	To see if a user can access the user settings from the main list page.
Test Environment	Hardware: Custom built PC running Windows 10 with active internet connection.
Test Steps	From the main list page click the options button and then settings.
Expected Result	Upon pressing the settings option, the user is brought to the user settings page.
Actual result	The result complied with the expected result. Upon pressing the setting option, the user is taken to the settings page.
Suggested action	Test successful, no actions required.
Resolution	Test successful, no actions required.



Captions left to right: main list page, opened options button, settings page.

Black Box Test 9			
AUT Name	Access User settings from user list	**Version**	1.0
Iteration ID	1.0	**Date of Test**	02/05/17

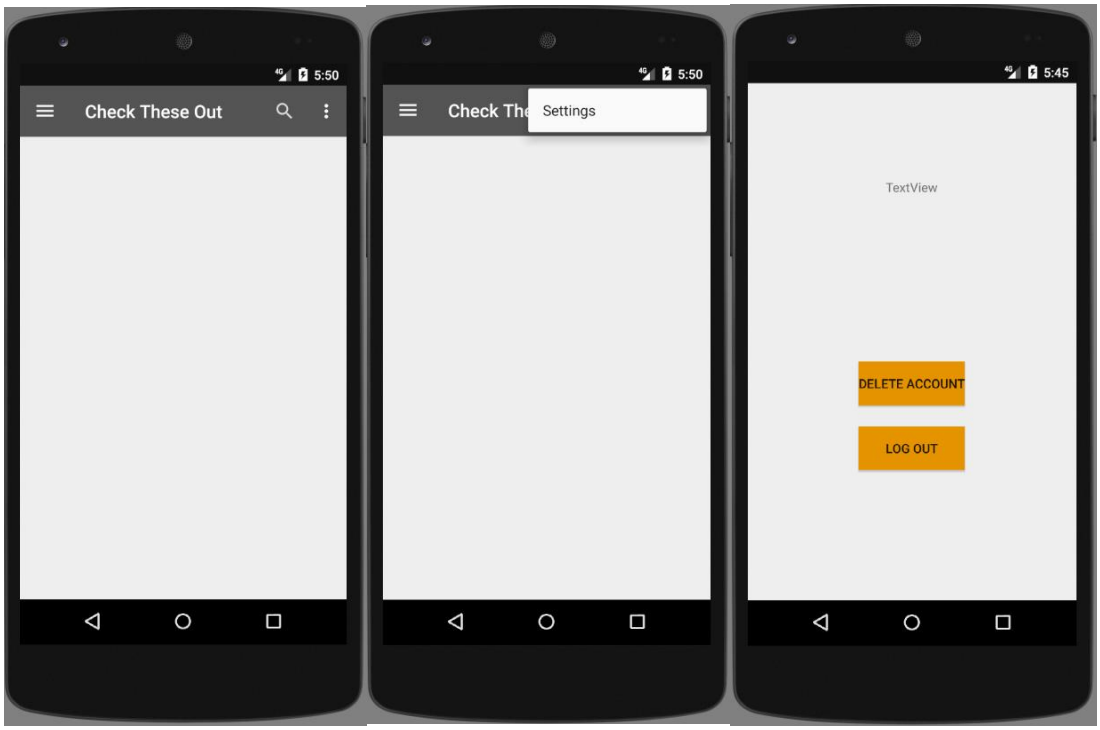
Test ID	BB9
Purpose of Test	To see if a user can access the user settings from the user list page.
Test Environment	Hardware: Custom built PC running Windows 10 with active internet connection.
Test Steps	From the user list page click the options button and then settings.
Expected Result	Upon pressing the settings option, the user is brought to the user settings page.
Actual result	The result complied with the expected result. Upon pressing the setting option, the user is taken to the settings page.
Suggested action	Test successful, no actions required.
Resolution	Test successful, no actions required.



Captions left to right: user list page, opened options button, settings page.

Black Box Test 10			
AUT Name	Access User settings from suggested list	**Version**	1.0
Iteration ID	1.0	**Date of Test**	02/05/17

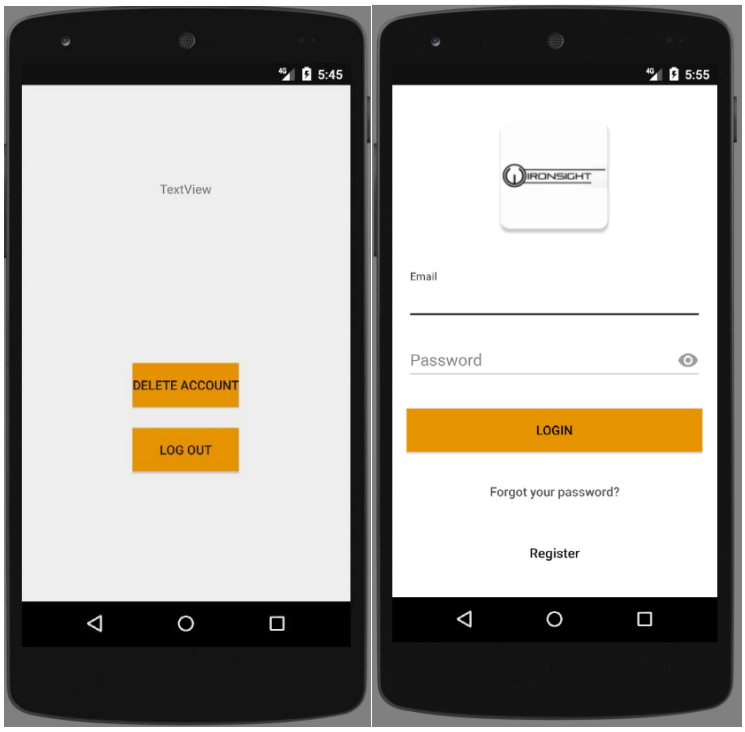
Test ID	BB10
Purpose of Test	To see if a user can access the user settings from the user suggested page.
Test Environment	Hardware: Custom built PC running Windows 10 with active internet connection.
Test Steps	From the suggested list page click the options button and then settings.
Expected Result	Upon pressing the settings option, the user is brought to the user settings page.
Actual result	The result complied with the expected result. Upon pressing the setting option, the user is taken to the settings page.
Suggested action	Test successful, no actions required.
Resolution	Test successful, no actions required.



Captions left to right: suggested list page, opened options button, settings page.

Black Box Test 11			
AUT Name	Log out user	**Version**	1.0
Iteration ID	1.0	**Date of Test**	03/05/17

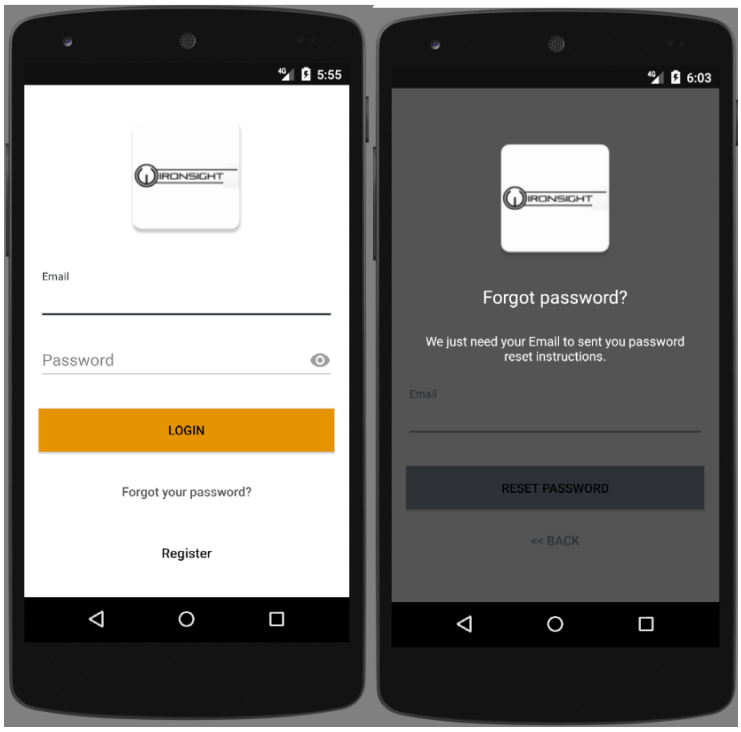
Test ID	BB11
Purpose of Test	To see if logging out will terminate session and return user to login page.
Test Environment	Hardware: Custom built PC running Windows 10 with active internet connection.
Test Steps	From the user settings page, the user clicks the log out button.
Expected Result	Upon pressing the log out button, the user session is terminated and the user is returned to the log in page.
Actual result	The result complied with the expected result. Upon pressing the log out button, the session is terminated and the user to taken back to the login in screen.
Suggested action	Test successful, no actions required.
Resolution	Test successful, no actions required.



