



South East Technological University

Functional Specification

Bridge Puzzle Game App for Elementary OS

Student Name: Ming Kit Choy

Student Number: C00246492

Table of Contents

1.	Introduction	5
2.	System Architecture	6
2.1	Optimizations	7
3.	Design Pattern	8
4.	Platform Integration	9
4.1	Leveraging Platform APIs	9
5.	Development Tools Integration	10
5.1	Vala Languages	10
5.2	Elementary OS	10
5.3	Doxygen	10
5.4	Firebase	11
6.	Adhering to Human Interface Guidelines	12
7.	Internationalization and Localization Design	13
8.	Sequences Diagrams	14
9.	Class Diagram	26
10.	Database Layout	27
11.	UI/UX	28
12.	Prototype Screens	29
12.1	Main Page	29
12.2	Login Page	30
12.3	Reset Password Page	31
12.4	Registration Page	32
12.5	Home Page	33
12.6	Play Game Page	34
12.7	Level Mode Page	35
12.8	Game Page	36
12.9	How to Play Page	37
12.10	Multiplayer Page	38
12.11	Invite Player Page	39
12.12	Add Players Page	40
12.13	Quick Match Page	41
12.14	Profile Page	42
12.15	Badges Page	43
12.16	Share Page	44

12.17	LeaderBoard Page	45
12.18	Setting Page	46
12.19	Feedback & Support Page	47
12.20	Inbox Page	48
12.21	Message Page	49
13	Conclusion	50

Table of Figures

figure 1 System Architecture Diagram	6
Figure 2 Sequence Diagram 1 - Registration	14
Figure 3 Sequence Diagram - Login	15
Figure 4 Sequence Diagram - Logout	16
Figure 5 Sequence Diagram - Play Game	17
Figure 6 Sequence Diagram - Multiplayer Game Mode	18
Figure 7 Sequence Diagram - View Profile	19
Figure 8 Sequence Diagram - View Leaderboard	20
Figure 9 Sequence Diagram - Share Badges	21
Figure 10 Sequence Diagram - Submit Feedback & Support	22
Figure 11 Sequence Diagram - View Inbox	23
Figure 12 Sequence Diagram - Report Error	24
Figure 13 Sequence Diagram - Manage Content	25
Figure 14 Class Diagram	26
Figure 15 Database Layout Diagram	27
Figure 16 Ui Colour Template	28
Figure 17 Prototype Screen - Main Page	29
Figure 18 Prototype Screen - Login Page	30
Figure 19 Prototype Screen – Reset Password Page	31
Figure 20 Prototype Screen - Register Page	32
Figure 21 Prototype Screen - Home Page	33
Figure 22 Prototype Screen - Game Option Page	34
Figure 23 Prototype Screen - Level Mode Page	35
Figure 24 Prototype Screen - Game Page	36
Figure 25 Prototype Screen - Rule Page	37
Figure 26 Prototype Screen - Multiplayer Page	38
Figure 27 Prototype Screen - Invite Player Page	39
Figure 28 Prototype Screen - Add Player Page	40
Figure 29 Prototype Screen - Search For Matching Page	41
Figure 30 Prototype Screen – View Profile Page	42
Figure 31 Prototype Screen - View Badges Page	43
Figure 32 Prototype Screen - Share Badges Page	44
Figure 33 Prototype Screen - View Leaderboard Page	45
Figure 34 Prototype Screen - Setting Page	46
Figure 35 Prototype Screen - Feedback & Support Page	47
Figure 36 Prototype Screen - Inbox Page	48
Figure 37 Prototype Screen – Message Page	49

1. Introduction

The primary objective of this design document is to provide a comprehensive overview of the Bridge Puzzle Game application. This document will outline the application's various features while also delving into the underlying architecture and design patterns that will be employed.

To better illustrate the system's performance, this document will include system sequence diagrams that detail the various interactions between the system components. Moreover, a class diagram will be presented to provide a detailed view of the system structure and the relationships between different classes.

In addition to the diagrams, this document will also contain a detailed database layout that will outline how data is being stored and managed. This will provide a clear understanding of the data flow within the system and how it is being processed.

Finally, screen layouts will be included as a prototype to demonstrate how the application will look and function in practice. This will help to provide a more realistic view of the application and how it will be used by end-users.

2. System Architecture

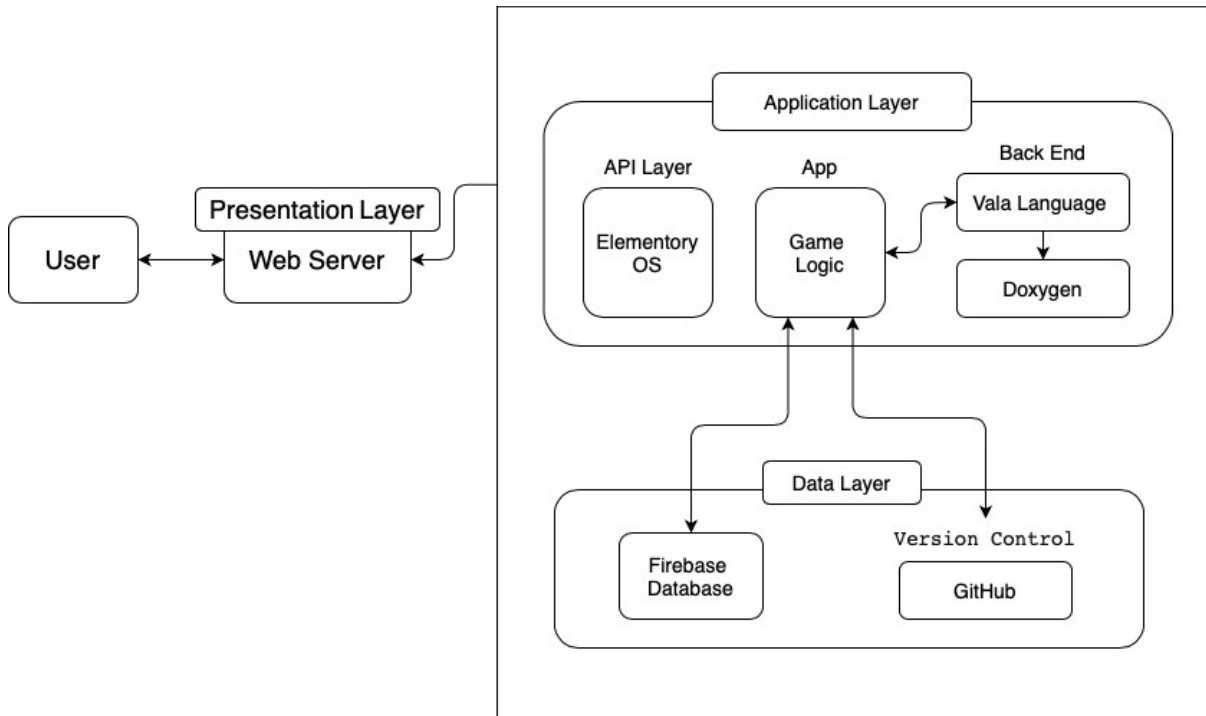


Figure 1 System Architecture Diagram

The user interface will be designed for players who interact with the game on Elementary OS. The Elementary OS will furnish a dependable desktop milieu for data acquisition and user management.

The application layer will consist of game logic, implemented in the Vala language, which will contain the core mechanics of the game. Doxygen will be utilized to generate documentation from the Vala source code.

The data layer will use the Firebase Database to store and retrieve game data, such as player scores and game state.

GitHub will be used for version control, collaborative work, and source code management.

2.1 Optimizations

To guarantee a seamless gaming experience on all devices, I will implement a range of strategies:

Caching: Utilized SQLite storage to cache game levels, badges, and other persistent entities locally, which eliminates the need for duplicate API fetches and ensures quicker load times. To maintain cache coherence, it has set up timeout durations and versioning to refresh stale data.

Profiling: The profiler tool tracks CPU/GPU usage metrics during gameplay. We have optimized performance-intensive sections, such as drawing complex level grids, using processor-efficient data structures and algorithms.

Animation: Taken advantage of hardware acceleration capabilities to render game effects and animations, which reduces the burden on the main processor.

3. Design Pattern

The Model-View-Controller (MVC) pattern is a widely used architectural pattern that is designed to break down an application into three interconnected components. These components are responsible for managing distinct aspects of the application logic.

1. **The Model** is responsible for managing the data, logic, and business rules of the application. This includes everything from level generation rules and bridge connections to user moves and scoring algorithms.
2. **The View** is responsible for defining how the user interface looks and appears to the players. It renders models into an interactive interface with components controlling UI elements like grids, bridges, and number displays.
3. **The Controller** handles inputs from the player and conveys required actions to the Model and View parts of the app by connecting the view inputs with appropriate model updates. For instance, tapping to add a bridge triggers model validation and view updates.

Benefits of MVC:

1. It allows for the separation of concerns, making it easier to develop and test business logic independently from UI. This means that models and views can be reused and replaced without affecting other components, making it simple to add capabilities over time.
2. It helps coordinate complex interactions between data and presentation layers by placing a central controller between them. This ensures that data is passed between the different components in a consistent and organized manner, making it easier to manage and maintain the codebase over time.

4. Platform Integration

4.1 Leveraging Platform APIs

The upcoming game will seamlessly incorporate Elementary grains and services. Specifically, it will utilize GtkScene game canvas, GtkGrid containers, and Granite widgets such as Avatar and toast notifications, providing an OS-consistent interface. Additionally, the contract provider in Contrax will allow for registering app actions for shortcuts, application menu contributions such as badges, and game discovery for store listings.¹

Sample Code:

```
var canvas = new Gtk.Scene ();
canvas.add_child(new Gtk.Grid());
var toast = new Granite.Widgets.Toast (_("Level complete!"));
```

Code copy

¹ Elementary, Inc. (2022). *Developer APIs and Docs*. <https://docs.elementary.io/develop>.

5. Development Tools Integration

5.1 Vala Languages

For this project, I have chosen to use the Vala Language due to its high performance and ease of integration with Elementary OS. Vala is a modern, high-level programming language, which simplifies and accelerates the development of applications for developers who are interested in creating desktop applications for the GNOME ecosystem.

Vala code is compiled into C code, making it highly compatible with existing C libraries and enabling efficient integration with the C ecosystem. Vala is closely integrated with GObject, which is the object system used in the GNOME platform. This language provides developers with a seamless way to create GObject-based components.²

5.2 Elementary OS

This project will use Elementary OS, an open-source software which is freely available for users to download, use and modify. Elementary OS is a Linux distribution that prioritises user-friendliness and offers a clean and intuitive interface. It is an ideal choice for users who want a polished and aesthetically pleasing computing experience. Additionally, Elementary OS is committed to user privacy and security, providing features such as data collection controls and regular security updates.³

5.3 Doxygen

Doxygen is a powerful and open-source tool that developers can use to generate code documentation written in various programming languages. It can automatically extract comments and documentation from the Vala code. It makes it more accessible for developers to create well-documented software.

Doxygen simplifies the process of creating consistent, organised, and easily navigable documentation for software projects, including source code comments, class hierarchies, and function references. It also generates clear and accessible documentation in HTML, PDF or other formats, giving users all the information they need to understand and use the software effectively.⁴

² Vala GitHub Repository: Access the Vala source code and contribute to its development on GitHub.

³ Elementary OS on GitHub: Access the source code and contribute to the development of the operating system on GitHub.

⁴ Doxygen on GitHub: Access the source code, report issues, and contribute to the development of Doxygen on GitHub.

5.4 Firebase

Firebase is a platform by Google that helps develop mobile and web apps. In this project, I will use Firebase as it serves Backend-as-a-Service (BaaS) and provides features such as real-time database, authentication, cloud hosting, cloud functions, storage and analytics.⁵

In this project, I will be utilizing Google Firebase's advanced real-time cloud database. This database will be used to store and synchronize game data across multiple devices, including user profiles, game state, progress, scores, and other persistent information. Firebase offers a highly adaptable and scalable database that comes with built-in authentication and offline functionality, making it an ideal choice for developing gaming applications. Furthermore, the application code will be optimized to ensure seamless integration of Firebase SDKs.

⁵ Firebase GitHub Repository: Access Firebase SDKs, samples, and open-source projects on GitHub.

6. Adhering to Human Interface Guidelines

To ensure a consistent user experience across all devices, the game will adhere to the Elementary HIG specifications for visual design, layout, iconography, and responsiveness.

Typography: To establish a clear hierarchy between titles, labels, and body content, all text will use the standard system font Noto Sans with appropriate sizes as per HIG. The text contrast levels will be maintained at a minimum of 7:1 against background colours, with dynamic colours for improved visibility. Custom game fonts may be used for specific UI elements to establish brand styling.

Grids and Spacing: Pixel-based grid guidelines will be used for margins, canvases, and alignment of buttons, cards, and other containers to ensure proper spacing and alignment. Whitespace, padding, and density will comply with specifications to reduce visual noise.

Iconography: Icons will follow style guidelines to be simple, intuitive, and consistent through components like toolbar buttons, side navigation, and dialogues. Custom game icons will be designed to be minimal and adaptable to themes, with colour accents to draw attention.

Responsiveness: The interface will dynamically scale the game grid, rearrange layouts, and adapt interactions across laptops, tablets, and mobiles. Text and touch targets will resize appropriately to be usable on smaller screens while retaining information hierarchy.

By rigorously complying with standard HIG practices, the game will mesh visually and functionally with the OS while maintaining high quality, usability, and accessibility across diverse system configurations.

7. Internationalization and Localization Design

The game is being designed with global users and multiple languages in mind, starting from the ground up.

Externalizing Strings: To ensure consistent localization of all user-facing text in the app, App. pot template files will be externalised using gettext utilities. The source code will reference the strings by using the gettext method for dynamic translations.

Translation Workflows: Translators can produce .po files that contain language-specific strings. These files can be merged to facilitate collaboration among multiple translators. Additionally, it is possible to send strings to third-party or community translation services.

Build Automation: The build and release pipeline for translations will generate binaries for all supported languages through GitHub workflows. The .po file's translated strings will be compiled into efficient .gmo files.

RTL Support: The UI will dynamically adjust layout mirroring and text alignment to seamlessly support right-to-left languages like Arabic without requiring code changes.

Pluralization: To support pluralization in different languages, multiple string variations can be provided by translators. At runtime, the correct string is programmatically selected based on the context.

Font Support: The game will automatically embed appropriate fonts for languages requiring special glyph support, such as Chinese and Japanese. Fallback fonts will handle unavailable glyphs.

Automation: Elementary emphasizes that automating translation compilation allows effortless scaling to multiple languages for developers.

With this comprehensive i18n/l10n readiness, launching the game globally will be accelerated by facilitating community translations and automated build workflows.

