CLASSIC REALM

IN THE PARTIAL FULFILLMENT FOR THE AWARD OF THE DEGREE

OF

BACHELOR OF TECHNOLOGY

IN

INFORMATION TECHNOLOGY

Submitted by

KHUSHBU KOUL (ROLLNO: 2006943)

SANDEEP SINGH (ROLLNO: 2006947)

SHIVAM NAGI (ROLLNO: 2006948)

MANPRIT SINGH BAHL (ROLLNO: 2106793)

DEPARTMENT OF INFORMATION TECHNOLOGY

RAYAT BAHRA INSTITUTE OF ENGINEERING & NANOTECHNOLOGY

HOSHIARPUR –146104, PUNJAB (INDIA)

NOVEMBER, 2023

ACKNOWLEDGEMENT

We express my sincere gratitude to our respected guide Ms Kanchan Sharma, Rayat Bahra Institute of Engineering & Nano Technology, Hoshiarpur, for giving us stimulating guidance, continuous support and supervision throughout the course of present work.

We would like to place on record my deep sense of thankfulness to Er. Simarpreet Singh, Assistant Professor & Head of the Department, Department CSE & IT, Rayat-Bahra Institute of Engineering & Nanotechnology, Hoshiarpur, for his generous guidance, help and useful suggestions.

We express my sincere gratitude to my respected teacher, Dr. Jyotshna, Director Principal, Rayat-Bahra Institute of Engineering & Nanotechnology, Hoshiarpur, for giving us stimulating guidance, continuous support and supervision throughout the course of present work.

We also wish to extend my thanks to all faculty members, Rayat-Bahra Institute of Engineering & Nanotechnology, Hoshiarpur, their insightful comments and constructive suggestions to improve the quality of this research work.

We are extremely thankful to Dr. Chander Mohan, Campus-Director, Rayat-Bahra Institute of Engineering & Nanotechnology, Hoshiarpur, for providing me infrastructural facilities to work in, without which this work would not have been possible.

Khushbu Koul

Sandeep Singh

Shivam Nagi

Manprit Singh Bahl

ABSTRACT

The ClassicRealm project is an application that revives the charm of classic games in a modern, digital platform. Developed Java and Google Firebase in Android Studio, the application offers a seamless and engaging gaming experience.

One of the standout features of ClassicRealm is its ability to generate a unique game code for online mode. This feature enhances the flexibility of the gaming experience, enabling players to connect with friends and compete against each other, regardless of their location.

Another key feature is the inclusion of a scoreboard. This not only allows players to track their performance but also adds a competitive edge to the gaming experience, encouraging players to continually improve their skills and aim for higher scores.

ClassicRealm successfully combines the nostalgia of classic games with the convenience of modern gaming. It caters to a wide range of players, from those seeking to relive the golden era of gaming to the new generation of gamers looking to experience these timeless classics.

The success of this project demonstrates the potential of classic games in the modern gaming market and opens up opportunities for further development and expansion. The project team's commitment to delivering a high-quality gaming experience is evident in the application's design and functionality.

In conclusion, ClassicRealm is more than just a gaming application; it is a celebration of classic games and their enduring appeal. It stands as a testament to the timeless nature of these games and their ability to bring people together, making it a valuable addition to the gaming community.

TABLE OF CONTENT

LIST OF FIGURES

CHAPTER-1 OVERVIEW OF TECHNOLOGY USED FOR MAKING PROJECT

1.1 Android Studio

Figure 1.1 Android Studio

Android Studio is the official integrated development environment (IDE) for Google's Android operating system, built on JetBrains' IntelliJ IDEA software and designed specifically for Android development. It is available for download on Windows, macOS and Linux based operating systems. It is a replacement for the Eclipse Android Development Tools (E-ADT) as the primary IDE for native Android application development.

Android Studio was announced on May 16, 2013, at the Google I/O conference. It was in early access preview stage starting from version 0.1 in May 2013, then entered beta stage starting from version 0.8 which was released in June 2014. The first stable build was released in December 2014, starting from version 1.0. At the end of 2015, Google dropped support for Eclipse ADT, making Android Studio the only officially supported IDE for Android development.

On May 7, 2019, Kotlin replaced Java as Google's preferred language for Android app development. Java is still supported, as is C++.

The following features are provided in the current stable version:

- Gradle-based build support.
- Android-specific refactoring and quick fixes.
- Lint tools to catch performance, usability, version compatibility and other problems.
- ProGuard integration and app-signing capabilities.
- Template-based wizards to create common Android designs and components.
- A rich layout editor that allows users to drag-and-drop UI components, option to preview layouts on multiple screen configurations.
- Support for building Android Wear apps.

- Built-in support for Google Cloud Platform, enabling integration with Firebase Cloud Messaging (Earlier 'Google Cloud Messaging') and Google App Engine.
- Android Virtual Device (Emulator) to run and debug apps in the Android studio.

Android Studio supports all the same programming languages of IntelliJ (and CLion) e.g. Java, C++, and more with extensions, such as Go; and Android Studio 3.0 or later supports Kotlin, and "all Java 7 language features and a subset of Java 8 language features that vary by platform version." External projects backport some Java 9 features. While IntelliJ states that Android Studio supports all released Java versions, and Java 12, it's not clear to what level Android Studio supports Java versions up to Java 12 (the documentation mentions partial Java 8 support). At least some new language features up to Java 12 are usable in Android.

