CS6051

Coursework 1

Final Submission

Development Report

Game Review Application

'GGReviews'

Name: Lauren Spruce

ID Number: 18011848

Date: 12/04/21

Table of Contents

Table of Figures

Tables

1: Introduction

1.1 Background

Before studying the module CS6051, a lot of time was spend considering what would be an interesting mobile application, not only for personal interests, but to please a large-scale audience.

In my free time, I often play games on my PC. It is estimated that, in the year 2020, there was *"2.69 billion video gamers worldwide".* (Clement, 2021) This shows that gaming can create endless amounts of development ideas, as there is a high demand in related content. By creating a game related application, I could create something that was interesting to me, as well as a large-scale audience. It is estimated that the number of gamers *"is expected to rise to over three billion gamers by 2023."* (Clement, 2021) With many new consumers being introduced to the gaming market, I feel like a gaming review application would be perfect to develop, as it allows new consumers an introduction to the gaming communities opinion on current games, so therefore they can make smart financial choices when purchasing from a developer. This is important, as the total revenues of the gaming industry reached *"£4.55bn in 2020."* (ukie, 2021) This shows that consumers are willing to spend a lot of money across many gaming developers. This means that lots of money could potentially be wasted on games that do not meet the current satisfaction standards, consumers deserve the right to research before buying.

A conclusion was made. The idea for a game review application was created. My application is named GGReviews. The 'GG' stands for 'Good game.' I used this acronym as *"in multiplayer competitive games, GG is used as a mark of sportsmanship and an acknowledgment that you had fun while battling against your opponents."* (Vicente, 2020) I felt like this acronym signifies the importance of collaboration between opponents, whilst also maintaining respect for one another. This marked the birth in GGReviews. It allows for teammates and opponents to come together, giving their honest feedback on a game, in doing so helping fellow consumers out. This will therefore improve the gaming community, allowing consumers voices to be heard, and important changes to be made within development teams.

1.2 Aims and Objectives of Application

Aim

- The most important aim is to develop an android application which will allow consumers to create and read game reviews, these games will be listed on the application and only contain most popular and up to date.

Objectives

- Develop an activity which has a login feature with authentication.
- Develop an activity which has a register feature with authentication.
- Develop an activity which allows users to change account details, with authentication.
- Develop an activity which allows users to create a review with authentication.
- Develop an activity which allows for users to read other user's reviews
- Development an activity which allows for the individuals review to be updated and deleted.
- Complete SQLite database, in which can store the game reviews and user details.

Smart Objectives

1. Develop an activity which has a login feature with authentication.

- **Is it specific to the application?**

  The login system will allow for authentication, asking users for username and password. It will also contain validation.

- **How will it be measured**

  It will be measured through the SQLite database when the username and password is entered. SQLite database will also have authentication.

  Is it appropriate?

  A suitable login system is a requirement for this application.

  Is it realistic?

  It can be completed easily.

- **Is it Time Related?**

  It will only take a day or so to implement this feature into the application.

2. Develop an activity which has a register feature with authentication.

- **Is it specific to the application?**

  The register system will allow for authentication, asking users for username, first name, last name, email, and password. It will also contain validation.

- **How will it be measured**

  It will be measured through the SQLite database when the user credentials are entered. SQLite database will also have authentication.

- **Is it appropriate?**

  A suitable register system is a requirement for this application.

- **Is it realistic?**

  Yes, it can be easily implemented.

- **Is it time related?**

  Will only take a day or so to implement.

3. Develop an activity which allows users to change account details, with authentication.

- **Is it specific to the application?**

  The update account system will allow for authentication, asking users for username, email, and password. It will also contain validation.

- **How will it be measured**

  It will be measured through the SQLite database when the user credentials are entered. SQLite database will also have authentication. Uses CRUD operation update to change user's credentials.

- **Is it appropriate?**

  A suitable update account system is a requirement for this application.

- **Is it realistic?**

  Yes, it can be easily implemented.

- **Is it time related?**

  Will only take a week or so to implement.

4. Develop an activity which allows users to create a review with authentication.

- **Is it specific to the application?**

  The create review system will allow for authentication, asking users for game name, title, rating, and description. It will also include validation.

- **How will it be measured**

  It will be measured through the SQLite database when the reviews are entered.
  SQLite database will also have authentication.

- **Is it appropriate?**

  A suitable create review system is a requirement for this application.

- **Is it realistic?**

  Yes, it can be easily implemented.

- **Is it time related?**

  Will only take a week or so to implement.

5. Develop an activity which allows for users to read other user's reviews.

- **Is it specific to the application?**

  The read other users' system will allow users to view all user reviews within a list displayed on the application.

- **How will it be measured**

  It will be measured through the SQLite database; it will read all current reviews. It will also display a model list of the reviews on the UI.

- **Is it appropriate?**

  Reading all user reviews is a requirement for this application.

- **Is it realistic?**

  Yes, it can be easily implemented.

- **Is it time related?**

  Will only take a few weeks to implement.

6. Development an activity which allows for the individuals review to be updated and deleted.

- **Is it specific to the application?**

  Updating and deleting reviews is a requirement as it allows for the CRUD operations, also gives the user option to delete a no longer wanted review or edit one in which they made a mistake.

- **How will it be measured**

  The SQLite DB will have queries in place to update and delete reviews using CRUD operations.

- **Is it appropriate?**

  Updating and deleting user reviews is a requirement for this application.

- **Is it realistic?**

  Yes, it can be easily implemented.

- **Is it time related?**

  Will only take a few weeks to implement.

7. Complete SQLite database, in which can store the game reviews and user details.

- **Is it specific to the application?**

  Supports the consumers as it can store all their details and game reviews, also allows for consumers to look at past reviews to help make their mind up on purchasing a game.

- **How will it be measured**

  Once the SQLite database as been programmed and all queries are functioning.

- **Is it appropriate?**

  Yes, this is required.

- **Is it realistic?**

  Yes.

- **Is it time related?**

  It will take 1-2 weeks to complete.

1.3 Development Purpose

GGReviews will be completely developed in Android Studio (IDE), alongside an SQLite database containing two tables. These tables are users and reviews. It will allow for users to register an account, have it stored within the users table, so therefore they can log in and out using their saved credentials. It will also provide users the ability to create and store game reviews, which will be listed under all reviews. Any user can read the reviews; however, users can only edit and delete their own reviews. This will all be supported by the CRUD operations, these will be implemented into queries and methods to create, read, update, and delete both user credentials and reviews.

GGReviews will be targeting game consumers, which will focus on the current popular games, and allow for users to give their feedback for any game they choose to. It will support future games, as they can be included within the database at any time, so therefore users have many games to search and review. It allows many consumers to use this application, as focuses purely on game reviews, therefore only the appropriate target audience will interact with this application.

GGReviews also allows users the comfort of using their mobile device, which is has clean and professional user interface. It is easy to use and more appealing to casual or competitive gamers.

2: Design and Development

2.1 Software Development Lifecycle

GGReviews followed a Unified Process model for its software development cycle. Unified process is *"based on the enlargement and refinement of a system through multiple iterations, with cyclic feedback and adaptation."* (Osis & Donins, 2017) I felt this was the best model to use for the development, as there are multiple stages within unified process, which meant each stage is short and brief, so therefore reflection and testing could be performed at each one. Within unified process, there are 4 stages, which are as follows:

Inception

Inception is the shortest phase, however the most crucial. *"Preparation of business case, establishing project scope and setting boundaries, outlining key requirements."* (Osis & Donins, 2017) This was crucial for this development as the purpose of the project needed to be established quickly, alongside recognising the importance of the application. GGReviews importance was to allow consumers to review games, this purpose stayed true throughout the entirety of the project.

Elaboration

"During this phase, the project team is expected to capture a majority of system's requirements." (Osis & Donins, 2017) This is completed within section 2, as I explore the use case diagram and data architecture of GGReviews.

Construction

"During this phase, the design of the system is finalized and refined, and the system is built." (Osis & Donins, 2017) This is shown in section 2, as all designing of the application and database is shown here. In section 3, the testing is completed, this was all thanks to constructing the application by using Java, SQLite, and XML.

Transition

Transition is **"the final project phase which delivers the new system to its end-users."** (Osis & Donins, 2017) This final project is shown in section 3, testing and within the viva which will be performed explaining the features and functions which were successfully implemented into the application. The user guide also explains how to deploy the application and use it.

2.2 Software Architecture

2.2.1 Use Case Diagram

This is a use case diagram of GGReviews. It shows the interaction of the application between a registered user, and a not registered user. Within the software architecture, the Use case diagram will be shown alongside the use case descriptions for each use case.

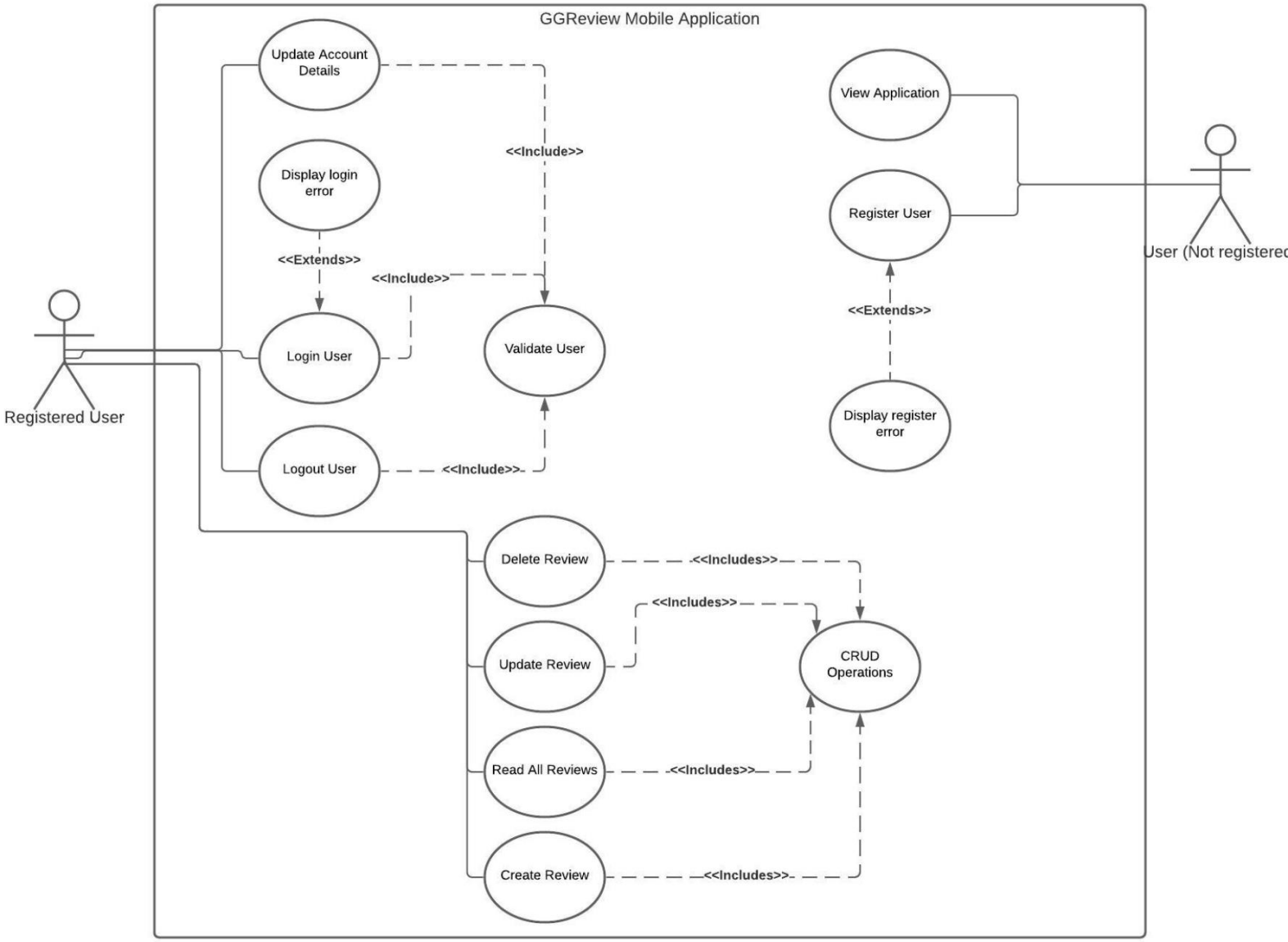


Figure 1: Use Case Diagram of GGReviews

2.2.2 Use Case Descriptions

1. **Use Case:** Register User

Actor: User (Non-registered)

Description: User can view application and decide if they would like to sign up. They can access register page by clicking on the register, within the login activity. They can fill out the form and successfully register to the application.

Event Table:

Table 1: Use Case: Register User

Actor	System Responses
1. The user navigates to the application form and fills out their details.	2. System will check if the username and password are approved.
	3. Checks the user's username, first name, last name, email, and password are all correct and suitable.
	4. Informs the user know all their details are correct and proceeds to finalise the registration.
5. User will be then redirected to the login page where they will re-enter their details to login.	

Other Actions:

Action 1.1: When filling out the form, the user leaves fields blank. System cues user to fill out all the fields.

Action 1.2: User has inputted an incorrect format of one of the personal details. System will cue user to re-enter.

Action 2.1: Username is already in use; system will cue the user to provide another username. This will also be checked if suitable.

Action 2.2: Password is not suitable as it does not include the standard format of capitals and numbers. System will cue user to re-enter a stronger password.

Action 4.1: The registration might not be confirmed on the application due to a possible error, therefore the user will have to try again, ending the use case.

Action 4.2: There could be an error with communication between the SQLite DB and the application. In this case, the user will be told to register again, or wait some time before trying again. This will cause the use case to end.

Action 4.3: External factors, including a mobile battery dying could cause the application to lose the user's information before submitting, therefore the user will have to re-enter all their details. The use case will then end.

Pre-conditions:

The user must have GGReviews installed on their mobile device.

Post conditions:

The new registered user can now successfully login and view home activity of the application.

2. **Use Case:** View Application

Actor: User (Non-registered)

Description: Any user can visit the application, they will only be allowed access to the login activity and register activity prior to registering an account.

Event Table:

Table 2: Use Case: View Application

Actor	System Responses
1. The non-registered user will download the application.	2. The application will get deployed and start up.
3. The non-registered user will click on the login activity or register activity.	4. This would then load for the non-registered user.
5. If the non-registered user has not registered or is done looking at the at the application, then they will close the application.	

Other Actions:

Action 1: The application is currently down, as the server hosting it is having technical difficulties. The application will display an error message, this will cue the user to reload the page or try again later, ending the use case.

Action 2: The links on the application may not all be working as supposed to. The application will show an error message, ending the use case.

Action 3: The application is not loading all the components; therefore, the user is prompted to reload or use the application again later. This will end the use case.

Pre-condition:

The user must have a mobile device and the application installed.

Post condition:

Not Applicable

3. **Use Case:** Login

Actors: Registered User

Description: A user can use the login activity, they will then be prompted to enter their credentials, this being their chosen username and password. If it is successful, they will be redirected to the home activity.

Event Table:

Table 3: Use Case: Login

Actor	System Responses
1. The registered user will load up the application and navigate to the log in activity where they will fill in their credentials, username, and password.	2. The system will check that both fields have been filled with appropriate data types.
	3. The system will then authenticate if the username and password match.
	4. Displays a success message to the user.
	5. Redirects user to the home activity.

Other Actions:

Action 1: The user leaves empty fields for either username and password or both. The system will cue the user to fill out all the fields.

Action 1.2: The user is not registered, the system will display a message informing the user that their username does not exist, prompting the user to register and then try to login in. This will end the use case.

Action 3: The application is taking too long to respond. The system will then display a message to the user to prompt them to try again later or reload the application. This will end the use case.

Action 5: The applications main page is not loading. This will make the system display an error message to the user, prompting them to reload the page or try again. The use case ends.

Pre-condition:

The user will need to be registered successfully as a user.

Post condition:

They will be able to access the application.

4. **Use Case:** Update Account Details

Actors: Registered User

Description: The user can click onto the navigation bar which will display an option to change account details. When the user has changed their desired credential, they can then click on save changes, this will then display a success message to confirm that the changes have been confirmed.

Event Table:

Table 4: Use Case: Update Account Details

Actor	System Responses
1. The registered user will select the update account details on the navigation.	2. The system will load the current stored credentials of that user.
3. The user will change different details and then proceed to click the save changes button.	4. The system will check if the updated details are suitable and correct, the system will then ask the user to confirm for a final time.
5. User confirms the changes	6. The system will modify the credentials and then allow the user to view the changes.

Other Actions:

Action 2: The application might not be able to load to the change detail form, displays an error message for the user telling to reload or try again, this will end the use case.

Action 3: The user leaves one or all the detail field empty. This will cue an error message, making the user fill in all the text fields.

Action 3.1: The password might not be suitable; error message tells user to re-enter stronger password.

Action 3.2: The user might try to put an illegal datatype in one of the fields, error message will inform the user to re-enter with correct details.

Action 3.3: User might try to take a username which is already in use, error message will inform user to choose another name.

Action 6: The application could crash, this would mean any unsaved detail changes would be lost, therefore the user would need to fill out the form again, ending the use case.

Pre-condition:

The user will need to be registered on the application already, the user will also need a suitable mobile device to access the website from.

Post Condition:

The user will be able to modify their details and they will be successfully changed.

5. **Use Case:** Logout

Actors: Registered User

Description: The user can click the log out link on the navigation bar to close their session.

Event Table:

Table 5: Use Case: Logout

Actor	System Responses
1. A logged in user navigates to the log out button, clicks it.	2. System will authenticate the request.
3. User redirected to the login activity of the application, logged out.	

Other Actions:

Action 1: The application could stop responding, this will show an error message to inform the user to reload the page, this would end the use case.

Action 2: The server might be down so therefore the system cannot authenticate the request, ending the use case. An error message will be displayed.