8. Sequences Diagrams

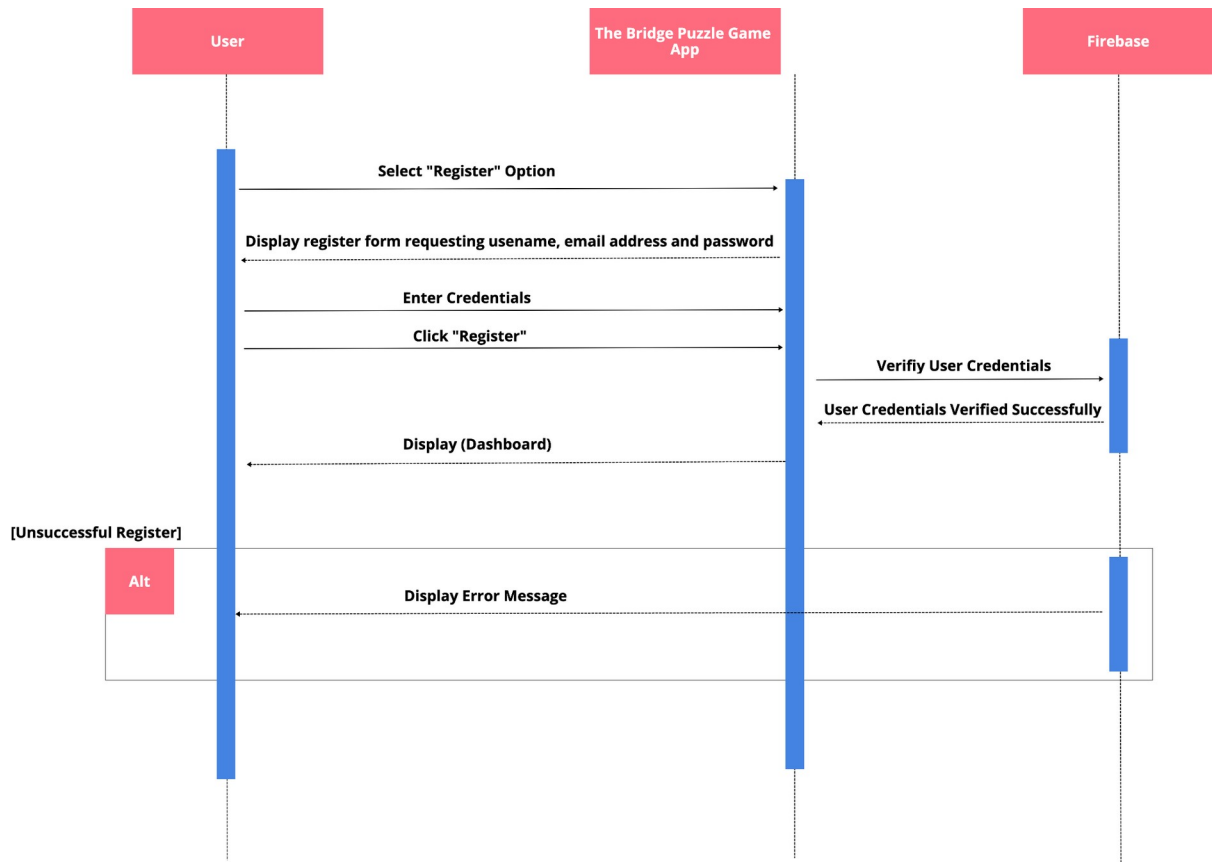


Figure 2 Sequence Diagram 1 - Registration

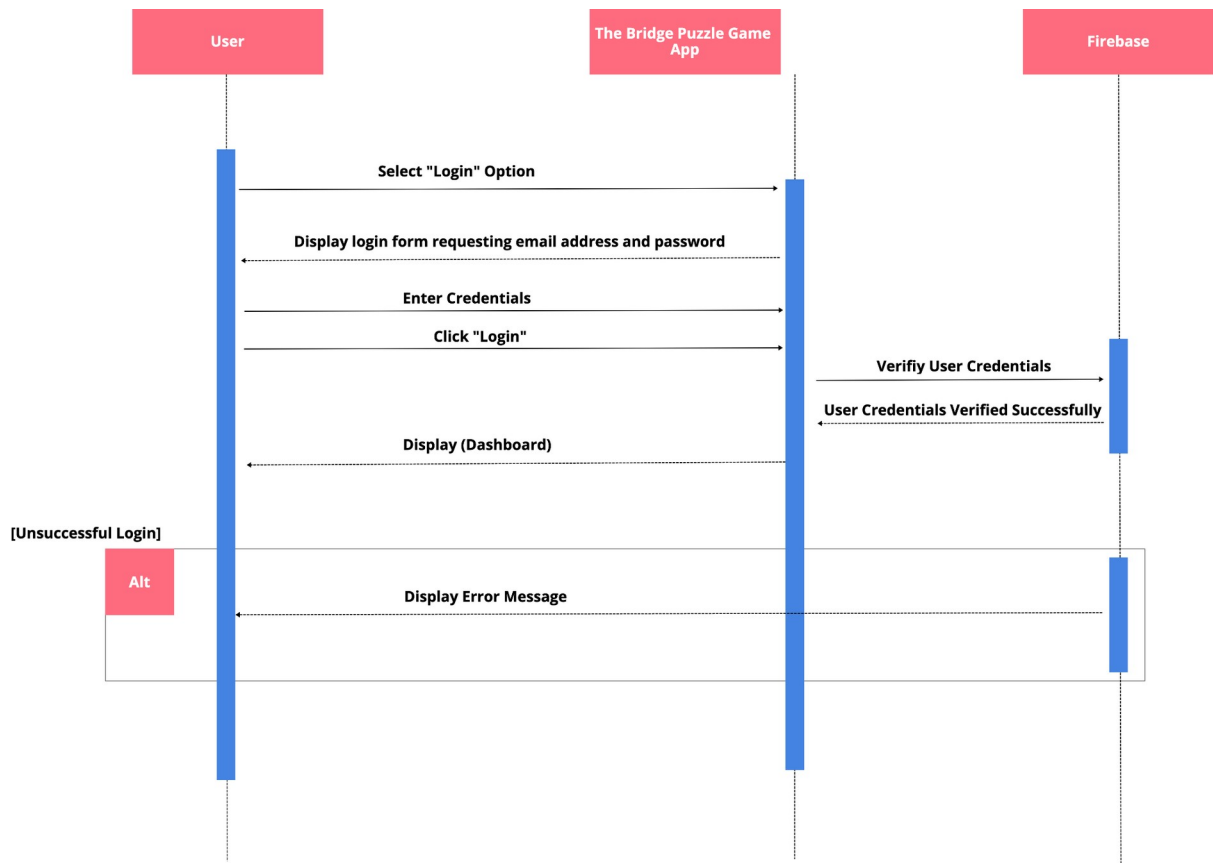


Figure 3 Sequence Diagram - Login

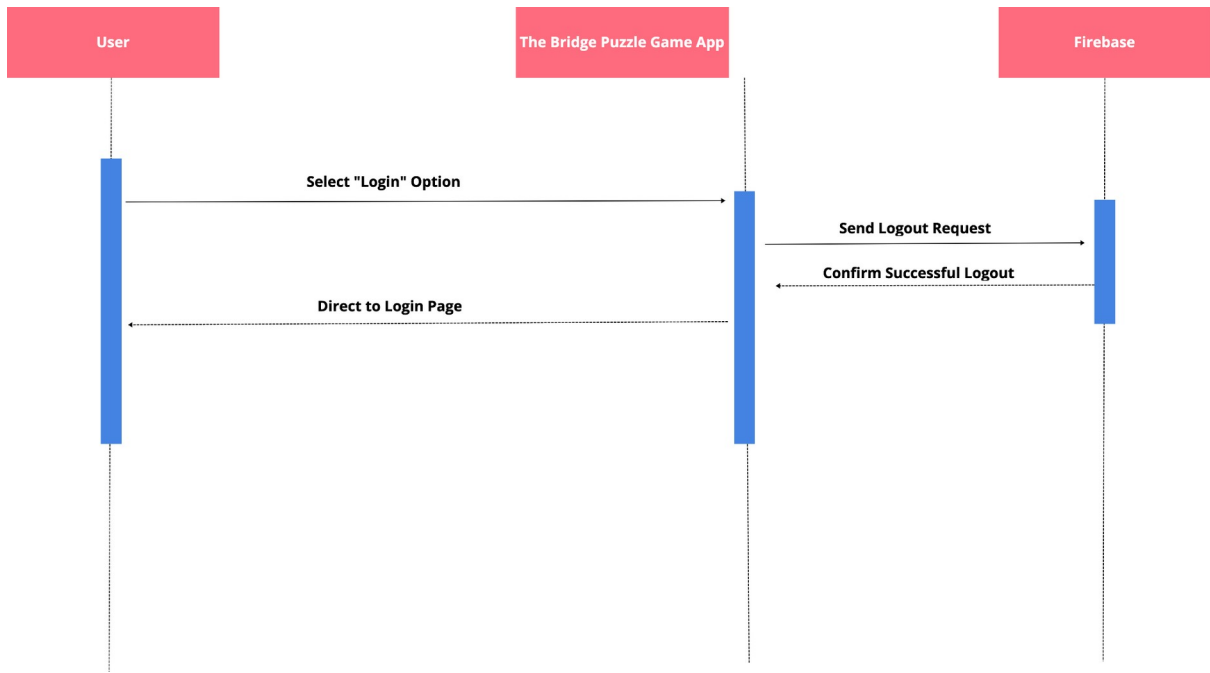


Figure 4 Sequence Diagram - Logout



Figure 5 Sequence Diagram - Play Game

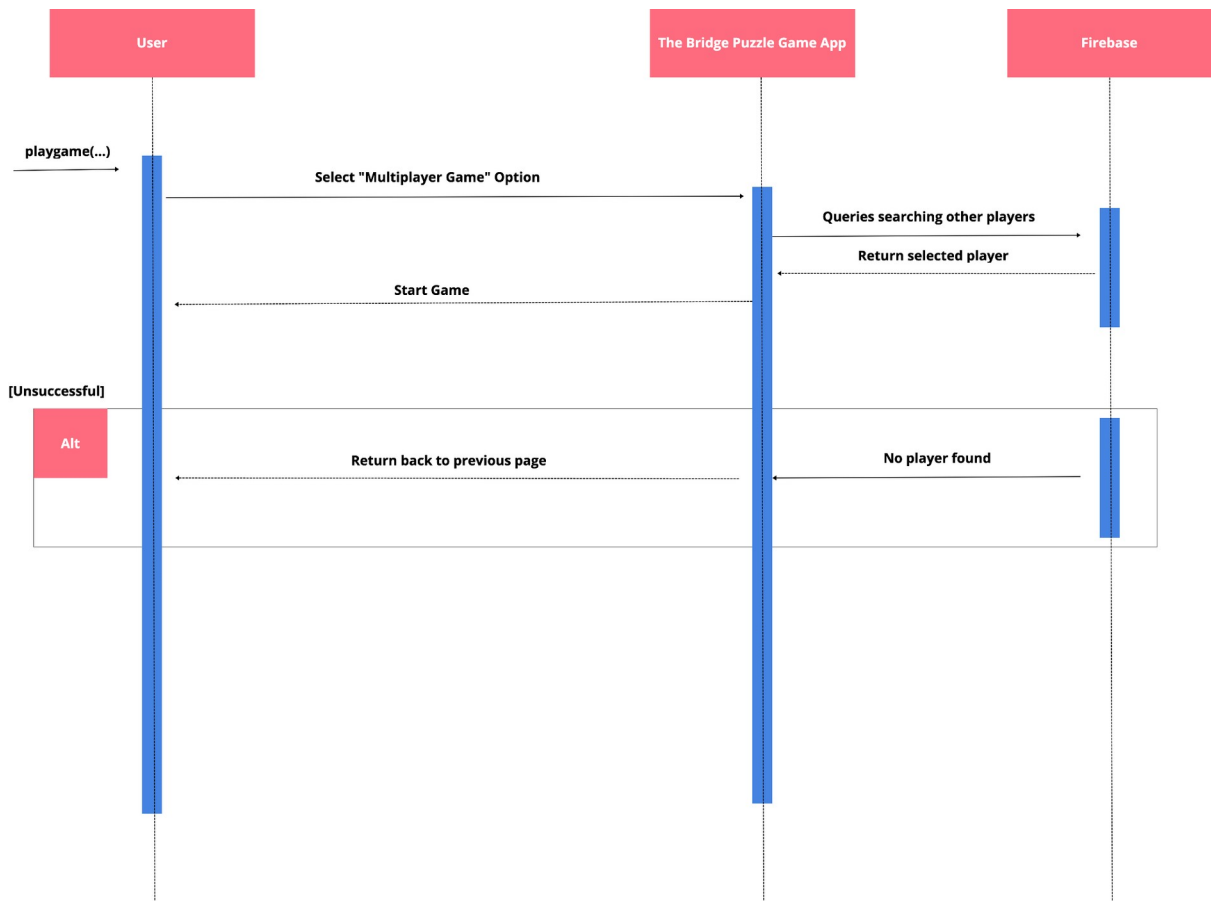


Figure 6 Sequence Diagram - Multiplayer Game Mode



Figure 7 Sequence Diagram - View Profile

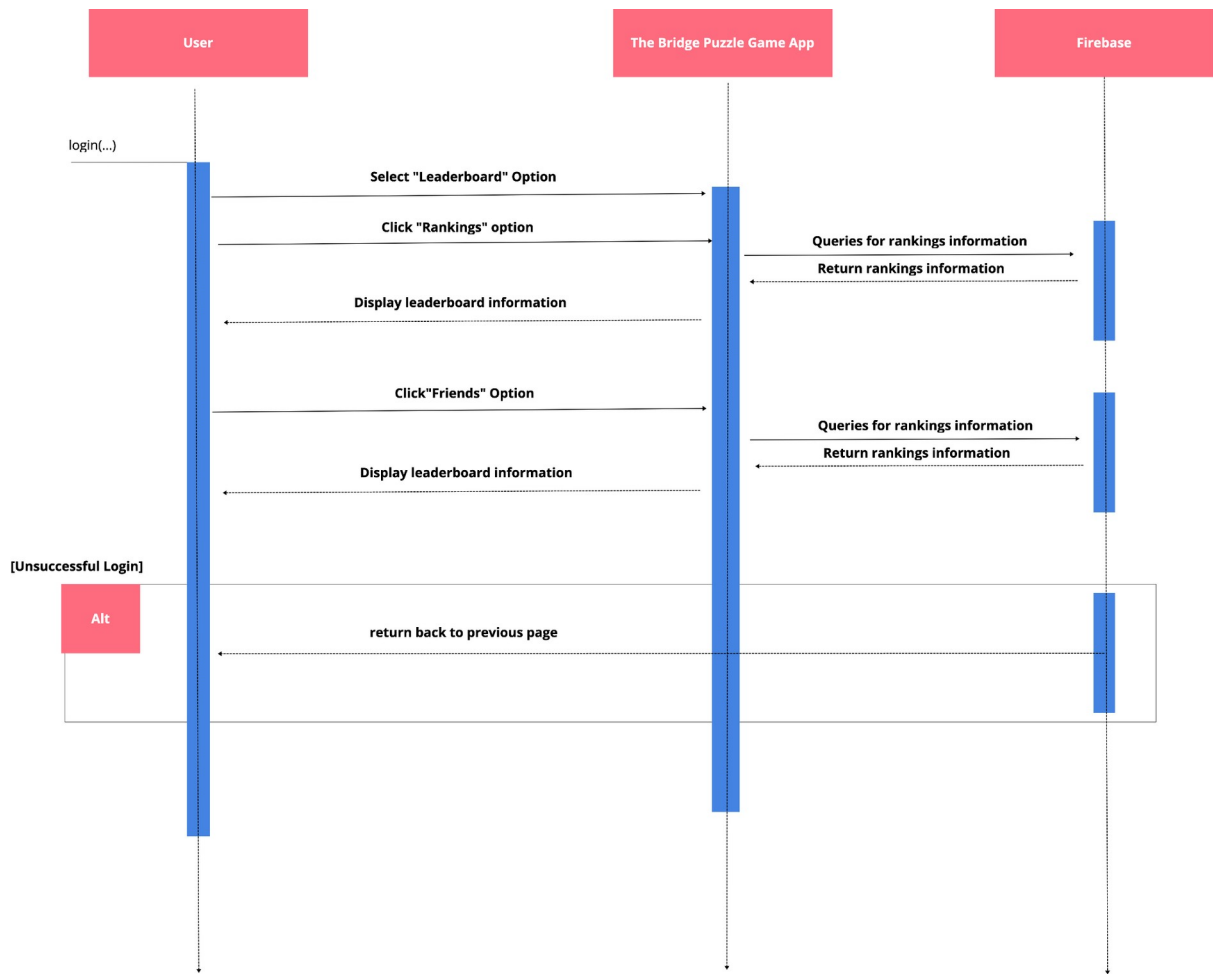


Figure 8 Sequence Diagram - View Leaderboard

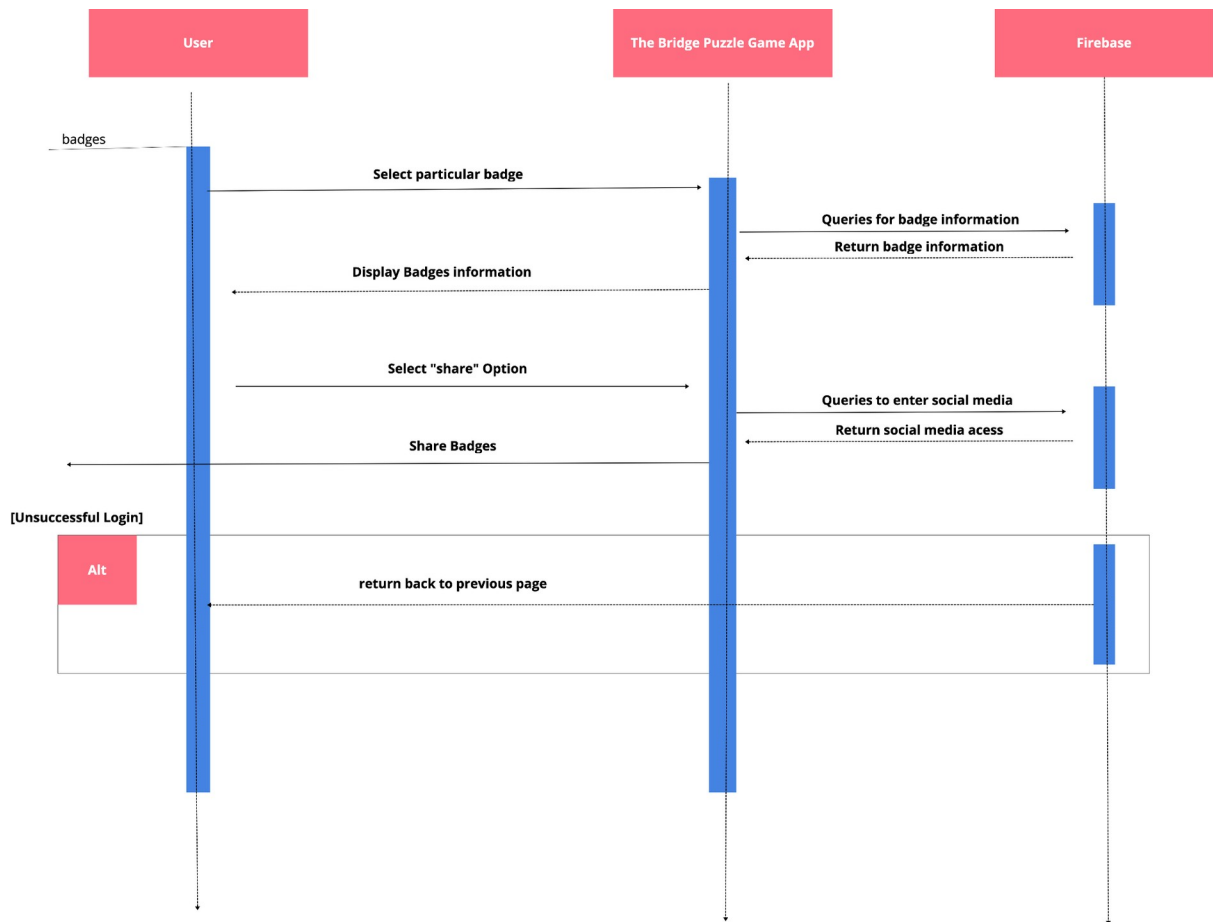


Figure 9 Sequence Diagram - Share Badges

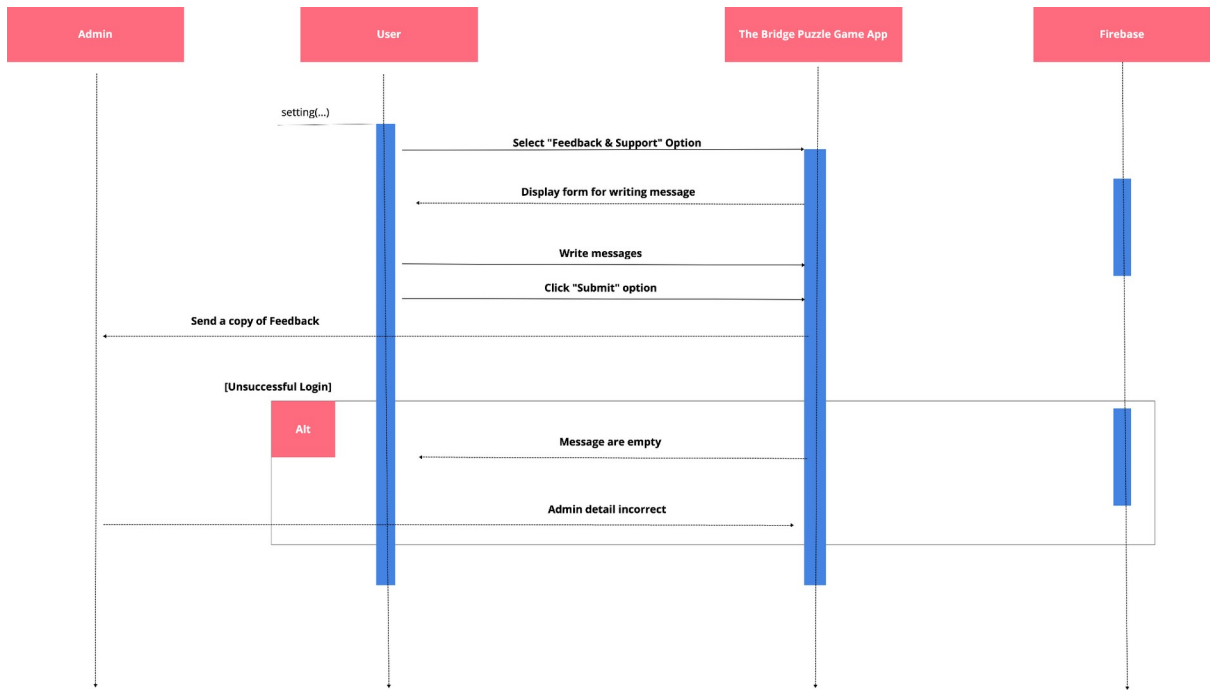


Figure 10 Sequence Diagram - Submit Feedback & Support

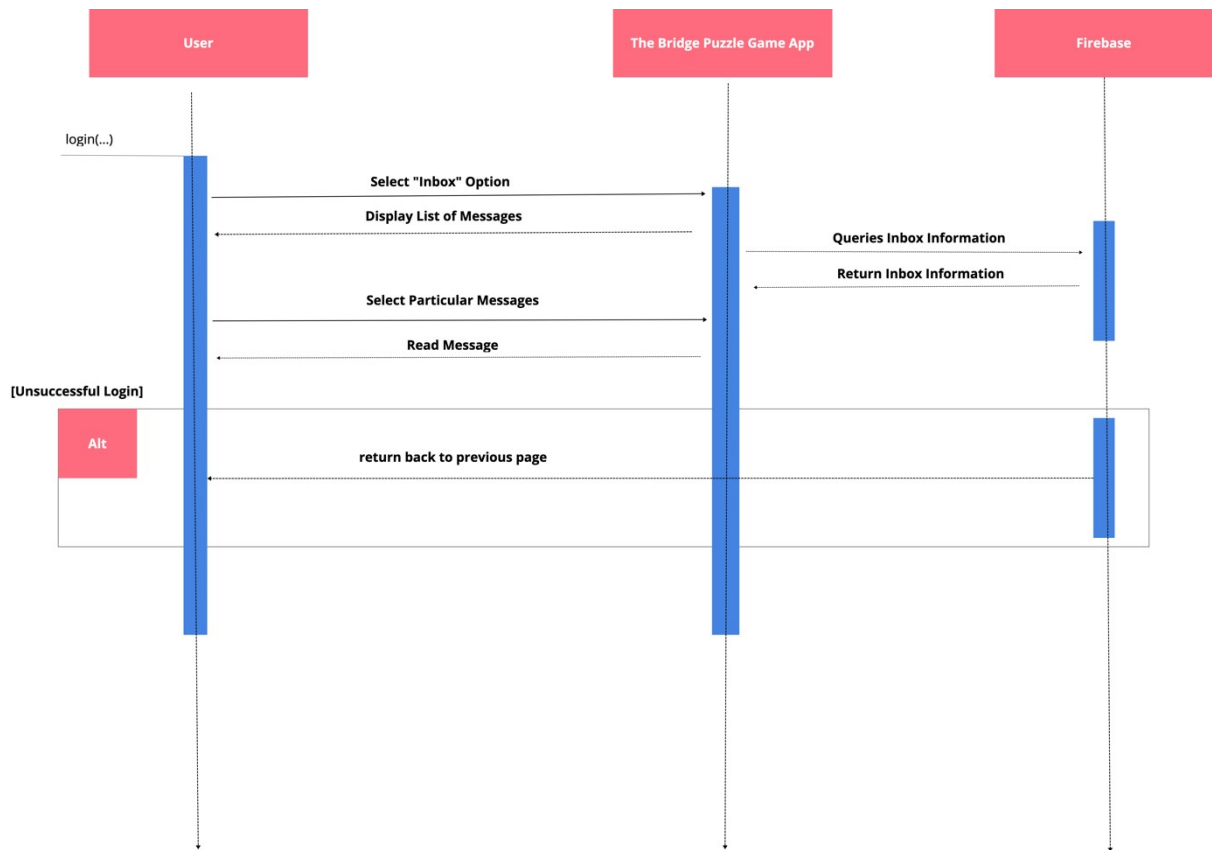


Figure 11 Sequence Diagram - View Inbox

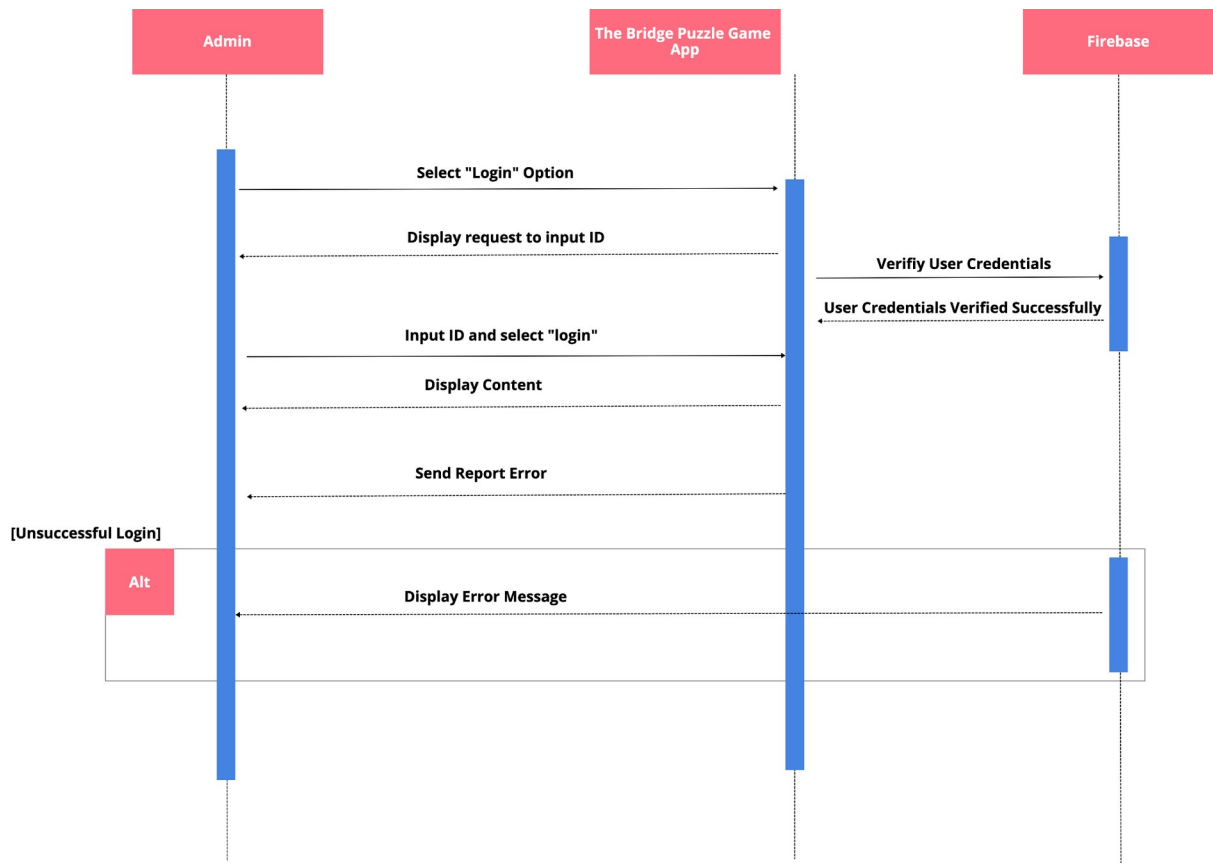


Figure 12 Sequence Diagram - Report Error



Figure 13 Sequence Diagram - Manage Content

9. Class Diagram

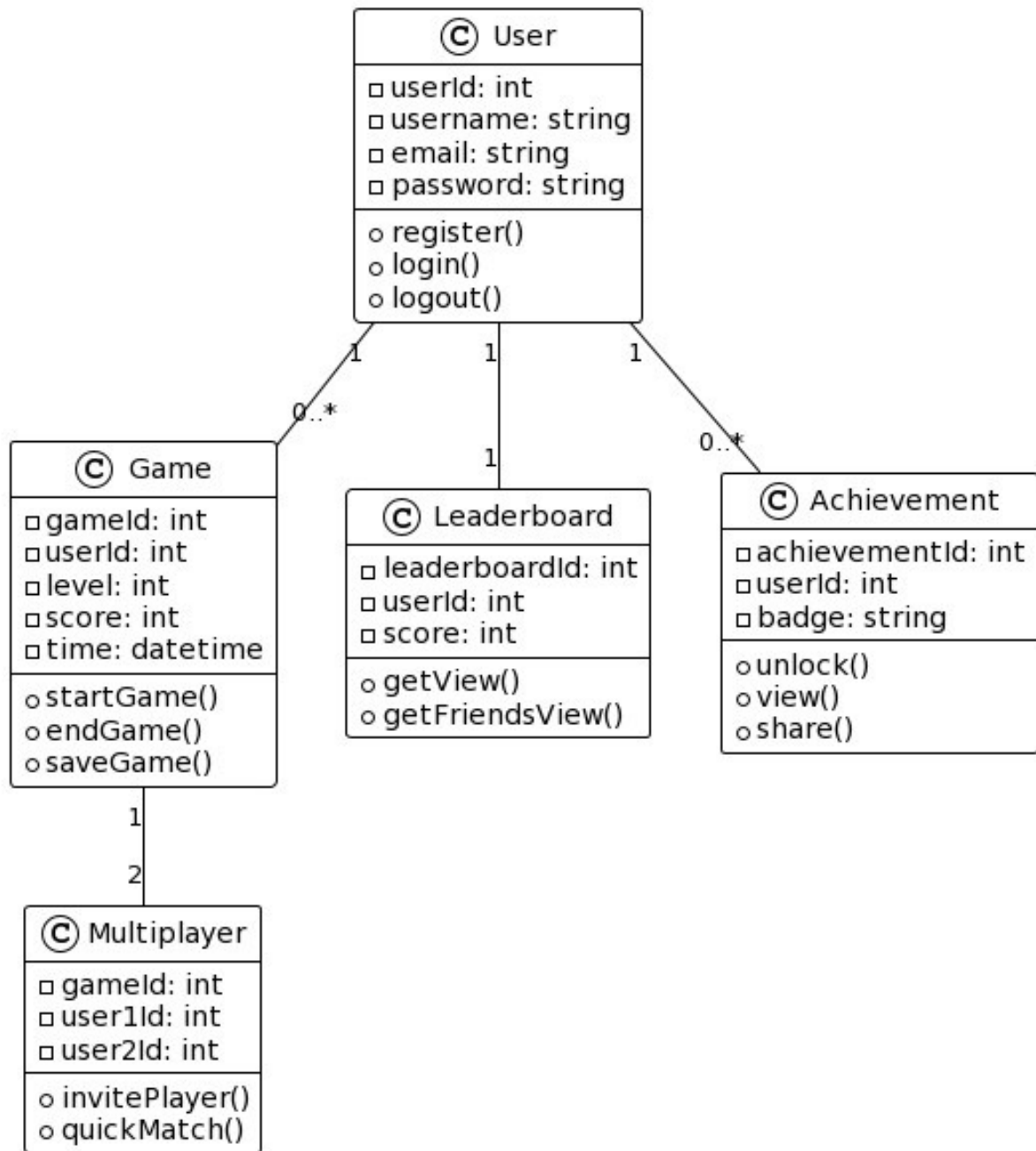


Figure 14 Class Diagram

10. Database Layout

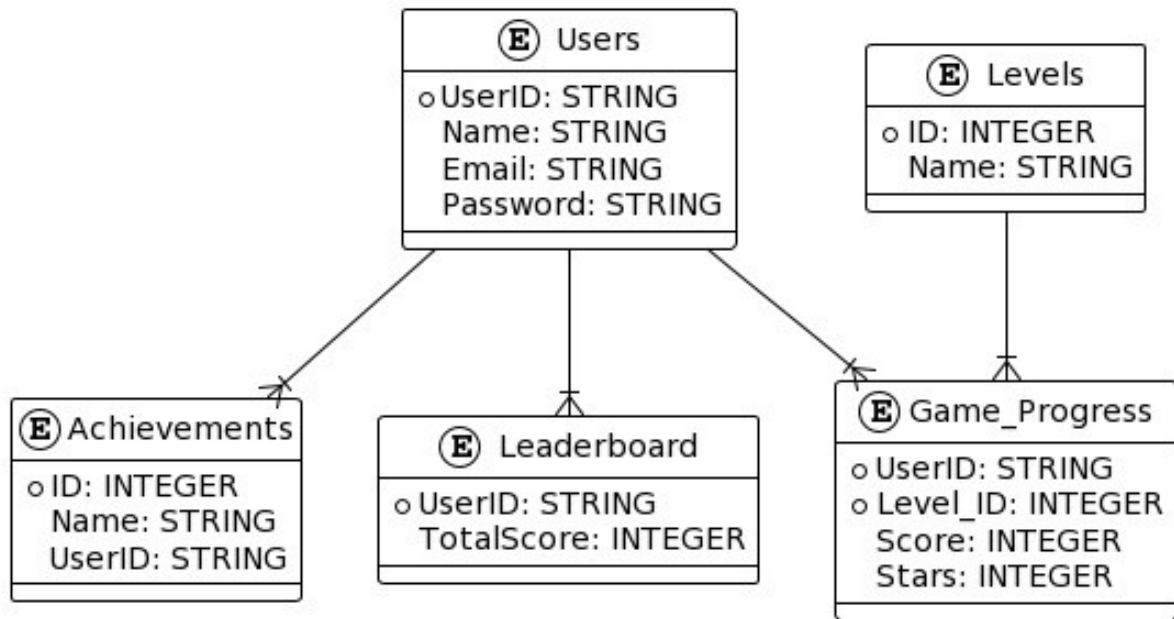


Figure 15 Database Layout Diagram

11. UI/UX

For the Bridge Puzzle Game UI/UX design, colours should balance aesthetics and usability. Here are some chosen colours and their reasons:

1. White: Represents purity, simplicity, and cleanliness. It can create a sense of space and make other colours stand out. In UI design, white is often used for backgrounds to enhance readability and create a modern, clean look.