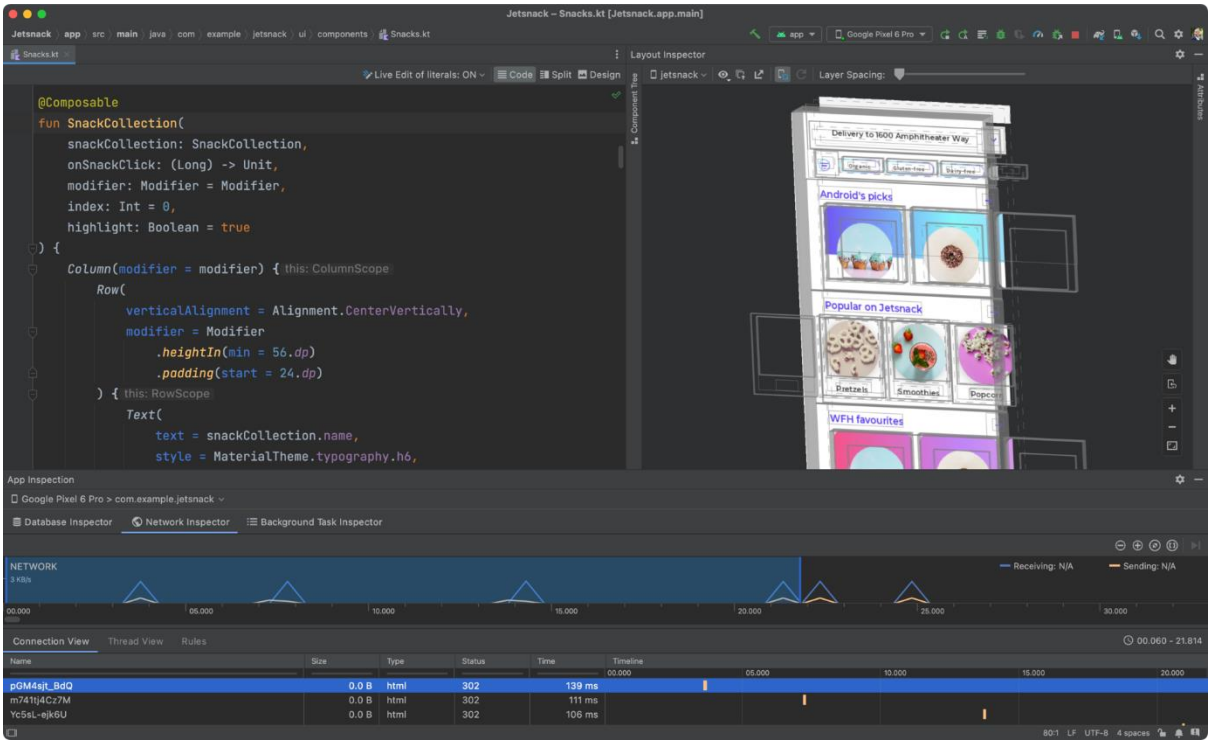

Figure 1.2 Android Studio Interface

1.2 Java

Java is a high-level, class-based, object-oriented programming language that is designed to have as few implementation dependencies as possible. It is a general-purpose programming language intended to let programmers write once, run anywhere (WORA), meaning that compiled Java code can run on all platforms that support Java without the need to recompile. Java

Figure 1.3 Java

applications are typically compiled to bytecode that can run on any Java virtual machine (JVM) regardless of the underlying computer architecture. The syntax of Java is similar to C and C++, but has fewer low-level facilities than either of them. The Java runtime provides dynamic capabilities (such as reflection and runtime code modification) that are typically not available in traditional compiled languages.

Java gained popularity shortly after its release, and has been a very popular programming language since then. Java was the third most popular programming language in 2022 according to GitHub and it is ranked fourth on TIOBE index as of October 2023. Although still widely popular, there has been a gradual decline in use of Java in recent years with other languages using JVM gaining popularity.

Java was originally developed by James Gosling at Sun Microsystems. It was released in May 1995 as a core component of Sun Microsystems' Java platform. The original and reference implementation Java compilers, virtual machines, and class libraries were originally released by Sun under proprietary licenses. As of May 2007, in compliance with the specifications of the Java Community Process, Sun had relicensed most of its Java technologies under the GPL-2.0-only license. Oracle offers its own HotSpot Java Virtual Machine;however, the official reference implementation is the OpenJDK JVM which is free open-source software and used by most developers and is the default JVM for almost all Linux distributions.

Java is the official language for Android App Development. Most of the apps in the Google Play Store are built with Java and it is also the most supported language by Google.

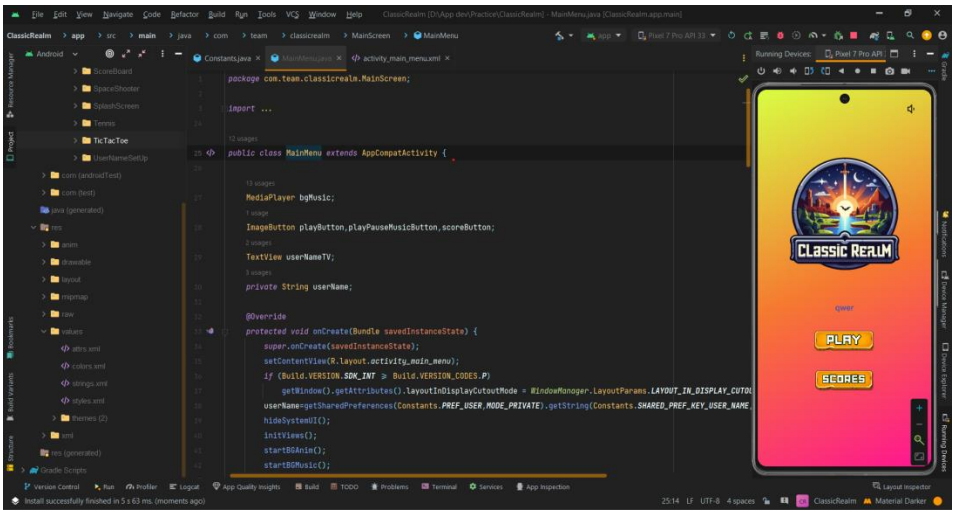

Figure 1.4 Java and Android Studio

1.3 Firebase

Firebase, Inc. is a set of backend cloud computing services and application development platforms provided by Google. It hosts databases, services, authentication, and integration for a variety of applications, including Android, iOS, JavaScript, Node.js, Java, Unity, PHP, and C++.

Figure 1.5 Firebase

Firebase evolved from Envolve, a prior startup founded by James Tamplin and Andrew Lee in 2011. Envolve provided developers an API that enables the integration of online chat functionality into their websites. After releasing the chat service, Tamplin and Lee found that it was being used to pass application data that were not chat messages. Developers were using Envolve to sync application data such as game state in real time across their users. Tamplin and Lee decided to separate the chat system and the real-time architecture that powered it. They founded Firebase as a separate company in 2011 and it launched to the public in April 2012.

- **Authentication**. Firebase provides a secure and easy way for users to sign into their app. Developers can use Firebase Authentication to support email and password login, Google Sign-In, Facebook Login and more.
- **Cloud Firestore**. Cloud Firestore is a flexible, scalable database for mobile, web, and server development from Firebase and Google Cloud Platform.
- **RealtimeDatabase**. The Firebase Realtime Database is a cloud-hosted NoSQL database that lets organizations store and sync data in real time across all of their users' devices. This makes it easy to build apps that are always up to date, even when users are offline.
- **Crashlytics**. Firebase Crashlytics is a service that helps organizations track and fix crashes in their app. Crashlytics provides detailed reports on crashes, so they can quickly identify the root cause and fix the problem.

- **Performance Monitoring.** Firebase Performance Monitoring provides insights into the performance of their app. Organizations can use Performance Monitoring to track metrics like CPU usage, memory usage and network traffic.

1.3.1 Firebase Realtime database

The Firebase Realtime Database is a cloud-hosted database. Data is stored as JSON and synchronized in realtime to every connected client. When you build cross-platform apps with our Apple platforms, Android, and JavaScript SDKs, all of your clients share one Realtime Database instance and automatically receive updates with the newest data.

