1. INTRODUCTION

ClassicRealmis 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 is a great application for gaming because it combines different categories of classic games into one application which can be enjoyed by every kind of user. 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 has 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.

1.1 Literature Survey

Exiting Systems

In the android gaming ecosystem, numerous applications offer a compilation of classic android games. While these applications excel at preserving classic titles for modern devices, they often fall short in providing essential features for user engagement.

- Missing multiplayer functionality.
- Lack of real time scoreboards.

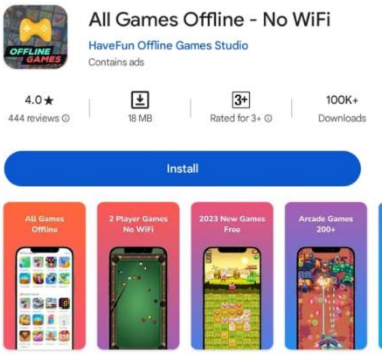


Figure 1 All Games Offline

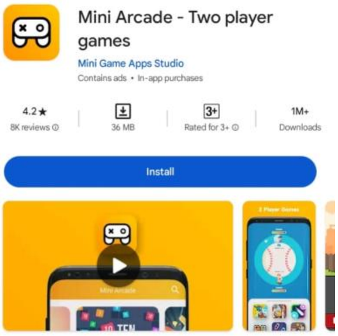


Figure 2 Mini Arcade

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 withGoogle 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.

1.2 Technology used

1.2.1 Java

Java is a most popular, object-oriented, widely used programming language and platform that is utilized for Android development, web development, artificial intelligence, cloud applications, and much more. So, mastering this gives you great opportunities in bigger organizations. Java was developed by Sun Microsystems (which is now the subsidiary of Oracle) in the year 1995. James Gosling is known as the father of Java. Before Java, its name was Oak. Since Oak was already a registered company, so James Gosling and his team changed the name from Oak to Java.

Figure 3 Java

Some important Java key points are given below:

- Object Oriented − In Java, everything is an Object. Java can be easily extended since it is based on the Object model.
- Platform Independent − Unlike many other programming languages including C and C++, when Java is compiled, it is not compiled into platform specific machine, rather into platform independent byte code. This byte code is distributed over the web and interpreted by the Virtual Machine (JVM) on whichever platform it is being run on.

- Secure − With Java's secure feature it enables to develop virus-free, tamper-free systems. Authentication techniques are based on public-key encryption.
- Portable − Being architecture-neutral and having no implementation dependent aspects of the specification makes Java portable. Compiler in Java is written in ANSI C with a clean portability boundary, which is a POSIX subset.
- Robust − Java makes an effort to eliminate error prone situations by emphasizing mainly on compile time error checking and runtime checking.
- Multithreaded − With Java's multithreaded feature it is possible to write programs that can perform many tasks simultaneously. This design feature allows the developers to construct interactive applications that can run smoothly.

1.2.2 FireBase

Google Firebase is a set of cloud-based development tools that helps mobile app developers build, deploy and scale their apps.Firebase's first product was the Firebase Realtime Database, an API that synchronizes application data across iOS, Android, and Web devices, and stores it on Firebase's cloud. The product assists software developers in building real-time, collaborative applications.

Figure 4 Firebase

Some important key features of Google firebase are given below:

- Cloud Messaging
- Authentication
- Test Lab
- Hosting
- Remote Configuration
- Dynamic Links
- Crash Reporting
- Real-time Database
- Storage
- Notifications

1.2.3 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.

Figure 5 Android Studio

At the end of 2015, Google dropped support for Eclipse ADT, making Android Studio the only officially supported IDE for Android development.

Some important features of android studio:

- Instant App Run
- Visual Layout Editor
- Intelligence Code Editor
- Help to Build Up App for All Devices
- Help to Connect with Firebase
- Support KOTLIN
- Color Previews
- Maven Repository

2. Feasibility Study

2.1Technical Feasibility

ClassicRealm will use technology such as

- Java
- Firebase
- Android Studio

which are publicly available to all the users, and anyone can use these technologies with basic specifications of computer.

Furthermore, these technologies are not difficult to learn and information about these technologies are readily available, making**ClassicRealm** a technically feasible project.

2.2 Economical Feasibility

All of the technologies used or related to software's development such as Java (the language), Android Studio (the IDE) are made to use by users free of cost that will be used to make this project are free of cost. Firebase provides free plan with limited resources.

2.3Operational feasibility

This project will have a simpler user interface which will makes it easy to use by any type of users older or younger. User of the age 12+ can easily play and use this application without any problem.

3. Objectives

Objectives of any project reflects developers point of view of how they want to create the project written down in simpler points. 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 setup 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.