2. Black: Symbolizes sophistication, elegance, and depth. It can add a touch of mystery and formality. Black is often used for text and other UI elements to create contrast and hierarchy.
3. Light Green (6D9773): Associated with nature, growth, and tranquillity. Light green, in particular, can evoke a sense of freshness and relaxation. This colour choice might contribute to a calming and pleasant gaming experience.
4. Dark Green (0C3B2E): Adds depth and contrast to the lighter green. Darker greens can convey a sense of stability and richness. It may be used to create boundaries or emphasize certain elements in your game.
5. Brown (BB8A52): Often associated with earthiness, warmth, and reliability. In the context of a game like Bridge Puzzle Game, brown could represent the wooden bridges or paths that players are creating. It adds a natural and grounded feel to the design.
6. Yellow (FFBA00): Represents energy, optimism, and happiness. In the context of a game, yellow can draw attention to important elements or actions. It might be used for highlights, prompts, or interactive elements, providing a visual focal point.

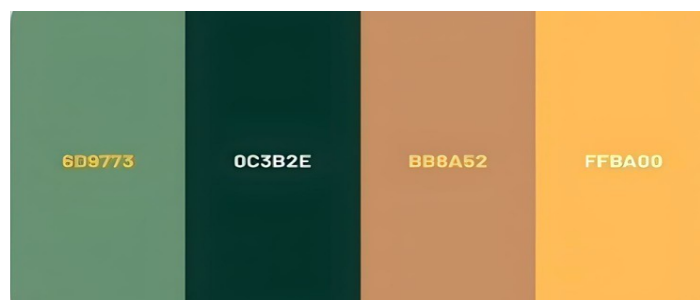


Figure 16 UI Colour Template

12. Prototype Screens

