Chor Sipahi Gaming App

A PROJECT REPORT
Submitted By

NIRANJAN DARSHAN
University Roll No 1900290140021

ABHISHEK KUMAR YADAV
University Roll No 1900290140002

DIVYANSH PANDEY
University Roll No 1900290140014

SAURABH VISHWKARMA
University Roll No 1900290140030

Submitted in partial fulfillment of the
Requirements for the Degree of

MASTER OF COMPUTER APPLICATIONS

Under the Supervision of
DR. VIPIN KUMAR
ASSOCIATE PROFESSOR

Submitted to

DEPARTMENT OF COMPUTER APPLICATIONS
KIET Group of Institutions, Ghaziabad
Uttar Pradesh-201206
(AUGUST 2021)

CERTIFICATE

Certified that **Niranjan Darshan (Enrollment No. 1900290140021), Divyansh Pandey (Enrollment No.1900290140014), Saurabh Vishwkarma (Enrollment No. 1900290140030), Abhishek Kumar Yadav (Enrollment No. 190029014005145)** have carried out the project work having " **Chor Sipahi** " for Master of Computer Applications from Dr. A.P.J. Abdul Kalam Technical University (AKTU), Technical University, Lucknow under my supervision. The project report embodies original work, and studies are carried out by the student himself/herself and the contents of the project report do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University/Institution.

Date:

Niranjan Darshan
University Roll No. 1900290140021
Divyansh Pandey
University Roll No. 1900290140014
Saurabh Vishwkarma
University Roll No. 1900290140030
Abhishek kumar Yadav
University Roll No. 1900290140002

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Date:

Dr. Vipin Kumar
Associate Professor
Department of Computer Applications
KIET Group of Institutions, Ghaziabad

Signature of Internal Examiner **Signature of External Examiner**

Dr. Ajay Shrivastava
Head, Department of Computer Applications
KIET Group of Institutions, Ghaziabad

ABSTRACT

Raja, Mantri, Chor, Sipahi (King, Minister, Thief and Police) is a wonderful four people guessing game is where the ‘Mantri’ has to identify the ‘Chor.’ This is a popular indoor game played by 6 to 12 year olds.

The game begins with making 4 chits namely Raja (1000 points) , Mantri (800 points) , Chor(0 points) and Sipahi (500 points). Each chit is folded and the 4 players are asked to pick one chit. Players open their chit to find out their character. The Raja exclaims “Mera Mantri Kaun Hai?” meaning ‘Who is my Minister?’ The Mantri then has to identify the ‘Chor’ from the other 2 remaining players. If the Mantri, guesses correctly then the points are retained or else he/she surrenders them to the Chor. Several rounds of this game are played before counting the points. The player with the highest score wins the game.

An extension of this game is seen when played in larger groups. New characters are added to accommodate more players. The players predetermine the character that will have to be identified, and also the characters that would have to be revealed in the beginning of the game. The guessing game is similar to what is played with 4 people but older children increase the difficulty level of this game by including more unrevealed characters thereby making it difficult to guess.

The game enhances face reading skill and helps develop cognitive thinking by requiring deductive reasoning. This entertaining game can be learned quickly and is a tool to make new friends. This game provides a platform for social interaction and enhances communication skills

ACKNOWLEDGEMENTS

Success in life is never attained single handedly. My deepest gratitude goes to my thesis supervisor, **Dr. Vipin Kumar** for his guidance, help and encouragement throughout my research work. Their enlightening ideas, comments, and suggestions.

Words are not enough to express my gratitude to **Dr. Ajay Kumar Shrivastava, Professor and Head, Department of Computer Applications**, for his insightful comments and administrative help at various occasions.

Fortunately, I have many understanding friends, who have helped me a lot on many critical conditions.

Finally, my sincere thanks go to my family members and all those who have directly and indirectly provided me moral support and other kind of help. Without their support, completion of this work would not have been possible in time. They keep my life filled with enjoyment and happiness.

Niranjan Darshan

Divyansh Pandey

Abhishek Kumar Yadav

Saurabh Vishwkarma

Table of Contents

List of Figures

INTRODUCTION

1.1 PROJECT DESCRIPTION

This project has been developed to override the problems prevailing in the practicing manual system. This application is supported to eliminate and in some cases reduce the hardships faced by the existing system. Moreover, this application is designed for moving the young generations toward the tradition in this world of digitalization and carry out operations in a smooth and effective manner.

No formal knowledge is needed for the user to use this system. Thus, by this all it proves it is user friendly. So, This “Chor Sipahi Gaming Application” can lead to error-free, secure, reliable and fast system.

This application provides the way to build cognitive and analytical skills among the players . Through this applications we are approaching towards the less use of papers by virtue of which we are indirectly sustaining the environment.

Chapter 2
PROJECT CATEGORY

2.1 TECHNOLOGIES USED

Android Studio :-Android Studio is the official Integrated Development Environment (IDE) for Android app development, based on IntelliJ IDEA . On top of IntelliJ's powerful code editor and developer tools, Android Studio offers even more features that enhance your productivity when building Android apps, such as:

- A flexible Gradle-based build system
- A fast and feature-rich emulator
- A unified environment where you can develop for all Android devices
- Apply Changes to push code and resource changes to your running app without restarting your app
- Code templates and GitHub integration to help you build common app features and import sample code
- Extensive testing tools and frameworks
- Lint tools to catch performance, usability, version compatibility, and other problems

TECHNOLOGIES USED

APPLICATION	: Android Studio
DESIGNING	: XML
Programming Language	: Java

2.2 Language Used (Designing and Developing)

This project has been developed XML and Java.

•XML:- **XML** stands for extensible markup language. A markup language is a set of codes, or tags, that describes the text in a digital document. The most famous markup language is hypertext markup language (HTML), which is used to format Web pages..

- **JAVA:-** Java is an object-oriented programming language developed by **Sun Microsystems**, and it was released in 1995.
- James Gosling initially developed Java in **Sun Microsystems** (which was later merged with **Oracle Corporation**).
- Java is a set of features of C and C++. It has obtained its format from C, and OOP features from C++.
- Java programs are platform independent which means they can be run on any operating system with any processor as long as the Java interpreter is available on that system.
- Java code that runs on one platform does not need to be recompiled to run on another platform; it's called **write once, run anywhere(WORA)**.

Chapter 3

SOFTWARE REQUIREMENT SPECIFICATION

3.1 GENERAL DESCRIPTION

Chor Sipahi is a gaming Android Application whose main objective is to Connect the young generation which is moving toward the Virtual/ Digital World about their traditional game. It is the Four Player game. It is developed and designed in Java and xml using Android Studio.

3.1.1 PROBLEM STATEMENT:

The problem occurred before having computerized system includes:

- Seeking for the help to play this traditional game.
- Excessive use of Paper for making chit and updating score.
- More chance of Unfairness while giving marks due to biasness.
- Helping the friend to identify thief with the use of symbol violate the rule of the game.

3.2 SYSTEM OBJECTIVES

3.2.1 Improvement in control and performance

The system is developed to cope up with the current issues and problems of forgetting the traditional game. The system identify who is accessing the game and the score will be updated on the scoreboard. To declare the winner of the game the game is played for 10 round and after 10 round the winner of the game will be disclosed according to score board.

Save cost

As this gaming apps will be free so the people for their wards will not have to pay anything for it. It will be available to all without any cost.

Save Time

People at any location will able to play this game by creating or joining the room.

3.2.2 Requirement Specification:

The application requirement specification is produced at the analysis task. The function and performance allocated to application as part of system engineering are refined by establishing a complete information description, a detailed functional and behavioral description, an indication of performance requirements and design constraints.

3.3 Functional Requirements:

Modes of Play:

The application will work both on Online as well as offline mode.

Internet Connectivity:

As discussed that Application will work on Online mode so it need regular Internet Connectivity to play the Game online.

Name and Number of the Players:

To Play manually or Offline mode there must be four players and Each player must give there name so that they can be easily identified while playing and guessing and for ranking on the scoreboard

Email id of the Players :

To play online the user have to first login with their valid mail id required either for creating room or joining room.

Those who will create the room must have the email of all the players to add them in a room.

3.4 Non-functional Requirements:

Performance Requirements

- User friendly**:** The system should be user friendly so that it can easily be understand by the user without any difficulty.
- Ease of maintenance :- System should be easy to maintain and use.
- Less time consuming: The system should be less time consuming which could be achieved by good programming.
- Error free: The system should easily handle the user error in any case.
- Static: Application runs on stand alone machine . Support only single user.

3.5 SOFTWARE AND HARDWARE REQUIREMENTS

This section describes the software and hardware requirements of the system .

3.5.1 SOFTWARE REQUIREMENTS

· **Operating system**- Android Operating system is used to operate and support the application.

Database: Firebase is used as database it is easy to maintain and retrieve data . It is easy to understand and easy to write as the predefined guide are present to assist you.

· **Development tools and Programming language**- XML is used to write the whole designing code. Java is the main language on which this gaing application is developed.

3.5.2 HARDWARE REQUIREMENTS

- Android Mobile Phone of API Level 16 and Onward.
- Memory used 6.31MB with 5.98 MB of application and 340 KB of Data.

3.6 EXISTING VS PROPOSED SYSTEM

Existing system does not have a facility of Chor Sipahi gaming application with fairness in game whereas proposed system has a facility of gaming application with the similar fairness as we do in playing manually.

Existing system does not have any facility of Playing game Online whereas proposed system is working on the facility of playing game online by creating / Joining room.

Existing System does not have the facility of Declaring winner after particular numbers of round Whereas proposed system declare the winner after 10 round.

3.7 . Software System Attributes

- **Portability**:- The system should be machine independent.

- **Security**:- The system is designed in such a way that it will store the recorded data in the system of the owner. The system will be secure from unauthorized access of the application.