4. Methodology/ Plan of work

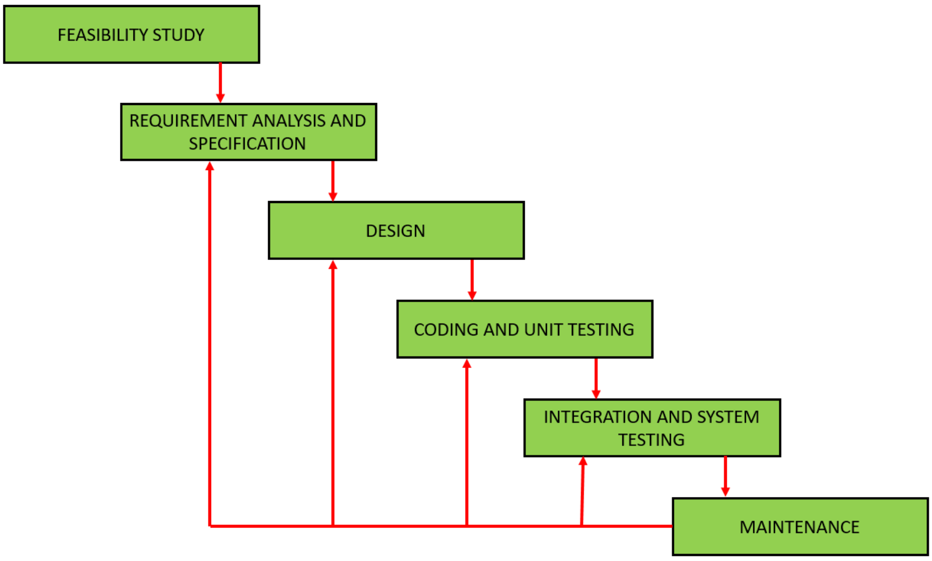

FEASIBILITY STUDY
REQUIREMENT ANALYSIS AND SPECIFICATION
DESIGN
CODING AND UNIT TESTING
INTEGRATION AND SYSTEM TESTING
MAINTENANCE

Figure 6 Iterative Waterfall Model

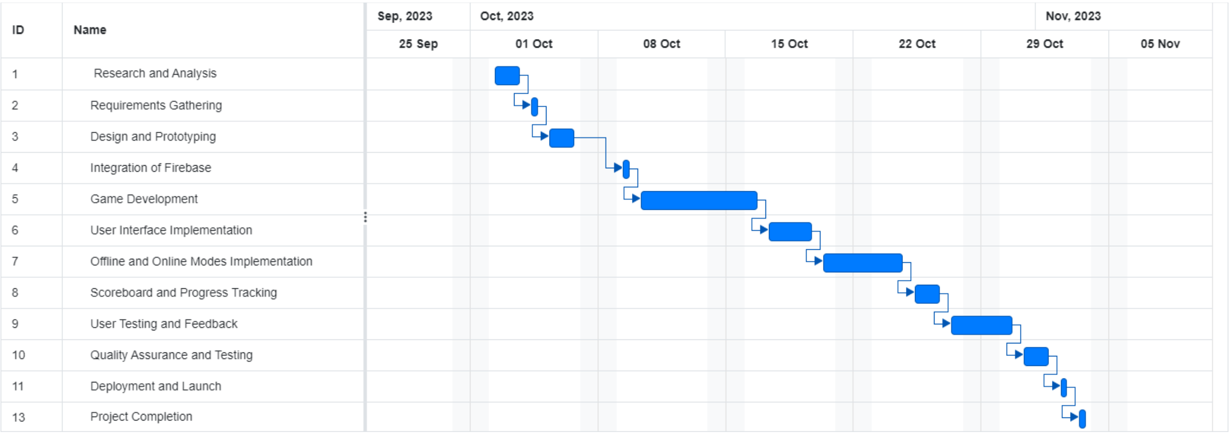
Phase 1: Project Planning

1.Requirements Gathering:

- Identify and document the specific classic multiplayer games to be included in the project.
- Define the features and functionalities of the games.

2. Project Schedule:

- Develop a detailed project timeline with milestones for each phase and activity.
- Allocate resources, including developers, designers, and testers, as per the schedule.



ID	Name
1	Research and Analysis
2	Requirements Gathering
3	Design and Prototyping
4	Integration of Firebase
5	Game Development
6	User Interface Implementation
7	Offline and Online Modes Implementation
8	Scoreboard and Progress Tracking
9	User Testing and Feedback
10	Quality Assurance and Testing
11	Deployment and Launch
13	Project Completion

Figure 7 Gantt Chart

Phase 2: System Design

3. Game Design:

- Create comprehensive game design documents for each classic game, specifying rules, gameplay, and user interfaces.

- Design the visual elements, including graphics and user interfaces for the games.

4. Technical Design:

- Develop technical architecture and infrastructure requirements.

- Define the database schema and data storage strategy.

- Select and integrate the Google Firebase services for real-time database, authentication, and cloud functions.

Phase 3: Implementation

5. Game Development:

- Begin the development of each classic game according to the design specifications.

- Follow coding standards and best practices in Java for Android development.

- Integrate Firebase SDKs for authentication and real-time database functionality.

Phase 4: Testing and Quality Assurance

6. Unit Testing:

- Conduct unit testing for individual game components and features.

- Ensure code correctness, functionality, and responsiveness.

7. Integration Testing:

- Test the integration of multiple games and the overall system.

- Verify that Firebase services work seamlessly with the application.

Phase 5: Deployment

8. Alpha Testing:

- Invite a select group of users or testers to participate in alpha testing.
- Gather feedback and identify any issues or bugs.

9. Beta Testing:

- Release a beta version to a larger group of users for further testing.
- Continue to gather feedback and make necessary improvements.

Phase 6: Finalization

10. Bug Fixes and Improvements:

- Address and resolve any issues, bugs, or user feedback.
- Ensure the application meets the desired performance and quality standards.

11. Iterative Improvement:

- Plan for periodic updates and enhancements to add new games, features, and improvements based on user feedback and emerging trends.

5. Facilities required for proposed system / Hardware and software requirements.

5.1 HardwareRequirement

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

2. **Android Device**
   - Android 7Nougat or above.

5.2 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).**

Bibliography

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.