The following illustrations represent preliminary user interface (UI) designs developed for The Bridge Puzzle Game App. It should be noted that the final UI version will vary significantly from the prototypes presented below.

12.1 Main Page

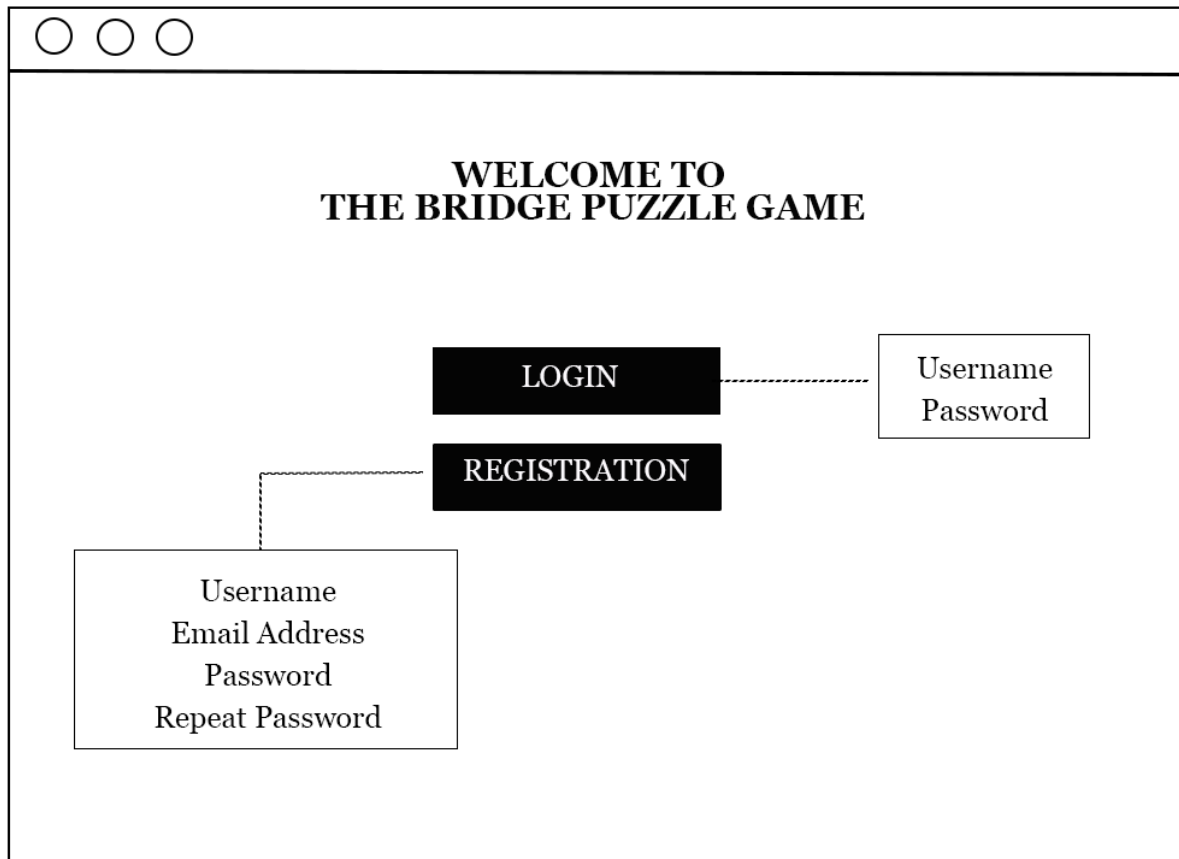


Figure 17 Prototype Screen - Main Page

The initial interface that users encounter when opening the application requires them to either log in to their existing profile or register a new account to store their database. Firebase Backend-as-a-Service (BaaS) will provide all the required data. The web hosting and database access will only be available to the application's developer.

12.2 Login Page

○ ○ ○

Login

Username

Password

Forget Password ?

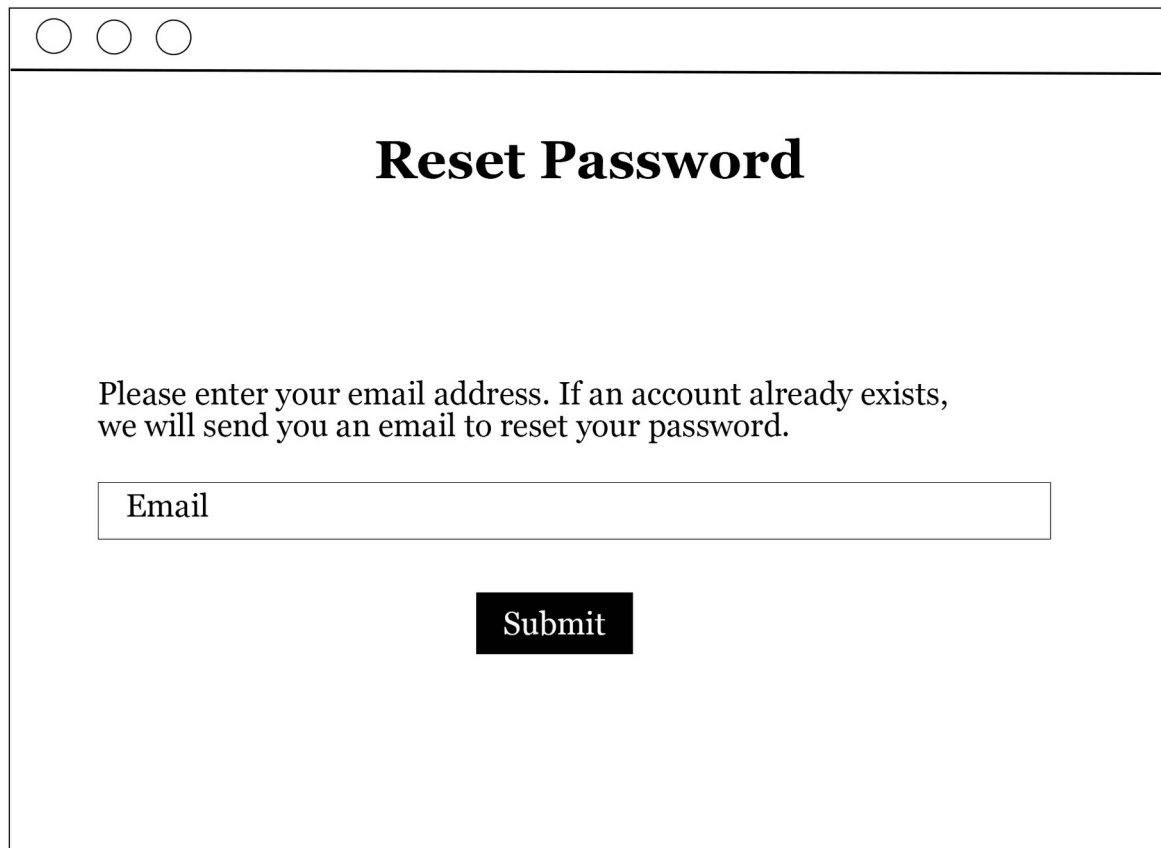
Login

Dont have an account ? Register

Figure 18 Prototype Screen - Login Page

To access the system, the user must enter their username and password, which are securely stored in the Firebase Cloud Functions and Firebase Storage and managed by the Firebase Authentication. The “Register” button is available for users who do not have an existing account and need to create one. Clicking on this button takes the user to the registration page, where they can fill out the necessary details. If the user forgets their password, they can simply click on the “Forget Password?” button, which will redirect them to the password recovery page where they can retrieve their password by following the instructions.