Action 3: The application might not redirect the user to the main page logged out, an error message will be displayed, ending the use case.

Pre-conditions:

The user will need to be registered and already logged in.

Post Conditions:

The user will successfully be logged out and can then register another account or log in again.

6. **Use Case:** Delete Review

Actors: Registered User

Description: The user can delete their review by clicking on their reviews and using the delete icon to confirm the deletion.

Event Table:

Table 6: Use Case: Delete Review

Actor	System Responses
1. A logged in user navigates to their current reviews.	2. System will authenticate the request.
3. User will click the delete button on the review.	4. System will ask user to confirm deletion.
5. User will click yes.	6. System will delete the review from the database.
7. Review will be deleted and updated onto the UI for the user to see.	

Other Actions:

Action 1: The application could stop responding, this will show an error message to inform the user to reload the application, this would end the use case.

Action 2: The SQLite might be down so therefore the system cannot authenticate the request, ending the use case. An error message will be displayed.

Pre-conditions:

The user will need to be registered and already logged in and have a review already made.

Post Conditions:

The user's review will be successfully deleted.

7. **Use Case:** Update Review

Actors: Registered User

Description: The user can update their review by clicking on their reviews and using the update icon to confirm the update.

Event Table:

Table 7: Use Case: Update Review

Actor	System Responses
1. A logged in user navigates to their current reviews.	2. System will authenticate the request.
3. User will click the update button on the review.	4. System will take user to update review activity.
5. User will input the new data.	
6. The user will click update button.	7. System will update the review details
8. The user will be able to see the updated review.	

Other Actions:

Action 1: The application could stop responding, this will show an error message to inform the user to reload the application, this would end the use case.

Action 2: The SQLite might be down so therefore the system cannot authenticate the request, ending the use case. An error message will be displayed.

Pre-conditions:

The user will need to be registered and already logged in and have a review already made.

Post Conditions:

The user's review will be successfully updated.

8. **Use Case:** Read All Reviews

Actors: Registered User

Description: The user can read all reviews by clicking the read all reviews tab.

Event Table:

Table 8: Use Case: Read all Reviews

Actor	System Responses
1. A logged in user navigates to the current reviews.	2. System will authenticate the request.
3. User will be shown all listed reviews	

Other Actions:

Action 1: The application could stop responding, this will show an error message to inform the user to reload the application, this would end the use case.

Action 2: The SQLite might be down so therefore the system cannot authenticate the request, ending the use case. An error message will be displayed.

Pre-conditions:

The user will need to be registered and already logged in.

Post Conditions:

The user will be able to view all the reviews.

9. **Use Case:** Create Review

Actors: Registered User

Description: The user can create a review by clicking ‘make’ on the bottom navigation bar.

Event Table:

Table 9: Use Case: Create Review

Actor	System Responses
1. A logged in user navigates to and clicks ‘make’ review.	2. System will authenticate the request.
3. User will input all the review data.	
4. User will click submit review.	5. System will create the review.
6. The user will be redirected to the review confirmation.	7. System will add the review to the SQLite review database.

Other Actions:

Action 1: The application could stop responding, this will show an error message to inform the user to reload the application, this would end the use case.

Action 2: The SQLite might be down so therefore the system cannot authenticate the request, ending the use case. An error message will be displayed.

Pre-conditions:

The user will need to be registered and already logged in.

Post Conditions:

The user’s review will be successfully created and inserted into the SQLite database.

2.3 Data Architecture

Within this section of the report, I will now be showcasing my database used for GGReviews. This will include the ER diagram and the data dictionary involved in the application.

2.3.1 Entity Relationship Diagram

Within the entity relationship diagram, it shows GGReview's database named 'GameReviewApplication.' Within this database, there are two tables, these are named 'tableregister,' and 'tablereview.'

Within the Entity Relationship diagram, it shows a one-to-one relationship between the usernames in the user table and the review table.

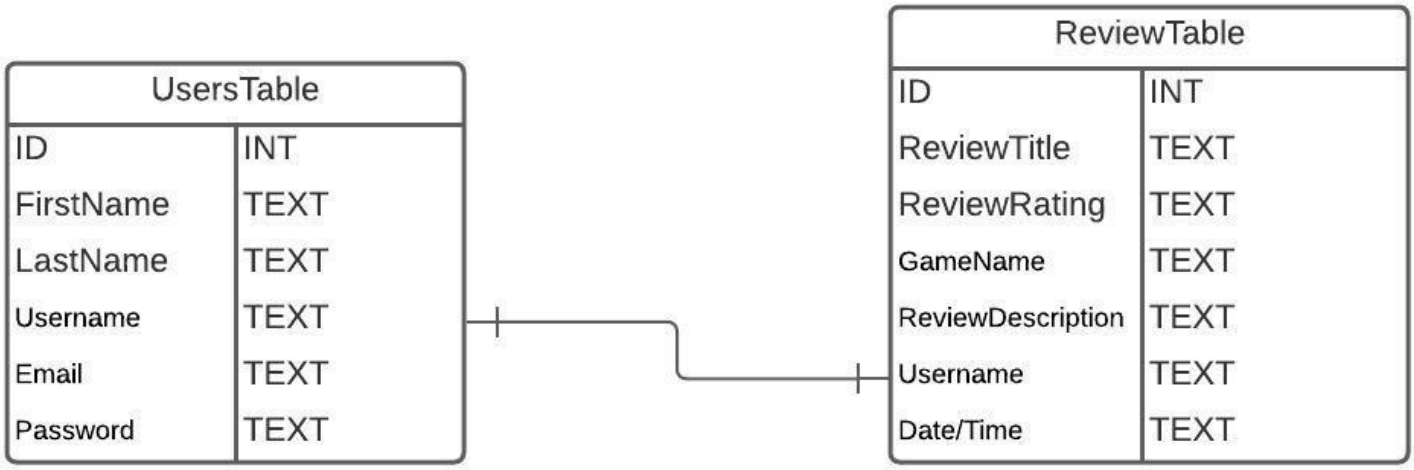


Figure 2: Entity Relationship Diagram for GGReviews

2.3.2 Data Dictionary

Table: Users

This table is used for storing all the users which have registered to the application, this includes the login credentials.

Table 10: User table

Data	Data Type	Max field Size	PK/FK
ID	INT	**10**	PK
Firstname	TEXT	**25**	
Lastname	TEXT	**25**	
Username	TEXT	**25**	
Email	TEXT	**25**	
Password	TEXT	**255**	

Table: Reviews

This table is used for storing all the users reviews which they have created while on the application.

Table 11: Reviews table

Data	Data Type	Max field Size	PK/FK
ID	INT	**10**	PK
Reviewtitle	TEXT	**25**	
Reviewrating	TEXT	**25**	
gamename	TEXT	**25**	
reviewdescription	TEXT	**255**	
username	TEXT	**25**	
datetime	TEXT	**50**	

2.4 UI Design

2.4.1 Login

When a user opens the application, they will be greeted with the login page. The login page is simple, user's will only need to enter their username and password they used to register to the application. Once the user successfully logs in, it will redirect to the main homepage. The final design stays true to the initial design, using all aspects of the design, however a password visibility function has been implemented. This is to ensure the user's password is kept private. Once the user has logged in successfully, it will take them to the home activity.

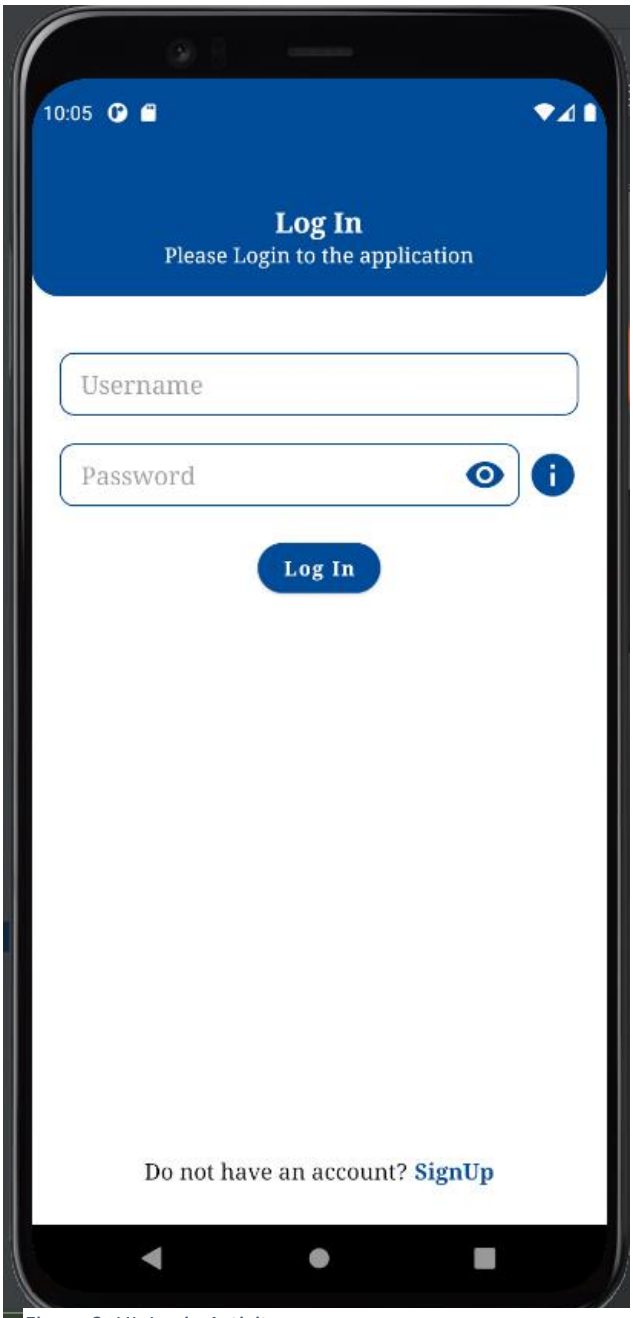


Figure 3: UI: Login Activity

2.4.2 Register Design

For the register activity, the user can enter their details, these include, username, first name, last name, and email. Once entered, the user will need to click the ‘sign up’ button. If the user has already, registered on the web application, they can click the ‘sign in’ link, this will re-direct them to the login activity. All fields are required, the user has the option to change them later in the ‘account details’ activity. The final design stays true to the initial design. Password visibility was also implemented within this page to ensure the privacy of the user.

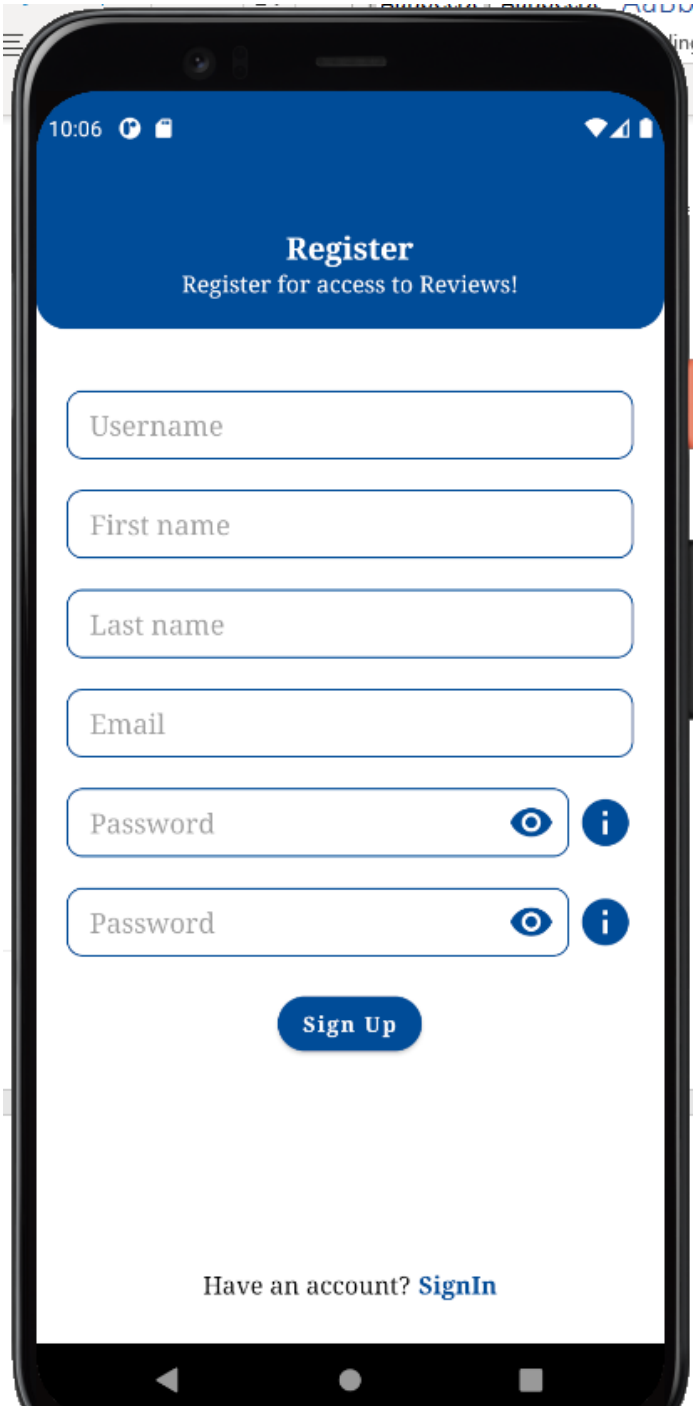


Figure 4: UI: Register Activity

2.2.3 All User Reviews/Homepage Design

For the homepage, it was decided to incorporate all user reviews here, as the initial homepage design felt empty. Within the homepage, the user has access to all the current reviews stored within the application. Users can only edit and delete their own reviews, which is shown by the icons next to the current user's review in the figure below. The home activity also shows the current user's name in the top left corner, also the navigation bar in the top right and main links at the bottom. Once a user clicks on a review, it will show the review, they can also edit and delete reviews from this activity.

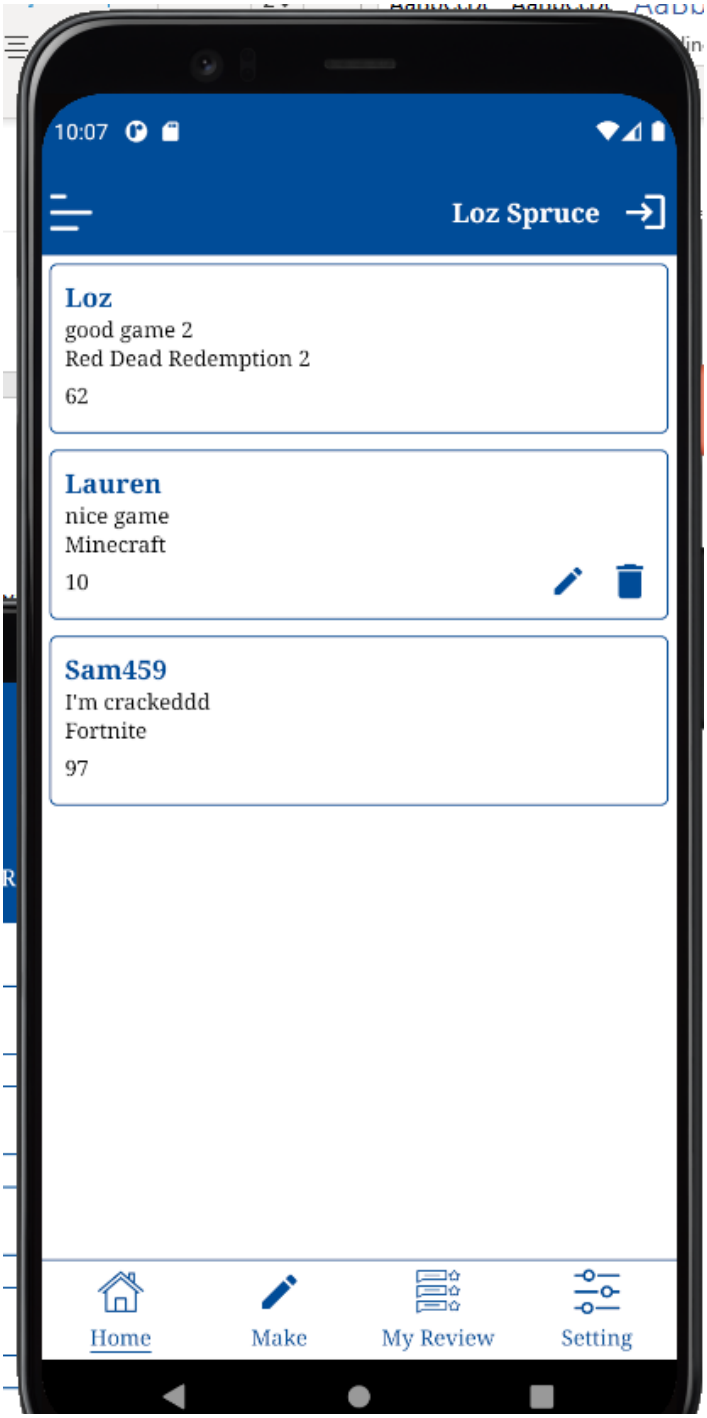


Figure 5: UI: All Reviews Activity

2.4.4 Create a Review

Within this activity, the user can create their own game review. The list of games can be clicked, to display a drop-down box of all current games stored within the review table. The user can also enter a review title, and a description.

The user also has the option to rate a game, between 0 and 100. I decided to use a larger scale as it gives a more precise review to the user. This can be manually typed in, or the user can use the plus and minus icons to increase of decrease the rating. There is also a clear feature implemented which allows for the user to clear all the data within the review and start over, if necessary.

Figure 6: UI: Create a Review Activity

2.4.5 Current User Reviews

Within this activity, the user can only view the reviews that they have currently submitted to the application. They are arranged in a list, showing the username, title, game name and rating. For all of the reviews, the user is able to edit and delete the reviews, which will start appropriate activities using intents.