Captions left to right: settings page, log in page.

Black Box Test 12			
AUT Name	Forgotten password	**Version**	1.0
Iteration ID	1.0	**Date of Test**	03/05/17

Test ID	BB12
Purpose of Test	To see if a user can reset their password from the “forgot password” link in login page.
Test Environment	Hardware: Custom built PC running Windows 10 with active internet connection.
Test Steps	From the log in page the user clicks “forgot your password?”, from the forgot password screen, enter the users email address and submit. Check user emails for reset password mail. In mail, click link and from the web page shown, enter a new password and save.
Expected Result	The user will be taken to the “forgot password” screen from the log in page. After having entered the users email and sending, the user receives an email to reset their password. Clicking the link in the email will bring the user to the reset password web page where a new password can be entered and saved.
Actual result	The result complied with the expected result. All steps outlined above were passed and the user has reset their password.
Suggested action	Test successful, no actions required.
Resolution	Test successful, no actions required.



noreply@ironsight-afcad.firebaseapp.com

to me

Hello,

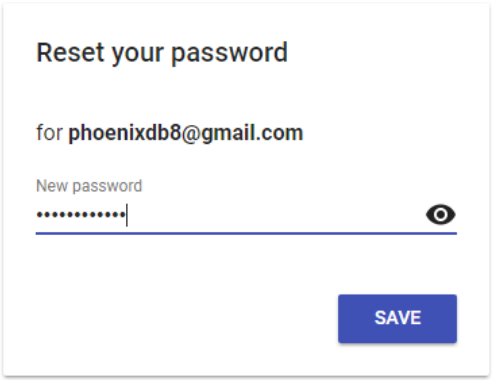
Follow this link to reset your IronSight password for your phoenixdb8@gmail.com account.

https://ironsight-afcad.firebaseapp.com/__/auth/action?mode=resetPassword&oobCode=mKi

If you didn't ask to reset your password, you can ignore this email.

Thanks,

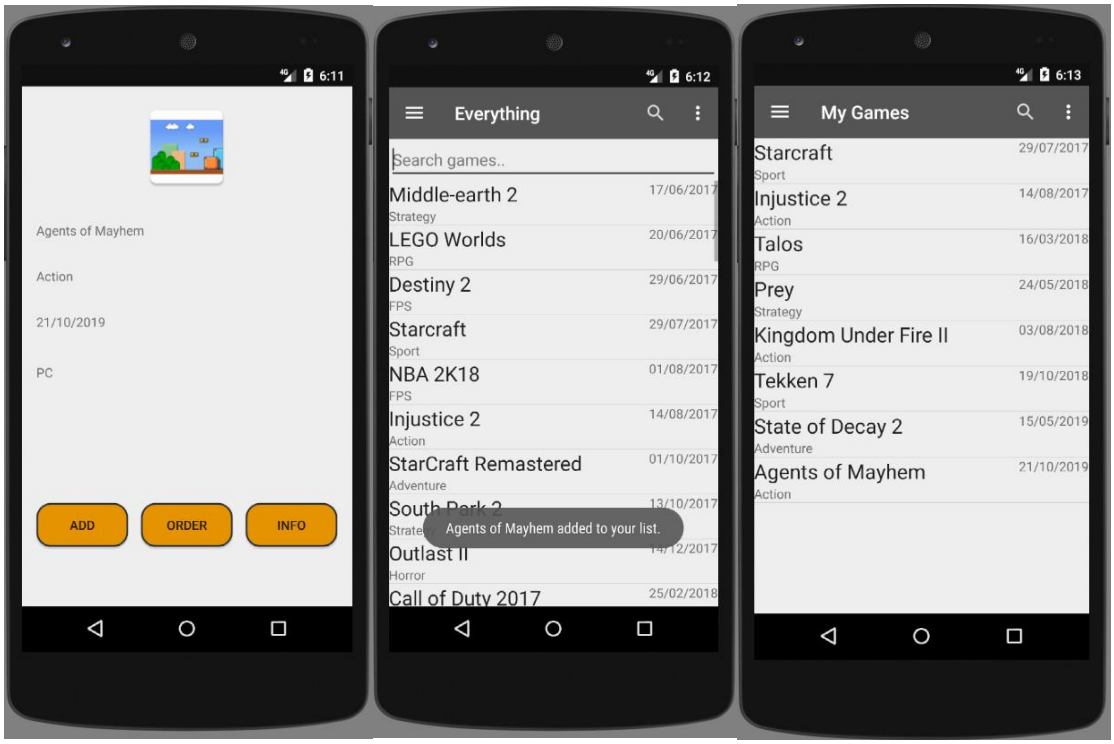
Your IronSight team



Captions top to bottom: login page, forgot password page, email to user, reset password web page.

Black Box Test 13			
AUT Name	Add game to user list	**Version**	1.0
Iteration ID	1.0	**Date of Test**	03/05/17

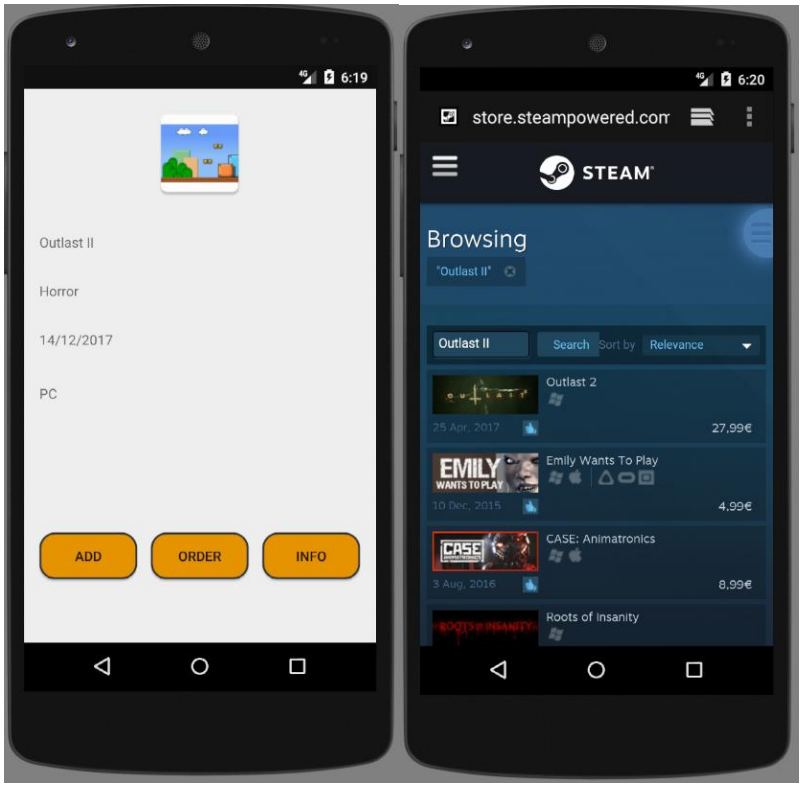
Test ID	BB13
Purpose of Test	To see if a user can add a game from the game profile page in main list to their own list.
Test Environment	Hardware: Custom built PC running Windows 10 with active internet connection.
Test Steps	From the game profile page in the main list of games, the user clicks the “add” button.
Expected Result	The user is shown an on screen message to say that the game has been added to their list and the user is brought to the main list page immediately after pressing the button. Navigating to the user list will show the new entry from main list.
Actual result	The result complied with the expected result. All steps outlined above were passed.
Suggested action	Test successful, no actions required.
Resolution	Test successful, no actions required.



Captions left to right: game profile page, game added message, user list page.