12.3 Reset Password Page

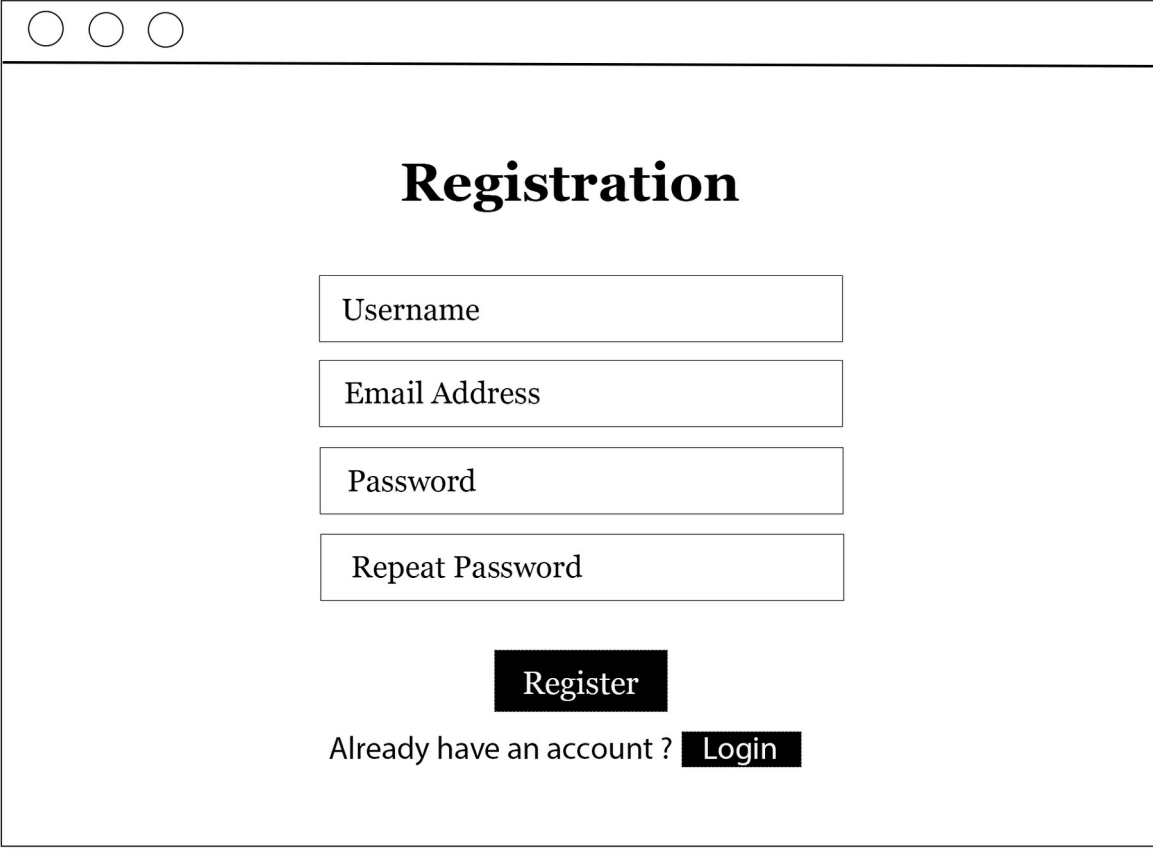


A prototype of a web page for resetting a password. The page is enclosed in a rectangular frame with three small circles in the top-left corner, representing window control buttons. The main heading is "Reset Password" in a large, bold, black serif font. Below the heading is a paragraph of text: "Please enter your email address. If an account already exists, we will send you an email to reset your password." Underneath this text is a single-line text input field with the placeholder text "Email". Below the input field is a black rectangular button with the word "Submit" written in white, bold, sans-serif font.

Figure 19 Prototype Screen – Reset Password Page

User can reset their password by entering their registered email. A password reset link will be sent to their email.

12.4 Registration Page



The image shows a registration page prototype within a browser window. The window has three circular window control buttons in the top-left corner. The page content is centered and features the title "Registration" in a large, bold, black font. Below the title are four vertically stacked text input fields, each with a thin black border and a light gray background. The labels for these fields are "Username", "Email Address", "Password", and "Repeat Password", positioned to the left of the input area. Below the input fields is a solid black rectangular button with the word "Register" in white text. At the bottom of the form area, the text "Already have an account ?" is displayed in a standard black font, followed by a solid black rectangular button with the word "Login" in white text.

Figure 20 Prototype Screen - Register Page

To create an account, the user must generate a unique username, provide a valid email address, configure a robust security password, and verify the password by re-entering it. These data will be securely stored in Firebase Storage and accessed through Firebase Authentication. The “Login” button is exclusively available for users who have already registered an account.

12.5 Home Page

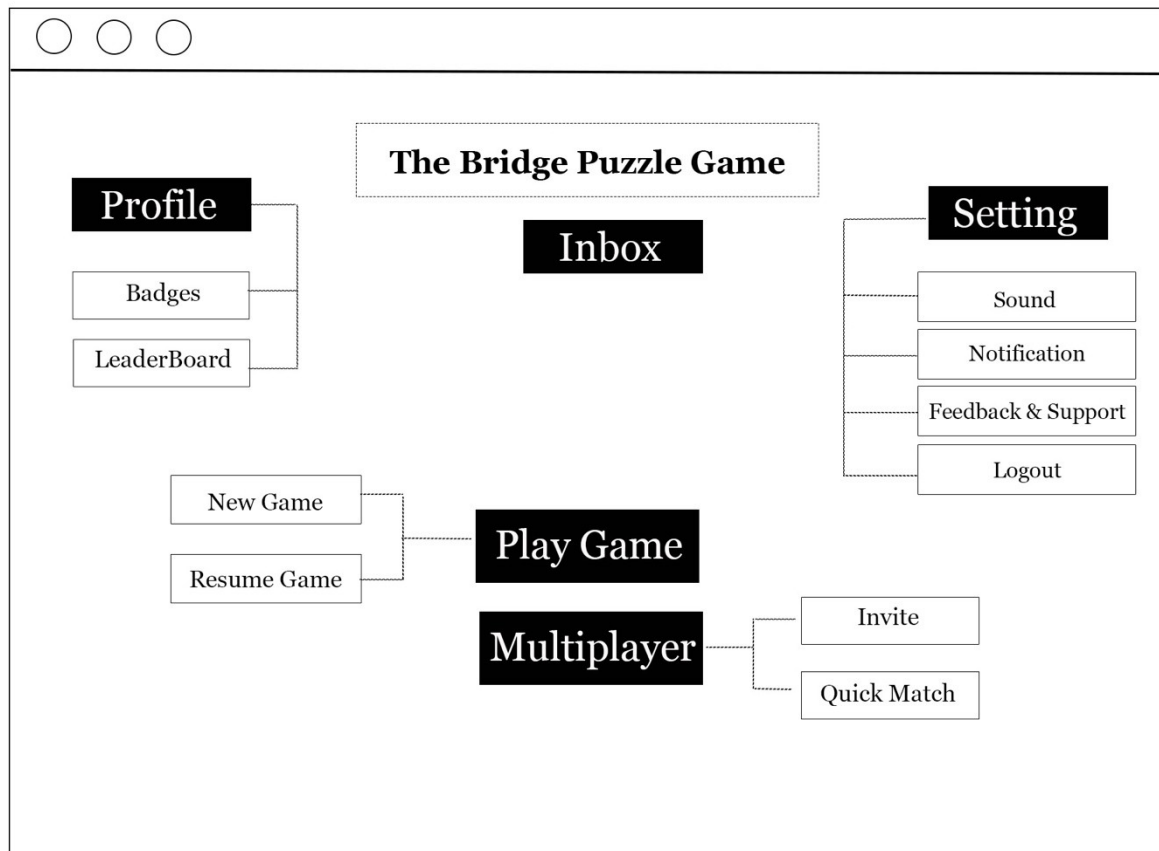


Figure 21 Prototype Screen - Home Page

Upon entering the main page of the game application, users are granted access to their database. From there, they can click on the "Profile" button to view their earned badges and access the leaderboard to compare their score with other users that sync in the real-time database.

The "Settings" button, powered by Firebase Storage, allows users to personalise their settings, including sound, notifications, feedback and support and options to log out from the account.

The "Inbox" button to receive messages from the app server.

To start playing the game, users can click on the "Play Game" button, while the "Multiplayer" button allows users to play games with their friends.

12.6 Play Game Page



Figure 22 Prototype Screen - Game Option Page

The gameplay page offers users the option to either begin a new game or continue playing from where they left off in their last saved game, which is stored in Firebase Storage. This feature allows users to seamlessly pick up their game progress without losing any previous achievements or progress.

The button “How to Play” provides instructions on the rules.

12.7 Level Mode Page

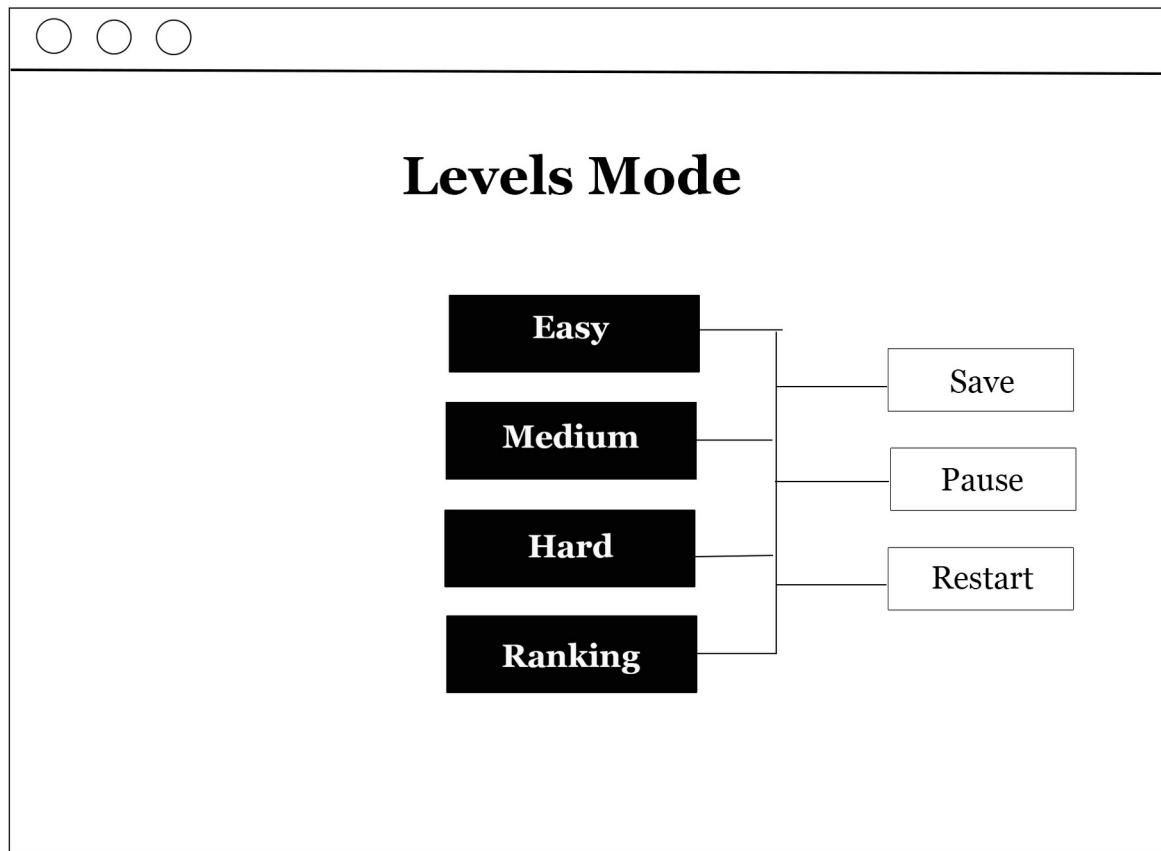


Figure 23 Prototype Screen - Level Mode Page

To begin playing the game, the user can select a level to start with or opt for the ranking mode.

The ranking mode in this game operates by randomly selecting a level for the user to play. To improve their ranking, the user must compete against the clock, striving to complete the level as quickly as possible. Once completed, the user's score will be shown on the leaderboard, where they can compare their performance with other users who have also played the ranking mode. The leaderboard ranks users from highest to lowest score, allowing players to see where they stand compared to others.

12.8Game Page

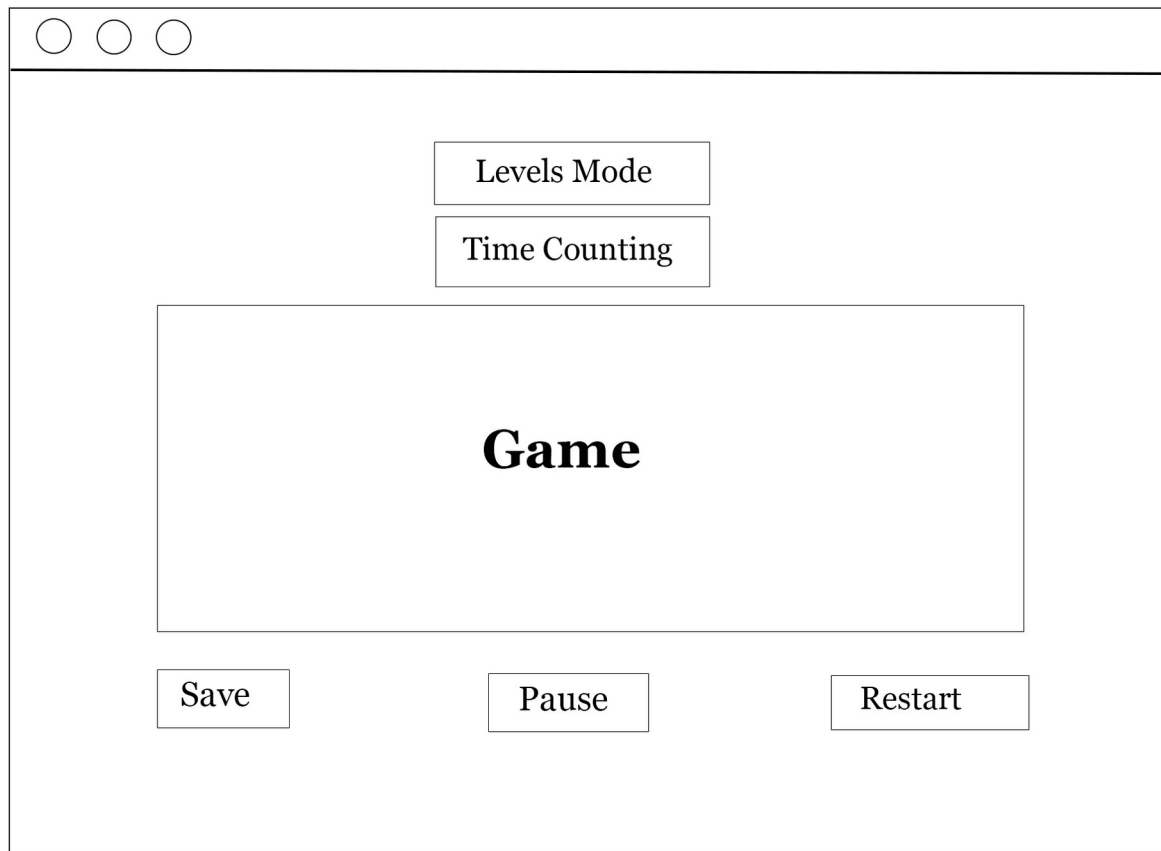


Figure 24 Prototype Screen - Game Page

On the game page, it will display the selected level and keep track of the time that has elapsed. To save the progress, pause the timer, and an option to restart the level is available.