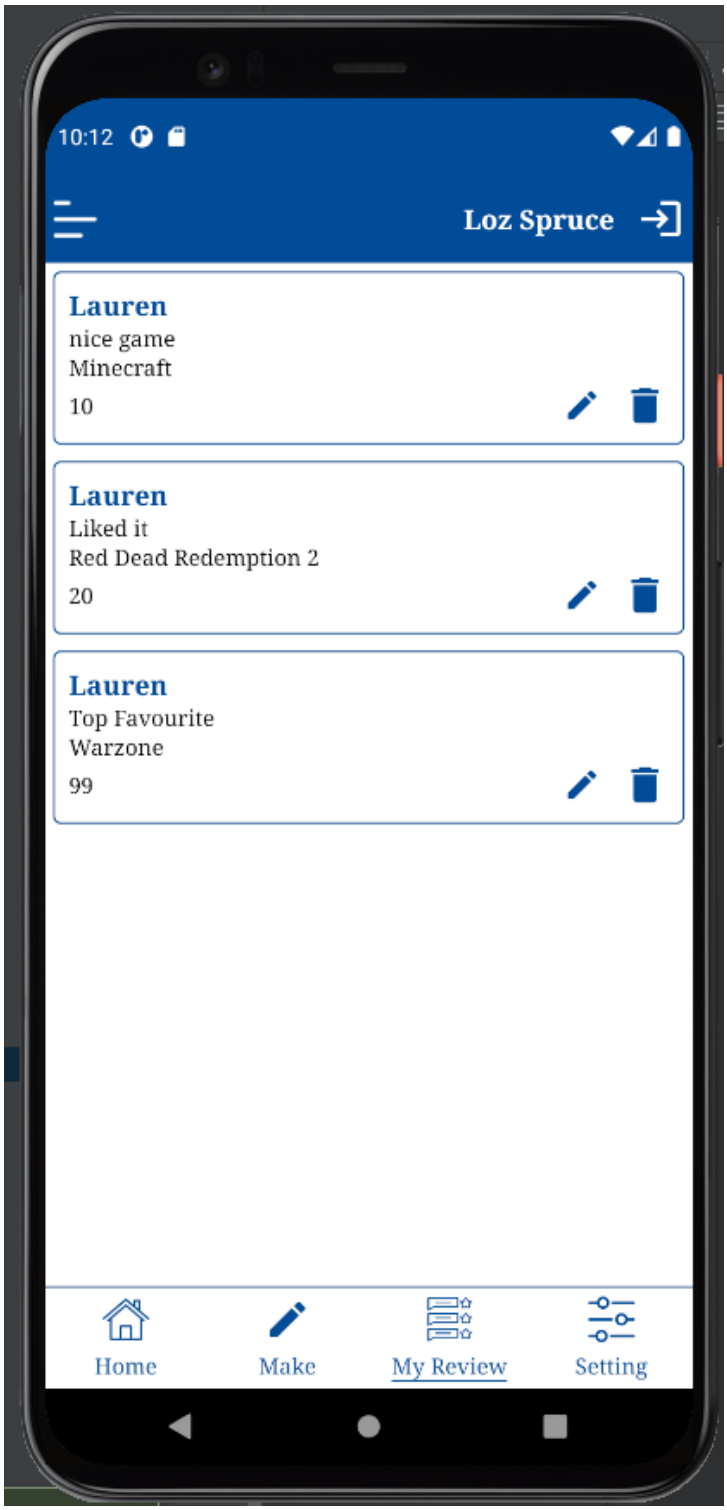


Figure 7: UI: Current User Reviews

2.4.6 Navigation Bar

The navigation bar can be accessed by clicking the 3 lines in the top right-hand corner of the application. The navigation bar displays the logo of the site, along with all the different pages the application has to offer. Exit app allows for the user to terminate the application and return to their mobile home screen.

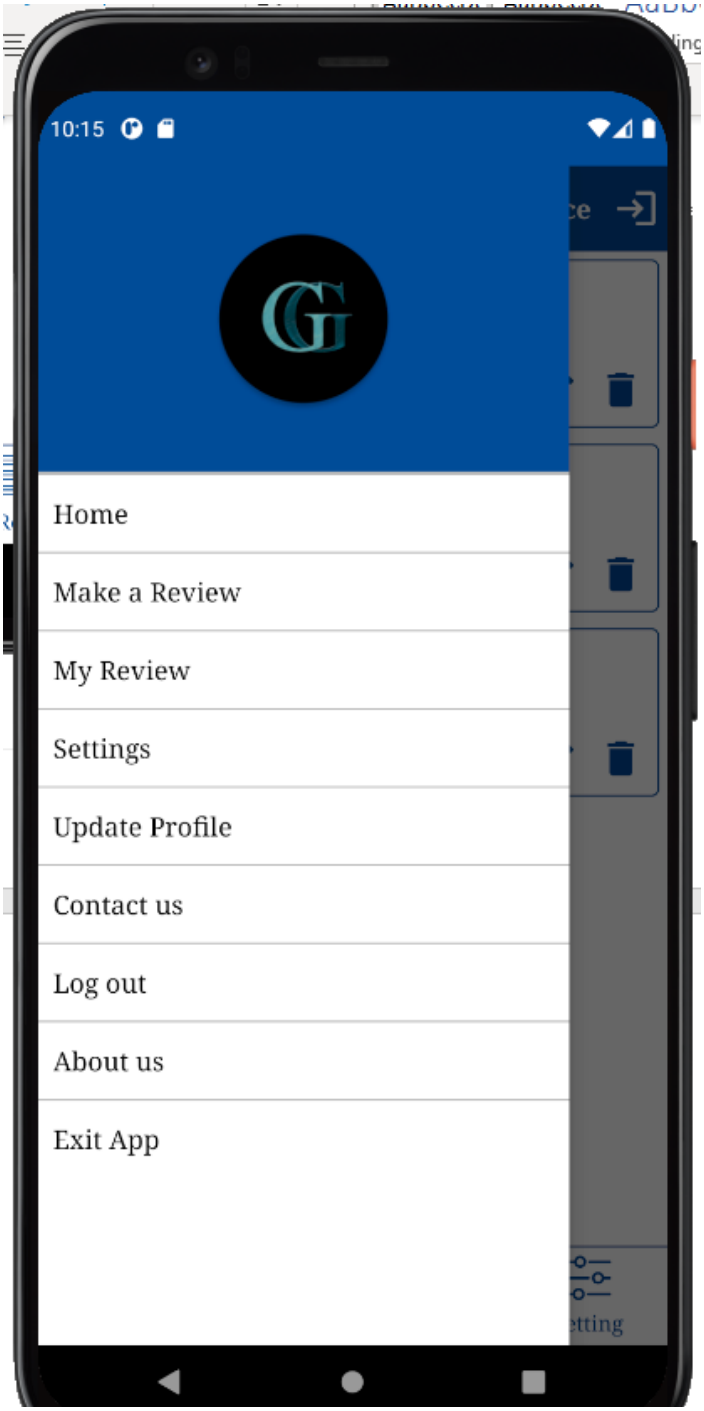


Figure 8: UI: Navigation Bar

2.4.7 Settings

The settings fragment can be accessed from the bottom navigation bar, also through the main navigation bar. It gives users more accessibility to the different activities within the application.

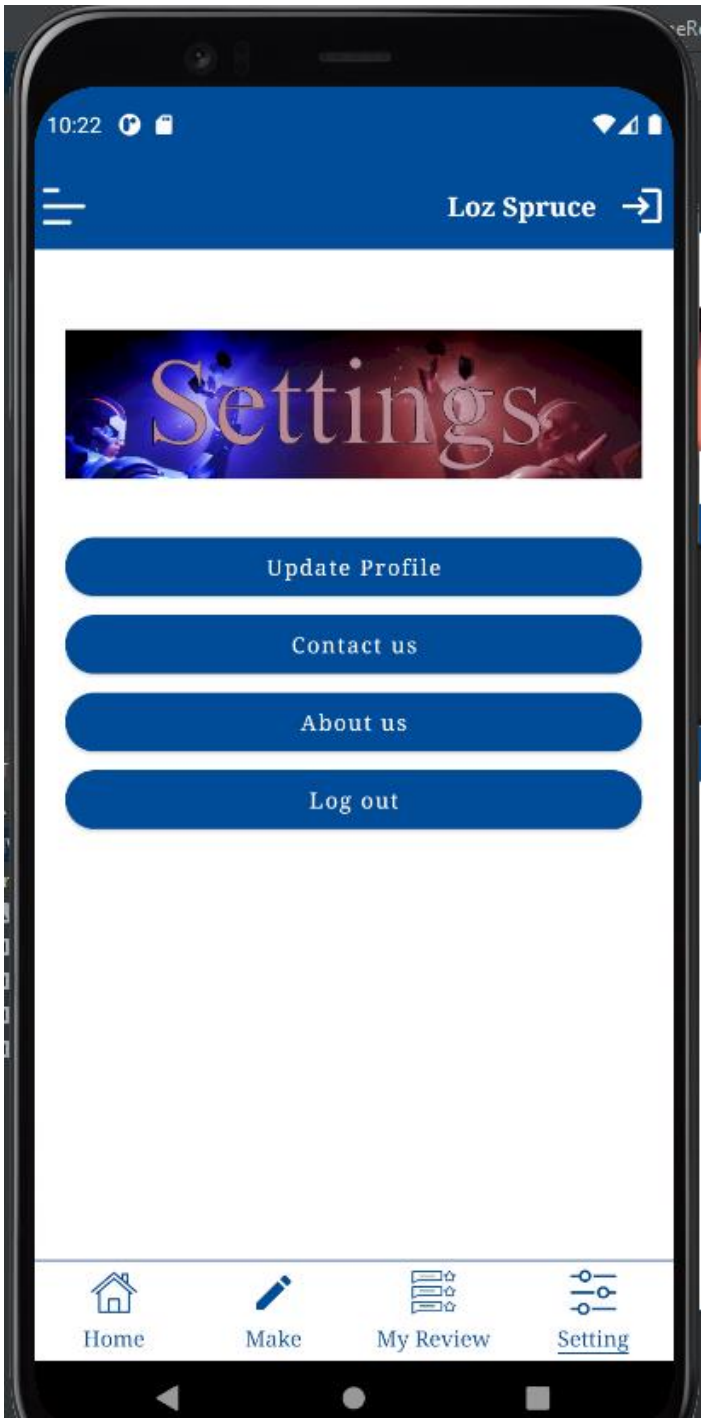


Figure 9: UI: Settings Activity

2.4.8 Update Account

Within the Update Account activity, it allows the current user to update their first name, last name, and email address. It has complete validation and ensures that the user cannot leave any field empty.

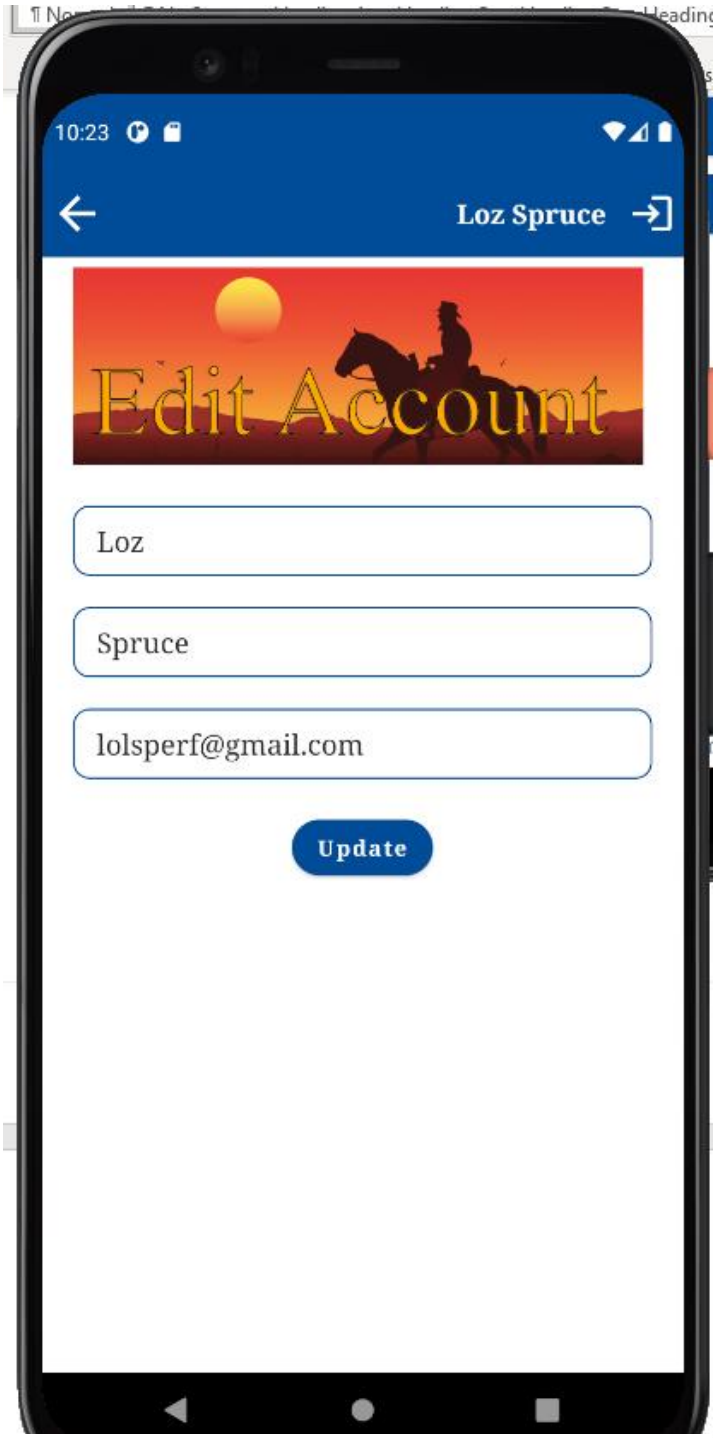


Figure 10: UI: Update User Details Activity

2.4.9 Contact Us

The contact us activity is designed to give users contact information if there were to be a bug or error within the application, to help them contact the support team. It includes a banner designed specifically for the application, along with a test phone number and email.

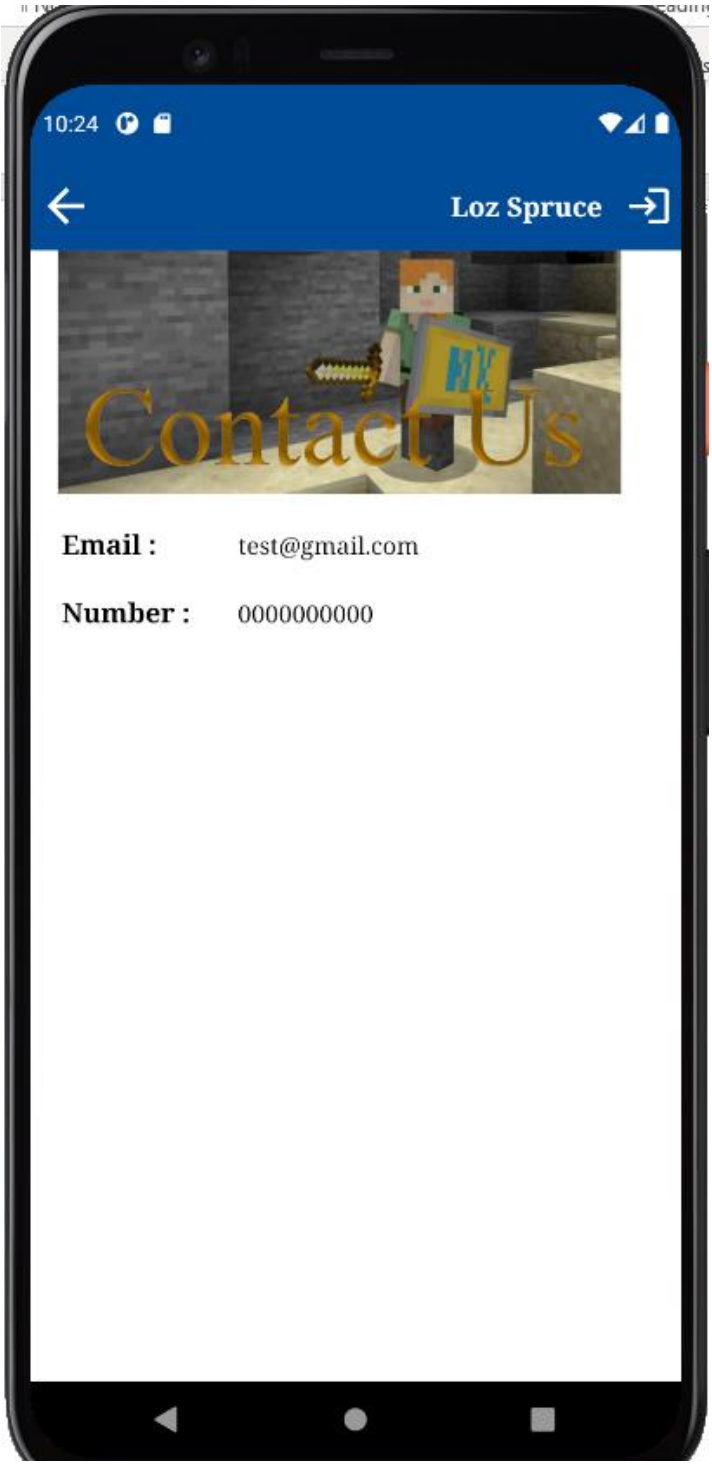


Figure 11: UI: Contact Us Activity

2.4.10 About Us

The about us page gives the user details on the purpose of the application and how to use it. It also informs users about the navigation bar and how to navigate around the application. Another custom banner is used to make the activity more appealing to the eye.