- **Maintainability**: The system will be designed in a maintainable order. The system can be easily modified and renewed according to the need of the organization.

3.8 . Feature of Chor Sipahi game

- no internet connection required play against the computer
- multiple users can play on their own phone or all on the same phone
- you can invite and play with school or college friends
- emoji stickers can be used to tease other players during the game
- graphics with a classic look and the feel of a royal game
- classic scoreboard to display scores of each round
- its also called as raja mantri chor police in india
- its also called as chor police game in india
- security of data to be stored
- ensures data accuracy (number of alert generated)
- minimize manpower
- minimize manpower
- minimize time consumption
- greater efficiency
- fast
- better services
- user friendliness and interactive
- minimum time requried
- easy to update
- user friendly
- free for the user

chor sipahi is a family game as well as kids game that was once played by kings and now it can be enjoyed by you and your family and friends. while the chor sipahi gameplay might seem simple at first, the chor sipahi game .

chor siphai is a perfect time pass game of raja mantri chor siphai chit game. you played chor siphai in your childhood with papers, now play chor sipahi on your phone and tablet.ready to shuffle The chits! guess The Thief or become the king of raja mantri chor sipahi

3.9. Preliminary investigation:

Fact Finding:

After obtaining the background knowledge, we began to collect data on the existing system.

The tools that are used in information gathering are as follows:

- Online Apps observation.
- Review of the peoples.

The model we have used is Incremental Model. In this model, first of all the existing system is observed, then customer requirements are taken in consideration then planning, modelling, construction and finally deployment and again adding the new system if asked by the customer to do so.

3.10. Model used: Incremental Model

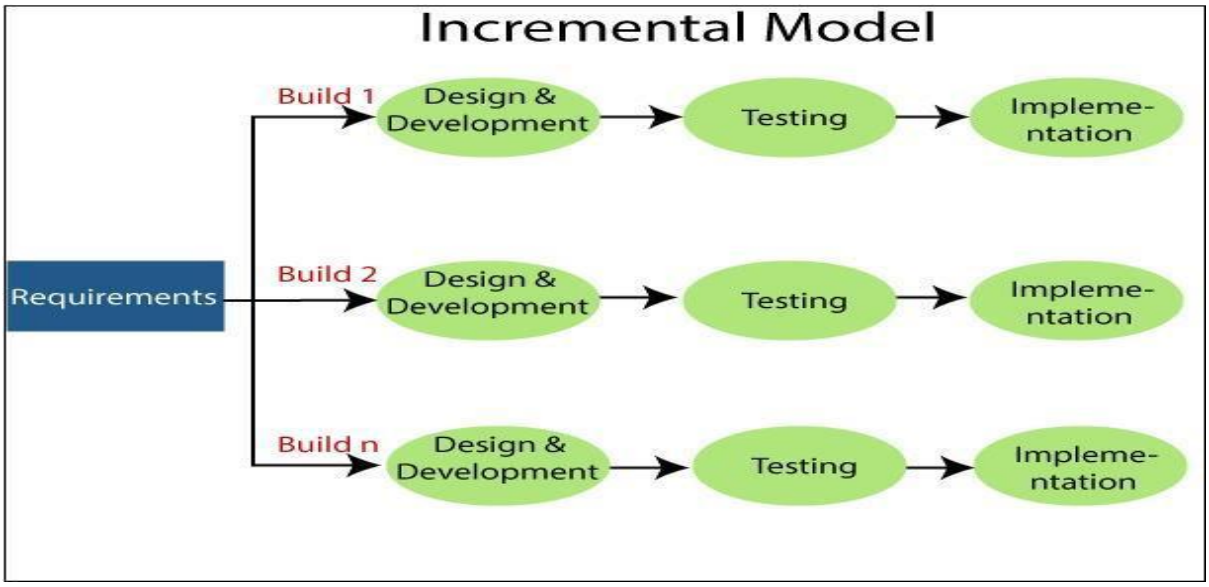


Fig 1.0: Incremental Model

Incremental Model is a software development process where requirements are divided into several stand-alone software development modules. In this project

the first increment is often a core product where the basic requirements are addressed, and supplementary features are added in the next increments.

3.11 Preliminary Description:

The first step in the system development life cycle is the preliminary investigation to determine the feasibility of the system. The purpose of preliminary investigation is to evaluate project requests. It is not a design study nor does it include the collection of details to describe the system in all respect. Rather, it is the collecting of information that helps committee members to evaluate the merits of project request and make an informed judgement about the feasibility of the proposed project.

Analyst working on the preliminary investigation should accomplish the following objectives:

- Clarify and understand the project request.
- Determine the size of the project.
- Access costs and benefits of alternative approaches.
- Determine the technical and operational feasibility of alternative approaches.
- Report the findings to management with recommendations outlining the acceptance and rejection of the proposal

Chapter 4

Feasibility study

After studying and analyzing all the existing and requires functionalities of the system, the next task is to do the feasibility study for the project. Feasibility study includes consideration of all the possible ways to provide a solution to a given problem. The proposed solution should satisfy all the user requirements and should be flexible enough so that future changes can be easily done based on the future upcoming requirements.

- ## 4.1 Economical Feasibility:

  For the economic feasibility, Economic analysis or cost/benefits analysis is most frequently used technique the effectiveness of a proposed system. it is a procedure to determine the benefits and saving those are expected from the proposes system and compare them with cost .if the benefits outweigh the costs, a decision is taken to design and implement the system. otherwise, further justification or alternative in proposed system will have to be made if it is to have a chance of being approved this is ongoing effort that improves in accuracy at each phase of a system life cycle

- ## 4.2 Technical feasibility:

  This included the study of function, performance and constraints that may affect the ability to achieve an acceptable system. For this feasibility study, we studied complete functionalities to be provided in the system, as described in the System Requirement Specification (SRS), and checked if everything was possible using different type of front end and back end platform.

4.3 Operational Feasibility:

No doubt the technically growing world needs more enhancement in technology, this apps is very user friendly and all inputs to be taken all self-explanatory even to a layman. As far our study is concerned, the clients will be comfortable and happy as the system has cut down their loads and bring the young generation to the same virtual world they are growing drastically.

Operational feasibility cover two aspects.one technical performance

aspects and the other is acceptance within the organization.

Operation feasibility determine how the proposed the system will fit in with the current operation and what needs to implement the system

Chapter 5

Planning and scheduling

5.1 Gantt chart

A Gantt chart can be developed for the entire project or a separate chart can be developed for each function. A tabular form is maintained where rows indicate the task with milestones and columns indicate duration (Days).

Task Name	21 April	24 April	27 April	12 May	17 May	20 May
Welcome Page						
Module and Play						
List of players						
Game Page						
Create Room						
Join Room						

Fig 2.0 Gantt Chart

5.1.1 Software Requirements with specifications:

Name of Components	Specifications
Operating system	Android
Language	JAVA
Software Development kit	Android Studio
Markup Language Enable	Xml

5.1.2 Hardware Requirements with specifications:

Name of Components	Specifications
Android Mobile Phone	API level 16 Onward
Application	5.98 MB
Memory Used	6.31 MB

5.2 : DATA FLOW DIAGRAM

Are used to graphically represent the flow of data in a Chor Sipahi Gaming Application. DFD describes the processes that are involved in a system to transfer data from the input to the List of Player and transferring data to the Players game field and updating the data name in the Score borad.

Chor Sipahi Gaming Application when Played Online will have to perform to transfer data from input block to the database. Here The data in create room The user have to login first and the data will be stored to the database. The email of each invited player after accepting in join room will also be added and so the operation is done in the Application.

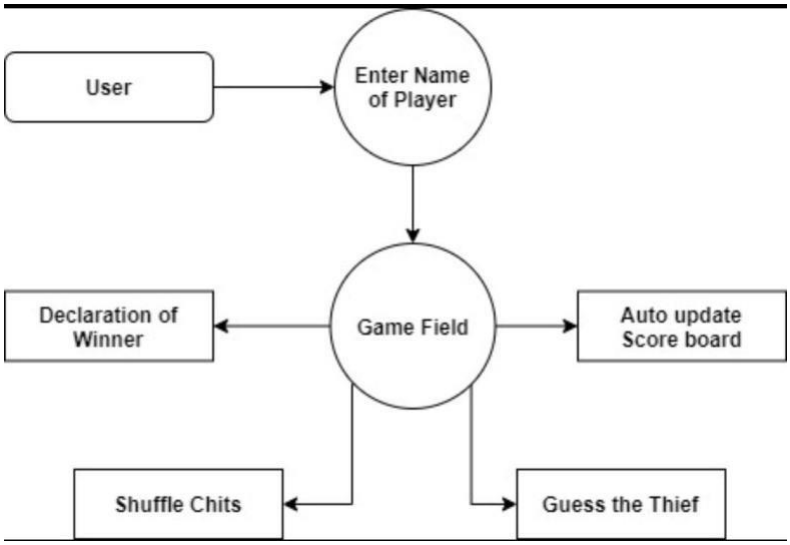


Fig 3.1 Offline Mode DFD

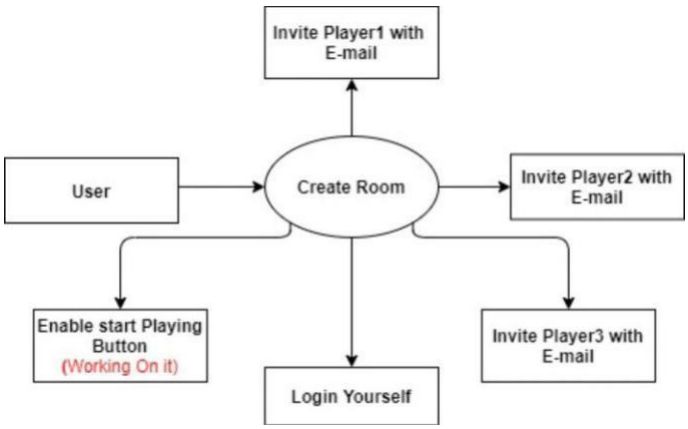


Fig 3.2 Online Mode Create Room

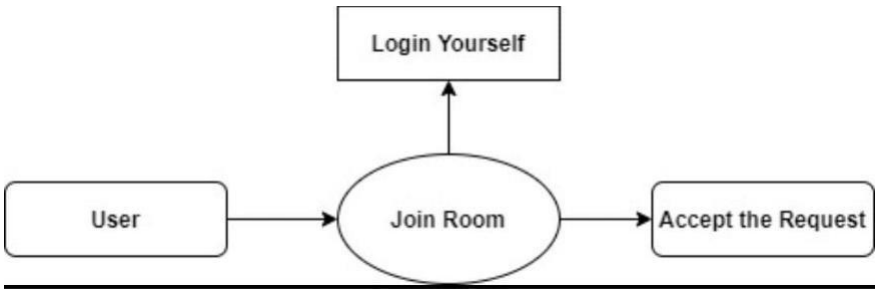


Fig 3.3 Online Mode Join Room DFD

5.3 ENTITY RELATIONSHIP DIAGRAM:

This ER Diagram represents the model of Chor Sipahi Gaming Applications Entity. The Entity Relationship Diagram show all visual instrument of Database table and relation between welcome page, modes of Play, List of Players, Online mode, offline mode, playing game field etc. All of it have Structured data and every entity may have some attributes.