12.9 How to Play Page

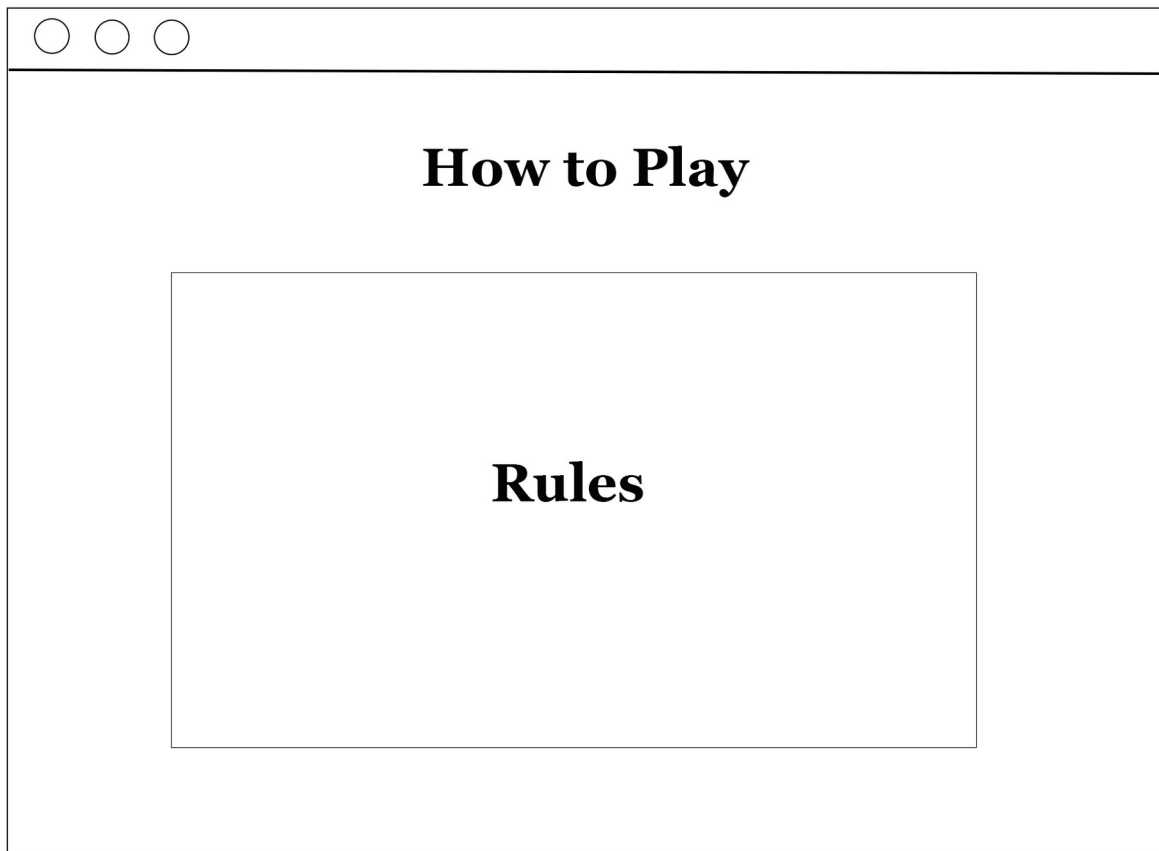


Figure 25 Prototype Screen - Rule Page

The rules of the game are explained on the page.

12.10 Multiplayer Page

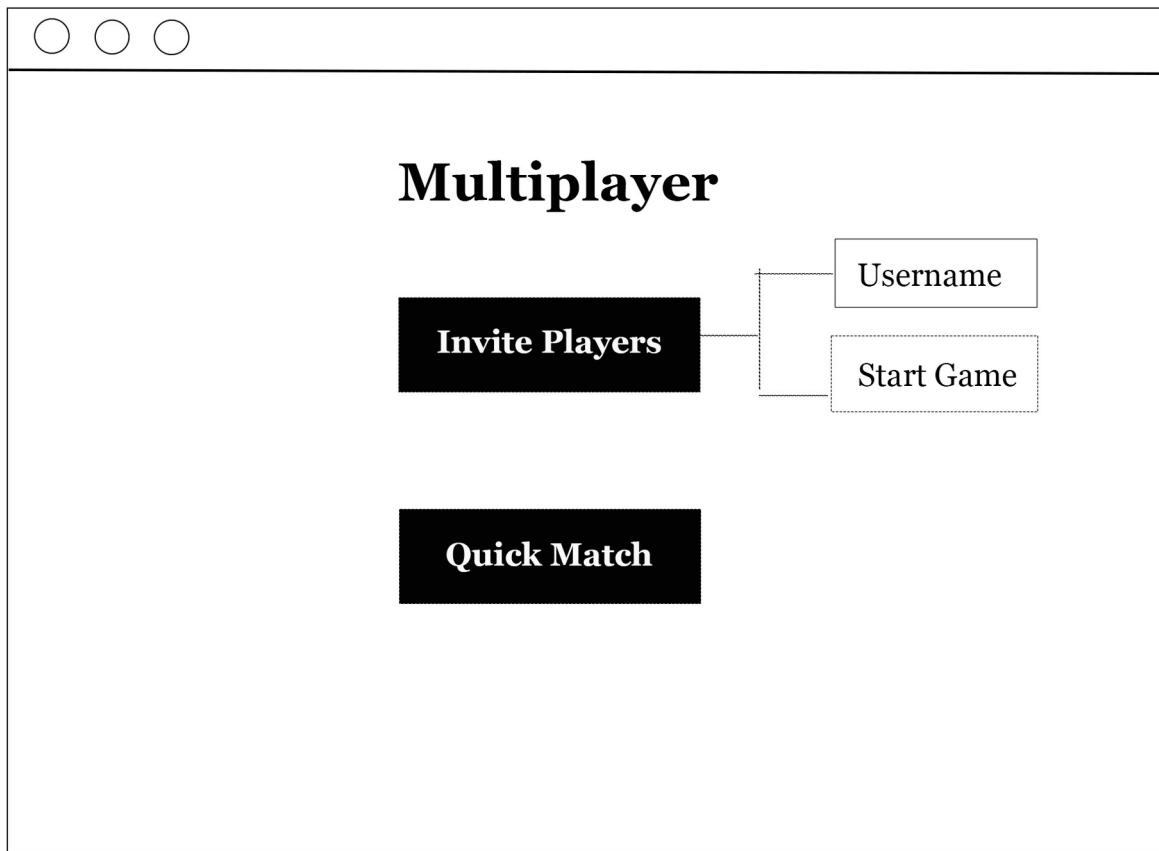


Figure 26 Prototype Screen - Multiplayer Page

Upon arriving at the Main Page, users will be presented with the opportunity to engage in multiplayer gameplay. To play with other users, they can either search for friends by typing their username into the “Invite Players” Button, or they can choose to be matched up with a random player online for a quick match. This feature allows users to connect and play with their friends or other players from all around the world.

12.11 Invite Player Page

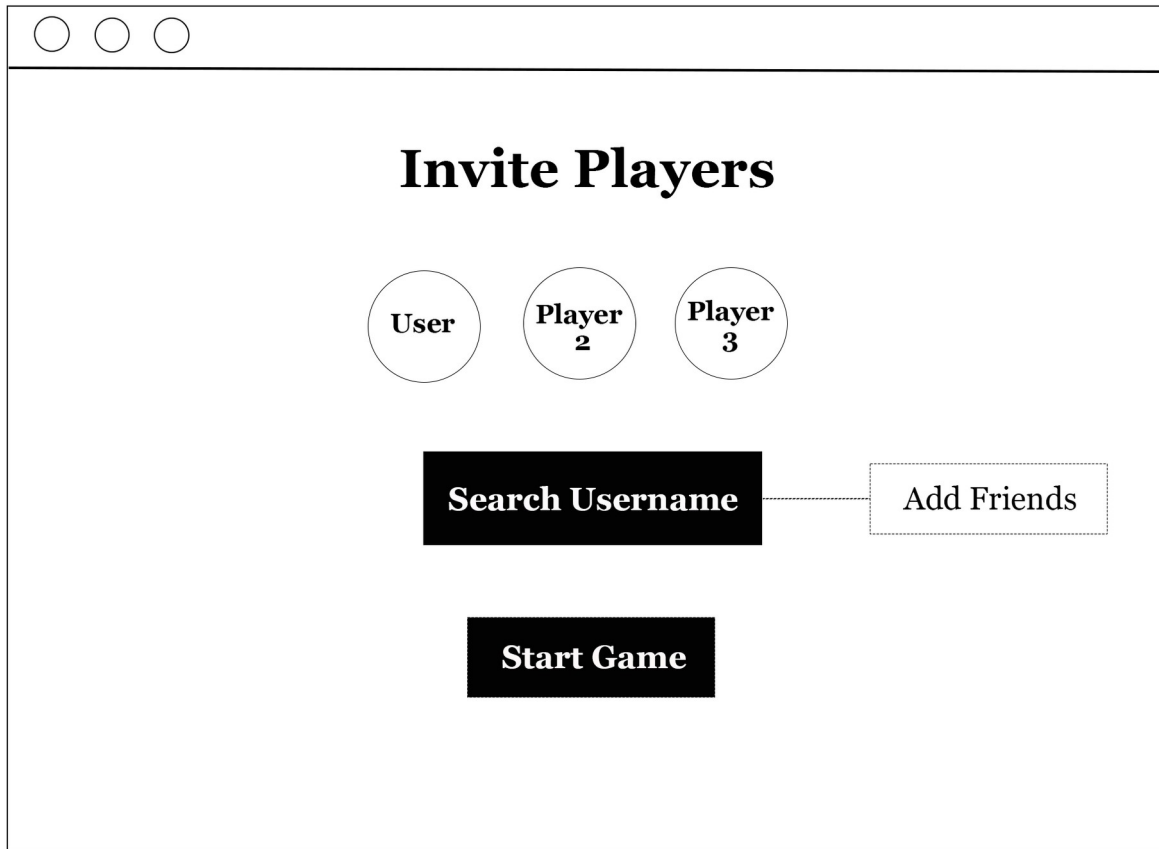


Figure 27 Prototype Screen - Invite Player Page

When inviting Players to join the game, the user invites another player by searching for their username. Once the other player accepts the invitation, the game can be started by the user. This allows for a seamless and efficient way to connect with other players and enjoy the gaming experience together.

12.12 Add Players Page

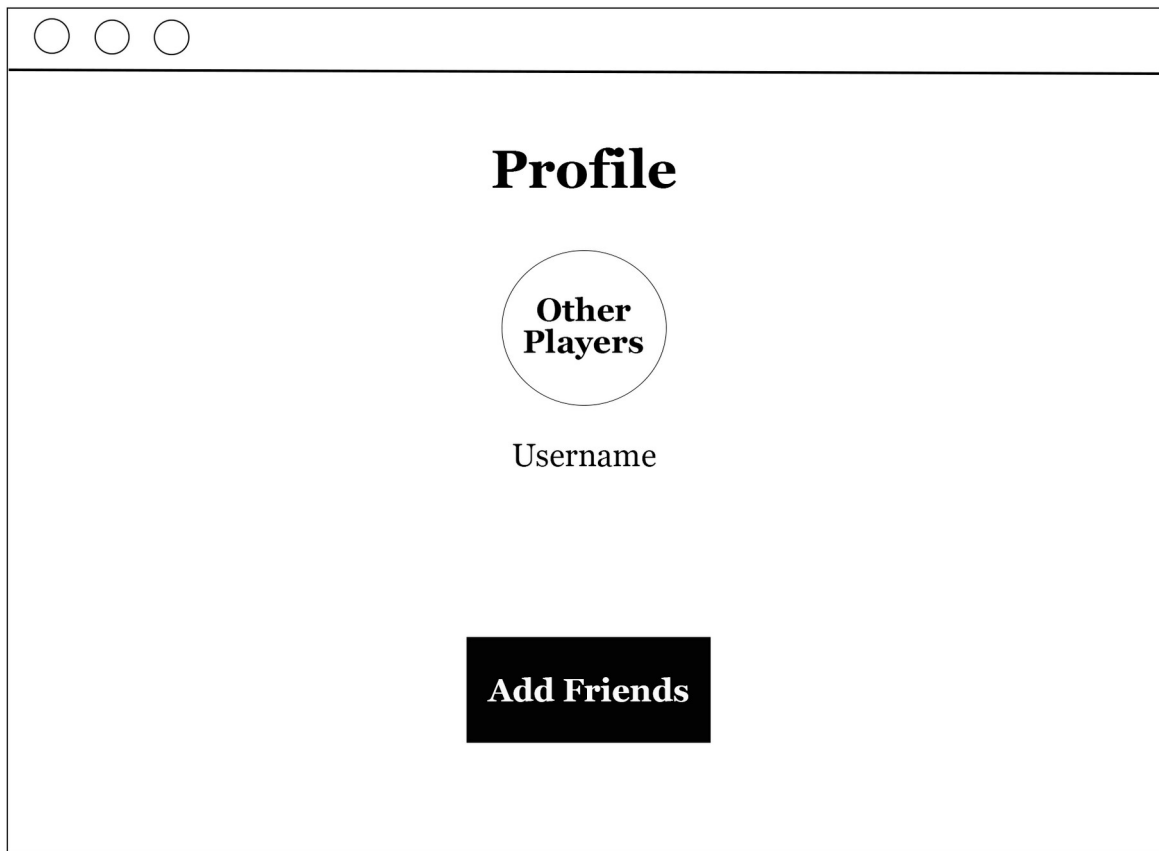


Figure 28 Prototype Screen - Add Player Page

The platform provides users with the ability to search for other players by their usernames. Upon locating a player, they have the option to establish a connection by adding them as a friend, which can facilitate future invitations to play. Additionally, the feature enables users to view the scores of their friends on the leaderboard, providing them with valuable insight into their performance and progress. This function enhances the overall user experience by facilitating social connections and by providing users with the means to track their performance metrics.