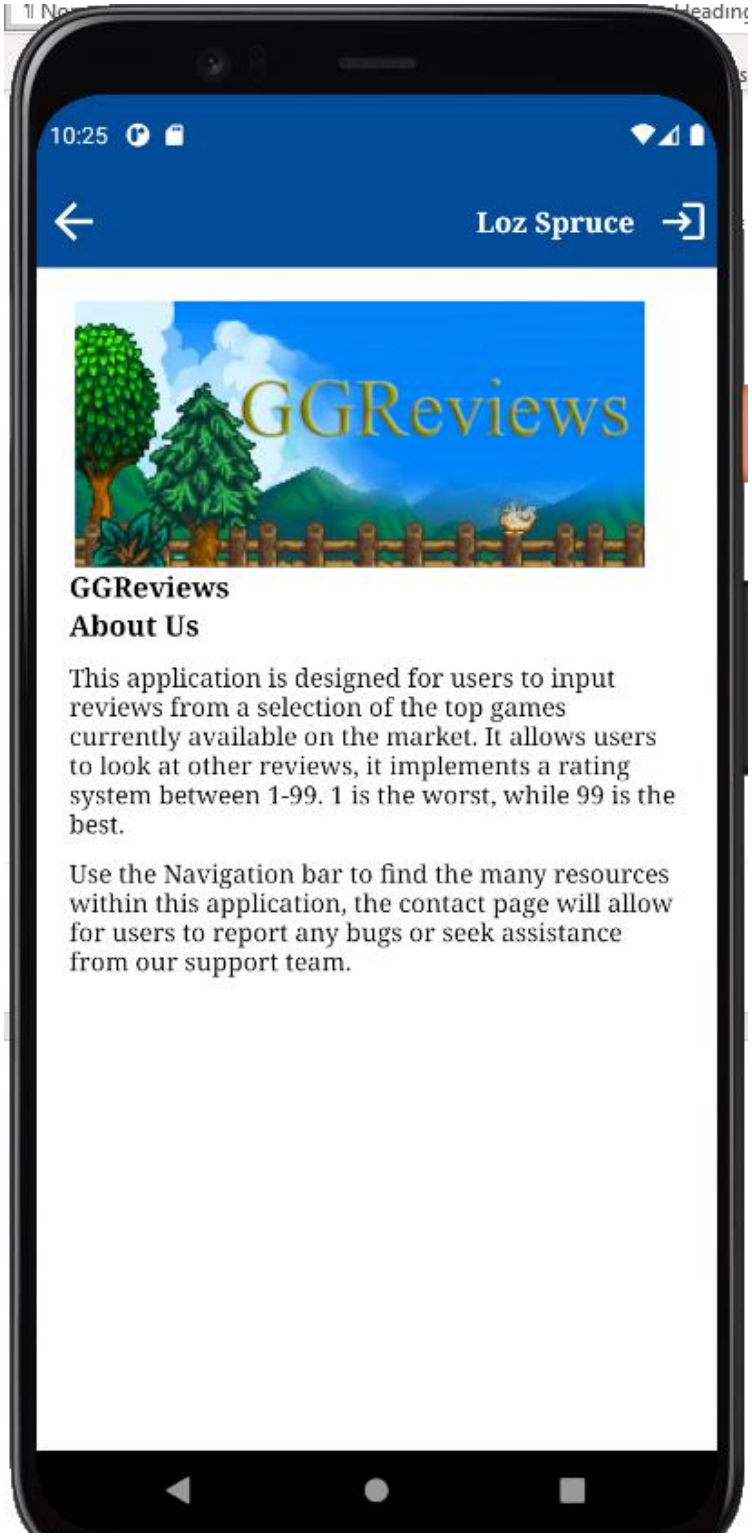


Figure 12: UI: About Us Activity

2.4.11 Review Confirmation

This activity is displayed after a user has successfully submitted a review. It gives the user an acknowledgement message, along with the option to use all reviews button to be redirected to their new review, alongside other user's reviews.

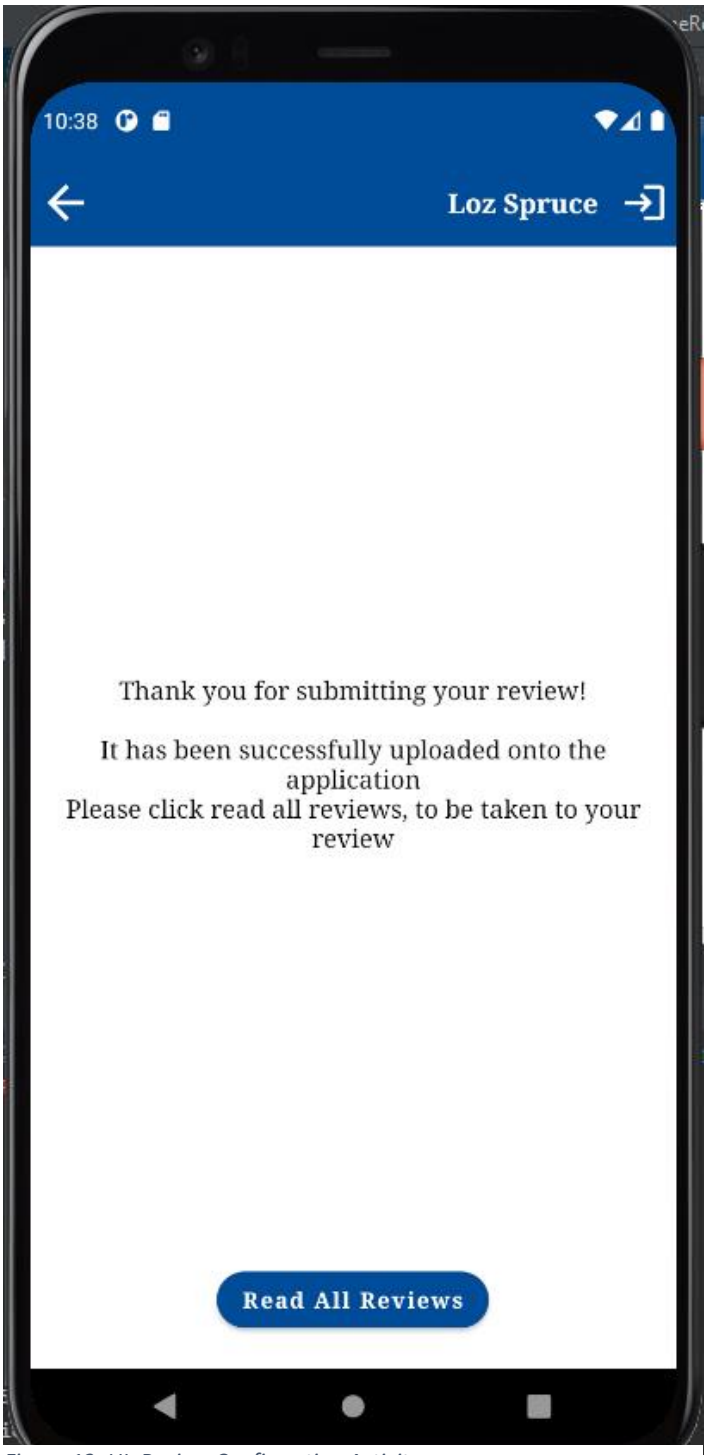


Figure 13: UI: Review Confirmation Activity

2.4.12 Logo Design

This is GGReviews main logo, which is displayed on the splash activity, when the application is started. It is also displayed on the main navigation bar. This logo was designed in photoshop and then imported into the design file in android studio.

Figure 14: UI: Logo Design

2.5 Class Diagram

The class diagram of GGReviews is big, as there are many classes and files involved in the development of the application. I will be showcasing a few of the class diagrams which were involved.

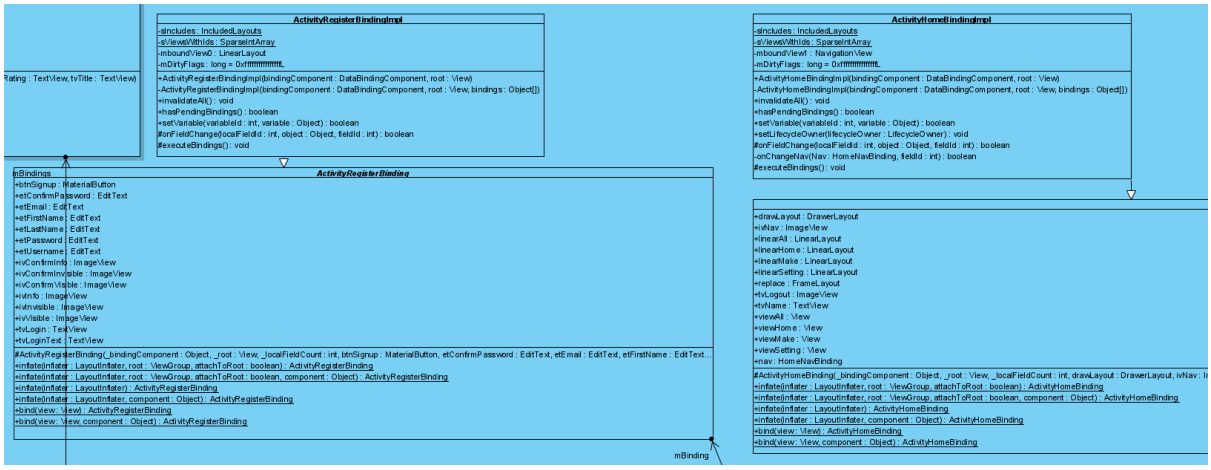


Figure 15: Class Diagram 1

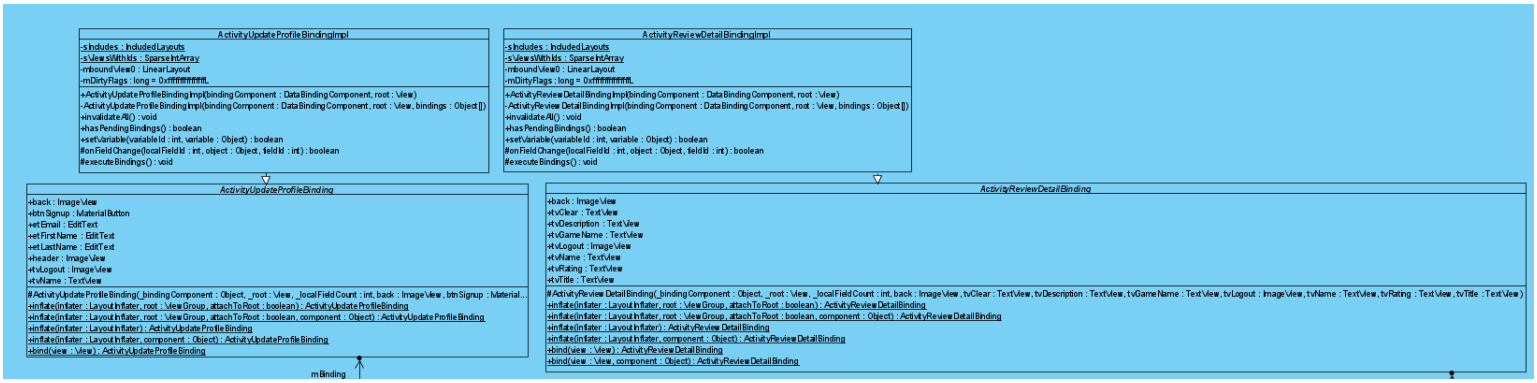


Figure 16: Class Diagram 2

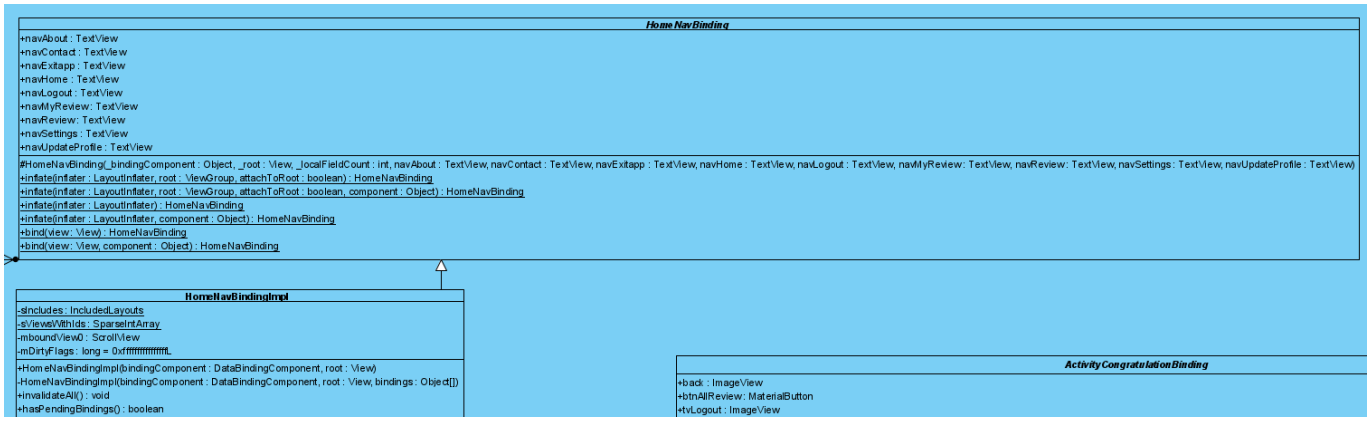


Figure 17: Class Diagram 3

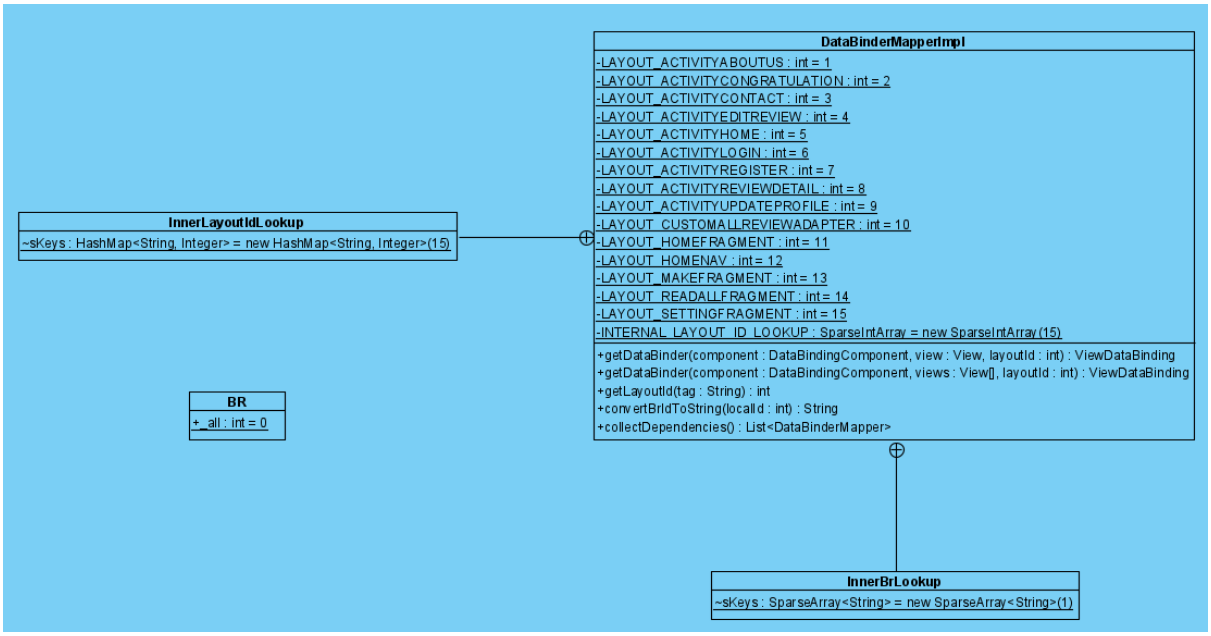


Figure 20: Class Diagram 4

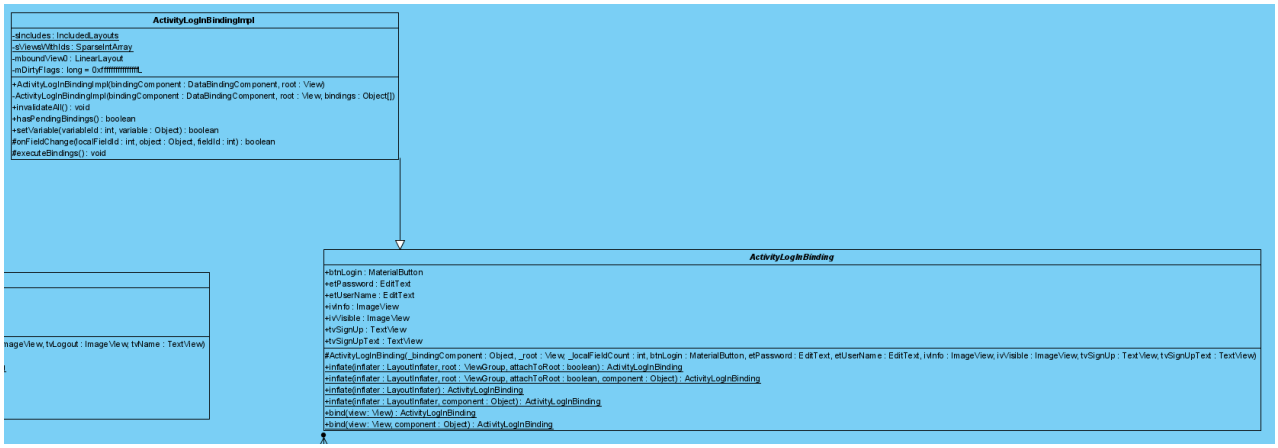


Figure 19: Class Diagram 3

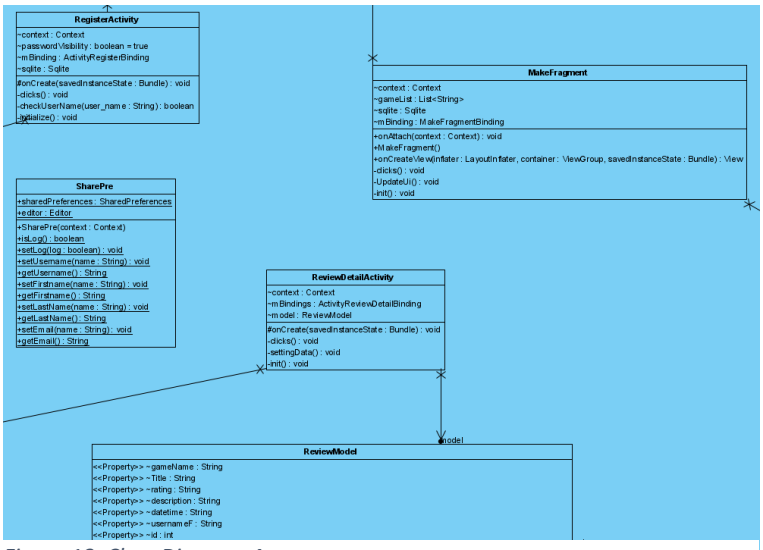


Figure 18: Class Diagram 4

2.6 Functional and Non-Functional Requirements

Functional Requirements of GGReviews

1. Login Activity requires user input of username and password
2. Register Activity requires user input of username, first name, last name, email, and password.
3. User can update their account
4. User can create a review
5. User can update their review
6. User can delete their review
7. User can view all their own reviews
8. User can view all reviews

Non-Functional Requirements of GGReviews

1. Scalability: Application must be able to scale up or down depending on how many new users and reviews are being created.
2. Performance: The application must be responsive to SQLite CRUD operations.
3. Security: The application will have strong password control when users set passwords while registering.
4. Data integrity: The SQLite and application data stays the same and consistent.
5. Availability: The SQLite database can be accessed at any time, the tables users and reviews are all manageable by the applications admin and support team.