Black Box Test 14			
AUT Name	Using Order button from main list game profile page.	**Version**	1.0
Iteration ID	1.0	**Date of Test**	03/05/17

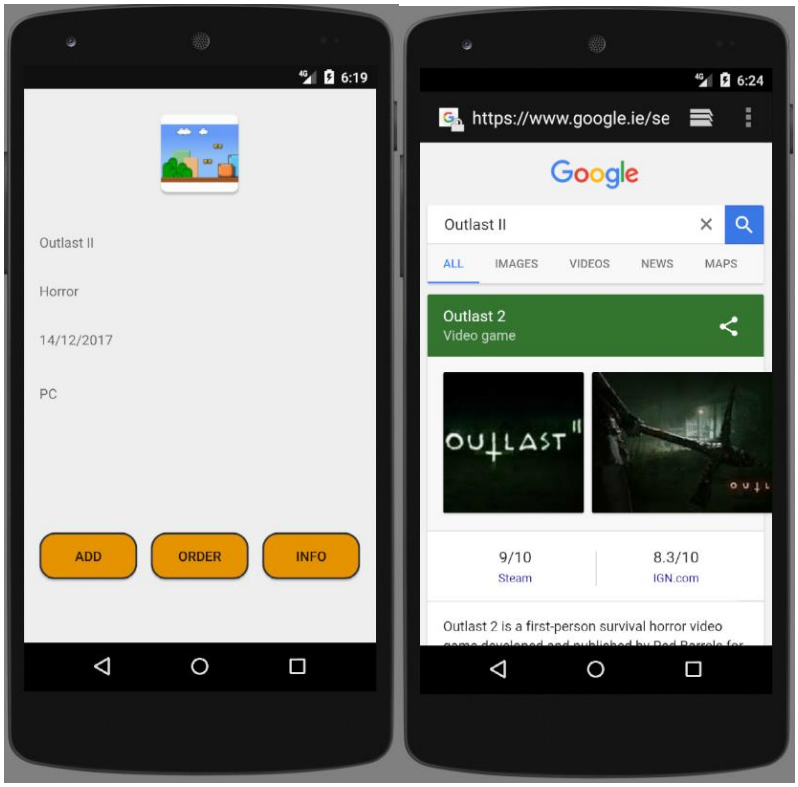
Test ID	BB14
Purpose of Test	To see if the order button will successfully take the user to the retailer page for that particular game.
Test Environment	Hardware: Custom built PC running Windows 10 with active internet connection.
Test Steps	From the game profile page in the main list of games, the user clicks the Order button.
Expected Result	The app will load a web page for that particular games retail page from game retailer Steam.
Actual result	The result complied with the expected result. All steps outlined above were passed.
Suggested action	Test successful, no actions required.
Resolution	Test successful, no actions required.



Captions left to right: game profile page from main list, steam purchase/ pre-purchase page.

Black Box Test 15			
AUT Name	Using Info button from main list game profile page.	**Version**	1.0
Iteration ID	1.0	**Date of Test**	04/05/17

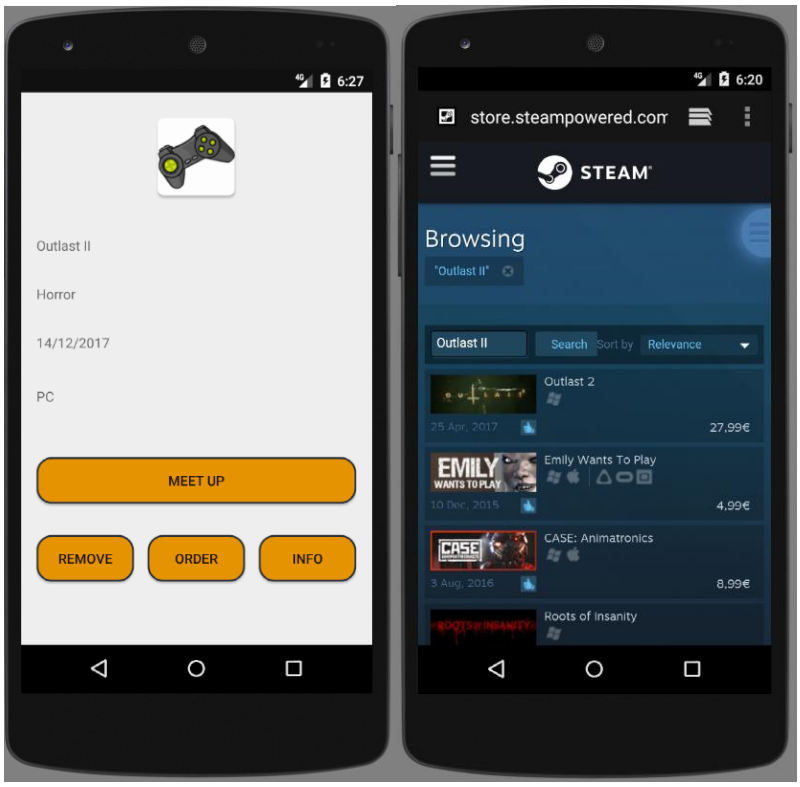
Test ID	BB15
Purpose of Test	To see if the info button will successfully take the user to an internet search of that game, display google summary of the game from the main list.
Test Environment	Hardware: Custom built PC running Windows 10 with active internet connection.
Test Steps	From the game profile page in the main list of games, the user clicks the Info button.
Expected Result	The app will load a web page for that particular games search results, the top of this page should display a summary of information for that game.
Actual result	The result complied with the expected result. All steps outlined above were passed.
Suggested action	Test successful, no actions required.
Resolution	Test successful, no actions required.



Captions left to right: game profile page from main list, search result for the game.

Black Box Test 16			
AUT Name	Using Order button from user list game profile page.	**Version**	1.0
Iteration ID	1.0	**Date of Test**	04/05/17

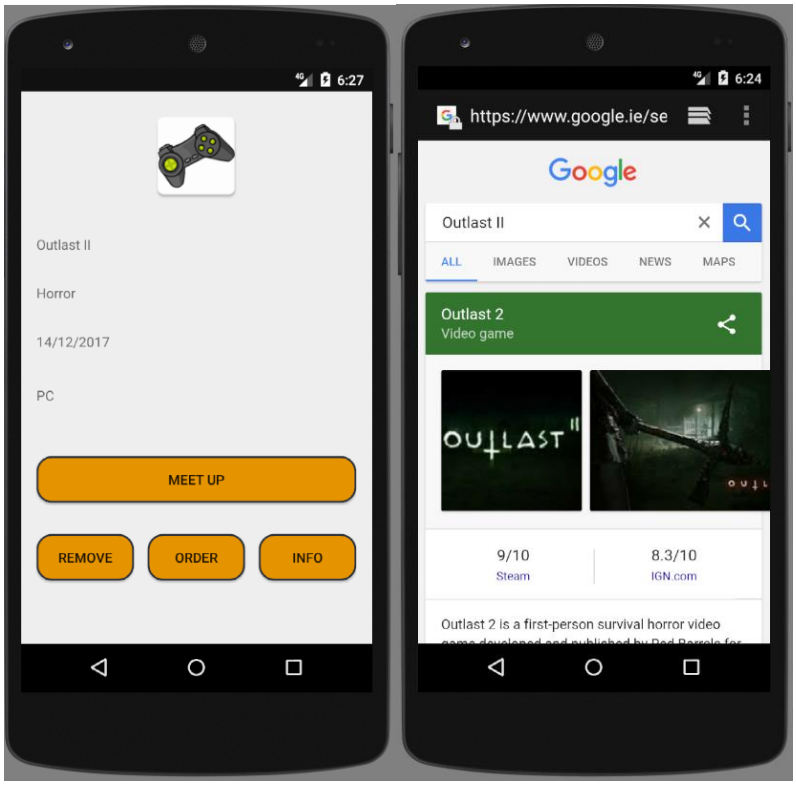
Test ID	BB16
Purpose of Test	To see if the order button will successfully take the user to the retailer page for that particular game.
Test Environment	Hardware: Custom built PC running Windows 10 with active internet connection.
Test Steps	From the game profile page in the user list of games, the user clicks the Order button.
Expected Result	The app will load a web page for that particular games retail page from game retailer Steam.
Actual result	The result complied with the expected result. All steps outlined above were passed.
Suggested action	Test successful, no actions required.
Resolution	Test successful, no actions required.



Captions left to right: game profile page from user list, steam purchase/ pre-purchase page.