12.13 Quick Match Page

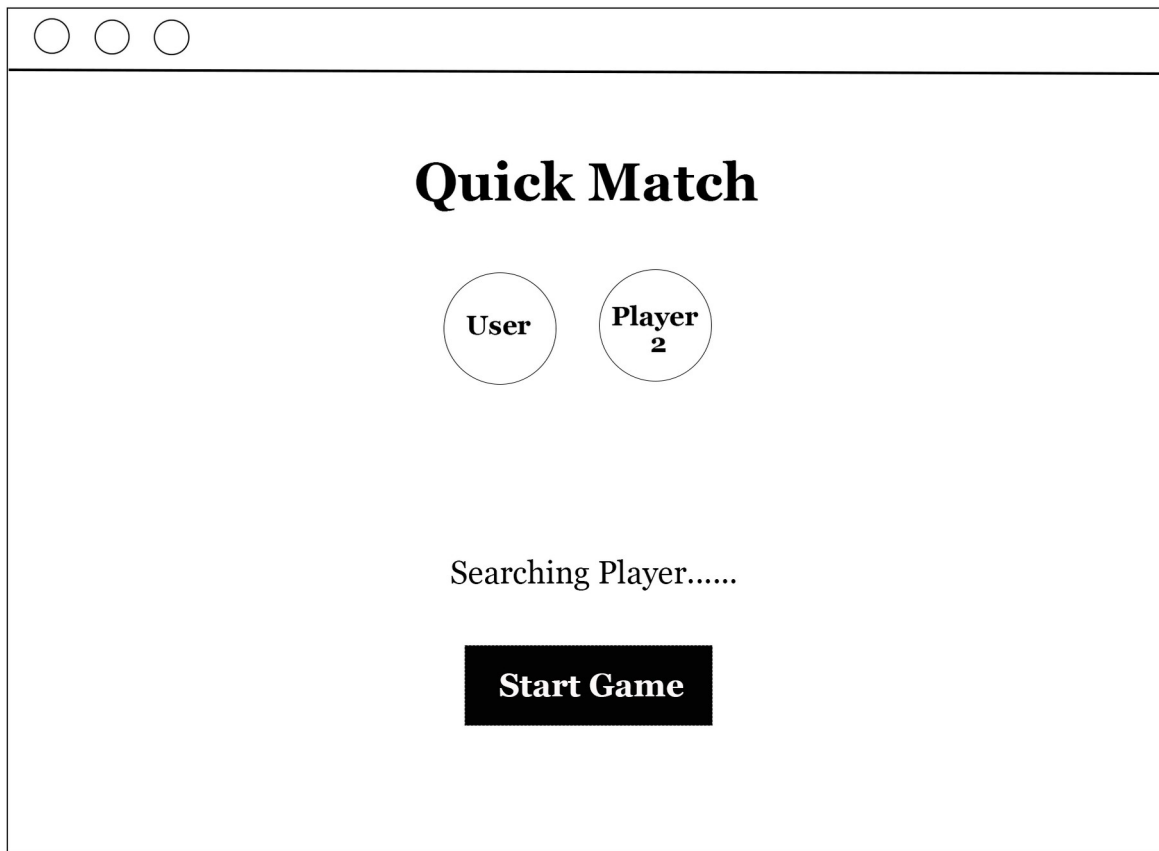


Figure 29 Prototype Screen - Search for Matching Page

When the user chooses the Quick Match option, the system initiates an automatic search to find other online players who are available to play a game. The search process aims to match the user with players who are of similar skill levels. Once a match is found, the user and the other player can begin playing the game together.

12.14 Profile Page

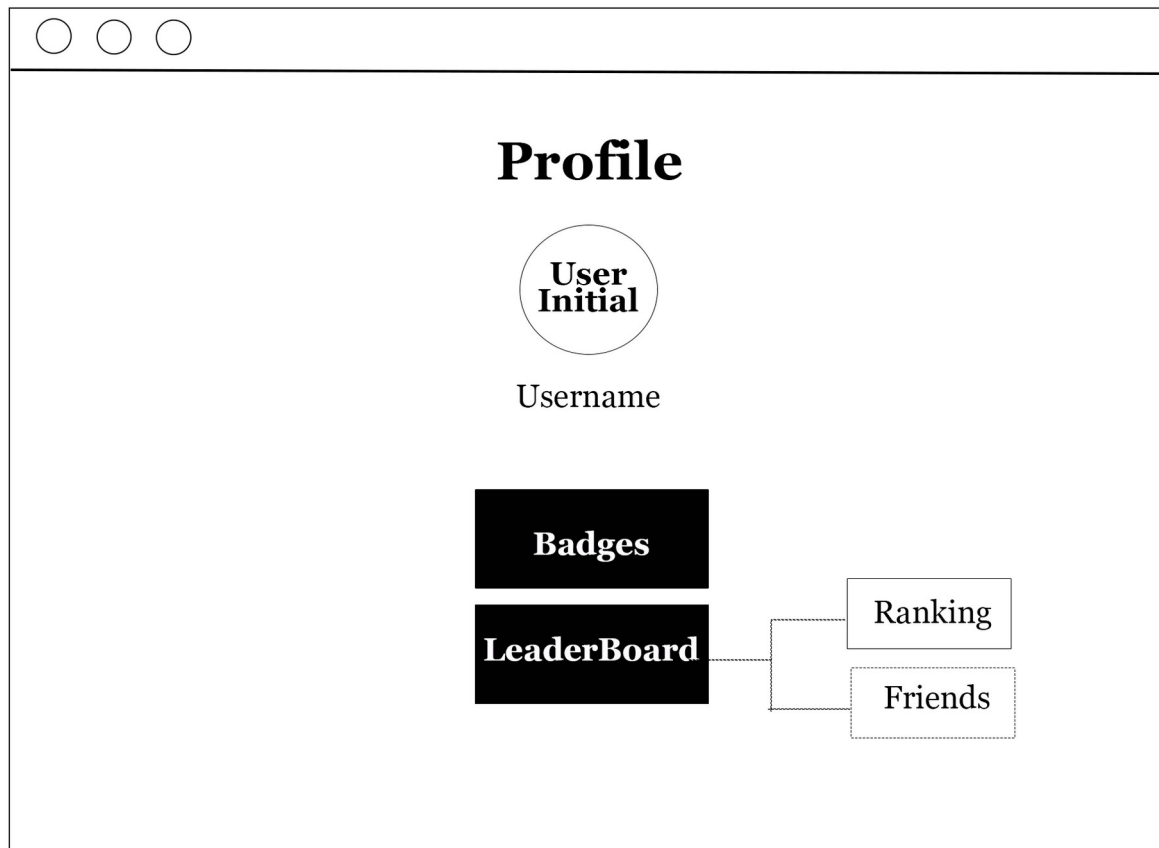


Figure 30 Prototype Screen – View Profile Page

The user's profile page is a dynamic and interactive interface that is constantly updated in real time using Firebase's powerful synchronization technology. It provides a comprehensive view of the user's achievements, displaying their badges of accomplishment and highlighting their progress on the leaderboard. With Firebase storage, all user data is securely stored and readily accessible, ensuring a safe and reliable platform for all users.

12.15 Badges Page

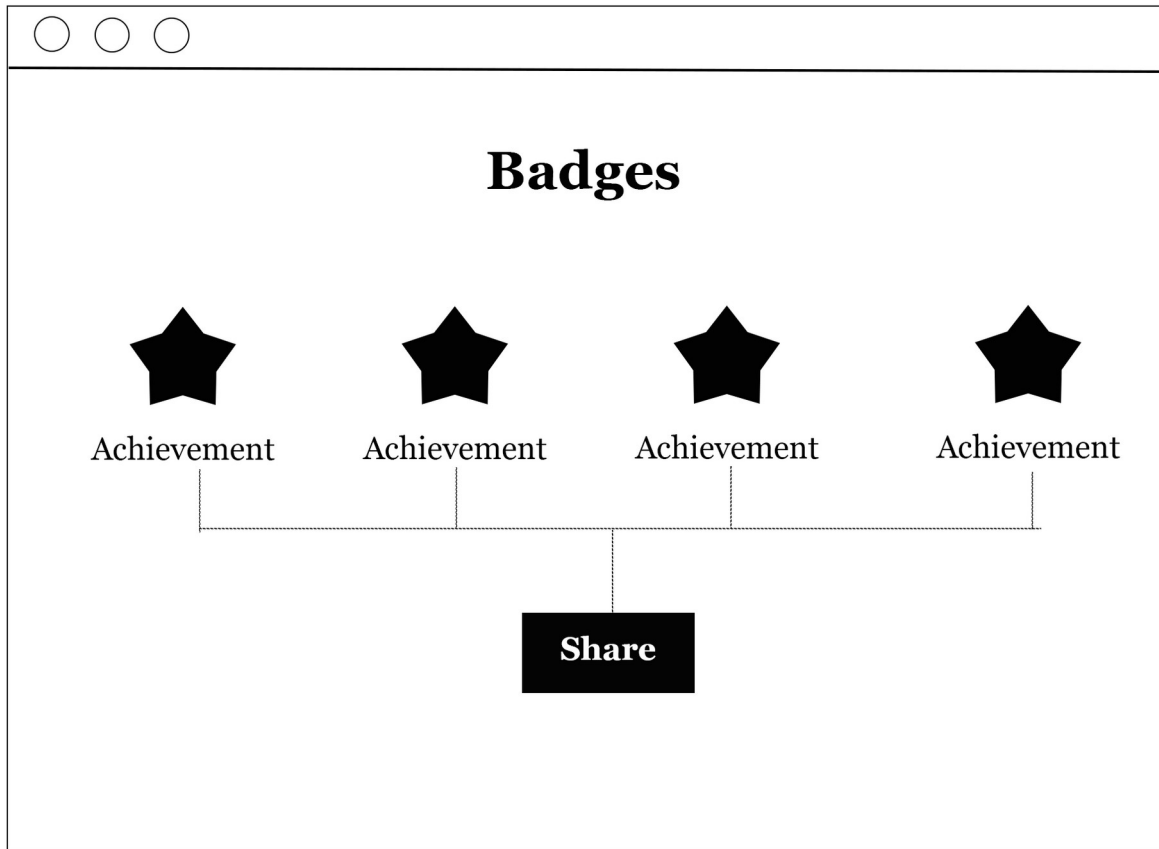


Figure 31 Prototype Screen - View Badges Page

These badges serve as a recognition of the user's achievements and progress in the game. The user can view and share badges through social features earned from their past games such as completing the level within a certain time etc.

12.16 Share Page

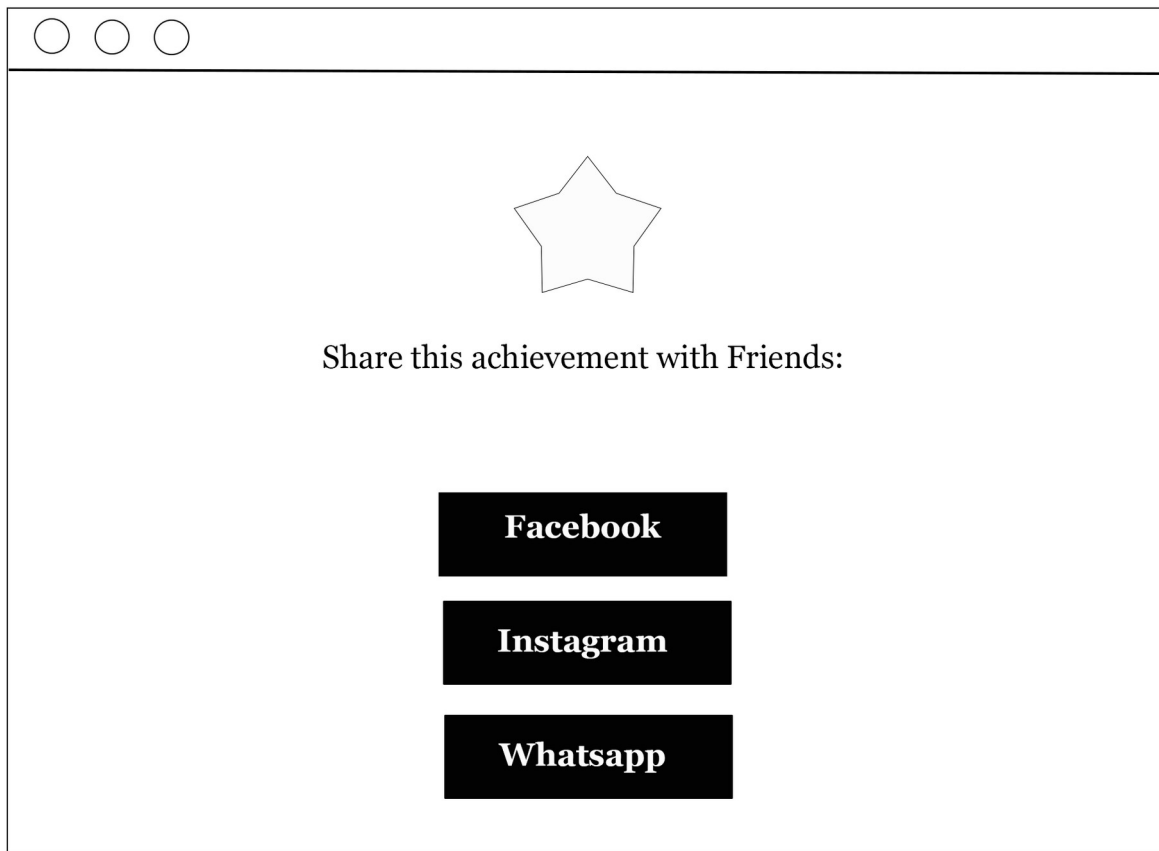


Figure 32 Prototype Screen - Share Badges Page

User can easily share their earned badges on social media platforms such as Facebook, Instagram, and WhatsApp.

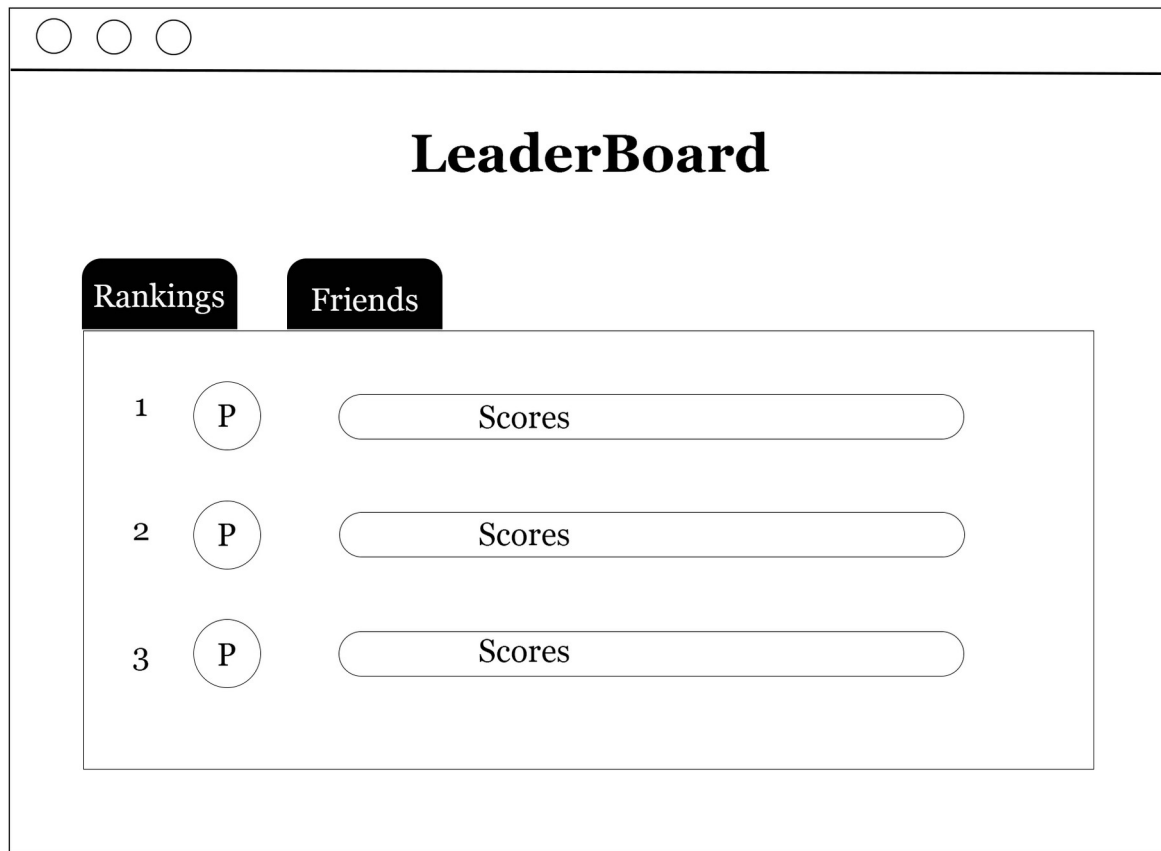


Figure 33 Prototype Screen - View Leaderboard Page

The leaderboard is divided into two sections: ranking and friends. The ranking section displays scores of global users who have opted for the ranking mode, sorted in descending order. By comparing their performance with others, users can gauge their skill level and identify areas that require improvement.