3: Testing and Evaluation

3.1 Log in

User's which have already successfully created an account will be able to use the login page. They must enter their Username and password they chose while registering to the site. Once the user has successfully logged in, it will redirect them to the home activity. The user must click the login button, once clicked, if the login has been successful then the home activity will start.

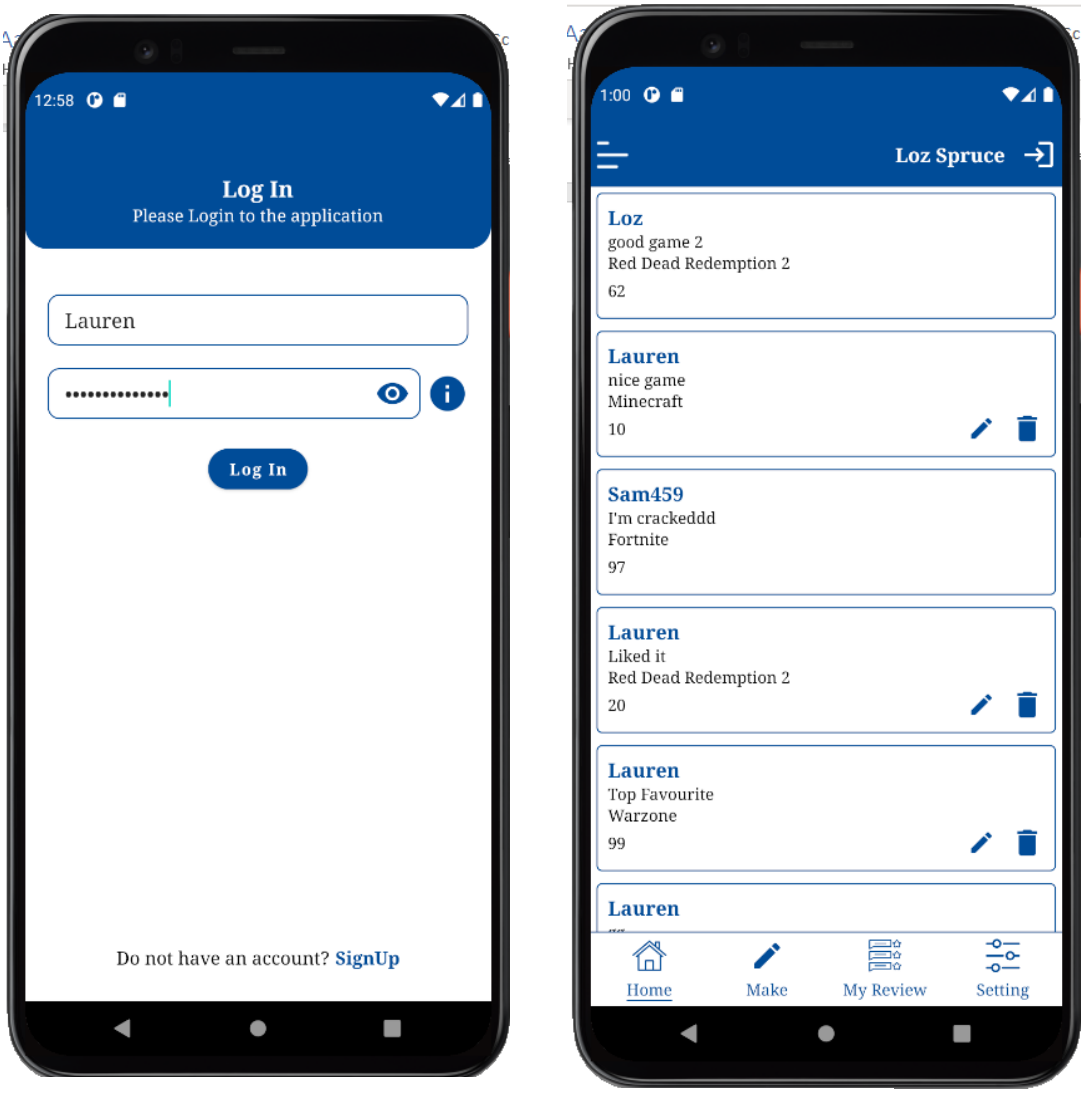


Figure 21: Login activity and the homepage being displayed after successful login

3.1.1 Incorrect Username or Password

If the user types in the incorrect username or password the following toast error message will be displayed, prompting the user to try again.

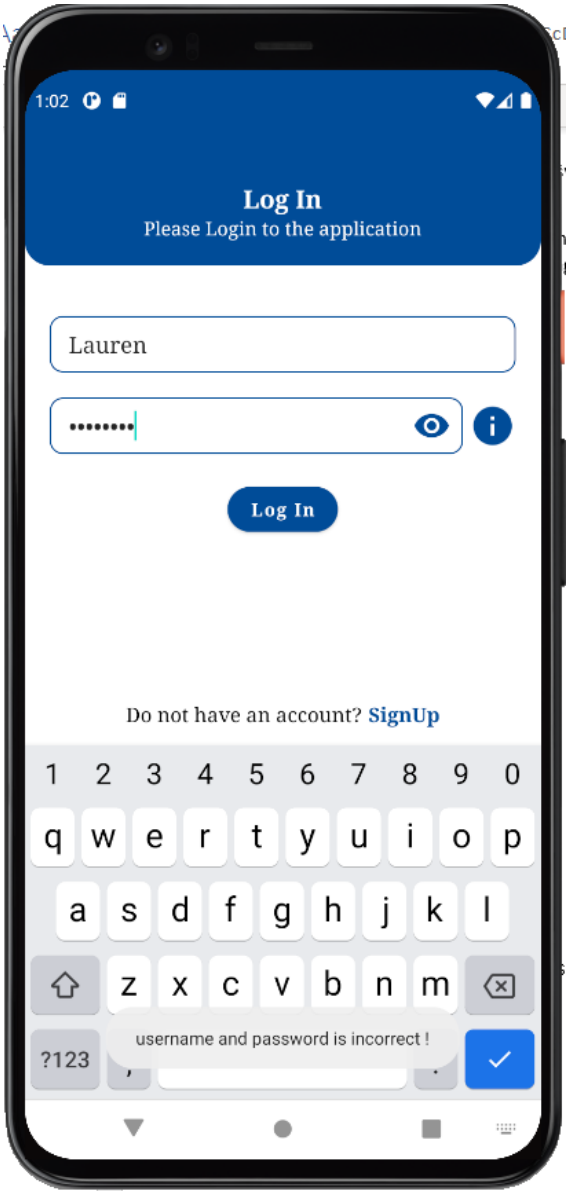


Figure 22: Username and Password validation

3.2 Registration

I will now be showing the registration activity. User's will be able to click 'Sign up,' from the login activity. Registration will allow for new users to create a profile by filling out all the required personal credentials. I will be showing some test's done within this page. Below is the UI design of the Registration form. User's must enter all field's correctly, otherwise the following error messages will be displayed. Once user has successfully entered all information correctly, the following dialogue box will appear on the below screenshot. The user's credentials will then be updated within the database within SQLite.

tableregister

Refresh table | Live updates

	id	firstname	lastname	username	email	password
1	1	Lauren	Spruce	Loz	lolspruce@hotmail.com	Lollipop45901.!
2	2	Loz	Spruce	Lauren	lolsperf@gmail.com	Charlie45901.!
3	3	Sam	Spruce	Sam459	Sam459@outlook.com	Charlie45901.!
4	4	Test	1	Test1	test@gmail.com	Charlie45901.!

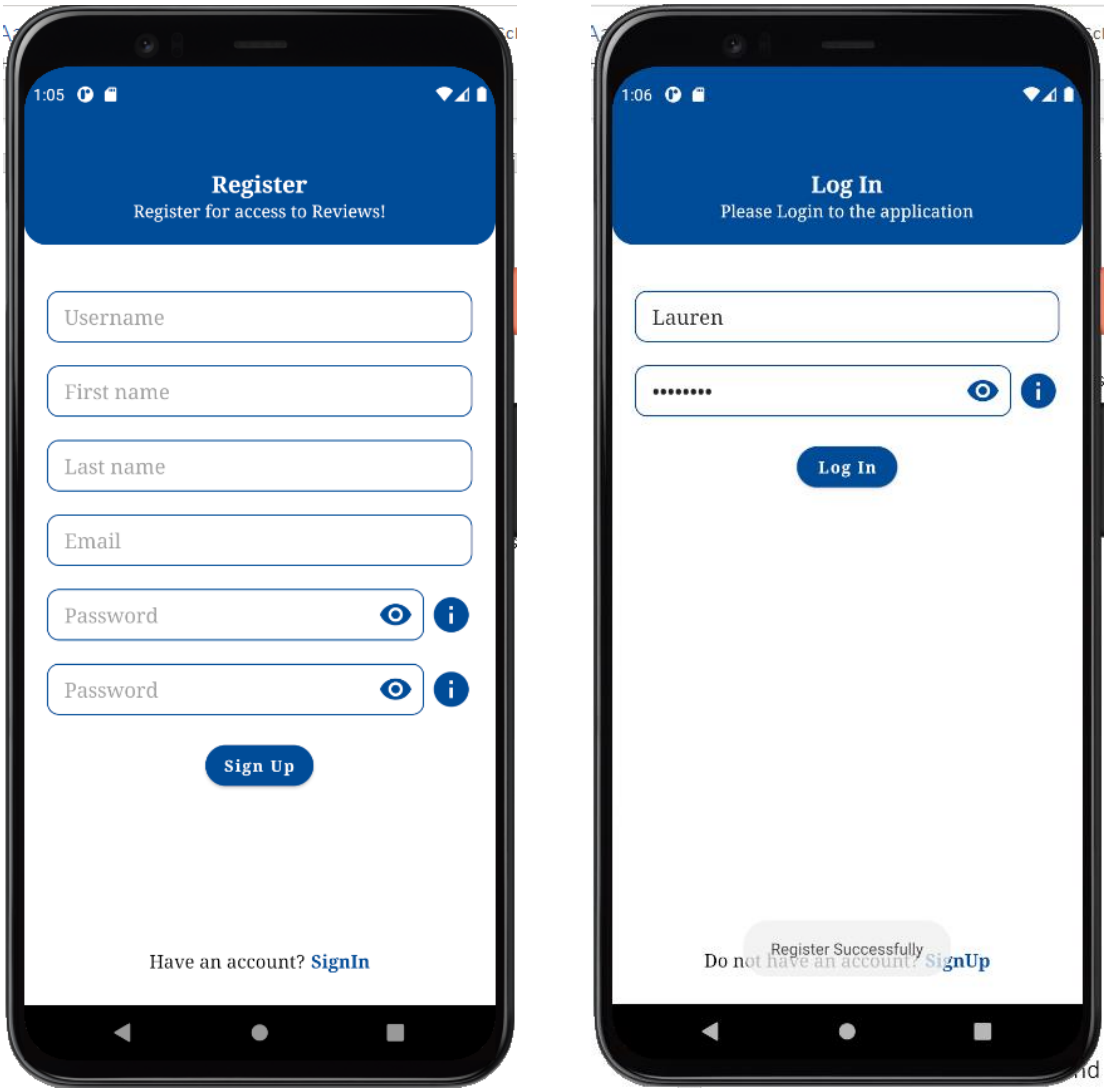


Figure 23: Registration testing

3.2.1 Password and Confirming password not matching

If the user does not enter the same password within each field, the following error toast message will be shown to the user. This prompts the user to retry until successful.

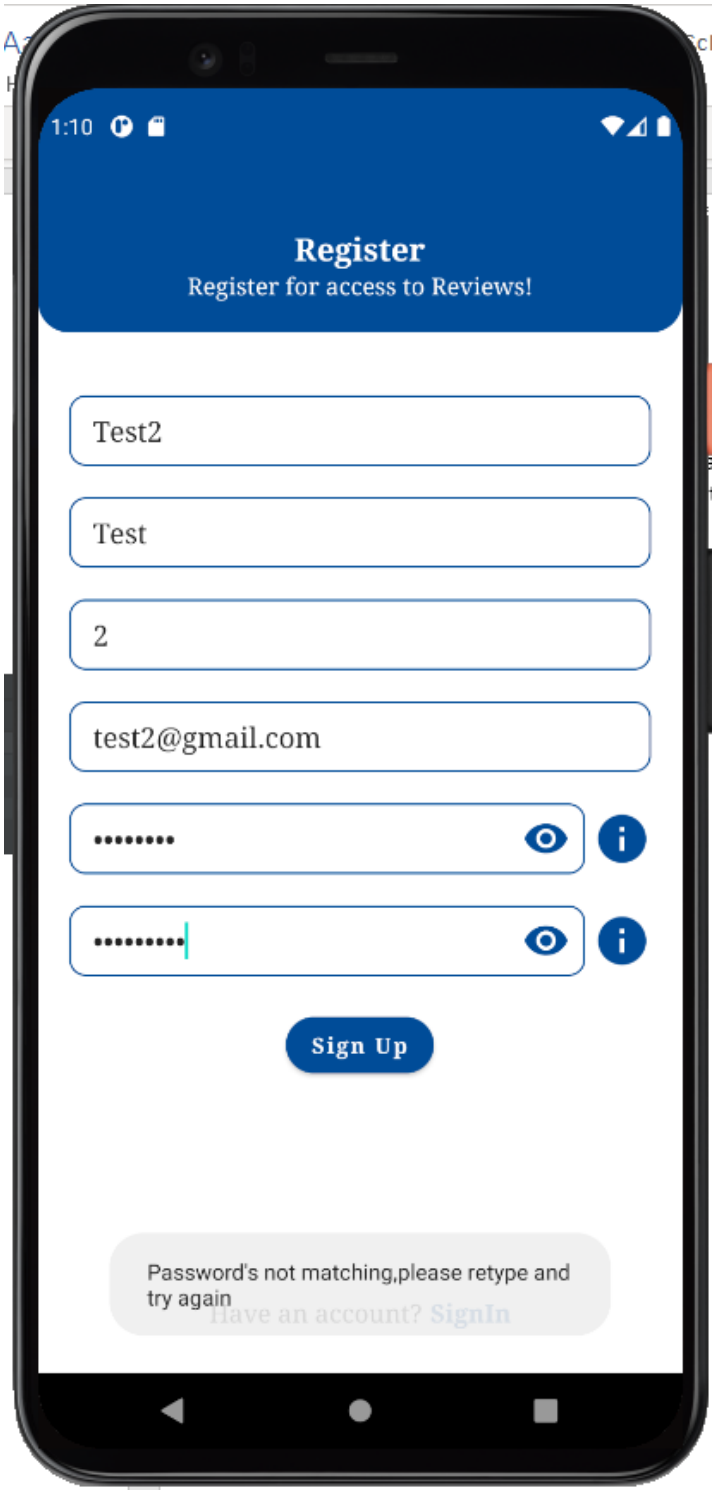


Figure 24: Password validation

3.2.2 Leaving fields empty

If the user leaves any field blank the following error message will appear. This is due to all fields being required to create the account. For the screenshot below, I leave all fields empty while trying to create the account. The appropriate message will appear for each of the empty fields if one is empty or all of them.