Chor Sipahi Gaming Application Entity and their Attributes:

1. Modes Of Play: Attribute of Modes Of Play : Online And Offline Mode.
2. List of players: Attributes are: Player1 Name,Player2 Name, Player3 Name,player4 Name, Start Playing.
3. Playing Game Field: Attributes are: Four Chits, shuffle, Guess, ScoreBoard.
4. Online Mode: Attributes are: Join Room and Create Room.
5. Join Room: Attributes of Join room are: Login user email, Login, Invitor mail id, Accept.

6. Create Room: Attributes are: Player1 email, Player2 email, player3 Email, Login Yourself ,Invite, Login.

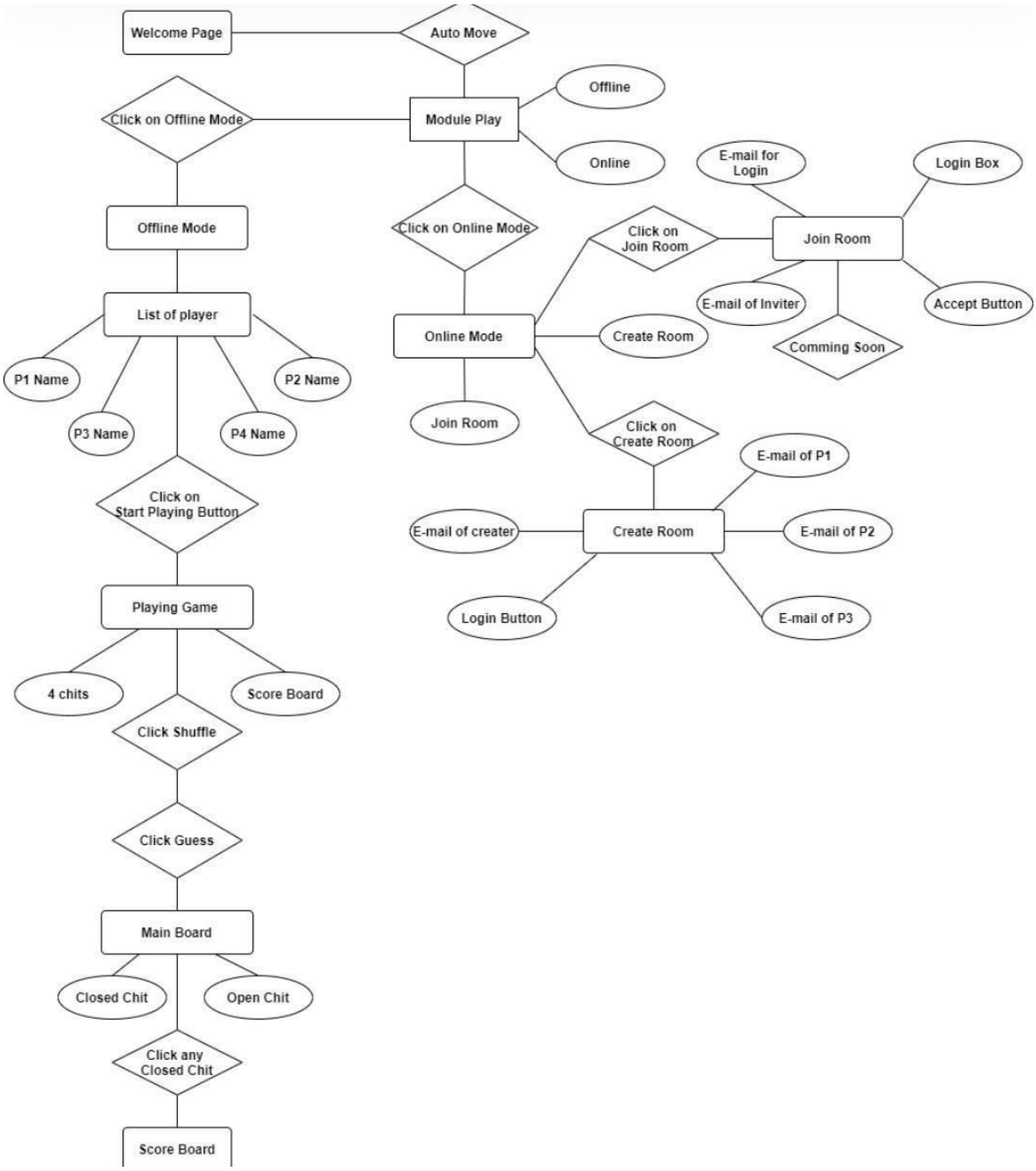


Fig 4.1 Entity relationship Diagram

5.4 Module Description

5.4.1. Welcome Page

Whenever the app will launch it will start with an introductory page. The question here is what will be there in Introductory page? Why it is included in any applications? The introductory page must have something that uniquely identify you mostly people use to include here logo of there application similarly we had also included Logo in the welcome or introductory page. The Application as a unique properties that take the people to the next Activity after 3 sec as the animation time to animate fully completed. Most Application including Chor Sipahi gaming Apps start there Application with the splash screen. A splash screen is **a graphical control element consisting of a window containing an image, a logo, and the current version of the software**. A splash screen can appear while a game or program is launching. A splash page is an introduction page on a website.

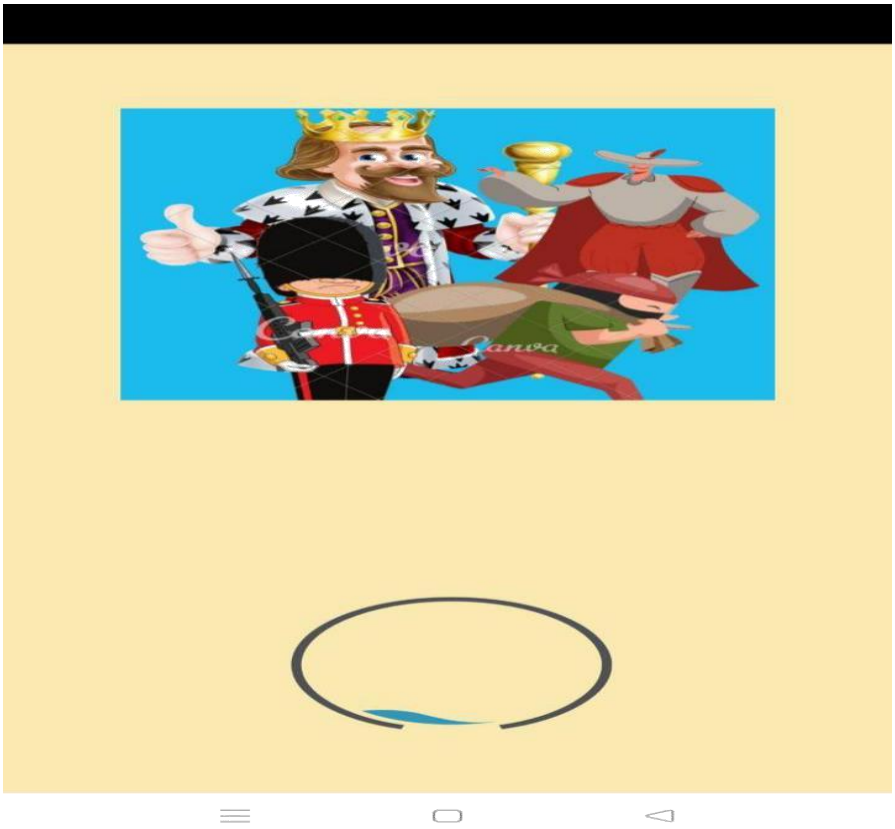


Fig 5.1 Welcome Page

Designing of Welcome Page

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#4BFFC107"

    tools:context=".Splashscreen">

    <com.airbnb.lottie.LottieAnimationView
```

```
    android:id="@+id/splash"
    android:layout_width="396dp"
    android:layout_height="297dp"

    app:layout_constraintBottom_toBottomOf="parent"

    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.466"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/imageView"
    app:layout_constraintVertical_bias="0.794"
    app:lottie_autoPlay="true"

    app:lottie_rawRes="@raw/splash" />

  <ImageView
    android:id="@+id/imageView"
    android:layout_width="265dp"
    android:layout_height="307dp"
    android:layout_marginTop="48dp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:srcCompat="@drawable/logo" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

Java Code of Welcome Page

```
package com.niru.newdemo;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.view.WindowManager;

public class Splashscreen extends AppCompatActivity {

    @Override
```

```
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_splashscreen);
        getSupportActionBar().hide();

getWindow().addFlags(WindowManager.LayoutParams.FLAG_FULLSC
REEN);

        Thread td=new Thread() {
            public void run() {
                try {
                    sleep(2200);

                } catch (Exception ex) {
                    ex.printStackTrace();

                } finally {
                    Intent intent=new Intent(Splashscreen.this,ModeOfPlay.class);
                    startActivity(intent);
                    finish();



                }
            }
        };
        td.start();
```

```
    }

}
```

5.4.2. Modes Of Play

After particular time duration the welcome page will take you to the intent page. The Activity with two button will be displayed Where the user have to select the modes in which they wants to play the Game Mostly it is found that user Play game in two modes :

Online Mode and Offline Modes.