The friends section of the leaderboard displays the rankings of the user's added friends. The Leaderboard update helps users keep track of their friends' progress and allows for friendly competition among them.

12.18 Setting Page

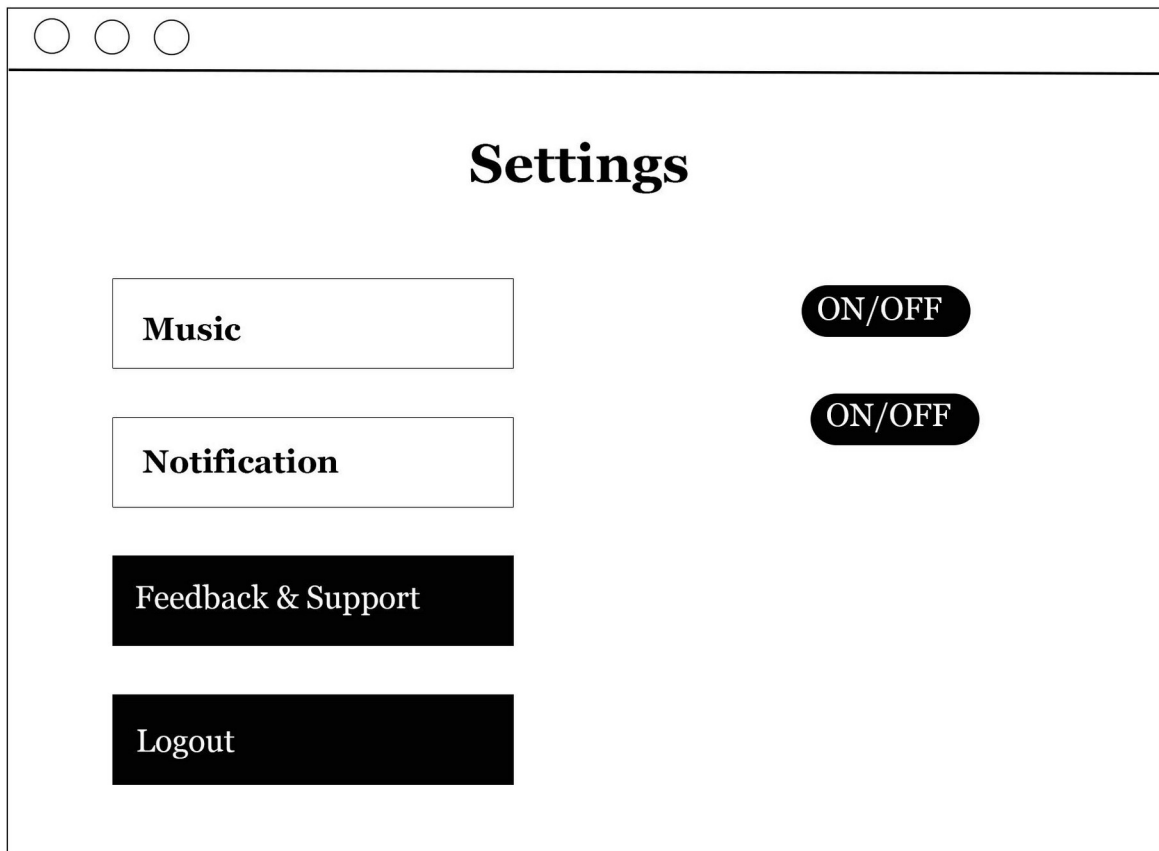
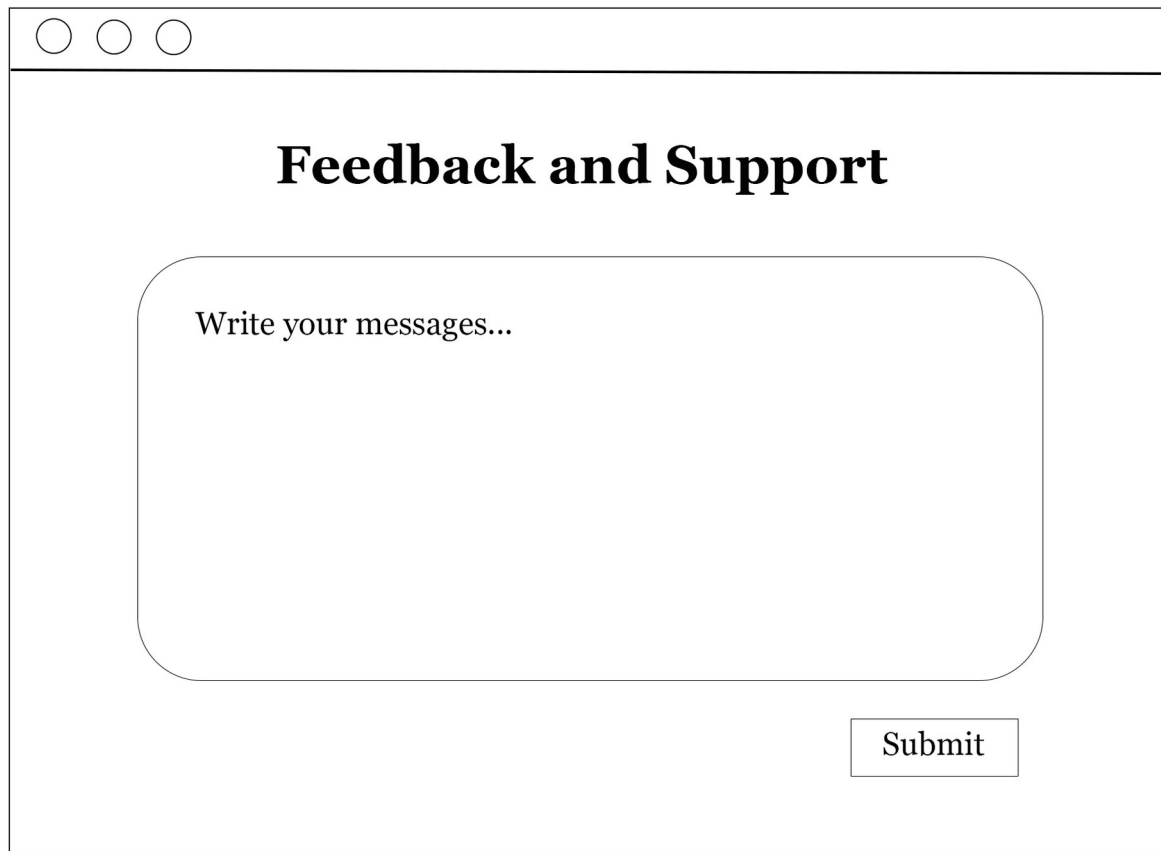


Figure 34 Prototype Screen - Setting Page

On the application's settings page, users are presented with the ability to manage the background music and notification preferences. Users can toggle these options on or off according to their preferences. Additionally, there is a "Feedback & Support" button that serves to encourage users to provide their valuable feedback to the application developer. By doing so, users can help to improve the application and ensure that it meets their needs and preferences. The user can log out to disconnect from their data and return to the main page.

12.19 Feedback & Support Page



○ ○ ○

Feedback and Support

Write your messages...

Submit

Figure 35 Prototype Screen - Feedback & Support Page

Users can anonymously submit a message using the provided box.

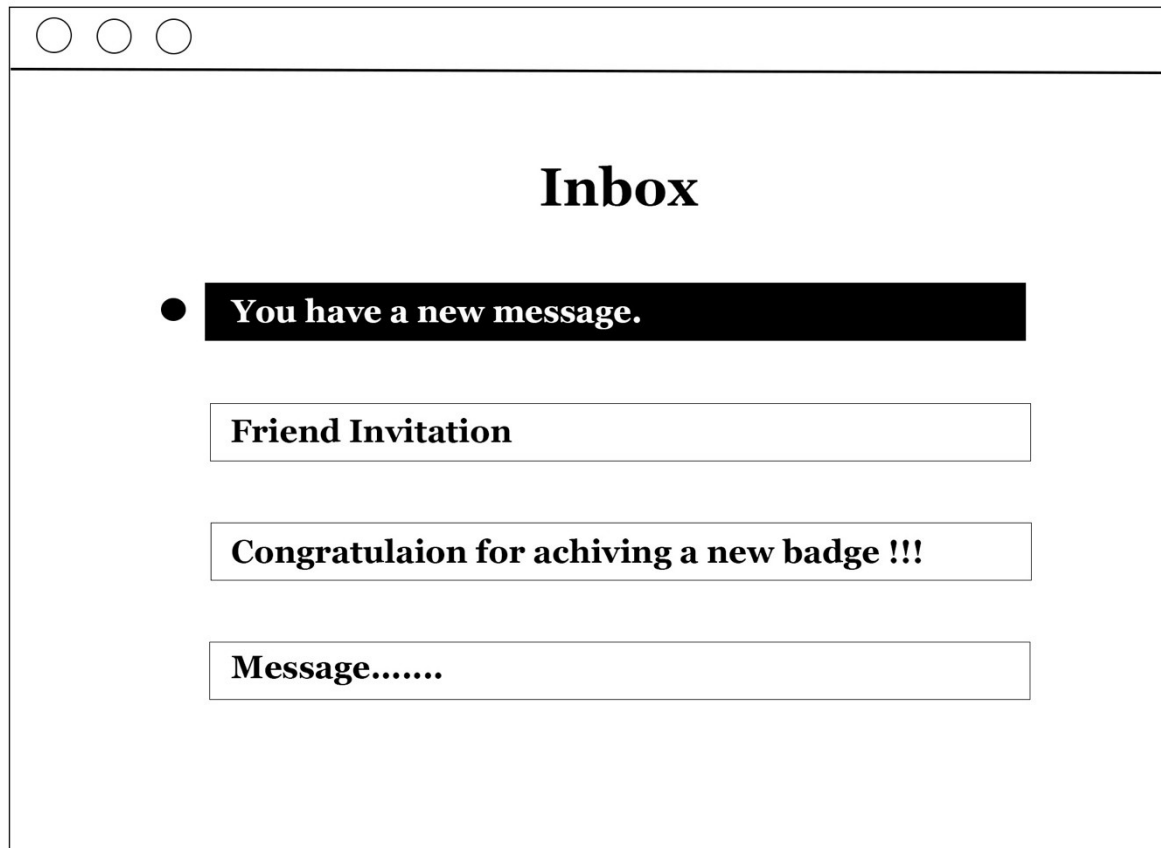


Figure 36 Prototype Screen - Inbox Page

The inbox is where users receive messages. A message that has not been read yet is indicated by a filled dot located next to it. This dot serves as a visual cue to let users know that they have not viewed the message yet. Once a message has been opened and read, the dot disappears, indicating that the message has been seen. This feature helps users to keep track of their unread messages and ensures that important messages are not missed.

12.21 Message Page

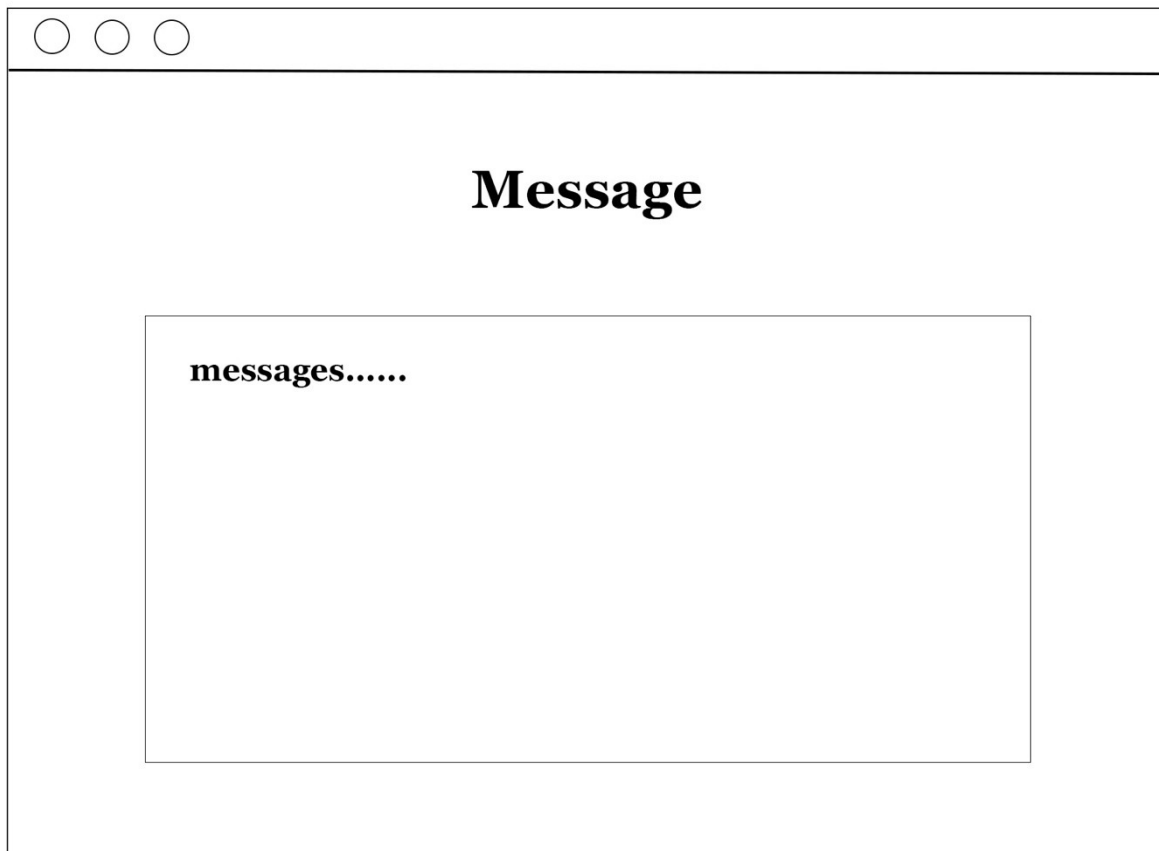


Figure 37 Prototype Screen – Message Page

The message on this page is meant to be read by users. However, there is no feature available for replying to this message.

13 Conclusion

This comprehensive document delves into the intricacies of the Bridge Puzzle Game application's design, encompassing its features, tools, technologies, and UI/UX. The application interactions are vividly illustrated through sequence diagrams and a class diagram. Additionally, the database layout serves to portray the storage of data. To top it off, a handful of prototype screens showcase the preliminary blueprints for the application. Overall, this document offers a lucid comprehension of the Bridge Puzzle Game application's design.