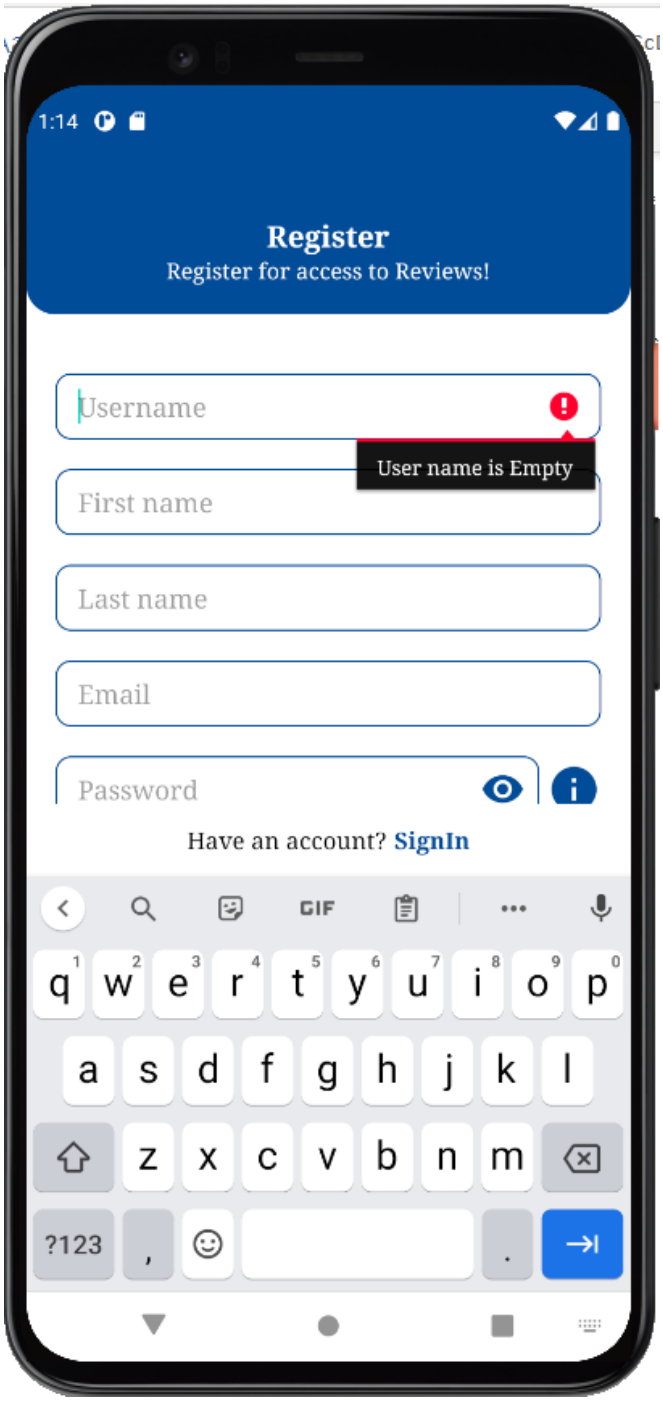


Figure 25: Empty field validation

3.2.3 Incorrect email format

If the user does not use the correct format for the email field it will display the following error message.

Figure 26: Email validation

3.3 Logging out

To log out, the user must click the top right hand logout icon, this will display the message, “Are you sure you want to log out of this session?” If the user clicks ‘yes,’ it will start the login activity. If the user selects ‘no,’ then the logout is cancelled and dismissed.

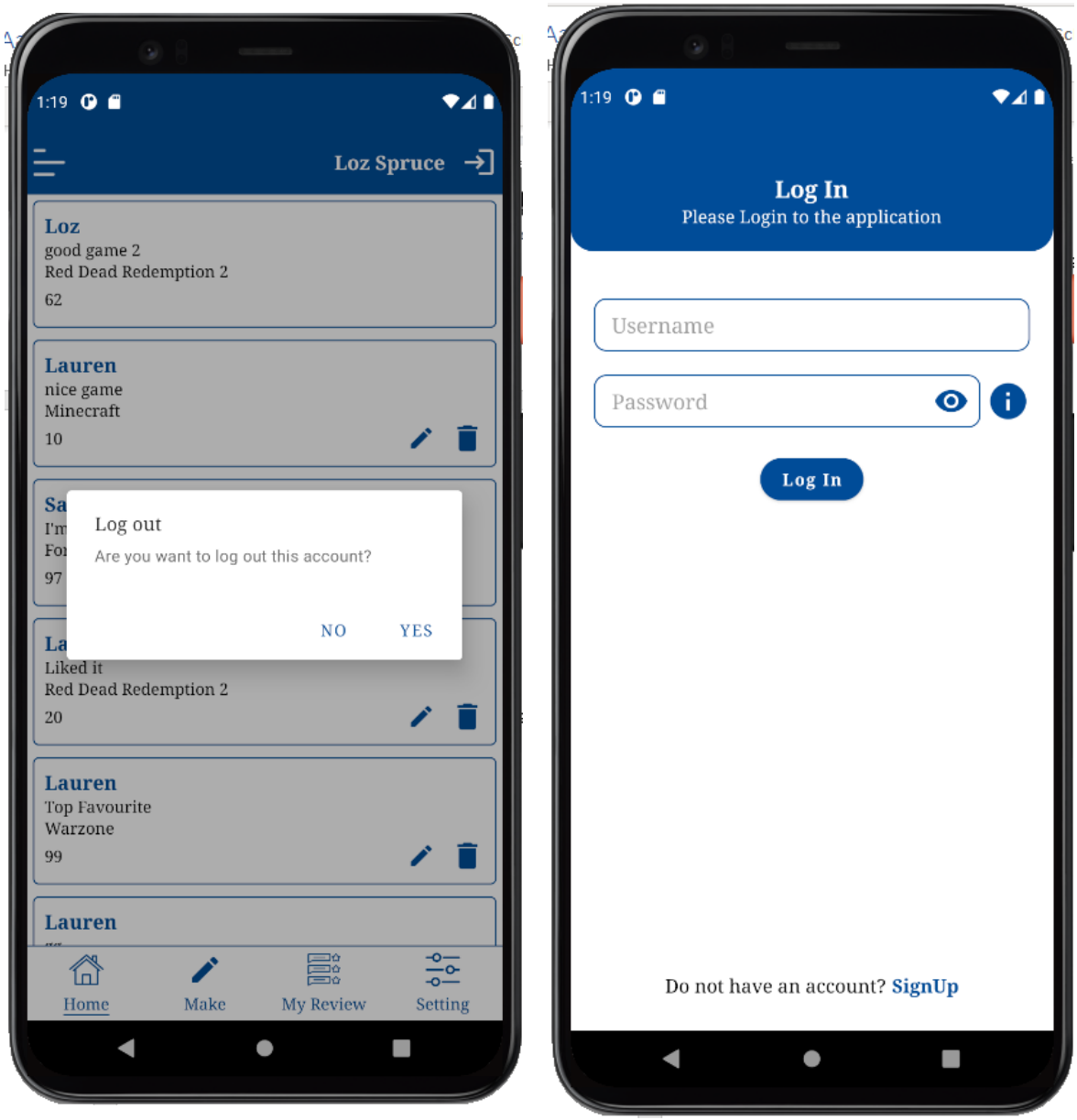
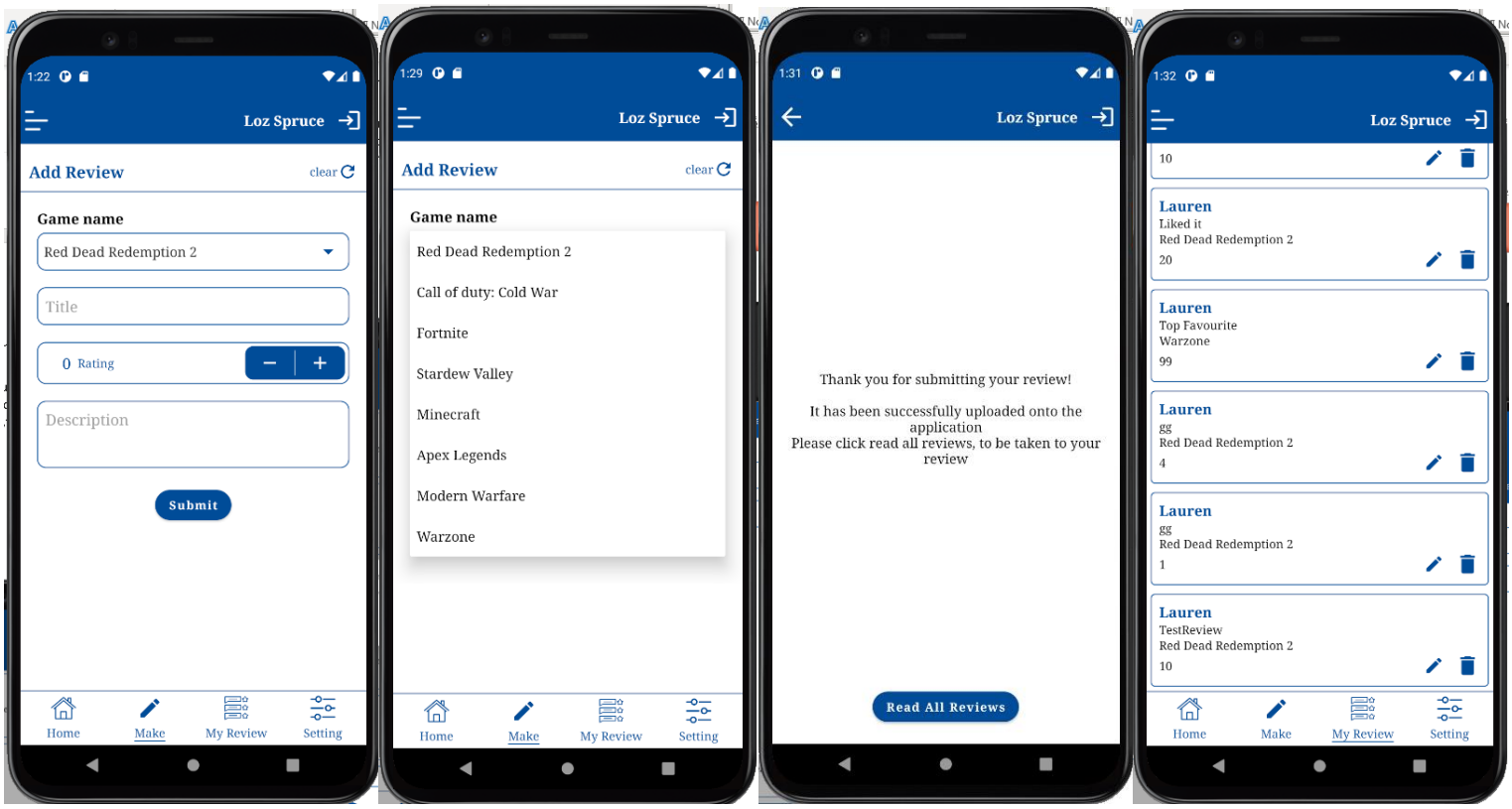


Figure 27: Logout implementation

3.4 Creating a review

The user must navigate to the ‘make’ icon on the bottom navigation, where they then can input the review data. A drop-down box shows the games list, users can select a game from there. After inputting review, they can press submit, once the review is submitted, a confirmation activity will appear, and the review will now be stored within the database and listed by ‘read reviews.’



tableregister | tablereview

Refresh table | Live updates

	id	reviewtitle	reviewrating	gamename	reviewdescription	username
1	1	good game 2	62	Red Dead Redemption 2	Good gg	Loz
2	2	nice game	10	Minecraft	Liked it	Lauren
3	3	I'm crackeddd	97	Fortnite	My name is Justin and I'm	Sam459
4	4	Liked it	20	Red Dead Redemption 2	It was okay	Lauren
5	5	Top Favourite	99	Warzone	My favourite game right now	Lauren
6	6	gg	4	Red Dead Redemption 2	good	Lauren
7	7	gg	1	Red Dead Redemption 2	ok	Lauren
8	8	TestReview	10	Red Dead Redemption 2	Test	Lauren

Figure 28: Creating a review

3.4.1 Empty Fields and Rating System

If any fields are left empty, an appropriate message will display to inform the user to fill in all details. Also, if the review is over 100, the correct error message will display to inform the user to use values between 0-100.

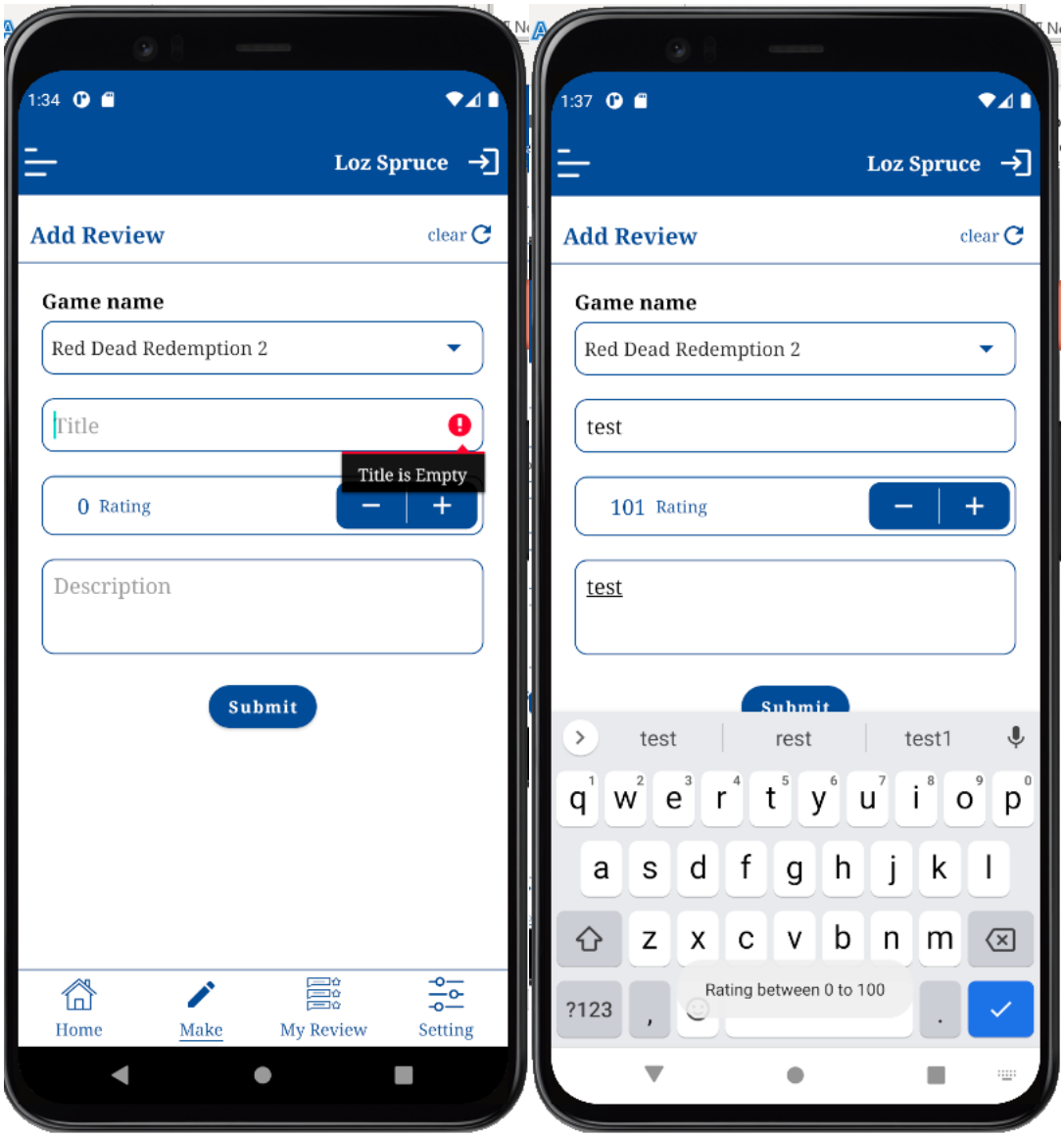


Figure 29: Empty fields and rating validation

3.5 Reading all Reviews

The user can view all reviews by clicking on the home activity. Here, all reviews will be listed, from all users. Note that the user can only update and delete their own reviews, so therefore only update, and delete icons will appear next to their own reviews.

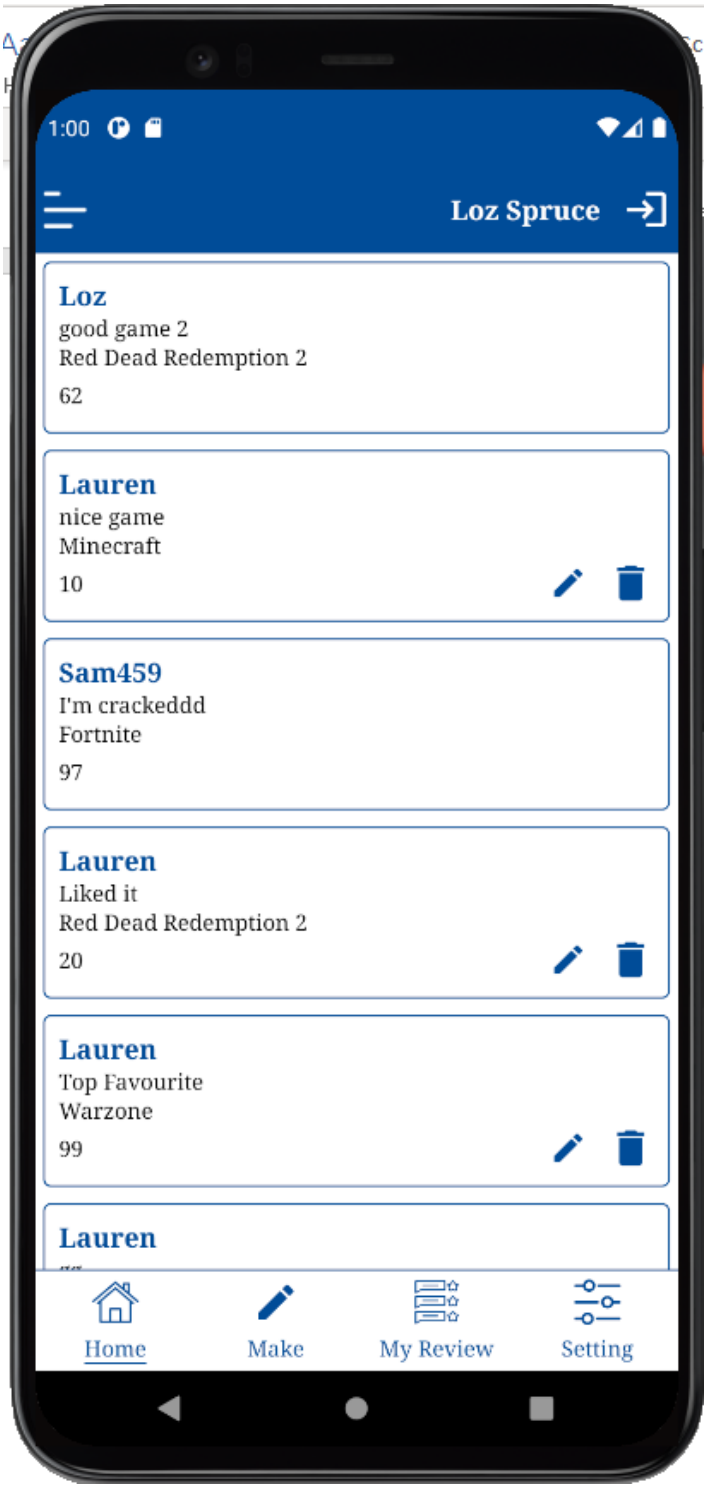


Figure 30: Viewing all reviews

3.6 Reading current user reviews

Current user reviews can be accessed from ‘My review,’ in the bottom navigation bar, also within the main navigation bar. Here, users can update and delete their reviews.