Black Box Test 17			
AUT Name	Using Info button from user list game profile page.	**Version**	1.0
Iteration ID	1.0	**Date of Test**	04/05/17

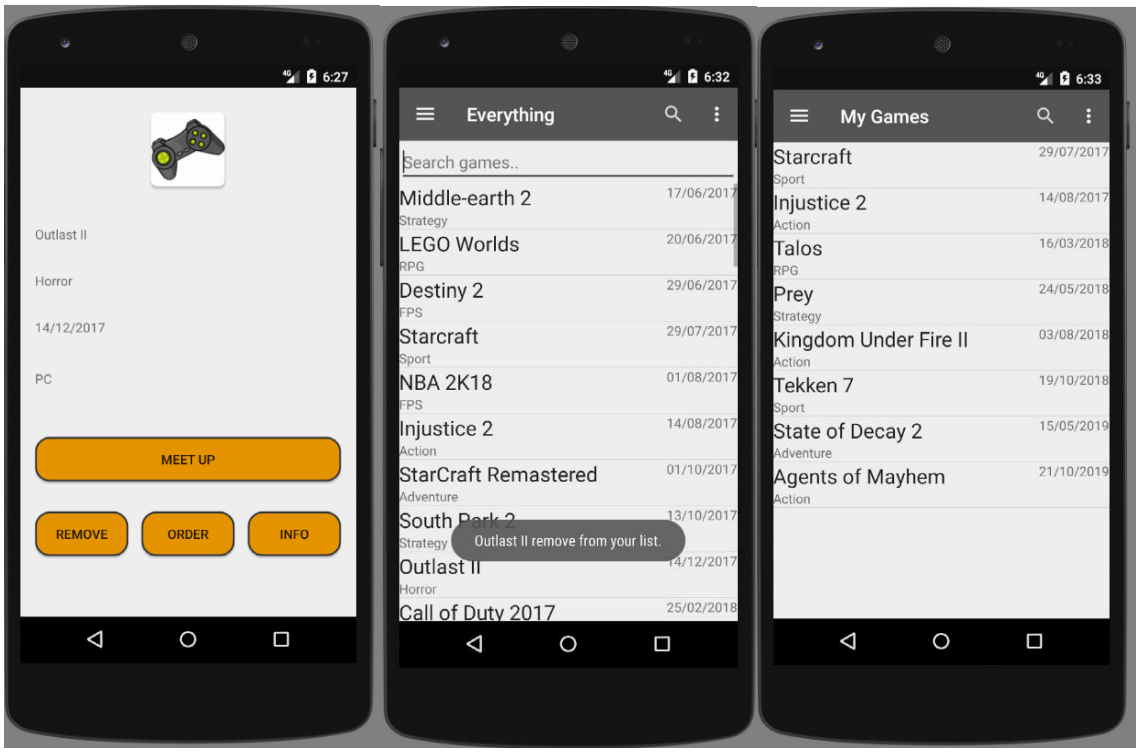
Test ID	BB17
Purpose of Test	To see if the info button will successfully take the user to an internet search of that game, display google summary of the game from the user list.
Test Environment	Hardware: Custom built PC running Windows 10 with active internet connection.
Test Steps	From the game profile page in the user list of games, the user clicks the Info button.
Expected Result	The app will load a web page for that particular games search results, the top of this page should display a summary of information for that game.
Actual result	The result complied with the expected result. All steps outlined above were passed.
Suggested action	Test successful, no actions required.
Resolution	Test successful, no actions required.



Captions left to right: game profile page from user list, search result for the game.

Black Box Test 18			
AUT Name	Add game to user list	**Version**	1.0
Iteration ID	1.0	**Date of Test**	04/05/17

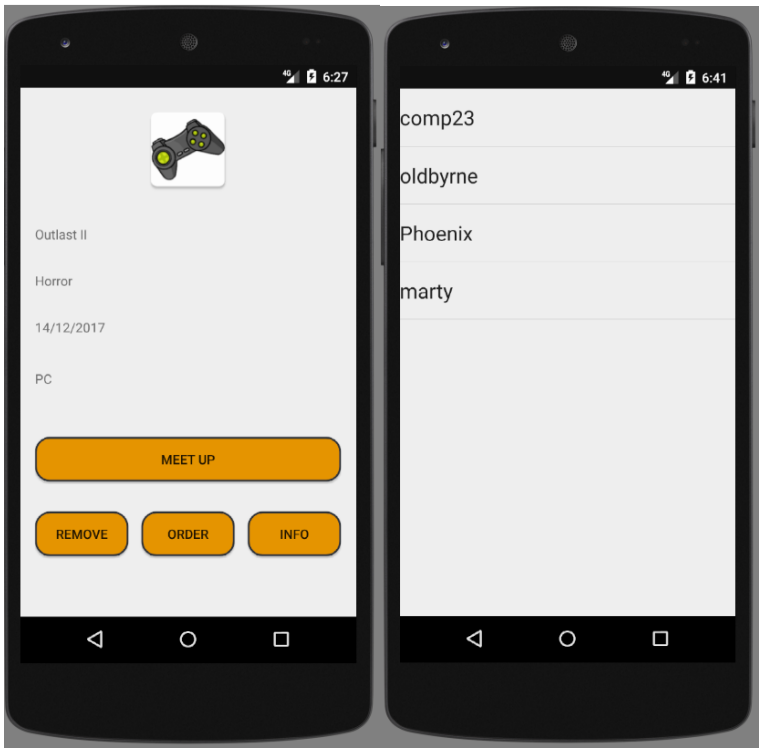
Test ID	BB18
Purpose of Test	To see if a user can remove a game from the game profile page in user list from their own list.
Test Environment	Hardware: Custom built PC running Windows 10 with active internet connection.
Test Steps	From the game profile page in the user list of games, the user clicks the “remove” button.
Expected Result	The user is shown an on screen message to say that the game has been removed from their list and the user is brought to the main list page immediately after pressing the button. Navigating to the user list will show the entry gone from user list.
Actual result	The result complied with the expected result. All steps outlined above were passed.
Suggested action	Test successful, no actions required.
Resolution	Test successful, no actions required.



Captions left to right: game profile page from user list, message showing game removed, updated user list.

Black Box Test 19			
AUT Name	Request to “meet up” with another user to play a specific game.	**Version**	1.0
Iteration ID	1.0	**Date of Test**	04/05/17

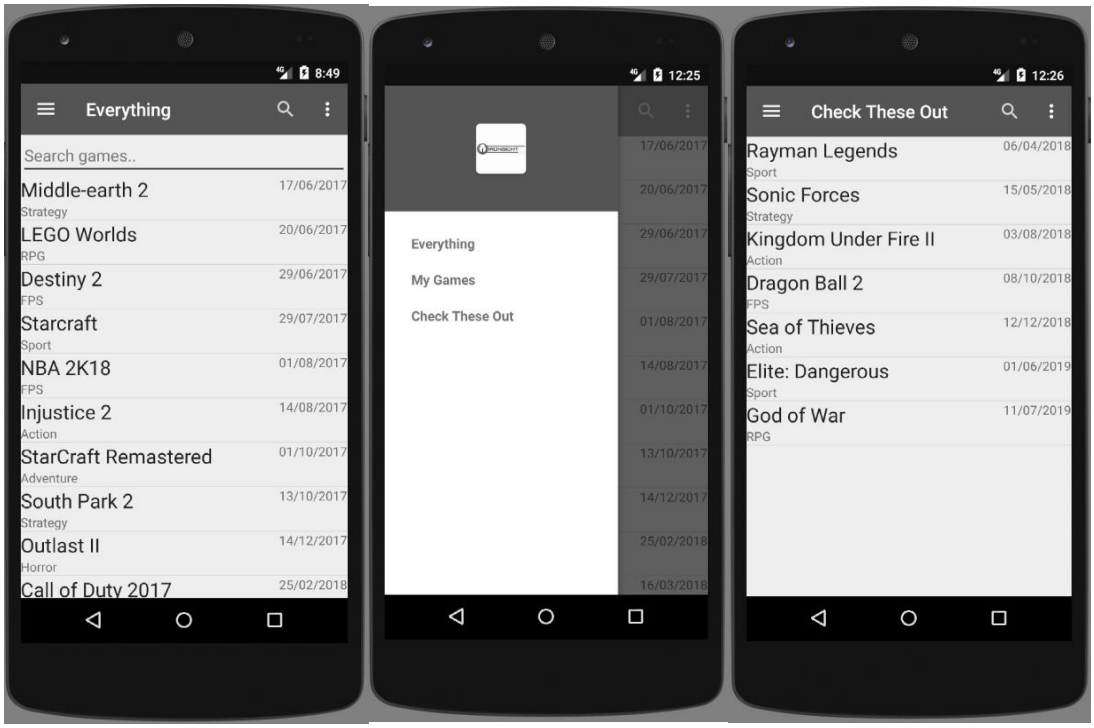
Test ID	BB19
Purpose of Test	To see if a user can request to meet up with another user to play a certain game.
Test Environment	Hardware: Custom built PC running Windows 10 with active internet connection.
Test Steps	From the game profile page in the user list of games, the user clicks the “meet up” button. The user is then directed to a list of users, click a user and the email application of the phone will launch. Press send on automatically populated email to send to user.
Expected Result	The user clicks on the meet up button and it navigates them to a page with a list of active users. From this page, the user clicks a user and it opens up the users email application where a pre-written email is created. The email includes the targeted users email address, a generated email subject with the active user’s username and game they wish to play. The body includes the same information as the subject only with slightly more information.
Actual result	The result complied with the expected result. All steps outlined above were passed.
Suggested action	Test successful, no actions required.
Resolution	Test successful, no actions required.



Captions left to right: game profile page from user list, the list of active users available to meet with.