The Firebase Realtime Database lets you build rich, collaborative applications by allowing secure access to the database directly from client-side code. Data is persisted locally, and even while offline, realtime events continue to fire, giving the end user a responsive experience. When the device regains connection, the Realtime Database synchronizes the local data changes with the remote updates that occurred while the client was offline, merging any conflicts automatically.

The Realtime Database provides a flexible, expression-based rules language, called Firebase Realtime Database Security Rules, to define how your data should be structured and when data can be read from or written to.

The Realtime Database is a NoSQL database and as such has different optimizations and functionality compared to a relational database. The Realtime Database API is designed to only allow operations that can be executed quickly. This enables you to build a great realtime experience that can serve millions of users without compromising on responsiveness.

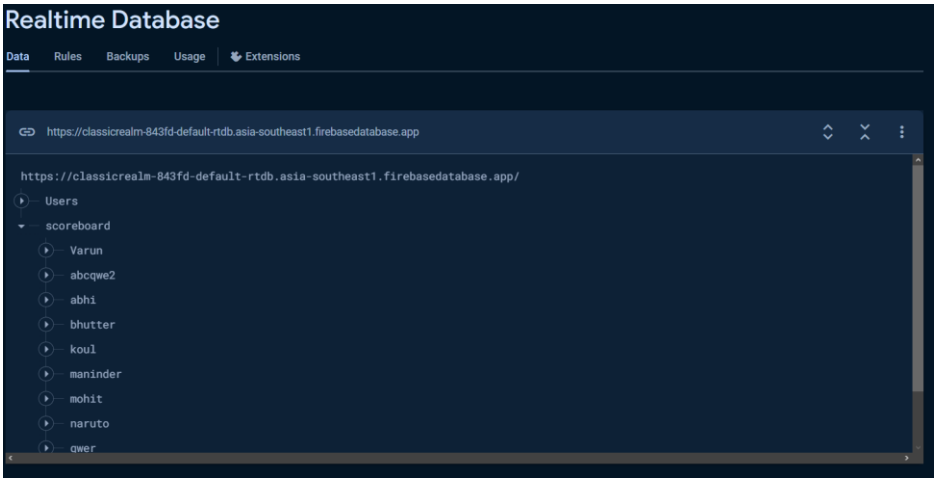

Figure 1.6 Firebase realtime database

CHAPTER-2 PROJECT REPORT WORK

2.1 INTRODUCTION

ClassicRealm is a classical multiplayer game application.

Our multiplayer classic game application is designed to bring together players from all over the world, providing them with a nostalgic gaming experience that transports them back to the golden era of gaming. The aim of this project is to recreate and celebrate the timeless classics that have stood the test of time, allowing players to connect and compete in beloved titles.

This application contains games like Tic-Tac-Toe, Break Out and more classical games. It is an application which has both single player and multiplayer games which players can play both off-line and on-line.

Moreover, player can generate Game code and join game by entering game code in on-line mode. This way players can play the game anytime and anywhere with friends.

This project is built as a part of our android development journey in the famous programming language Java along with Google Firebase (that helps mobile app developers build, deploy and scale their apps) all of this done in Android studio.

ClassicRealm solves this issue by combining different categories of classic games into one application. Games like Tic-Tac-Toe, which is a Multiplayer game to Break Out, which is a single player game, this app encapsulates a variety of games.

Furthermore, this application as a feature of keeping a score board, which helps player to keep track of their performance so that next time they can aim higher. This feature is rarely available in other applications that are available on the internet.

Lastly, ClassicRealm is a great application for anyone who wants to relive the experience of playing classical games with their friends or alone.

2.1.1LITERATURE SURVEY

- Classic Game Applications

Classic game applications have been studied extensively in the literature. A systematic literature review of game-based learning discusses how game-based learning, both in the

form of serious games and gamification, has developed rapidly and penetrated various fields of science.

- Multiplayer Game Applications

Research on multiplayer game applications has shown that they have many potential benefits. A systematic literature review shows that no type of play is overly dominant with the ratio between multi-player and single-player being 57:43. Another study on massively multiplayer online games concludes a significant positive relationship between playing multiplayer games and social well-being.

- Use of Firebase in Game Development

The use of Firebase in game development has also been explored in the literature. A study on game development with Google Firebase discusses how Firebase real-time database can compete with real DBs like MySQL and NoSQL Firebase. Another study demonstrates the application of Firebase in an Android app where images along with strings are loaded to Firebase and retrieved from Firebase similar to Instagram.

2.2.OBJECTIVES

The objectives of this project are simplified below and three major points are needed to cover, those are:

- To build an android multiplayer gaming application containing multiple single player and multiplayer games that can be played both off-line and on-line by using a set up of Game Code to enter the gaming environment.
- To provide a gaming platform with a score board section which will show users progress and create a multiplayer battle platform which users can use to track their progress and scores.
- To adopt an easy to use and understand User Interface so that user can have a wonderful gaming experience.