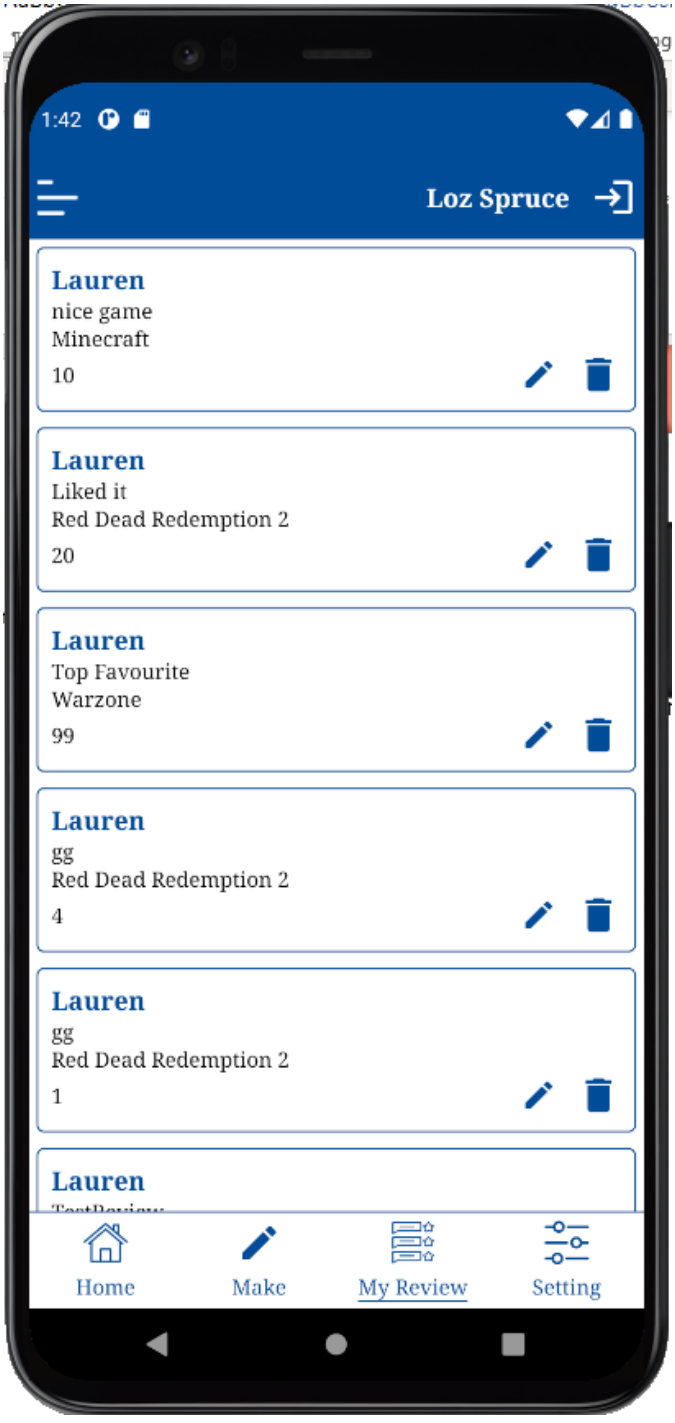
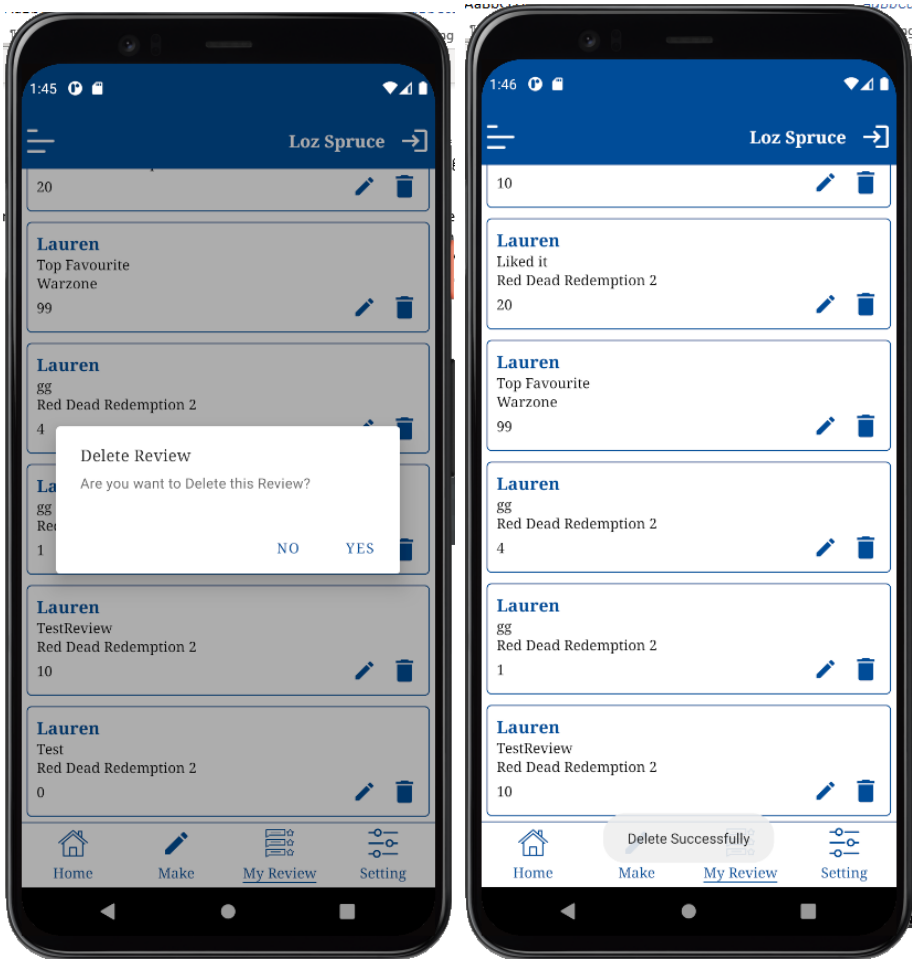


Figure 31: Viewing current user reviews

3.7 Deleting a review

Now I will be testing delete all reviews function. First, we will look at 'TestReview' within the database. This is located at the bottom of the screenshot. Now, the user must locate 'TestReview' within the 'My Review' tab. Once the review has been found, the user must click the delete icon, which will show the message, "Are you sure you want to delete this review?" If yes, the review is deleted, this is updated within the SQLite database. If 'no' is selected, then the review is kept.

	id	reviewtitle	reviewrating	gamename	reviewdescription	username
1	1	good game 2	62	Red Dead Redemption 2	Good gg	Loz
2	2	nice game	10	Minecraft	Liked it	Lauren
3	3	I'm crackeddd	97	Fortnite	My name is Justin and I'm	Sam459
4	4	Liked it	20	Red Dead Redemption 2	It was okay	Lauren
5	5	Top Favourite	99	Warzone	My favourite game right now	Lauren
6	6	gg	4	Red Dead Redemption 2	good	Lauren
7	7	gg	1	Red Dead Redemption 2	ok	Lauren
8	8	TestReview	10	Red Dead Redemption 2	Test	Lauren



tableregister | tablereview

Refresh table | Live updates

	id	reviewtitle	reviewrating	gamename	reviewdescription	username
1	1	good game 2	62	Red Dead Redemption 2	Good gg	Loz
2	2	nice game	10	Minecraft	Liked it	Lauren
3	3	I'm crackeddd	97	Fortnite	My name is Justin and I'm	Sam459
4	4	Liked it	20	Red Dead Redemption 2	It was okay	Lauren
5	5	Top Favourite	99	Warzone	My favourite game right now	Lauren

Figure 32: Deleting user review

3.8 Updating a review

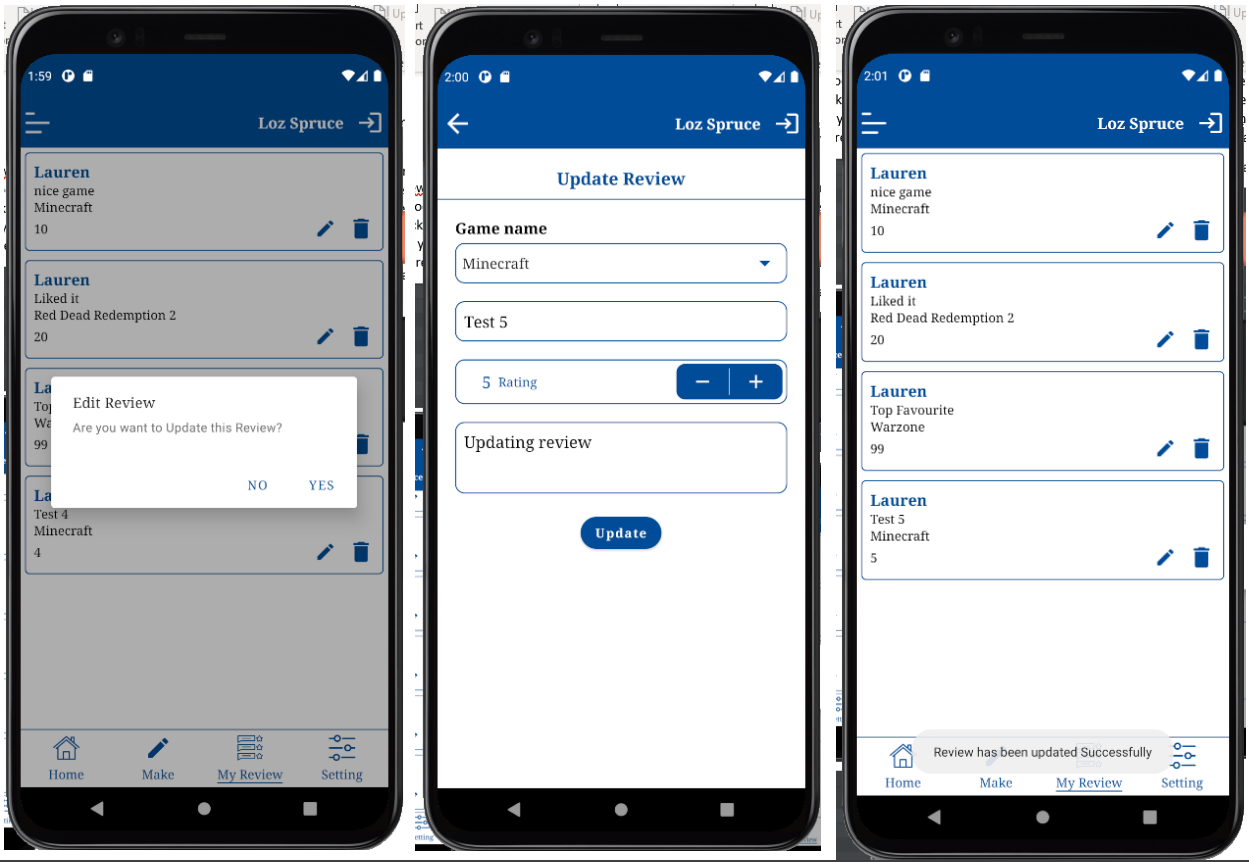
Now, updating a review will be tested. The user can click on their own review to update it. The update review activity will start, in which the user can input new review data. Once the review has been changed, the user can click the update review button. This will display a dialog message asking the user if they want to confirm changes. If yes, then the review is changed, and this is updated in the SQLite database. If no, then no changes are made.

I will be updating the 'Test 4' to 'Test 5' review for this demonstration.

tableregister | tablereview

Refresh table | Live updates

	id	reviewtitle	reviewrating	gamename	reviewdescription	username
1	1	good game 2	62	Red Dead Redemption 2	Good gg	Loz
2	2	nice game	10	Minecraft	Liked it	Lauren
3	3	I'm crackeddd	97	Fortnite	My name is Justin and I'm	Sam459
4	4	Liked it	20	Red Dead Redemption 2	It was okay	Lauren
5	5	Top Favourite	99	Warzone	My favourite game right now	Lauren
6	10	Test 4	4	Minecraft	Updating review	Lauren



tableregister | tablereview

Refresh table | Live updates

	id	reviewtitle	reviewrating	gamename	reviewdescription	username
1	1	good game 2	62	Red Dead Redemption 2	Good gg	Loz
2	2	nice game	10	Minecraft	Liked it	Lauren
3	3	I'm crackeddd	97	Fortnite	My name is Justin and I'm	Sam459
4	4	Liked it	20	Red Dead Redemption 2	It was okay	Lauren
5	5	Top Favourite	99	Warzone	My favourite game right now	Lauren
6	10	Test 5	5	Minecraft	Updating review	Lauren

Figure 33: Updating Review

3.9 Updating User Account

To update a user account, the user must navigate to 'Update Account' within the navigation bar. Once loaded, the user can input any changes to their first name, last name, and email. Once updated, the user can then click the update button. Once updated, a success message will appear, and the settings activity will launch. The account data will have been updated in the SQLite database. The account we will update is 'Loz Spruce,' username 'Lauren.' We will change the email and first name, which is shown below

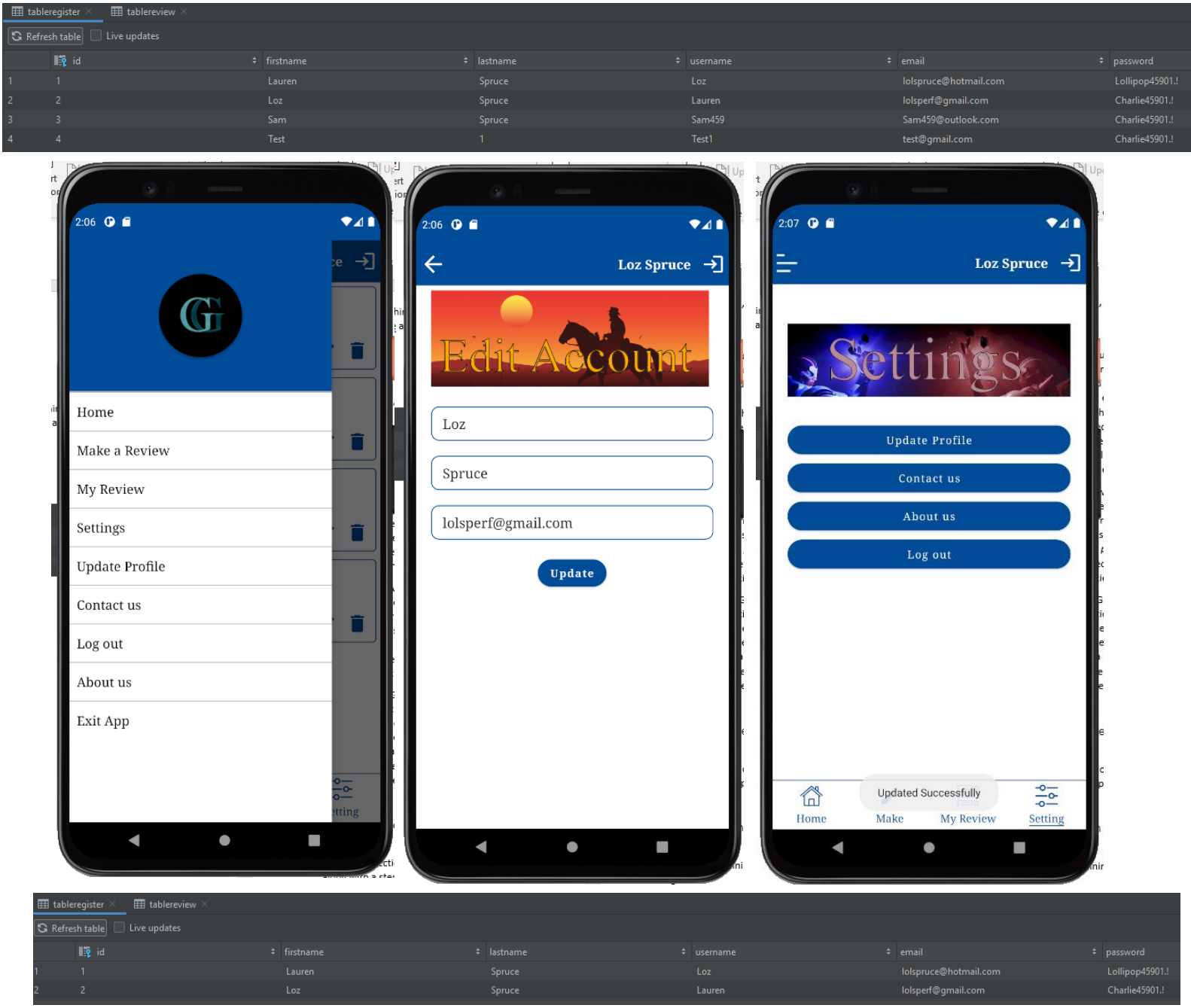


Figure 34: Updating Account

3.9.1 Empty Fields

If there are any empty fields, an appropriate message will be displayed informing users to input data.