Black Box Test 20			
AUT Name	Access “Check these out” list	**Version**	1.0
Iteration ID	1.0	**Date of Test**	04/05/17

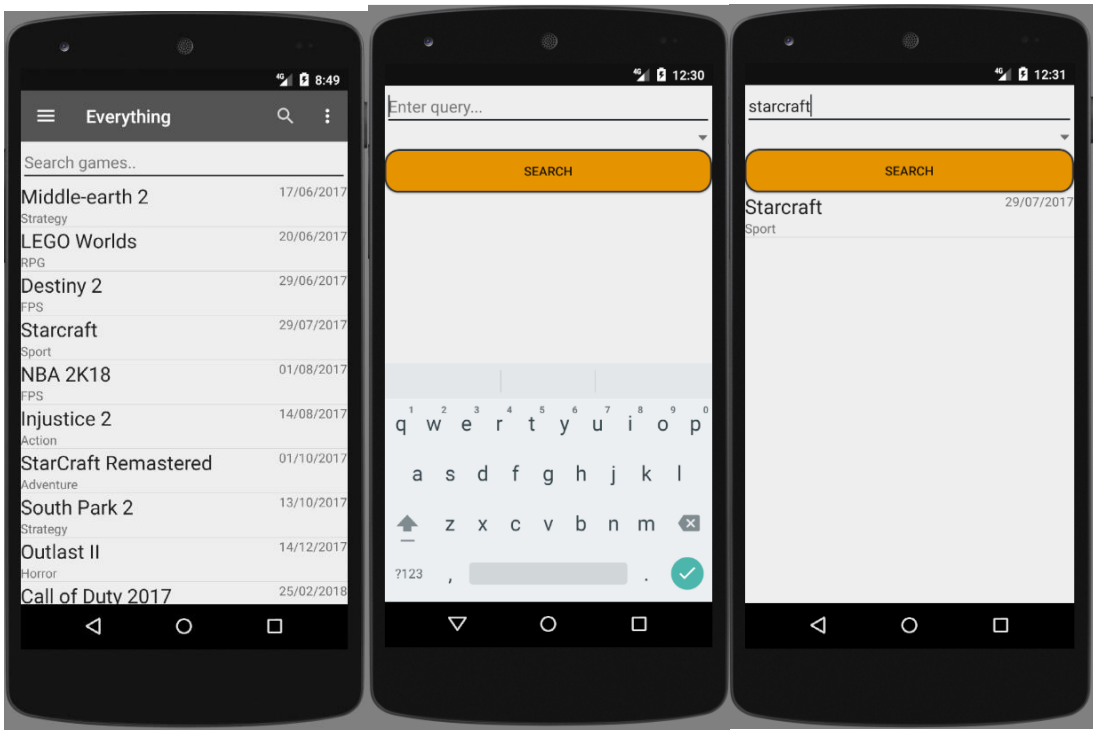
Test ID	BB20
Purpose of Test	To see if the app will generate a variety of randomized suggestions for the user in the check these out list.
Test Environment	Hardware: Custom built PC running Windows 10 with active internet connection.
Test Steps	From the main list the user must open the navigation drawer and click on the check these out option.
Expected Result	Upon clicking the check these out option, the user is taken to the check these out list.
Actual result	The result complied with the expected result. All steps outlined above were passed.
Suggested action	Test successful, no actions required.
Resolution	Test successful, no actions required.



Captions left to right: game profile page from main list, options drawer, check these out list.

Black Box Test 21			
AUT Name	Use search function on main list	**Version**	1.0
Iteration ID	1.0	**Date of Test**	04/05/17

Test ID	BB21
Purpose of Test	To see if the search function works on the main list page.
Test Environment	Hardware: Custom built PC running Windows 10 with active internet connection.
Test Steps	From the main list page, click the magnifying glass on the top right of the screen. From the search screen, enter a name of a game and then click the search button.
Expected Result	Clicking the mag glass will bring the user to the search page. After having entered a game name and pressing search, the user is shown a list of matching queries.
Actual result	The result complied with the expected result. All steps outlined above were passed.
Suggested action	Test successful, no actions required.
Resolution	Test successful, no actions required.



Captions left to right: game profile page from main list, search page, search results.

Describe any testing tools, test plans and test specifications used in the project

2.9 Evaluation

2.9.1 Heuristic Evaluation

Heuristic evaluation is a simple and easy way of evaluating the user interface of the application. The process is that of analysing the interface with regards to a set of defined headings. This evaluation will be carried out by a single individual.

2.9.1.1 Visibility of system status

The interface is all fairly self-explanatory. The first page I visited was the log in screen upon starting up the application. As I hadn't an account at that point, I naturally looked for a register option. The register button was at the bottom of the relatively clutter free page so I found it quite quickly. Having come on to the registration page, I also found it to be fairly self-explanatory. I was required to enter an email address followed by a password. I then submitted my details and was then prompted to enter a user name followed by a submit button. With this completed, I was brought to what seemed to be the main page of the site. Prior to this however I was flashed with a "registration

successful" message on screen. Before proceeding, I'll note that when registering my password, the app also flashed a messaged saying that it was too short so I entered a new one.

At the main page, it all seemed very familiar. The universal icon buttons for settings, menu and searching were all at the top navigation bar. The page itself had a list of games on it which were not ordered by name but by release date which was to the right of the name. I thought that made sense as the app aims to help users follow new releases.

Having clicked on an entry in the list, I was brought to a game profile page where there was more information about the game as well as a few button to interact with. The information given was the name of the game, the genre of it, the release date and platform. The buttons said Add, Order and Info. I clicked on Add and a message flashed up that the game was now added to my list. I then clicked on the Order and Info buttons and they brought me to a retailer site for that game and an information site about it. I then backed out of the game profile with my phones back button.

I then decided to explore the navigation options. I clicked on the 3 bars which usually represent navigation and a navigation area slid across the screen from the left. There were three options here, one of which I was already at, everything. The other two were My Games and Check These Out. I clicked on my games. The page was very similar to the "everything" page but only had a single list entry on it which was the game I had added a moment ago. I clicked on the entry and was once again brought to the profile page. Though now there was a fourth button "meet up". I clicked meet up, not being sure what it was meant to do and a list of users popped up. Having clicked a user, my email application opened and a readymade email to the user requesting to play my added game appeared. Without sending the request, I backed on to the app again. Here I clicked the Remove button which replaced the add button from earlier and I was told the game was now removed from my list and I was immediately redirected to the main page again.

Back on the navigation menu, I went to the Check These Out option. Here I was shown another list with seemingly randomised games though still ordered by release date. I assume these are titles that are suggested to me to investigate. The profile on these behaved the same as in the main list of "everything".

From the main page again, I decided to try the search button. When I pressed it I was taken to a search page. There was a prompt at the top of the page asking me to enter a query. So I entered the name of an upcoming game I knew about already and hit the search button. The game appeared as a list element below the search button and I clicked it. This action brought up the game profile screen again. I then backed to the main page again using the phones back button.

I then pressed the settings button and was brought to another page where I could log out. I did so and was taken back to the log in page. This page also asked "forgot your password?" I clicked this and was taken to a screen where I could enter my email. I entered it and received an email from IronSight to reset my password. I clicked the link

and was taken to a web page where I could enter a new password. Having done so and clicked save, the site informed me of a successful change.

2.9.1.2 Match between the system and the real world

I found all the language used in the app to be very simple and straightforward. Any problems with registry or failed logins were handled this way, by using friendly simple phrasing. Any time that an event occurred that seemed relevant to me and me knowing about, there was a little message that popped up for a second to tell me about. Overall navigating the site was a very non-traumatic experience. Any time I click a button, not only am I told what it has done (or failed to do), I am redirected to the next page that I would expect to be navigated to.

2.9.1.3 User freedom and control

Given the straightforward idea behind this app, there is little opportunity to take a wrong turn. All pages load extremely quickly so an accidental click navigating somewhere unwanted is quickly undone. The only action that may require some time to undo would be accidently adding a game to the personal list as you would then need to navigate to the list, find the game, go into its profile page and then remove. This could be fixed with a confirmation dialog when adding a game but I think this would affect users more often than accidently clicking the add button, particularly experienced users. Most pages can be accessed from the top navigation and side navigation bars which are almost always present, apart from game profile pages and meet up list.

2.9.1.4 Consistency and standards

All labels, links and buttons were using standard naming and conventions. Buttons and links were all labelled clearly and I do not recall any confusion as to what their functionality was. Meet up was initially a mystery but having explored its functionality, it became completely clear very quickly. I think there was a definite effort to maintain a simplistic and intuitive design when the app was being developed.