2.3METHODOLOGY

2.3.1 FLOWCHARTS / DFD'S

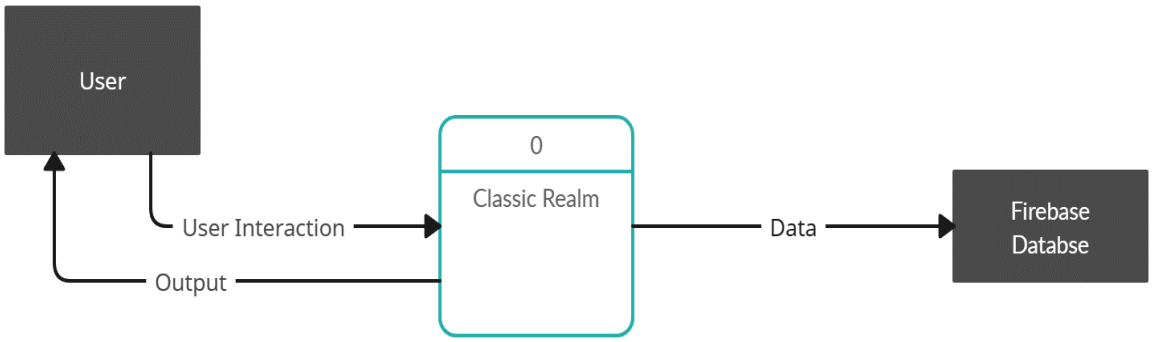

Figure 2.1 0-level DFD

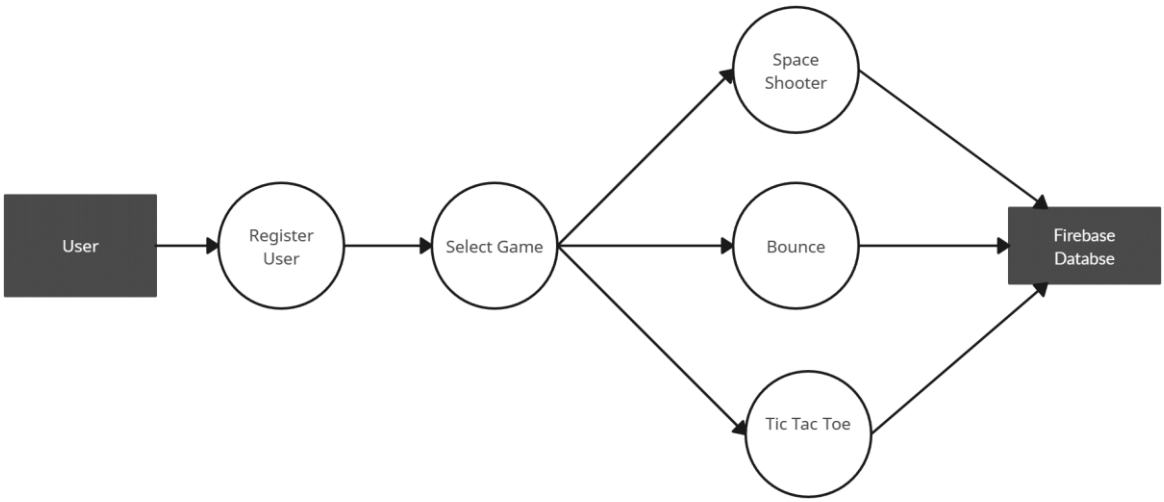

Figure 2.2 1-level DFD

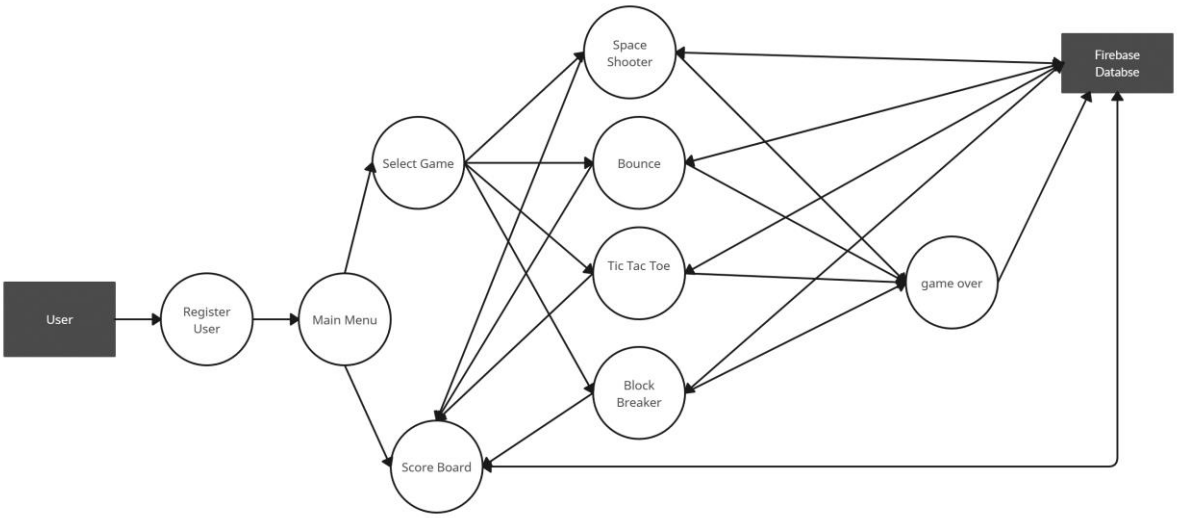

Figure 2.3

2.3.2 JASON TREE DIAGRAM

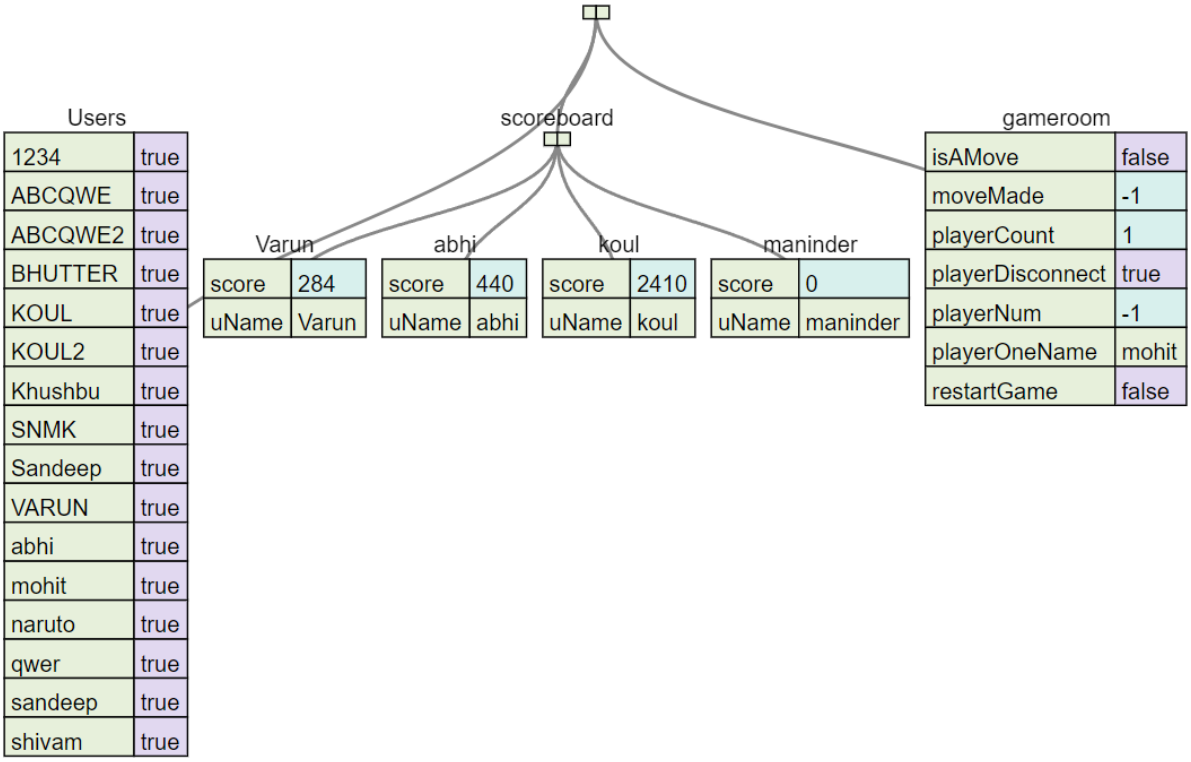

Figure 2.4 JASON TREE DIAGRAM

2.3.3 USE CASE DIAGRAMS

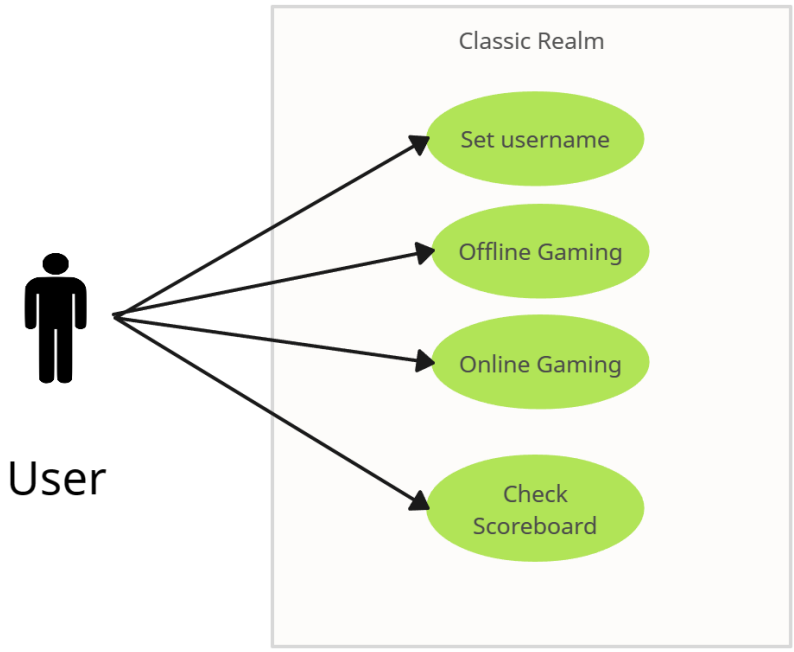

Figure 2.5 Use Case Diagram

2.3.4 DATABASE STRUCTURE

The Firebase Realtime Database is a cloud-hosted database. Data is stored as JSON and synchronized in realtime to every connected client. When you build cross-platform apps with our Apple platforms, Android, and JavaScript SDKs, all of your clients share one Realtime Database instance and automatically receive updates with the newest data.