Functionality of

Online Mode: The game will be played similarly among four player as played in Offline mode but all the player will be at different locations. In the world they need Internet connectivity till the end of the game to play without any hindrance.

Offline Mode: It is needed to list out the players name who are four in number and what ensure that everyone is at the same place and playing game on the Same device. It doesnot need any Internet Connectivity People can enjoy the game without internet connectivity

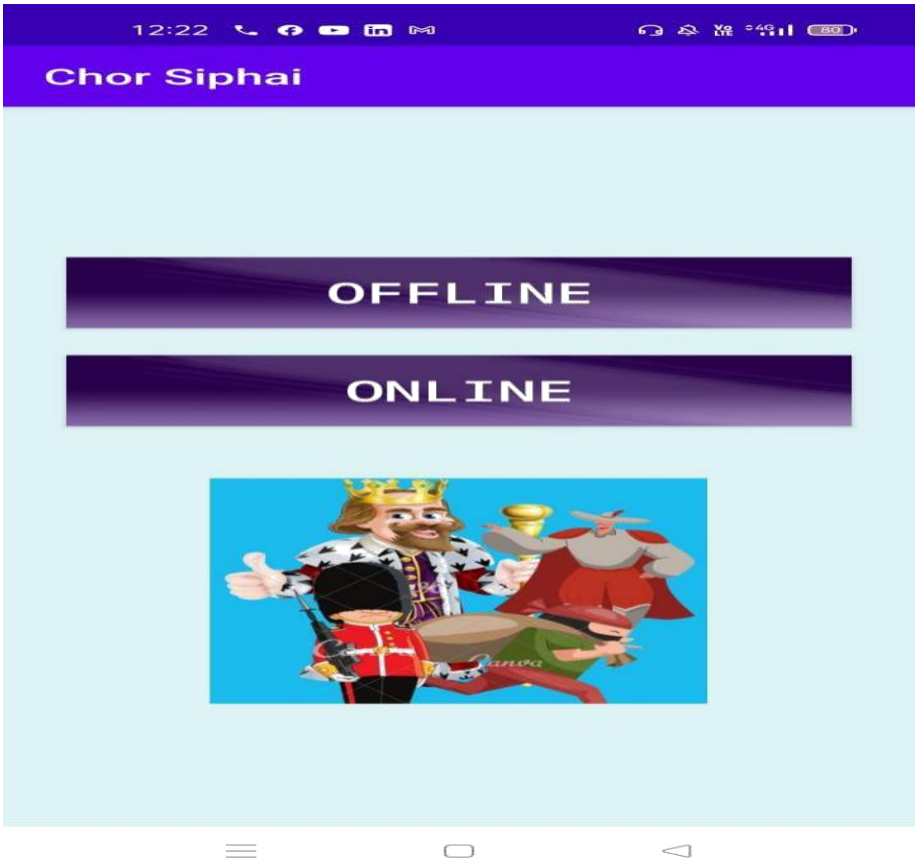


Fig 5.2 Modes Of Play

Designing of Modes of Playing

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:gravity="center"
    android:background="#5CACE6E8"
    tools:context=".ModeOfPlay">
```

```
<Button
    android:layout_width="match_parent"
    android:layout_height="65dp"
    android:text="Offline"
    android:id="@+id/offline"
    android:textSize="25dp"
    android:fontFamily="monospace"
    android:textStyle="bold"
    android:textColor="#fff"
    android:layout_marginTop="25dp"
    android:layout_marginLeft="25dp"
    android:layout_marginRight="25dp"
    android:gravity="center"
    android:background="@drawable/screenbg"/>
<Button
    android:layout_width="match_parent"
    android:layout_height="65dp"
    android:text="Online"
    android:id="@+id/online"
    android:textSize="25dp"
    android:fontFamily="monospace"
    android:textStyle="bold"
    android:textColor="#fff"
    android:layout_marginTop="25dp"
    android:layout_marginLeft="25dp"
    android:layout_marginRight="25dp"
    android:gravity="center"
```

```
        android:background="@drawable/screenbg"/>
    <ImageView
        android:id="@+id/imageView"
        android:layout_width="265dp"
        android:layout_height="207dp"
        android:layout_marginTop="48dp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:srcCompat="@drawable/logo" />

</LinearLayout>
```

Java Code for Modes of Play

```
package com.niru.newdemo;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;

public class ModeOfPlay extends AppCompatActivity
    { Button online,offline;
```

```
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_mode_of_play);
        online=findViewById(R.id.online);
        offline=findViewById(R.id.offline);
        online.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent=new
Intent(getApplicationContext(),MainActivity.class);
                startActivity(intent);
                finish();
            }
        });

        offline.setOnClickListener(new View.OnClickListener()
            { @Override
            public void onClick(View v) {
                Intent intent=new
Intent(getApplicationContext(),Playername.class);
                startActivity(intent);
                finish();
            }
        });
```

5.4.3. Offline Mode

When the user select the Offline Mode then the Intent Activity will open with the activity ask you to enter the name of the four player who wants to participate in game. It is required to have four player to be available for game.

List Of Player: In Offline Mode entering the name of the player to play game is the great requirement to identify each player. That is the reason why this Activity have four input field.It will be easy to declare and identify the player with there during turn or in Leaderboard.

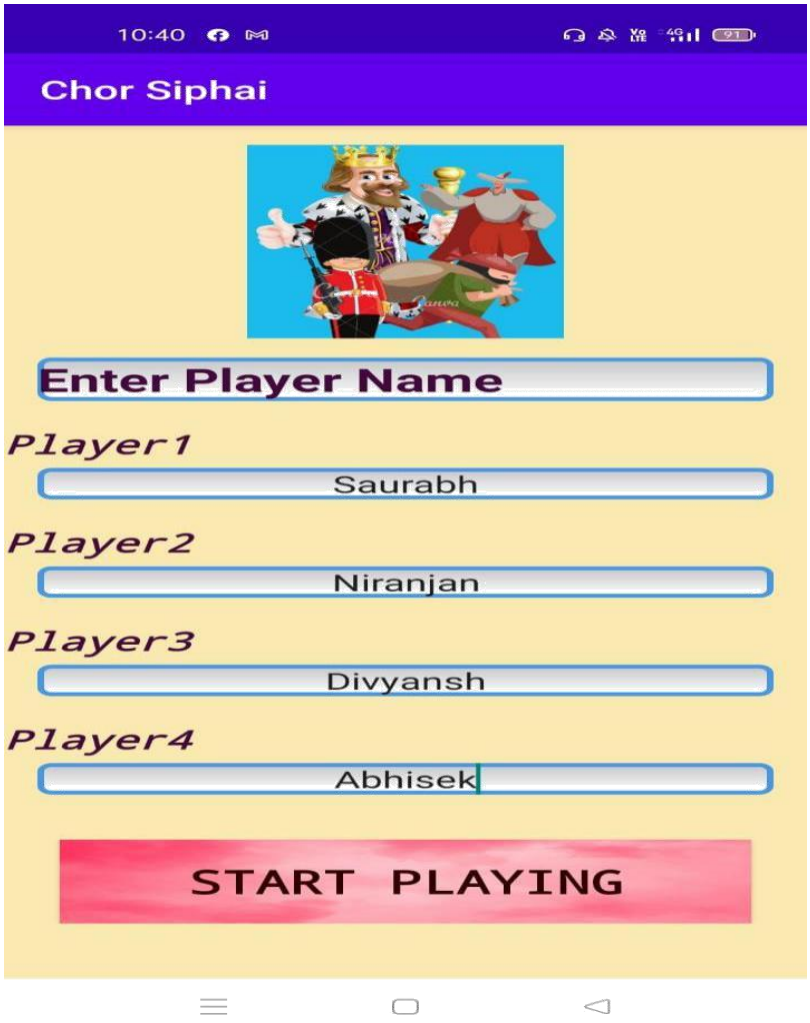


Fig 5.3 List of Players

Designing of List of Players

```
<?xml version="1.0" encoding="utf-8"?>
<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
```

```
android:layout_height="match_parent"
android:orientation="vertical"
android:background="#4BFFC107"
tools:context=".Playername">
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical">
    <ImageView
        android:layout_width="match_parent"
        android:layout_height="150dp"
        android:src="@drawable/logo"
        android:layout_marginTop="15dp"
        />
    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:background="@drawable/edittextstyle"
        android:text="Enter Player Name"
        android:textSize="25dp"
        android:layout_marginLeft="15dp"
        android:layout_marginTop="15dp"
        android:layout_marginBottom="20dp"
        android:layout_marginRight="15dp"
        android:textStyle="bold"
        android:textColor="#43093D"/>
    <TextView
```

```
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Player1"
    android:textSize="20dp"
    android:fontFamily="monospace"
    android:textStyle="bold|italic"
    android:textColor="#43093D"/>
<EditText
    android:id="@+id/player1"
    android:layout_marginLeft="15dp"
    android:layout_marginTop="5dp"
    android:layout_marginBottom="20dp"
    android:layout_marginRight="15dp"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:gravity="center"
    android:hint="Player1"
    android:inputType="text"
    android:background="@drawable/edittextstyle"/>
<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Player2"
    android:textSize="20dp"
    android:fontFamily="monospace"
    android:textStyle="bold|italic"
    android:textColor="#43093D"/>
```