2.9.1.5 Error Prevention

The only errors that I encountered were when adding or removing games from the personal list. While the action performed properly 90% of the time, there were still instances where the app would crash. Any other errors occurred during registration where details were not formatted correctly like the password or the email was not in the correct format. Though these are handled with error messaging and I simply had to retry with the correct format.

3 Conclusions

This report outlines how useful and flexible the Android Studio environment and Firebase online services can be. The IronSight gaming application is designed for gaming enthusiasts to manage a constantly growing gaming world. Where new games are constantly being released and many can slip under the radar. IronSight offers its users the ability to catalogue their own interests and also find likeminded people in the process. The gaming community is one that is constantly expanding and IronSight aims to expand with it.

The format of how Android Studio operates was a challenge in that it is unlike anything I had undertaken before. The system of activity, fragment and intent was something I had never come across previously but quickly made sense as something that works very well with mobile app development. Designing the layout for UI pages was done using XML which was also entirely new to me and something that needed practice but again is something that now makes complete sense. Linking these layout files to their java coded activities became second nature as time went on.

Using adapters to populate lists and listeners so the app is aware of any passive changes as well as processing button clicks was very useful for a lot of the apps functionality. But again not something I had used before. Using all of these new and interesting features along with the things I already knew about java programming opened up a lot of possibilities during development which caused slight deviations from the original proposal. I could think that these deviations are a bad thing as it's going against my core idea in a small way but I would be disappointed now if these changes didn't happen. The changes made represent the massive amounts of new things I have learned about software development over the past months.

I believe IronSight can become even more than what it already is and I intend to continue working on it, as not only do I find the process so interesting but I have a vested interest in what the app itself can offer as a tool.

There have been slight changes to the app throughout its development. Some due to restrictions in the development and some due to new innovative ideas but the core of what IronSight set out to be became a reality.

4 Further development

4.1 User profile pictures

Add in profile pictures for users that appear in the meet up list and on the side navigation menu. This picture could be changed after registration in the settings menu.

4.2 Game profile pictures

Each game would have its own image on its profile screen as well as thumbnail in the list pages.

4.3 Pricing

Add an attribute to the game entries in the database for pricing so users can see the price of the game depending on their region.

4.4 Alter meet up functionality to be totally in-app

Change the form of communication between app users from using email to using in-app communication, either through popups or an in-app chat room.

4.5 Enhance searching

Add the ability to search not only by game name but by genre or platform.

4.6 Additional user settings

Enable users to change their password, permanently delete their account and change their profile picture.

4.7 Styling of lists

Change the list items to support a small thumbnail image of the game and the possibly the font.

4.8 Refine "Check these out"

Alter this list to support games more closely related to the existing games already in the user's personal list.

4.9 Expand to other forms of media

IronSight can add in other forms of media quite easily by adding additional nodes in the database to support movies or books. While the database is designed to be open enough for this to be easily implemented, there would be some work required for this change with the user interface.

5 References

Ravi Tamada. 2016. *Android Getting Started with Material Design*. [ONLINE] Available at: http://www.androidhive.info/2015/04/android-getting-started-with-material-design/. [Accessed 6 May 2017].

Simplified Coding. 2016. *Firebase Realtime Database Tutorial for Android, Simplified Coding*. [ONLINE] Available at: https://www.youtube.com/watch?v=EM2x33g4syY. [Accessed 6 May 2017].

Firebase Google. 2017. *Add Firebase to Your Android Project*. [ONLINE] Available at: https://firebase.google.com/docs/android/setup. [Accessed 6 May 2017].

TVAC Studio. 2017. *Android Studio Tutorial - Firebase Tutorial*. [ONLINE] Available at: https://www.youtube.com/watch?v=i-gZAYBMuBs. [Accessed 6 May 2017].

Google Inc.. 2017. *Shrink Your Code and Resources*. [ONLINE] Available at: https://developer.android.com/studio/build/shrink-code.html. [Accessed 9 May 2017].

Google Inc.. 2017. *Material design*. [ONLINE] Available at: https://material.io/guidelines/material-design/introduction.html. [Accessed 9 May 2017].

6 Appendix

6.1 Project Proposal

Note: All number indexing has been removed to avoid interference with Technical Report numbering.

6.1.1 IronSight

David Byrne, 13109863, david.byrne5@student.ncirl.ie

BSc (Hons) in Computing

Software Developement

14/10/2016

6.1.2 Background

As it stands, on the web or through application providers on both Android and iPhone, there are no services providing comprehensive and concise information on video game release dates. While there are sites which provide extensive scheduling, there are none providing a complete package of detailed filtered release dates, pricing and available pre-ordering. In the current state of affairs, a consumer must navigate through popular websites often cluttered with verbose articles of content to get to the release date of the product they wish to purchase.

The absence of a platform to be able to track and schedule upcoming games can often lead to disappointment from the consumers point of view. Given the amount of games which are released over the course of even a month, there can be difficulties around keeping track of what people want and when they can get it.

Games in the modern market are being priced higher than ever before with new releases usually labelled at €75. Given the current arrangement of available information online, most people would struggle with effectively prioritising games that they can afford over a certain stretch of time. Often a person may see a good review of a game they had not previously been aware of and would go out and buy this game. For many people, this may result in not having the finance for a game they would much rather have purchased, released maybe a week later.

While there are some services providing lists of release dates online, these lists can be cluttered and lack any refined filtering options or else be lacking in some other features. The use of multiple services from different websites is often needed to get all the information desired. The ability to track release dates, be aware of the cheapest option available in the market and have a link to where this game can be bought is currently not available.

Digital video game distributor “Steam” are introducing on average 8 games per day to their library. The market is in need of a service that can track these games in a clear and concise manner while offering users the option to save their own personal lists rather than having to traverse massive amounts of data each time they log on. As well as providing further information about pricing and pre-ordering.

Within the gaming community there is a big divide between “indie” gaming and “AAA” gaming. “Indie” games are typically developed by a single person or small team and are centered primarily on an artistic environment with simple gameplay. While “AAA” games are developed by large companies made up of multiple teams of developers. These games emphasise the use of the latest technology available to game developers and so boast the best visuals and complex game mechanics. There are gamers who appreciate both of these styles of game though many people find themselves enjoying either one or the other. No current scheduling service divides these two styles and it is a feature which is sought after.

6.1.3 Objectives

6.1.3.1 Mobile application

The service will be provided to users through a mobile Android application created through the Android Studio program. The app will make use of a login for users to access their personal accounts. User interface will consist of multiple screens for the overall listing of games, account settings, personal lists and recommended games.

6.1.3.2 Database

A comprehensive database of both existing and upcoming video games will be integral to the functionality of the app. An API from “IGDB.com” will be used to populate the database, though will not include all necessary information around pricing or pre-order availability. Getting this information will require the use other sources such as Steam and GameStop.

The database will be searchable by using a deep filtering system which can sort and separate by release date, genre, title, platform, and by either “indie” or “AAA”. Tags on each game will also help separate what the user is searching for. For instance, a game may be tagged as “story-rich” or “online multiplayer”. This will enable the app to filter games even more or suggest similar games to the user with the same tags.

6.1.4 User Accounts

The main list page will be heavily populated with entries from the database and even though it will be refinable, the data must be manually refined each time a user wishes to view certain entries. To solve this issue, each user has their own account where a custom list of games will be displayed. The user will use the main database interface to locate games of interest. They will then have the option to add these games to their own list.

These personal lists will generate sub-lists of tags related to the games stored there. The tags will be used to generate “recommended games” associated with the specific users account.

6.1.5 Pricing feature

Each game will have a price attached to it so as users can plan ahead in terms of future purchasing while also being able to reference prices for games already released. Games are available for purchase digitally online and also through physical retailers so prices from retailer websites will be used to populate the price attribute in the app. The opportunity for implementing a price comparison feature may also be explored depending on limitations of available information.