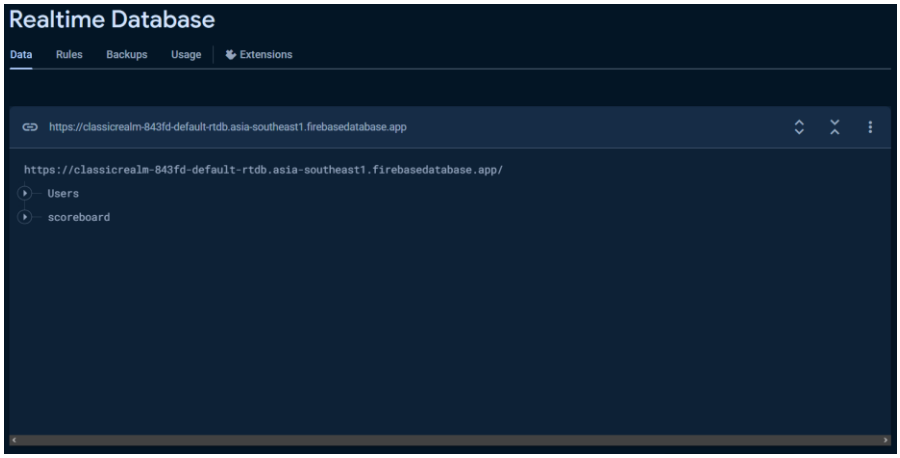

Figure 2.6 DATABASE STRUCTURE

2006943 2006947
2006948 2106793

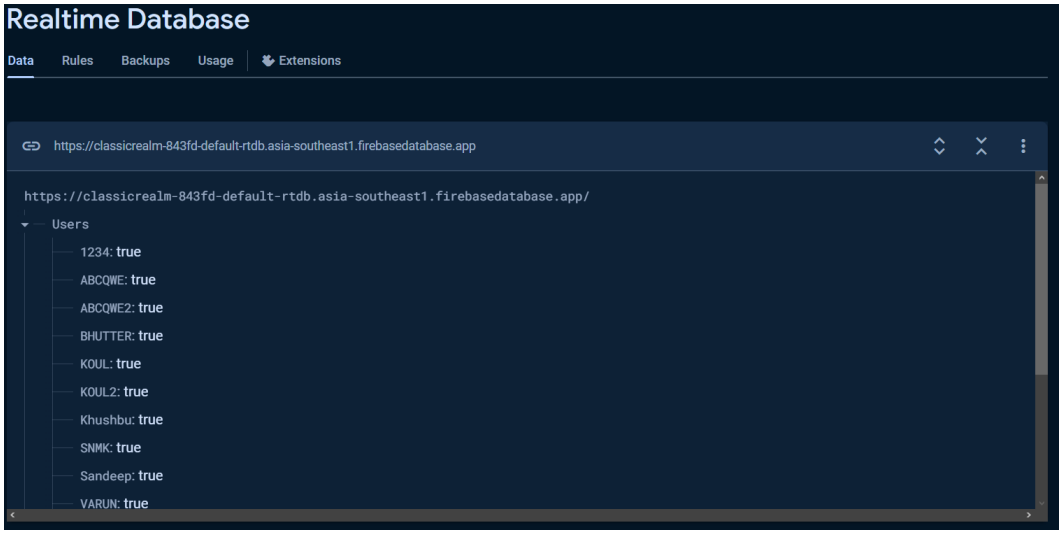

Figure 2.7 User Data

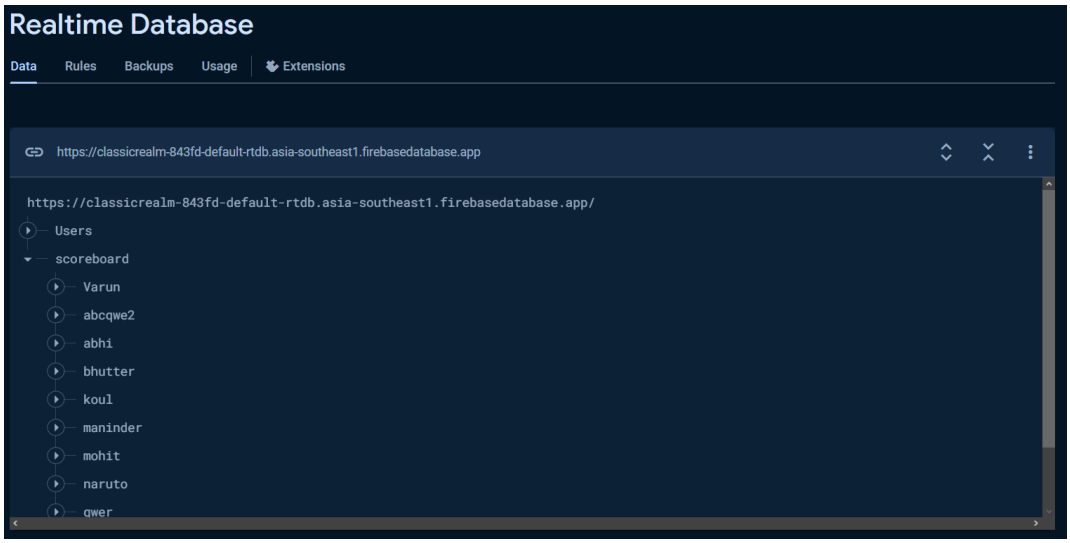

Figure 2.8 Scores

2.3.5 PROJECT PLANNING GANTT CHART

Figure 2.9 Gantt chart

2.4HARDWARE / SOFTWARE REQUIREMENTS

2.4.1 Hardware Requirement

1. Computer/Laptop
   - x86 64-bit CPU (Intel / AMD architecture).
   - 8GB RAM (16GB preferable).
   - 5GB free storage.