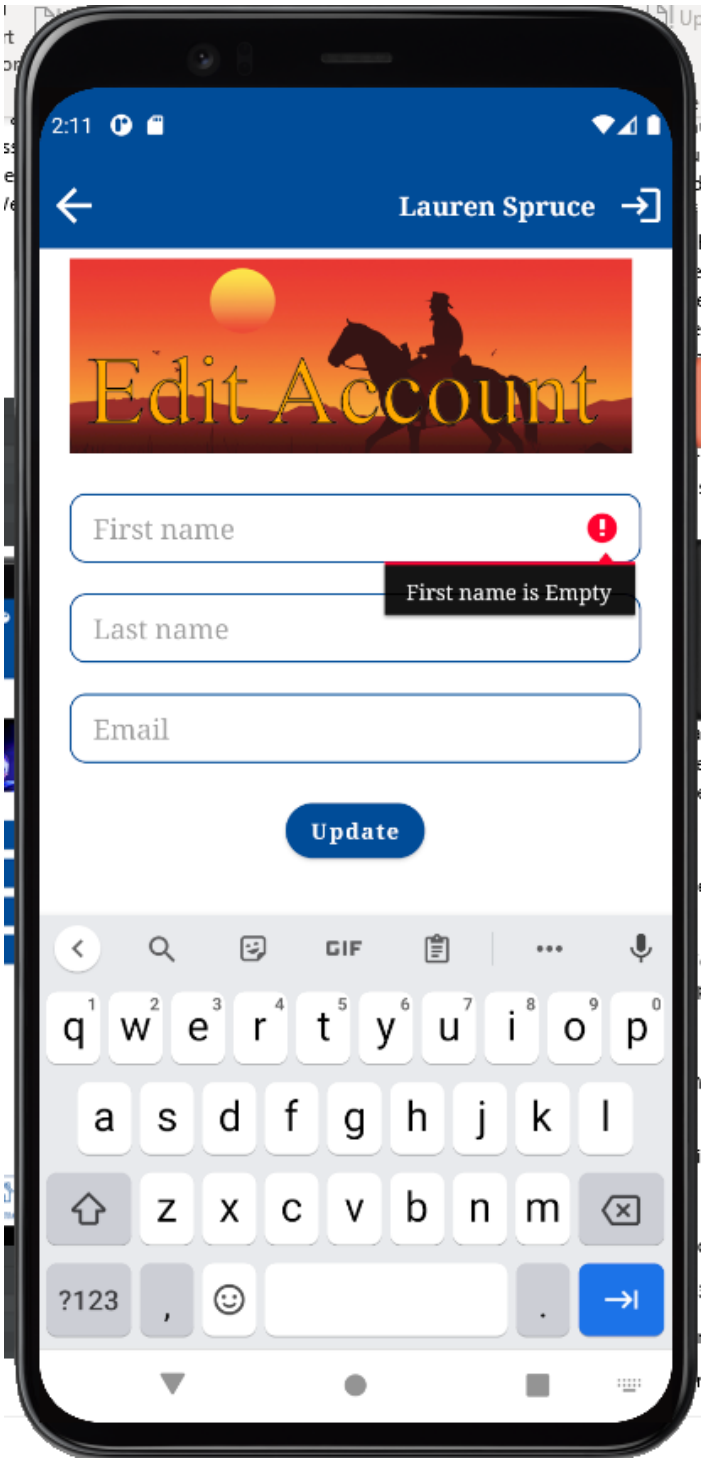


Figure 35: Empty fields update profile

4: Usability, Creativity and Reflection

GGReviews is an android application which is developed for consumers using an android device, these devices must be able to run 4.0 android onwards. GGReviews is also an application which allows for consumers to freely review games and read other reviews. The application has a professional and simple layout, also uses 2 navigation bars to allow users to navigate around the application with ease. It includes an about us page, which informs users on the details of the application, so therefore users have a solid idea on what the applications purpose is. All the banners and pictures used within GGReviews relate to games, the banners were edited using photoshop, these images are ones taken from some of the games that users may wish to review on the application. It relates back to the purpose of the site, and the bold and coloured text within the banners helps them stand out within each activity.

To use GGReviews, users will have to enter and provide personal information which will be stored within the SQLite database. If this application were ever deployed on the market, I would have to consider the current laws associated with Data Protection. For example, Microsoft teams have plenty of policies and security implemented, like Data Loss Prevention (DLP) and Information Barriers. *"These policies can impact users in 1:1 chat, group chats, or at a team-level."* (Microsoft, 2020) This protects users from their information being accessed by unwanted users and used for malicious activities.

If I were to put GGReviews on the market, I would add the following features. First, I would provide the user the option to delete all their data from the application, to ensure I was complying with the UK law, the General Data Protection Regulation. Secondly, I will ask the user before a successful registering of their account if they want their data to be used for further research and analytics, giving the user a choice. Finally, when registering, the user would be provided information on how their data will be stored and used in the application, giving them a clear understanding and chance to back out if they did not consent.

4.1 User Guide

Within this section of the report, the minimum software and hardware requirements will be shown, along with a step-by-step guide on how to install and deploy the application.

4.1.1 Minimum Requirements

Disk Space: Minimum 1GB HDD or SSD

Processor:

Intel: i3 Processor, or higher

AMD: Ryzen 3 1300X or higher

Operating System:

Windows 7 or preferably the latest version, Windows 10

Internet Connection: **Not required**

4.1.2 Software Requirements

User's will need these required software's installed on their PC:

- Android Studio
- SQLite Database

4.1.3 Installing

1. First, the user must download all necessary software to successfully run the application. This is listed within '4.1.2 Software Requirements.'
2. Once all software is downloaded, the user must download the Project file from my Weblearn submission, this is the 'GameReview' zip file. This must be extracted before opening within android studio.
3. Once the project file has been downloaded onto the user's PC, the user must then open Android studio. Once opened, the user must select 'Open project,' and click on the application.

4.1.4 Running the Application

1. Once the user has opened the file and android studio has finished downloading updates and content, the user must navigate over to the AVD drop down list. The user must select a suitable AVD to emulate the application onto.

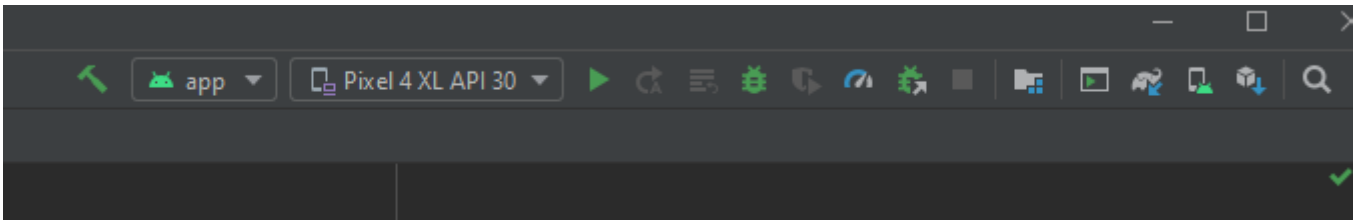


Figure 36: AVD Drop down menu

2. Once a suitable AVD has been selected, the user must then click on the 'Run application' button, which is located next to the AVD selector. Once the user has clicked this, they can check the build output, which will display that the build was successful. This is located at the bottom of android studio. The application can also be checked within the run window.

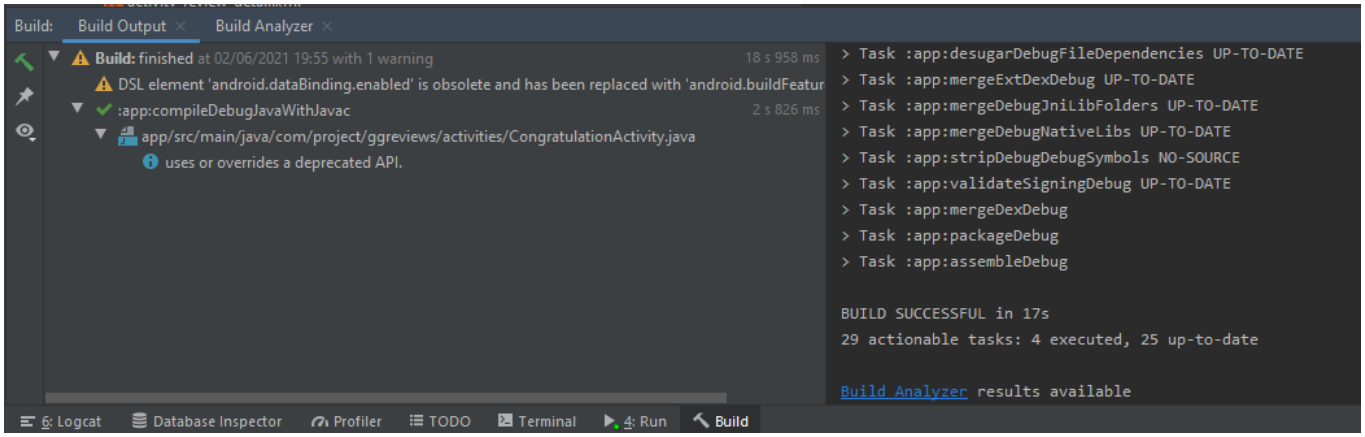


Figure 37: Android Studio Build Output

3. Once the build has been completed, the AVD will open on the user's navigation bar on their PC. Once the user clicks this, it will display the splash screen with the GGReviews logo. This will then show the login screen to the user, where they can log into the application.

   Alternatively, the user can check on their AVD that the application is there, by navigating to their home screen. One the list of applications, they should see GGReviews as an application, recognised by the logo.

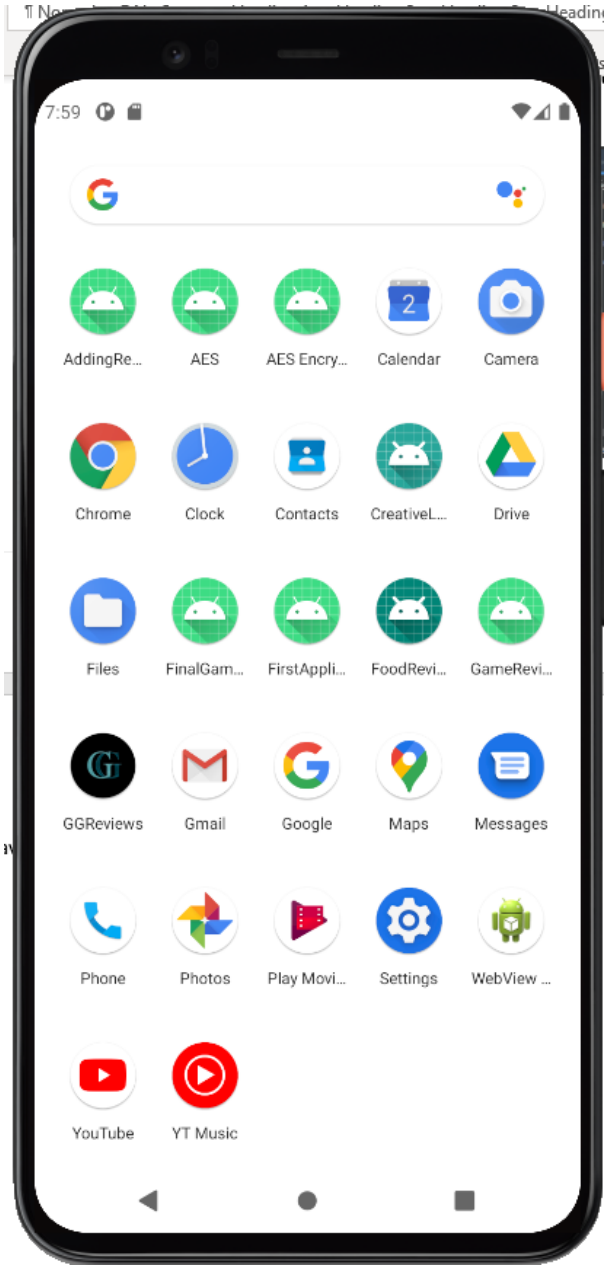


Figure 39: AVD Application List

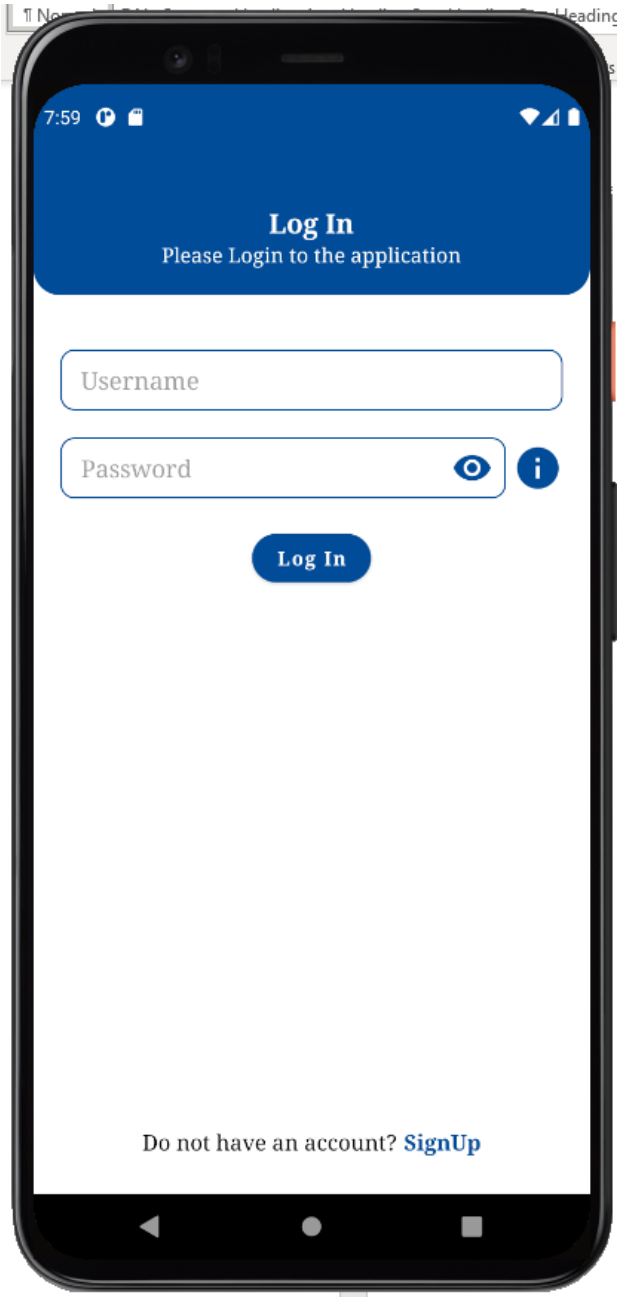


Figure 38: GGReviews Application Start Up

5: Conclusion

In conclusion, I found the development of GGReviews to be both rewarding and challenging. This is due to only recently being exposed to android studio as an IDE, and the fact that this is my first ever functioning mobile application.

Within GGReviews, I implemented a database, using SQLite and Java programming, as well as using XML to design the overall layout and looks of the application. This application allows game consumers to input feedback relating to current on market games and allow a consumer to read other user's reviews. It also allows for account customisation, updating and deleting reviews.

I felt like GGReviews accomplished its goal of being user friendly, as the layout is clean and professional. There are two main navigation bars, this way it is clear where each activity is, all activities are fully functioning and implemented correctly. I feel like the overall application design went well as I was able to create my own logo for the application, I felt like it gave the application a more unique touch and set it apart from the rest. I also enjoyed created the banners for each activity. However, if I were to complete this coursework again, I would spend more time designing, possibly change the background colour and add more images across the application.

I found that implementing the SQLite database was quite challenging, as it required a lot of methods for calling the data. I also found implementing multiple tables challenging as well, however, after completing this coursework I feel much more confident within SQL and database integration.

One problem I encountered was whilst using XML activity files. I had to experiment a lot with the different layouts available, in some cases even manually placing objects by dragging the arrows. This was very time consuming, however I learnt a lot about XML positioning because of it.

GGReviews has met all functional requirements as discussed in the proposal, however, I did modify the rating system from 0-10, to 0-100. I felt like this gave the user more freedom to choose a more specific rating. It allows for a more accurate review; I feel like this was a beneficial change to GGReviews.

Overall, I really enjoyed developing GGReviews, I have obtained many new skills relating to mobile application development, whilst polishing current database skills and Java coding. I feel like this coursework has inspired me to continue application development and I am excited to learn more.

6: References

Clement, J. (2021, June 1). *Number of active video gamers worldwide from 2015 to 2023*. Retrieved from Statista: https://www.statista.com/statistics/748044/number-video-gamers-world/#:~:text=The%20video%20gaming%20industry%20is,three%20billion%20gamers%20by%202023.

Microsoft. (2020, 11 12). *Security and compliance in Microsoft Teams*. Retrieved from Microsoft: https://docs.microsoft.com/en-us/microsoftteams/security-compliance-overview

Osis, J., & Donins, U. (2017). *Unified Process*. Retrieved from Science Direct: https://www.sciencedirect.com/topics/computer-science/unified-process#:~:text=Inception%3A%20defines%20the%20scope%20of%20the%20project%20and%20develop%20business%20case.&text=Elaboration%3A%20Plan%20project%2C%20specify%20features%2C%20and%20baseline%20t

ukie. (2021, March 21). *UK Games Industry Market Valuation 2020*. Retrieved from Ukie: https://ukie.org.uk/news/uk-games-industry-valuation-2020#:~:text=Game%20software%20revenues%20reached%20a,by%20format%20was%20Nintendo%20Switch.

Vicente, V. (2020, May 8). *What Does "GG" Mean, and How Do You Use It?* Retrieved from How-To-Geek: https://www.howtogeek.com/466547/what-does-gg-mean-and-how-do-you-use-it/