6.1.6 Notifications

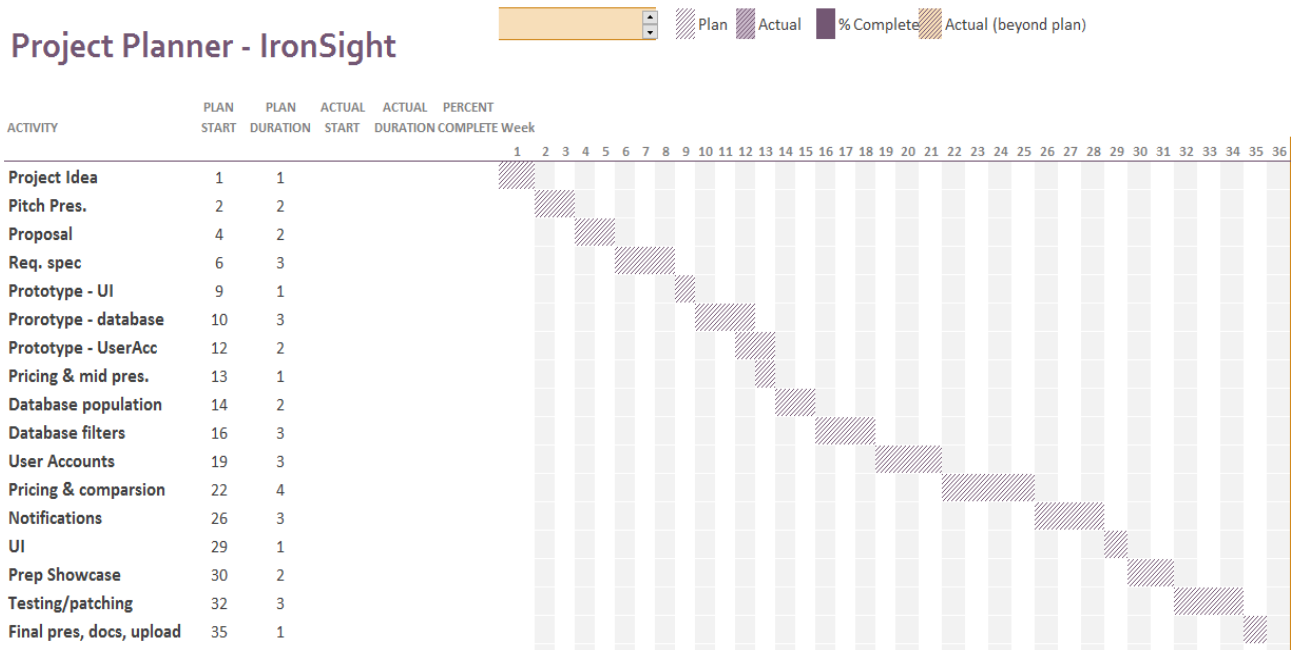
On the day of release for any games stored in user lists, a notification will be sent to the user advising them of the games release.

6.1.7 Google Calendar

Alternatively to in-app notifications, through use of the user accounts personal lists, a feature will be implemented where any games in a user's list would be sent to the users Google Calendar by release date. This would require permissions for access to Google Calendar.

Some users may prefer the option to use their own calendar to schedule game purchases rather than the in-app schedule.

6.1.8 Project Plan



Steps:

1. Project Idea: brainstorm ideas and settle on one with appropriate complexity and realistic achievement.
2. Pitch Presentation: prepare pitch presentation, a few slides and overview of key aspects of project.
3. Proposal: outline the purpose of the project, the features it will have and how it will be developed.
4. Requirements specifications: describe the requirements of the project.
5. Prototype UI: develop an early stage UI for midway presentation.
6. Prototype database: develop an early stage database for midway presentation.
7. Prototype User Accounts: develop an early stage user account for midway presentation.
8. Pricing & midway presentation: start development on pricing feature and prepare midway presentation.
9. Database population: populate the database with information from API and lay out appropriately.
10. Database filters: design search on DB incorporating tags.
11. User Accounts: further develop user accounts with favourited list and recommended games.

12. Pricing & comparison: implement pricing on all games, with price comparison feature between retailers.
13. Notifications: display notifications to users phone on day of favourited games release. Also implement notifications to users Google Calendar.
14. UI: complete user interface through Android Studio. Main list page. User account page. Recommended games page. Incorporating a search filter page/partition for main page.
15. Prepare Showcase: prepare for showcasing the project.
16. Testing: test all systems for bugs and fix.
17. Final presentation, documents and upload: prepare a final presentation, prepare 2 hard copies of final document, upload finished project.

6.1.9 Technical Details

Technologies in use:

1. Java through Android Studio
2. SQLite
3. IGDB API
4. Steam API
5. Google Calendar

6.1.9.1 Java through Android Studio

The skeleton of the app will be created using Java in the Android Studio IDE. Android Studio offers powerful UI tools with relative ease of use.

6.1.9.2 SQLite

SQLite is the ideal tool to manage the database as it is compact, does not require configuration and will use only the amount of memory it needs to operate.

6.1.9.3 IGDB API

The database will be pulling information on the games from this API which provides a vast quantity of information on video games, both upcoming and released.

6.1.9.4 Steam API

Steam API will be what the app will use to get information on game prices.

6.1.9.5 Google Calendar

A users personal Google Calendar will be updated by the app to show the release dates of the games the user has saved in their “favourite” games.

6.1.10 Evaluation

- The application will be tested through the perspective of a user. The user must be able to log in and remain logged in each time the app is launched unless set otherwise.
- The UI must display results from the database accurately according to the users filtering of tags and attributes.
- When on the personal list page, entries should be displayed by default in order of release date and it should be easy to set to any other order.
- Recommendations on the assigned page should be appropriate to the user's personal list and should not display recommendations unless sufficient data is acquired from personal list.
- Price labels should be accurate and up to date on all game entries.
- Notifications should send to Google Calendar in a concise and well-structured format.

6.2 Monthly Journals

6.2.1 Reflective Journal - September

Student name: David Byrne

Programme: BSc in Computing Evening

Month: September

Achievements

This month was only about coming up with the idea for the project. I had one or two things I wanted to go for but one of them has already been explored in a lot of depth by some other companies. It was going to be a restaurant reservation system where you can pick your own table at whatever time.

This being taken, I decided on a database for upcoming games. I don't keep track of games I want to get in the future and often can't afford them when they do come out. The app has a lot of potential around scheduling and offering cheapest prices available. I haven't started my proposal yet, it's on the substantial list of things to do though.

Reflection

I'm happy I went with the idea I had. Wasn't sure if it was enough but it turns out that all the "panel" were pretty enthusiastic about it. I actually managed to add a few layers of complexity on the fly in the presentation.

It's an area that I'm very interested in and honestly, this is an app that I would download myself. So the motivation, beyond getting a good result, is definitely there.

I did a small bit of market research before committing to it. A few of my friends are into games as well so I asked them and they were very supportive of it. They'd have no problem telling me if the idea was daft.

Intended Changes

No changes to the idea. I think what I have there is more than enough. There may be one or two features not good enough or overly complex and not worth the time. I will be looking into these finer details as time progresses.

Supervisor Meetings

Date of Meeting: NA

Items discussed: NA

Action Items: NA

6.2.2 Reflective Journal - October

Student name: David Byrne

Programme: BSc in Computing Evening

Month: October

Achievements

Plenty done this month in terms of planning and documentation. Both proposal and requirements specifications uploaded this month. I also started meeting with my supervisor, Michael, who gave me a lot of insight into what I'll need to do to accomplish what I'm aiming for. He advised on some frameworks to look at as well as maybe taking a step back from just game related products to maybe other media type products as well. He said I should consider looking into coding when I get a few requirements down but I haven't had much time for it, I'm really just aiming at what's around the next corner with other subjects.

Reflection

I wasn't too enthusiastic about the documentation work but I have to say, they opened my eyes to a lot of aspects of the project that I hasn't previously considered. Though I'm glad they're out of the way now, the value in having done them is very apparent to me now. Primarily around the requirement specification, things like what technologies I'll

need to use and what my priority requirements are have been made clearer. Though I realize it's very subject to change, having it there to refer to as I go is really helpful.

The meetings with Michael have been a great help too. It's obvious he knows the ins and outs of a lot of the things I want to accomplish with this project. He also has encouraged me to look at monetizing the application if possible and I won't resist him on that. We've scheduled to meet every 2 weeks and I'm happy to be seeing him so often. Being able to check in with what I have so far is reassuring. He says that I'm track at the moment so I just need to keep it that way now.

Intended Changes

The development environment and targeted platform. These are two things that I've been juggling around since September and I'm delighted to have pinned both down. I will be making a native android application using Android Studio. That's it set in concrete. I chose this because my language is Java and I really like Android Studio. So the course is set for the time being. Prototype next on the chopping block.

6.2.3 Reflective Journal - November

Student name: David Byrne

Programme: BSc in Computing Evening

Month: November

Achievements

This month was primarily based around designing the prototype and putting together the technical report. In terms of designing the prototype, I decided to stick with Android Studio and use an interesting technology called Material Design that works with Android Studio. It took a while to get the hang of and there's plenty more to discover with it, but al round the technology will definitely suit the UI features I want to bring to the app.