2. Android Device
   - Android 7 Nougat or above.

2.4.2 Software Requirement

1. Operating system:
   - Windows 10 or 11
   - Mac OS X 10.11 or higher, 64-bit
   - Linux: RHEL 6/7, 64-bit

2. Android Studio

3. Web Browser (For Firebase Console).

CHAPTER-3 USER GUIDE

3.1 SCREEN SHOTS WITH LABELLING

Figure 3.1 SplashScreen

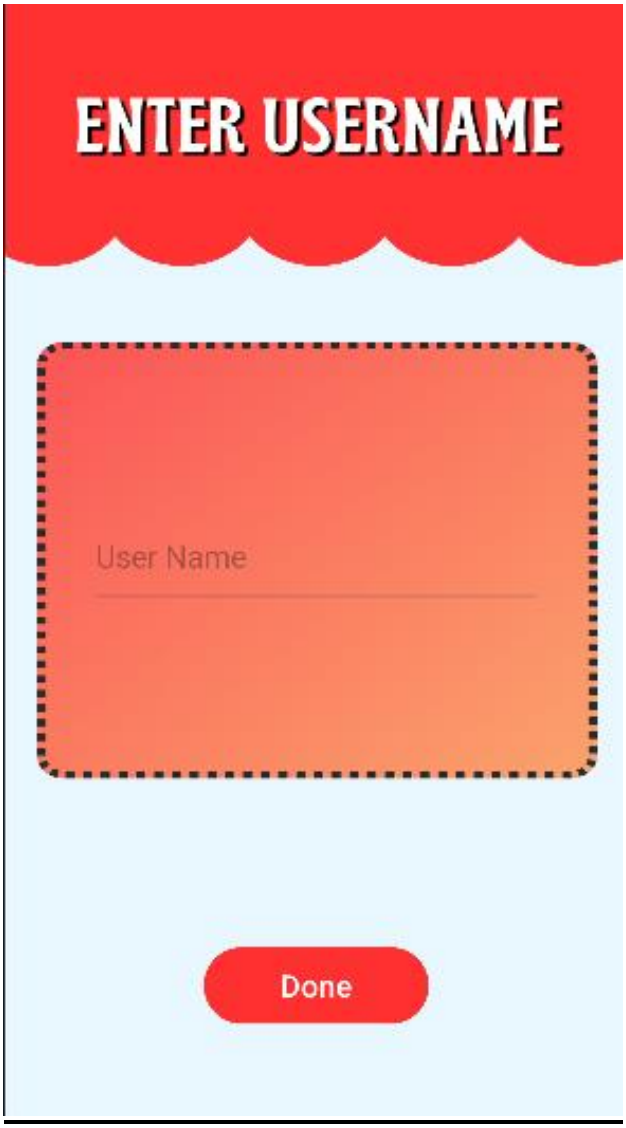

Figure 3.2 User name

2006943 2006947
2006948 2106793

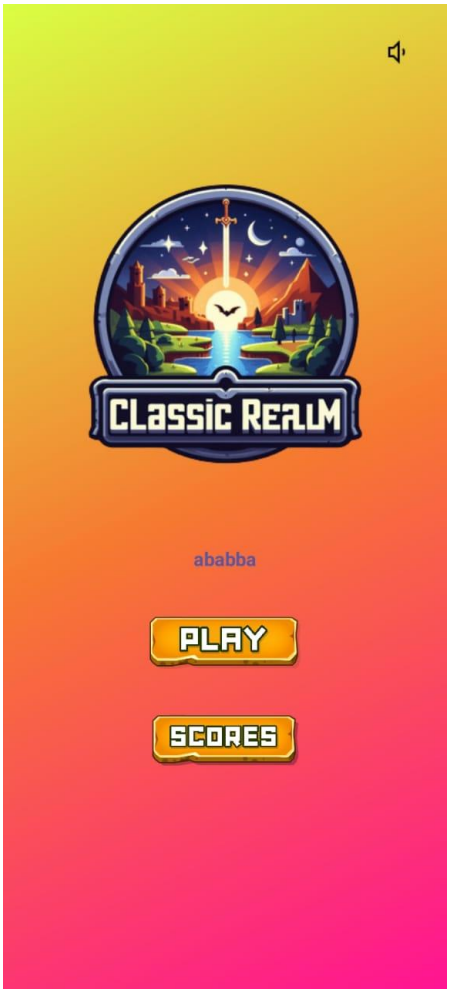

Figure 3.3 Main screen

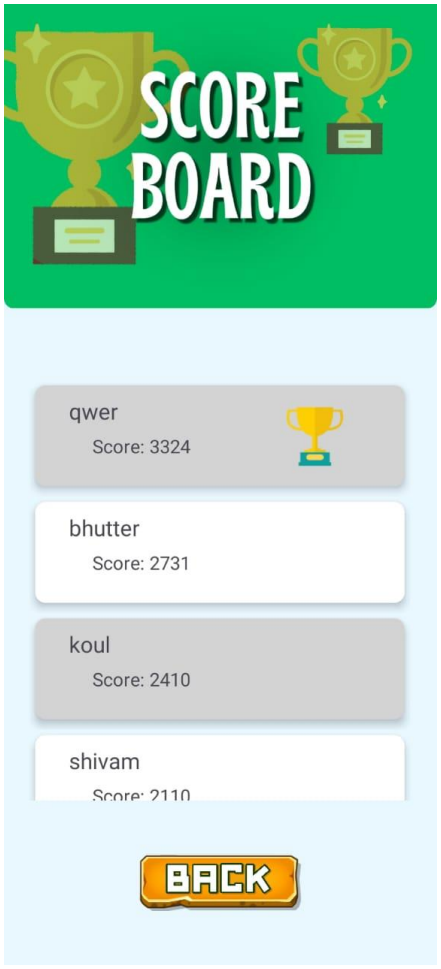

Figure 3.4 ScoreBoard

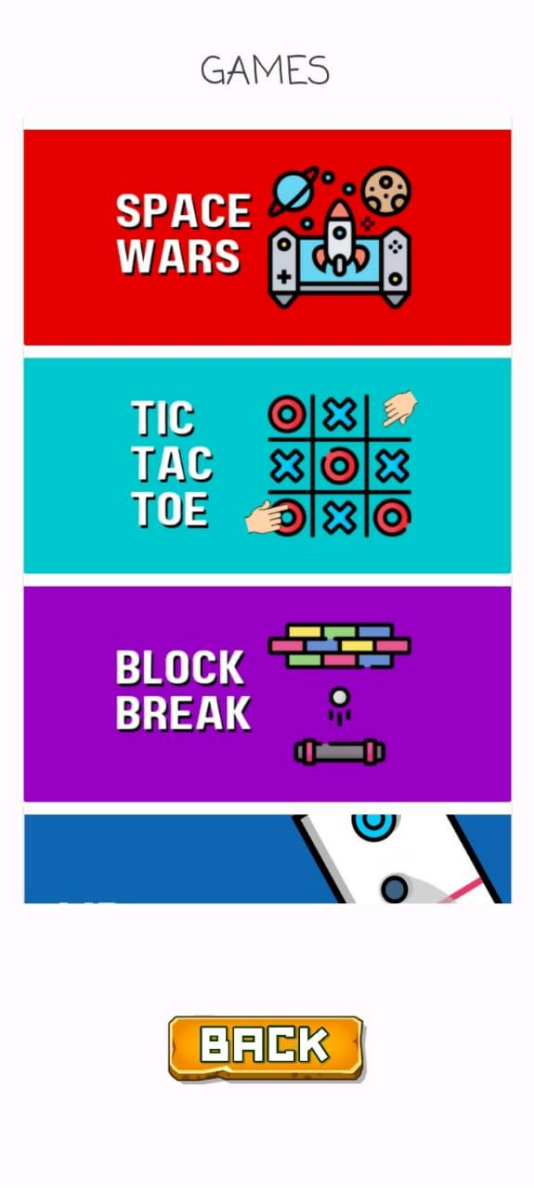

Figure 3.5 Games

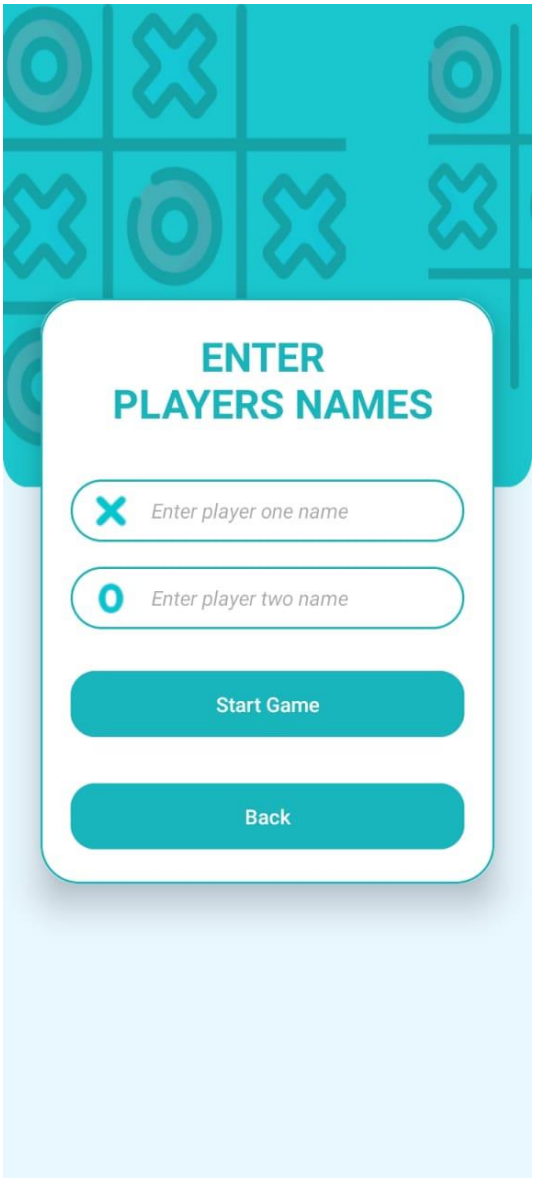

Figure 3.6 TicTac Offline

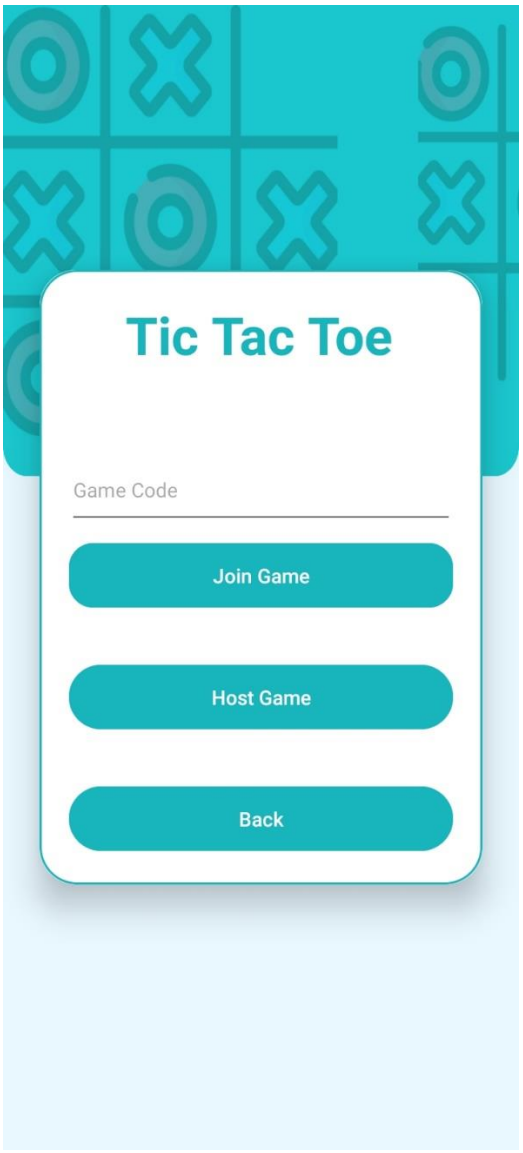

Figure 3.7 TicTac Online

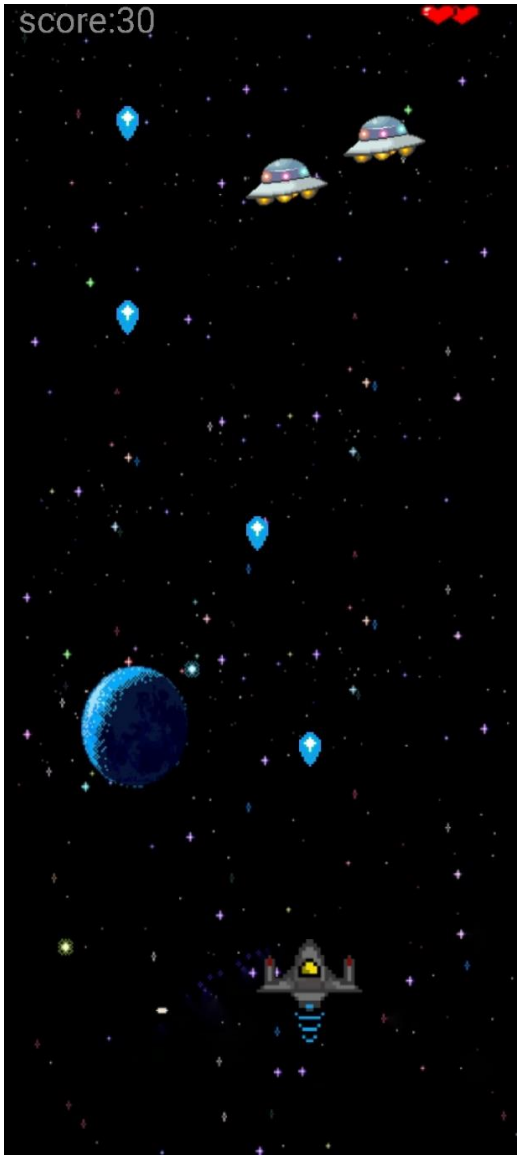

Figure 3.8 SpaceWars

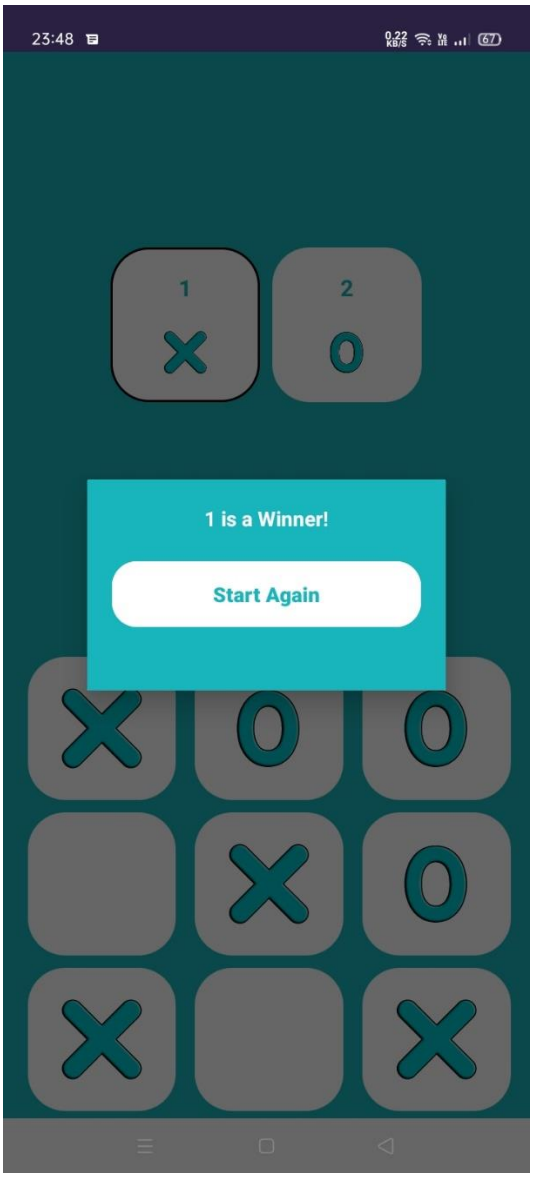

Figure 3.9 TicTac GameOver

Figure 3.10 SpaceWars GameOver

CHAPTER-4 CONCLUSION & FUTURE SCOPE

4.1CONCLUSION

In conclusion, the ClassicRealm project successfully achieves its goal of providing a platform for both single and multiplayer classic games. By leveraging Java and Google Firebase in Android Studio, the application offers robust performance and scalability. The unique feature of generating a game code for online mode enhances the flexibility and accessibility of the gaming experience, allowing players to connect with friends anytime, anywhere.