```
<EditText
    android:id="@+id/player2"
    android:layout_marginLeft="15dp"
    android:layout_marginTop="5dp"
    android:layout_marginBottom="20dp"
    android:layout_marginRight="15dp"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:gravity="center"
    android:hint="Player2"
    android:inputType="text"
    android:background="@drawable/edittextstyle"/>
<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Player3"
    android:textSize="20dp"
    android:fontFamily="monospace"
    android:textStyle="bold|italic"
    android:textColor="#43093D"/>
<EditText
    android:id="@+id/player3"
    android:layout_marginLeft="15dp"
    android:layout_marginTop="5dp"
    android:layout_marginBottom="20dp"
    android:layout_marginRight="15dp"
    android:layout_width="match_parent"
```

```
    android:layout_height="wrap_content"
    android:gravity="center"
    android:hint="Player3"
    android:inputType="text"
    android:background="@drawable/edittextstyle"/>
<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Player4"
    android:textSize="20dp"
    android:fontFamily="monospace"
    android:textStyle="bold|italic"
    android:textColor="#43093D"/>
<EditText
    android:id="@+id/player4"
    android:layout_marginLeft="15dp"
    android:layout_marginTop="5dp"
    android:layout_marginBottom="10dp"
    android:layout_marginRight="15dp"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:gravity="center"
    android:hint="Player4"
    android:inputType="text"
    android:background="@drawable/edittextstyle"/>
<Button
    android:layout_width="match_parent"
```

```
        android:layout_height="65dp"
        android:text="Start Playing"
        android:id="@+id/startplaying"
        android:textSize="25dp"
        android:fontFamily="monospace"
        android:textStyle="bold"
        android:textColor="#300101"
        android:layout_marginTop="25dp"
        android:layout_marginLeft="25dp"
        android:layout_marginRight="25dp"
        android:gravity="center"
        android:background="@drawable/buttonbackground"/>
    </LinearLayout>
</ScrollView>
```

Java Code for List of Players

```
package com.niru.newdemo;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent; import
android.os.Bundle;
import android.view.MenuItem;
import android.view.View;
```

```
import android.widget.Button;
import android.widget.EditText;

public class Playername extends AppCompatActivity
    { EditText player1,player2,player3,player4;
    Button startgame;



    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_playername);

        player1=findViewById(R.id.player1);
        player2=findViewById(R.id.player2);
        player3=findViewById(R.id.player3);
        player4=findViewById(R.id.player4);
        startgame=findViewById(R.id.startplaying);

        startgame.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String p1name=player1.getText().toString();
                String p2name=player2.getText().toString();
                String p3name=player3.getText().toString();
                String p4name=player4.getText().toString();
                Intent intent=new Intent(Playername.this,OfflinePlay.class);
```

```
            intent.putExtra("player1name",p1name);
            intent.putExtra("player2name",p2name);

            intent.putExtra("player3name",p3name);

            intent.putExtra("player4name",p4name);
            startActivity(intent);

            finish();
        }
    });

}


private void doExit(){
    startActivity( new Intent(getApplicationContext(),ModeOfPlay.class));
    finish();
}

@Override
public void onBackPressed() {
    super.onBackPressed();
    doExit();
}