For the prototype, I aim to be able to have all the main screens ready to demo. Though the app will not be running from a database, the information displayed to the users will be hardcoded in and the login screen will accept any form of input. Any entries on the main, personal and suggested lists will also be hardcoded in. Process on these actions have already begun. The most difficult aspect of this development is the integration with the design I have chosen to use. Normally Android Studio uses activities as the separate UI screens though I am using fragments which are broken up version of activities and this can cause problems but nothing I haven't been able to get around so far.

Reflection

Again Michael has been a great help with coming up with ideas to enhance the app. Some I'm thinking at the moment are pretty farfetched for my abilities though they're staying on the list of to-dos for now.

Initially I have to say that I was skeptical about using Android Studio as I wasn't sure if it could power the UI that I wanted. Though with the discovery of Material Design, I'm now fully settled and a lot less apprehensive.

Having completed a lot of my CAs in other classes, the time I have now to focus on the project is a little relieving. I never anticipated the workload involved in the year so far and getting to the end of this bloated semester is something I'm very much looking forward to.

Intended Changes

No huge changes to the app planned at the moment. I think everything I've decided upon so far is what I intend to stick with. Maybe as the real programming begins things may shift about but for now it's a straight course.

6.2.4 Reflective Journal - December

Student name: David Byrne

Programme: BSc in Computing Evening

Month: December

Achievements

Finished with the prototype. The vast majority of the project relies on the database to be fully functional so it was necessary to hard code in the data being shown on the presentation version. Overall it went quite well, the app was fully intractable with maybe of the end product UI features included. The login screen was there but accepted any form of input as an ID. The main, user and suggested lists were there though not populated by a database and do not currently interact with each other. The info for all 3 were pre-written.

I'm quite happy with all the different screen ideas I had coming together. Being able to combine various Android Studio technologies and get what I want for the app is both relieving and satisfying.

There were some issues in the presentation around wires and connections but all in all I think it went well. I was able to communicate what the app was about and how I aimed to get to where I want with it. The fact that I'm actively developing it while testing on a real mobile device is something that gives me comfort in that I know what I have so far will be functional on any android device.

Reflection

It's nice to have the prototype done and reviewed. At least I know I'm on the right track but I realise that most of the work is still ahead of me. Getting the IGDB database to integrate with the app and also the local DB will be a challenge. Primarily getting the

suggestion system to work is something that will likely take a good bit of time and this is pretty high up in terms of priority and overall functionality.

Exams starting soon in other subjects so I'll be looking at that mostly for the time being. There is a 2 week gap of "free time" though before the semester begins and after the exams finish so ample time to get stuck in to the project.

Intended Changes

I need to remove nearly all the game content from the app. Since it's all hard coded, it will need to be populated by the new databases. While it's nice to have the bones of the content structure there, the real job will be the databases. As well as those, the login will need to be working with real user accounts. Accounts being absolutely essential for saving personal lists and generating individual suggested lists.

6.2.5 Reflective Journal - January

Student name: David Byrne

Programme: BSc in Computing Evening

Month: January

Achievements

Genuinely not a lot done this month. When I first showed off my well thought out project plan to Michael, I got a veiled laugh from him. I thought to myself, surely it's a workable plan. He then pointed out that I had tasks for every day of the semester including time off over Christmas. I started to see where he was coming from and sitting here now in January I definitely see where he was coming from.

One thing that I am very happy with discovering though is Firebase. This collection of fantastic features include user login through a server, has a real-time database, supports my notifications, implements adverts which I'm hoping can be used to create sponsors entries in the lists, testing, crash reporting and many others. So implementing that and the IGDB database is the dream at the moment.

Reflection

I do wish I took less time off over Christmas because it seems like I have a lot ahead of me. I'm not panicking though as I've never done anything like this before so it could go either way. Either Firebase gets implemented easily and that a lot of ticked boxed or it doesn't and I'm spending a lot of time on it. Same situation with IGDB, feeding in the game info from the database. Could go either way. So all I can do now is get the head down and gets this baby working.

Intended Changes

- Introduce firebase with all its bells and whistles
- Get the IGDB database hooked up

- Develop the suggestion feature, like through a plugin

6.2.6 Reflective Journal - February

Student name: David Byrne

Programme: BSc in Computing Evening

Month: February

Achievements

I've managed to integrate the firebase real-time database into the app. After countless tutorials and trial and error I've managed to get the app to read off the database in a format suited to the application. As far as I can tell, the Game data model needs to match up with the JSON variable names in the database or the data won't map to the model correctly or even at all. It also has trouble with reading anything other than a flat database so myself and Michael ran through the database and identified what needed to be taken out of a nest and made more readily available for the app. Now to take those objects and use them for individual game pages where users will be able to add them to their own lists.

This may seem like a minor accomplishment but a lot of research and time went into it. Lots more to do from here.

Reflection

As Michael said to me, regardless of the amount of effort that went into where I've gotten to so far, these efforts may not be very apparent to anyone else. So I need to get the "flashier" aspects of the app done as well. Though those aspects are very reliant on the more boring parts that I'm currently working on. Having never made an app before, there's a lot more to it than I had anticipated. Features I want to implement are causing a lot more issues than I had thought they might. Going to be a long couple of months head.

Intended Changes

- Populate the firebase database and use that for the app info as opposed to feeding directly from IGDB
- Create the user list from the firebase storage by adding game objects to it from the main database
- Use material design notifications as they allow to better set dates

6.2.7 Reflective Journal - March

Student name: David Byrne

Programme: BSc in Computing Evening

Month: March

Achievements

Not nearly as much as I'd have liked. Database integration using firebase with an android app is not very straightforward and given that this app is very reliant on database interaction, there have been headaches. In terms of achievements, this paragraph will be rather short this month. I'm still trying to refine the data input from firebase to get everything in terms of game information. While this process is not at a standstill, it is moving slowly and with the time remaining, I'm incredibly eager to finish this off as there are many other features yet to begin.

Reflection

Feeling the pressure. And the frustration if I'm honest. Time and effort is getting pumped into this project at the expense of others and is not yielding nearly as much result as I'd like. Compromises are going to be met. Some things will not be a part of this app that were intended. Solely in the interest of getting something done that I can showcase. Oh god the showcase. Now all I can do is continue trying, whether something comes out of it at the end is yet to be seen. If I can scale one or two mountain between now and May, there remains some hope.

Intended Changes

- Format the data to suit the app
- Create the user list from the firebase storage by adding game objects to it from the main database
- Use material design notifications as they allow to better set dates

Declaration Cover Sheet for Project Submission

SECTION 1 *Student to complete*

Name:
Student ID:
Supervisor:

SECTION 2 Confirmation of Authorship

The acceptance of your work is subject to your signature on the following declaration:

I confirm that I have read the College statement on plagiarism (summarised overleaf and printed in full in the Student Handbook) and that the work I have submitted for assessment is entirely my own work.

Signature:_____ Date:_____

NB. If it is suspected that your assignment contains the work of others falsely represented as your own, it will be referred to the College's Disciplinary Committee. Should the

Committee be satisfied that plagiarism has occurred this is likely to lead to your failing the module and possibly to your being suspended or expelled from college.

Complete the sections above and attach it to the front of one of the copies of your assignment,

What constitutes plagiarism or cheating?

The following is extracted from the college's formal statement on plagiarism as quoted in the Student Handbooks. References to "assignments" should be taken to include any piece of work submitted for assessment.

Paraphrasing refers to taking the ideas, words or work of another, putting it into your own words and crediting the source. This is acceptable academic practice provided you ensure that credit is given to the author. Plagiarism refers to copying the ideas and work of another and misrepresenting it as your own. This is completely unacceptable and is prohibited in all academic institutions. It is a serious offence and may result in a fail grade and/or disciplinary action. All sources that you use in your writing must be acknowledged and included in the reference or bibliography section. If a particular piece of writing proves difficult to paraphrase, or you want to include it in its original form, it must be enclosed in quotation marks and credit given to the author.

When referring to the work of another author within the text of your project you must give the author's surname and the date the work was published. Full details for each source must then be given in the bibliography at the end of the project

Penalties for Plagiarism

If it is suspected that your assignment contains the work of others falsely represented as your own, it will be referred to the college's Disciplinary Committee. Where the Disciplinary Committee makes a finding that there has been plagiarism, the Disciplinary Committee may recommend

- that a student's marks shall be reduced
- that the student be deemed not to have passed the assignment
- that other forms of assessment undertaken in that academic year by the same student be declared void
- that other examinations sat by the same student at the same sitting be declared void

Further penalties are also possible including

- suspending a student college for a specified time,
- expelling a student from college,
- Prohibiting a student from sitting any examination or assessment.,
- the imposition of a fine and
- The requirement that a student to attend additional or other lectures or courses or undertake additional academic work.