The inclusion of a scoreboard adds a competitive edge and encourages players to continually improve their skills. This feature, which is rarely found in similar applications, sets ClassicRealm apart from its competitors.

ClassicRealm effectively combines nostalgia with modern gaming convenience, offering a unique and engaging gaming experience. It is a testament to the timeless appeal of classic games and their ability to connect players across generations. The project serves as a valuable contribution to the gaming community and stands as a remarkable achievement in the field of Android development.

4.2FUTURE SCOPE

1. Game Selection: The project should include a well-curated selection of classic multiplayer games. The scope could involve continually expanding the game library based on user demand.

2. Platform: ClassicRealm is designed for Android devices, but the scope may include potential expansion to other platforms like iOS or web-based versions in the future.

3. Gameplay Modes: The project includes both single-player and multiplayer modes, both online and offline. Expanding or enhancing these modes may be part of the scope.

4. Features: The scope encompasses features like chat functionality, user profiles, leaderboards, and customization options. Additional features may be added to enhance user experience.

REFERENCES

1. Android Documentation

- Author: Google

- URL: https://developer.android.com/docs

- Description: The official Android documentation provided by Google offers a wealth of information on Android app development, UI/UX design, and Firebase integration.

2. Firebase Documentation

- Author: Google

- URL: https://firebase.google.com/docs

- Description: Firebase's official documentation provides comprehensive guides and resources on integrating Firebase services into Android apps.

3. Firebase YouTube Channel

- Author: Firebase Team

- URL: https://www.youtube.com/c/Firebase

- Description: The official Firebase YouTube channel offers video tutorials and demos on various Firebase services, which can be helpful for learning implementation.

4. "Firebase in a Weekend" (Coursera Course)

- Author: Google

- URL: https://www.coursera.org/learn/firebase-android

- Description: A Coursera course offered by Google that teaches how to integrate Firebase into Android apps. It includes practical exercises and examples.

5. "Android Canvas and Drawables" (Official Android Developer Documentation)

- Author: Google

- URL: https://developer.android.com/guide/topics/graphics/2d-graphics

- Description: The official Android Developer Documentation provides an extensive guide on working with the Android Canvas and Drawables for 2D graphics. It covers essential concepts and techniques for creating graphics and games on the Android platform.