@Override
```

```
    public boolean onOptionsItemSelected(@NonNull MenuItem item) {
        onBackPressed();
        return true;
    }
}
```

5.4.4. Main Field Or Game Field

Here it is shown that After entering the name of the players , each player will move to the main Field Page where the four chits will be present and the counter to count the round and to identify whose turn is this and also updating the score. The whole chit here will be shuffled randomly and those who are king and minister will disclose themselves and the person who have minister in chit will guess so the turn ribbon will display whose turn is this and by clicking on guess rest to un reveled chits will be enabled to click by guess among two whatever the response will be according to the rule of this game the score of each person will get updated. After Ten round the Winner of the game and the ranking of the Player will be displayed.

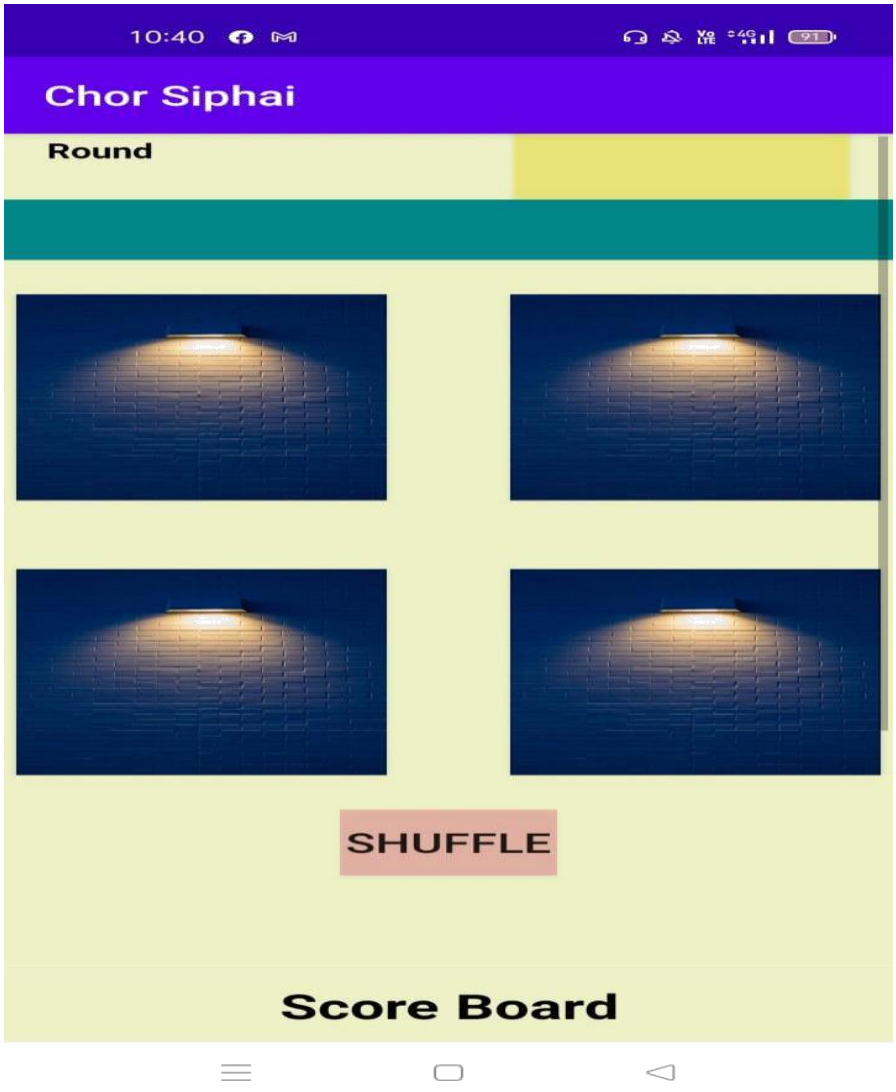


Fig 5.4 Main Field (i)

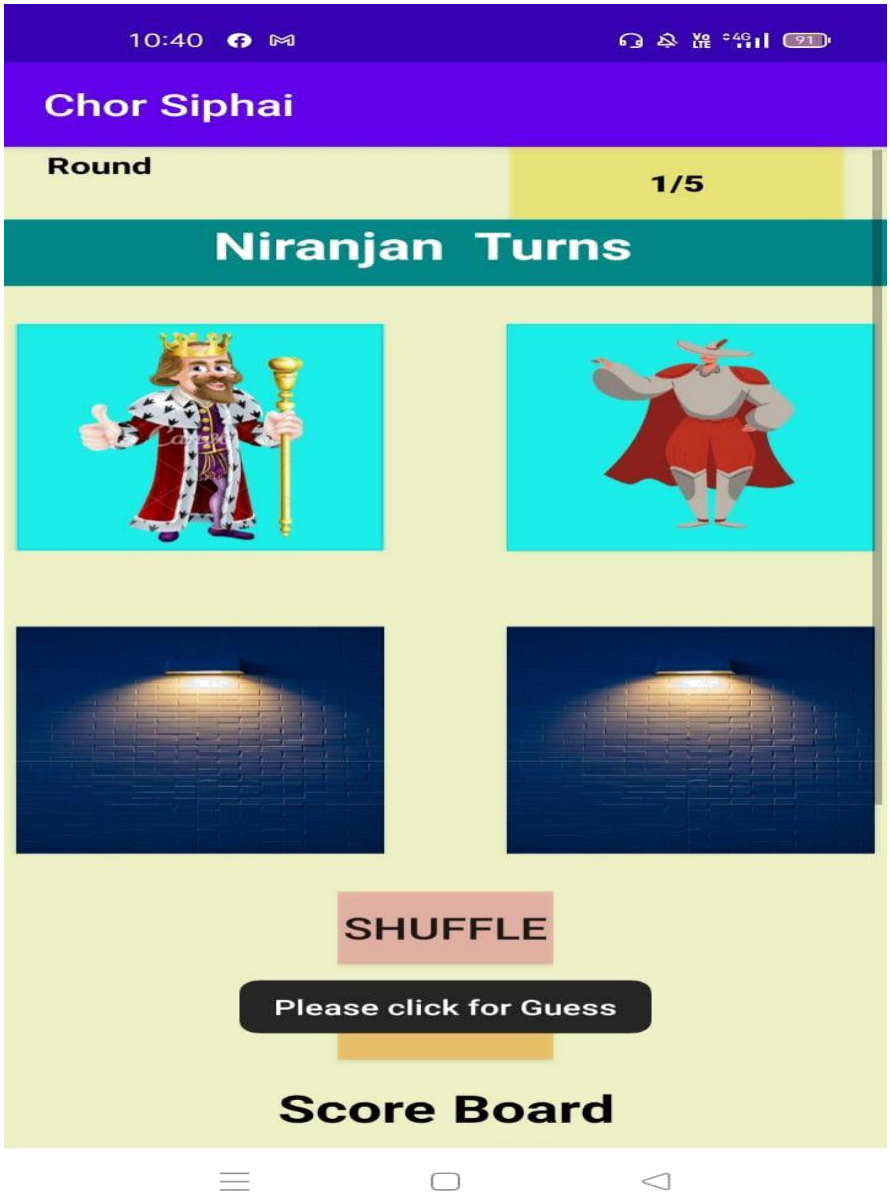


Fig 5.5 Main Field(ii)

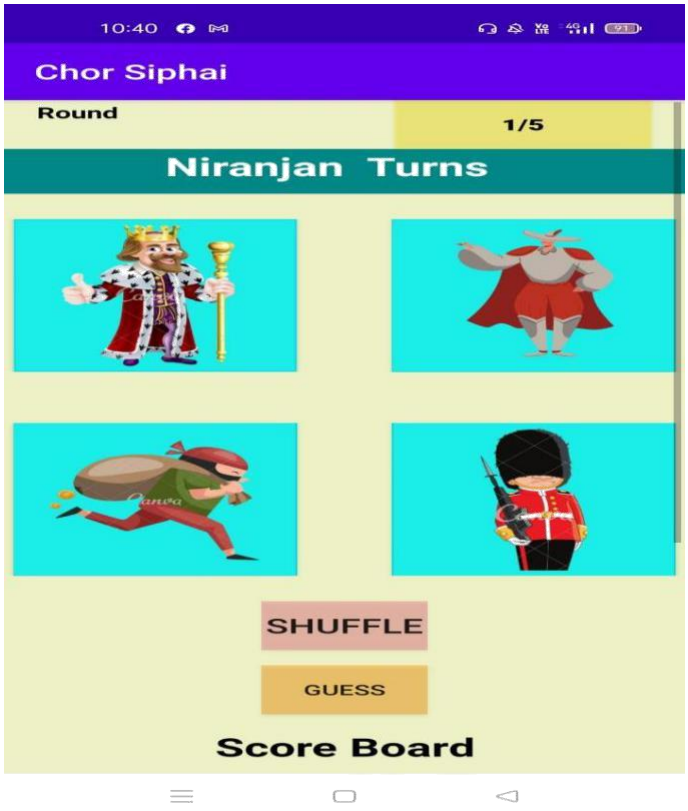


Fig 5.6 Main Field (iii)

Fig 5.7 Main Field(iv)

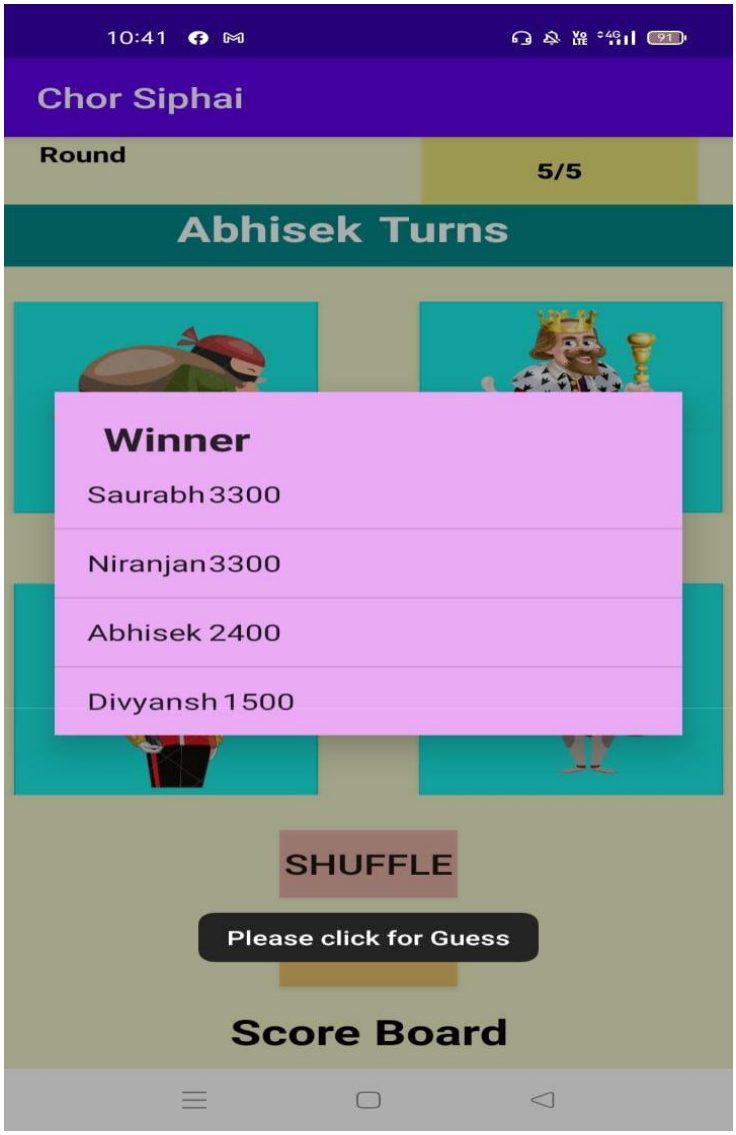


Fig 5.8 Main Field(v)

Designing of Main Field

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:background="#F6EDED"
android:gravity="center"
    tools:context=".MainActivity">

    <Button
        android:layout_width="match_parent"
        android:layout_height="65dp"
        android:text="Create"
        android:id="@+id/create"
        android:textSize="25dp"
        android:fontFamily="monospace"
        android:textStyle="bold"
        android:textColor="#fff"
        android:layout_marginTop="25dp"
        android:layout_marginLeft="25dp"
        android:layout_marginRight="25dp"
        android:gravity="center"
```

```
        android:background="@drawable/screenbg"/>
    <Button
        android:layout_width="match_parent"
        android:layout_height="65dp"
        android:text="Join"
        android:id="@+id/join"
        android:textSize="25dp"
        android:fontFamily="monospace"
        android:textStyle="bold"
        android:textColor="#fff"
        android:layout_marginTop="25dp"
        android:layout_marginLeft="25dp"
        android:layout_marginRight="25dp"
        android:gravity="center"
        android:background="@drawable/screenbg"/>


</LinearLayout>
```

Java Code of Main Field

```
package com.niru.newdemo;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.view.MenuItem;
import android.view.View;
import android.widget.Button;

public class MainActivity extends AppCompatActivity {

    Button create,join;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);



        create=findViewById(R.id.create);
        join=findViewById(R.id.join);
```

```
    create.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Intent intent=new Intent(MainActivity.this,Create.class);
            startActivity(intent);
            finish();

        }
    });

    join.setOnClickListener(new View.OnClickListener()
        { @Override
        public void onClick(View v) {
            Intent intent=new Intent(MainActivity.this, Comingsoon.class);
            startActivity(intent);
            finish();

        }
    });

}
private void doExit(){
    startActivity( new Intent(getApplicationContext(),ModeOfPlay.class));
    finish();
}

@Override
```

```
    public void onBackPressed() {
        super.onBackPressed();
        doExit();
    }

    @Override
    public boolean onOptionsItemSelected(@NonNull MenuItem item)
        { onBackPressed();
        return true;
    }
}
```

5.4.5 Online Mode

When the user select the Online Mode then the Intent Activity will open with the activity having two Button i.e. Join Room and Create Room. So in online mode we need to bind up all my four college to whom we want to play within one place so that they never get the useless activity to play.so We need to work within a room . So we use to perform

Functionality of :

Create room: To invite exactly the four person to whom you wants to play with. Here the user have to invite the person by sending him the request to join.

Join Room: To get connect in a single room other than the creator of room the rest 3 player have to join and accept the request the creator have sent to play game.

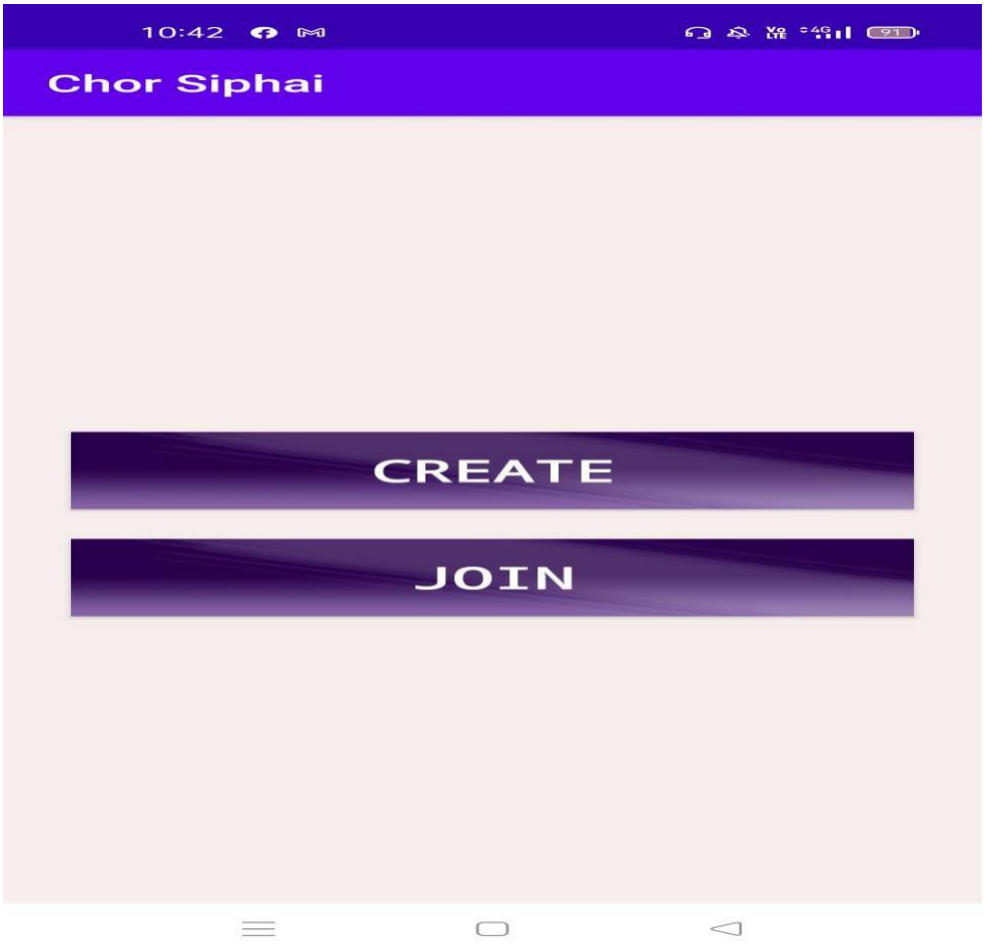


Fig 5.9 Online Function

5.4.6. Create Room

In the Online mode it’s a great need to make a room to so that rest of the player can available in the room to start the game. So it has One Login field where user have to login themselves with the email and the field will be registered by there email authentication and it will also have the field in which rest 3 players email will be entered by the creator of the room by clicking on invite button respective to the email the Request will be send to the person to join the room

Fig 5.10 Create Room

Designing of Create Room

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#EDF4A2" tools:context=".Create">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical">



        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="horizontal">

            <EditText
                android:layout_width="0dp"
                android:layout_height="wrap_content"
                android:layout_weight="2"
                android:hint="Enter user ID"
```

```
        android:id="@+id/edtInviteAccept1"
        android:textSize="15sp"
        android:layout_margin="5dp"/>



    <Button
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:gravity="center"
        android:id="@+id/btnInvite1"
        android:text="Invite"
        android:onClick="btnInvite1"
        android:background="@color/teal_700"
        android:textColor="@color/white"

        android:textSize="15sp"
        android:padding="3dp"
        android:layout_margin="10dp"/>



</LinearLayout>
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
```

```
android:orientation="horizontal">

<EditText
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_weight="2"
    android:hint="Enter user ID"
    android:id="@+id/edtInviteAccept2"
    android:textSize="15sp"
    android:layout_margin="5dp"/>



<Button
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:gravity="center"
    android:id="@+id/btnInvite2"
    android:text="Invite"
    android:onClick="btnInvite2"
    android:background="@color/teal_700"
    android:textColor="@color/white"

    android:textSize="15sp"
    android:padding="3dp"
    android:layout_margin="10dp"/>
```

```
</LinearLayout>
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal">

    <EditText
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight="2"
        android:hint="Enter user ID"
        android:id="@+id/edtInviteAccept3"
        android:textSize="15sp"
        android:layout_margin="5dp"/>



    <Button
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:gravity="center"
        android:id="@+id/btnInvite3"
        android:text="Invite"
        android:background="@color/teal_700"
```

```
        android:textColor="@color/white"

        android:onClick="btnInvite3"
        android:textSize="15sp"
        android:padding="3dp"
        android:layout_margin="10dp"/>

           </LinearLayout>



<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal">

    <EditText
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight="2"
        android:hint="Enter New ID"
        android:textSize="15sp"
        android:id="@+id/edtLogin"
        android:inputType="textEmailAddress"
        android:layout_margin="5dp"/>
```

```
        <Button
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:text="Login"
            android:background="@color/teal_700"
            android:textColor="@color/white"

            android:onClick="btnLogin"
            android:id="@+id/btnLogin"
            android:textSize="15sp"
            android:padding="3dp"
            android:layout_margin="10dp"
            android:gravity="center"/>


    </LinearLayout>
</LinearLayout>

</LinearLayout>
```

Java Code of Create Room

```
package com.niru.newdemo;

import androidx.annotation.NonNull;
import androidx.annotation.RequiresApi;
import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.graphics.Color;
import android.os.Build;
import android.os.Bundle;
import android.util.Log;
import android.view.MenuItem;
import android.view.View;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.analytics.FirebaseAnalytics;
import com.google.firebase.auth.AuthResult;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;
import com.google.firebase.database.DataSnapshot;
```

```
import com.google.firebase.database.DatabaseError; import
com.google.firebase.database.DatabaseReference; import
com.google.firebase.database.FirebaseDatabase; import
com.google.firebase.database.ValueEventListener;

import java.util.ArrayList;
import java.util.HashMap;
import java.util.Objects;

public class Create extends AppCompatActivity {



    EditText edtInviteAccept1,
edtInviteAccept2,edtInviteAccept3,edtLogin;
    Button btnSelected, btnReset, btnInvite1, btnInvite2, btnInvite3,
btnAccept1,btnAccept2,btnAccept3, btnLogin;
    //TextView txtName1, txtName2, txtScore1, txtScore2, txtSymbol;


    String userID;


    ArrayAdapter<String> adapter;
    ArrayList<String> arrayList;
    // ArrayList<String> listItems=new ArrayList<String>();
    //ArrayAdapter<String> adapter;


    Edit names = new Edit();
    String myEmail;
    FirebaseDatabase database = FirebaseDatabase.getInstance();
```

```
DatabaseReference myRef = database.getReference();
int activePlayer = 1; // 1 is for 1st Player , 2 is for 2nd Player
ArrayList<Integer> player1 = new ArrayList<>(); // Player 1 Data
ArrayList<Integer> player2 = new ArrayList<>(); // Player 2 Data
String playerSession = "";
//String mySample = "X";

private FirebaseAnalytics mFirebaseAnalytics;
private FirebaseAuth mAuth;
private FirebaseAuth.AuthStateListener mAuthListener;

@RequiresApi(api = Build.VERSION_CODES.LOLLIPOP)

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_create);

    Objects.requireNonNull(getSupportActionBar()).setElevation(0);

    // list = (ListView) findViewById(R.id.list);
    arrayList = new ArrayList<String>();
    edtLogin = findViewById(R.id.edtLogin);

    // txtSymbol = findViewById(R.id.txtSymbol);

    btnLogin = findViewById(R.id.btnLogin);
```

```
// edtLogin = findViewById(R.id.edtLogin);
edtInviteAccept1 = findViewById(R.id.edtInviteAccept1);
edtInviteAccept2 = findViewById(R.id.edtInviteAccept2);
edtInviteAccept3 = findViewById(R.id.edtInviteAccept3);

btnInvite1=findViewById(R.id.btnInvite1);
btnInvite2=findViewById(R.id.btnInvite2);
btnInvite3=findViewById(R.id.btnInvite3);
// btnLogin=findViewById(R.id.btnLogin);




mFirebaseAnalytics = FirebaseAnalytics.getInstance(this);
mAuth = FirebaseAuth.getInstance();



mAuthListener = new FirebaseAuth.AuthStateListener() {
    @Override
    public void onAuthStateChanged(@NonNull
FirebaseAuth firebaseAuth) {
        FirebaseUser user = firebaseAuth.getCurrentUser();


        if (user != null) {
            userID = user.getUid();
            myEmail = user.getEmail();
            btnLogin.setEnabled(false);
            edtLogin.setText(myEmail);
```

```
                myRef.child("New
Users").child(ExactName(myEmail)).child("Request").setValue(userID);
                GettingRequest();

            } else {
                Log.d("TAG ", "Null user!");
            }
        }
    };



}

private String ExactName(String string) {

    String[] exactName = string.split("@");

    return exactName[0];

}

private void NewUser(String email, String password) {

    mAuth.createUserWithEmailAndPassword(email, password)
            .addOnCompleteListener(this, new
OnCompleteListener<AuthResult>() {
                @Override
```

```
                public void onComplete(@NonNull Task<AuthResult> task)
{
                    if (task.isSuccessful()) {
                        // Sign in success, update UI with the signed-in user's
information
                        Log.d("Successful Login",
"createUserWithEmail:success");
                        FirebaseUser user = mAuth.getCurrentUser();


                    } else {
                        // If sign in fails, display a message to the user.
                        Log.w("Login Failed", "createUserWithEmail:failure",
task.getException());
                        Toast.makeText(getApplicationContext(),
"Authentication failed.",
                                Toast.LENGTH_SHORT).show();


                    }
                }
            });


    }



    @Override
    public void onStart() {
        super.onStart();
        // Check if user is signed in (non-null) and update UI accordingly.
```

```
        //FirebaseUser currentUser = mAuth.getCurrentUser();
        mAuth.addAuthStateListener(mAuthListener);
    }



    public void btnLogin(View view) {

        NewUser(edtLogin.getText().toString(), "1234567");

    }
    //////////////

    public void btnInvite1(View view) {

        myRef.child("New
Users").child(ExactName(edtInviteAccept1.getText().toString())).child("
Request").push().setValue(myEmail);

        StartGame(ExactName(edtInviteAccept1.getText().toString()) +
":" + ExactName(myEmail));

       /* myRef.child("New
Users").child(ExactName(edtInviteAccept2.getText().toString())).child("
Request").push().setValue(myEmail);

        StartGame(ExactName(edtInviteAccept2.getText().toString()) +
":" + ExactName(myEmail));
```

```
        myRef.child("New
Users").child(ExactName(edtInviteAccept3.getText().toString())).child("
Request").push().setValue(myEmail);


        StartGame(ExactName(edtInviteAccept3.getText().toString()) + ":"
+ ExactName(myEmail));


       */ // mySample = "X";


        //btnAccept1.setEnabled(false);
        // txtSymbol.setText("You have 'O'");



    }
//////////////////////////////

    public void btnInvite2(View view) {


        myRef.child("New
Users").child(ExactName(edtInviteAccept2.getText().toString())).child("
Request").push().setValue(myEmail);


        StartGame(ExactName(edtInviteAccept2.getText().toString()) +
":" + ExactName(myEmail));


       /* myRef.child("New
Users").child(ExactName(edtInviteAccept2.getText().toString())).child("
Request").push().setValue(myEmail);
```

```
        StartGame(ExactName(edtInviteAccept2.getText().toString()) +
":" + ExactName(myEmail));


        myRef.child("New
Users").child(ExactName(edtInviteAccept3.getText().toString())).child("
Request").push().setValue(myEmail);


        StartGame(ExactName(edtInviteAccept3.getText().toString()) +
":" + ExactName(myEmail));


      */ //   mySample = "X";


        //btnAccept2.setEnabled(false);
        // txtSymbol.setText("You have 'O'");



    }


    ////////////////////////////
    public void btnInvite3(View view) {


        myRef.child("New
Users").child(ExactName(edtInviteAccept3.getText().toString())).child("
Request").push().setValue(myEmail);


        StartGame(ExactName(edtInviteAccept3.getText().toString()) +
":" + ExactName(myEmail));
```

```
      /* myRef.child("New
Users").child(ExactName(edtInviteAccept2.getText().toString())).child("
Request").push().setValue(myEmail);


       StartGame(ExactName(edtInviteAccept2.getText().toString()) +
":" + ExactName(myEmail));


       myRef.child("New
Users").child(ExactName(edtInviteAccept3.getText().toString())).child("
Request").push().setValue(myEmail);


       StartGame(ExactName(edtInviteAccept3.getText().toString()) + ":"
+ ExactName(myEmail));


      */ // mySample = "X";


       //btnAccept3.setEnabled(false);
       // txtSymbol.setText("You have 'O'");



   }
   ////////////////////////
   private void StartGame(String playerGameID) {


       // playerSession = playerGameID;
       myRef.child("Playing").child(playerGameID).removeValue();
```

```
myRef.child("Playing").child(playerGameID).addValueEventListener(ne
w ValueEventListener() {
        @Override
        public void onDataChange(@NonNull DataSnapshot
dataSnapshot) {


            try {
                player1.clear();
                player2.clear();
             //   player3.clear();
            //    player4.clear();
                //  activePlayer = 2;
                HashMap<String, Object> hashMap =
(HashMap<String, Object>) dataSnapshot.getValue();


                if (hashMap != null) {


                    String value;


                    for (String key : hashMap.keySet()) {
                        value = (String) hashMap.get(key);



                        String[] splitID = key.split(":");
                        //                autoPlay(Integer.parseInt(splitID[1]));


                    }
```

```
                }

            } catch (Exception e) {
                Log.e("TAG", "Error " + e);
            }


        }

        @Override
        public void onCancelled(@NonNull DatabaseError
databaseError) {

        }
    });

}



  private void GettingRequest() {


    myRef.child("New
Users").child(ExactName(myEmail)).child("Request").addValueEventLis
tener(new ValueEventListener() {
        @Override
        public void onDataChange(@NonNull DataSnapshot
dataSnapshot) {
```

```
try {
    HashMap<String, Object> hashMap =
(HashMap<String, Object>) dataSnapshot.getValue();

    if (hashMap != null) {

        String value;

        for (String key : hashMap.keySet()) {
            value = (String) hashMap.get(key);
            edtInviteAccept1.setText(value);
            edtInviteAccept2.setText(value);

            edtInviteAccept3.setText(value);

            changeColor(edtInviteAccept1); // changing the
background color of edit text when someone sents invitation
            changeColor(edtInviteAccept2); // changing the
background color of edit text when someone sents invitation
            changeColor(edtInviteAccept3); // changing the
background color of edit text when someone sents invitation

            myRef.child("New
Users").child(ExactName(myEmail)).child("Request").setValue(userID);
            break;
        }

    }
```

```
            } catch (Exception e) {
                Log.e("TAG", "Error " + e);
            }



        }


        @Override
        public void onCancelled(@NonNull DatabaseError
databaseError) {


        }
    });


}


private void changeColor(EditText editText) {


    editText.setBackgroundColor(Color.YELLOW);


}


private void doExit(){
    startActivity( new
Intent(getApplicationContext(),MainActivity.class));
    finish();
}
```

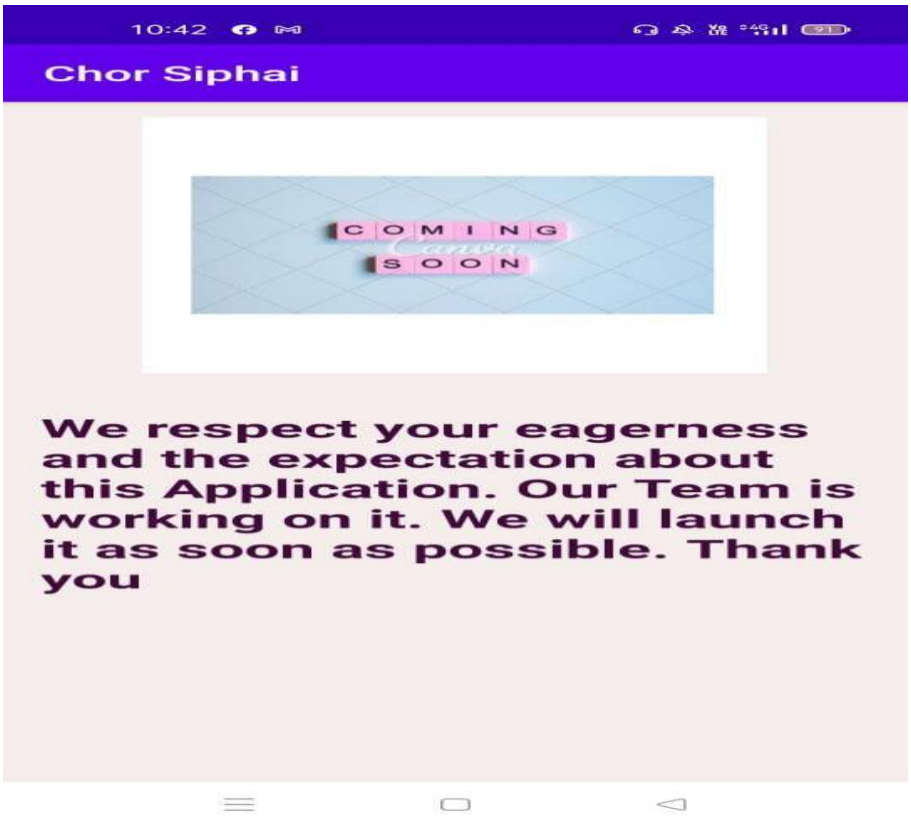
```
@Override
public void onBackPressed() {
    super.onBackPressed();
    doExit();
}

@Override
public boolean onOptionsItemSelected(@NonNull MenuItem item)
    { onBackPressed();
    return true;
}

}
```

5.4.7. Join Room

After the creation of the room To get connected in a single room other than the creator of room the rest 3 player have to join and accept the request the creator have sent to play game. There will be the Accept request field where the creator email id will be highlighted by green color it means that this person have send you request to play the game. By clicking on Accept button the person will be added in room and the database will get updated.(Work in Progress)



Designing of Join Room(Coming Soon)

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#EDF4A2" tools:context=".Join">
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical">


        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="horizontal">

            <EditText
                android:layout_width="0dp"
                android:layout_height="wrap_content"
                android:layout_weight="2"
                android:hint="Enter user ID"
```

```
        android:id="@+id/edtInviteAccept1"
        android:textSize="15sp"
        android:layout_margin="5dp"/>



    <Button
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:gravity="center"
        android:onClick="btnAccept"
        android:text="Accept"
        android:textColor="@color/white"
        android:padding="3dp"
        android:background="@color/teal_700"
        android:id="@+id/btnAccept"
        android:layout_margin="10dp"
        android:textSize="15sp" />



</LinearLayout>



<LinearLayout
```

```
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:orientation="horizontal">

<EditText
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_weight="2"
    android:hint="Enter New ID"
    android:textSize="15sp"
    android:id="@+id/edtLogin"
    android:inputType="textEmailAddress"
    android:layout_margin="5dp"/>

<Button
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:text="Login"
    android:onClick="btnLogin"
    android:background="@color/teal_700"
    android:textColor="@color/white"

    android:id="@+id/btnLogin"
    android:textSize="15sp"
    android:padding="3dp"
    android:layout_margin="10dp"
```

```
            android:gravity="center"/>



        </LinearLayout>
    </LinearLayout>

</LinearLayout>
```

5.5 Flow of Work :

Work Flow Diagram show the flow of work in any application. Chor Sipahi Gaming Applications work flow in correct sequence. All the activity are intent in such a way that the smoothness in the work flow can be felt easily and is user friendly.

Work Flow of Chor Sipahi Gaming Application:

1. Start your application
2. Soon the application launch there will be a splash screen for 3 sec .
3. Now the user have to select in which mode they wants to play i.e. Online Mode Or Offline Mode.
4. **(a) If Offline mode is selected:**

   i. The user have to enter the name of four players in the player list.

   ii. Then After clicking on Start Playing button the user will be directed to the Main field where one can enjoy playing game.

   iii. The game is are of 10 round now the player have to shuffle the chits.

   iv. The King and the minister will be disclosed and the one who is minister have the turn to find the thief to do so he/she have to click on the guess button and not he is free to guess any of the two undisclosed chits.

   v. If the answer is write then the respective score will be allotted in the leader board below. If not then the marks will be deducted from the one who is minister and score will be added to one who is thief.

   vi. Similarly the game will continue till the 10 rounds and the winner of the game will be disclosed and the ranking of the players will be disclose with there score as well.

5 . If Online Mode is selected:

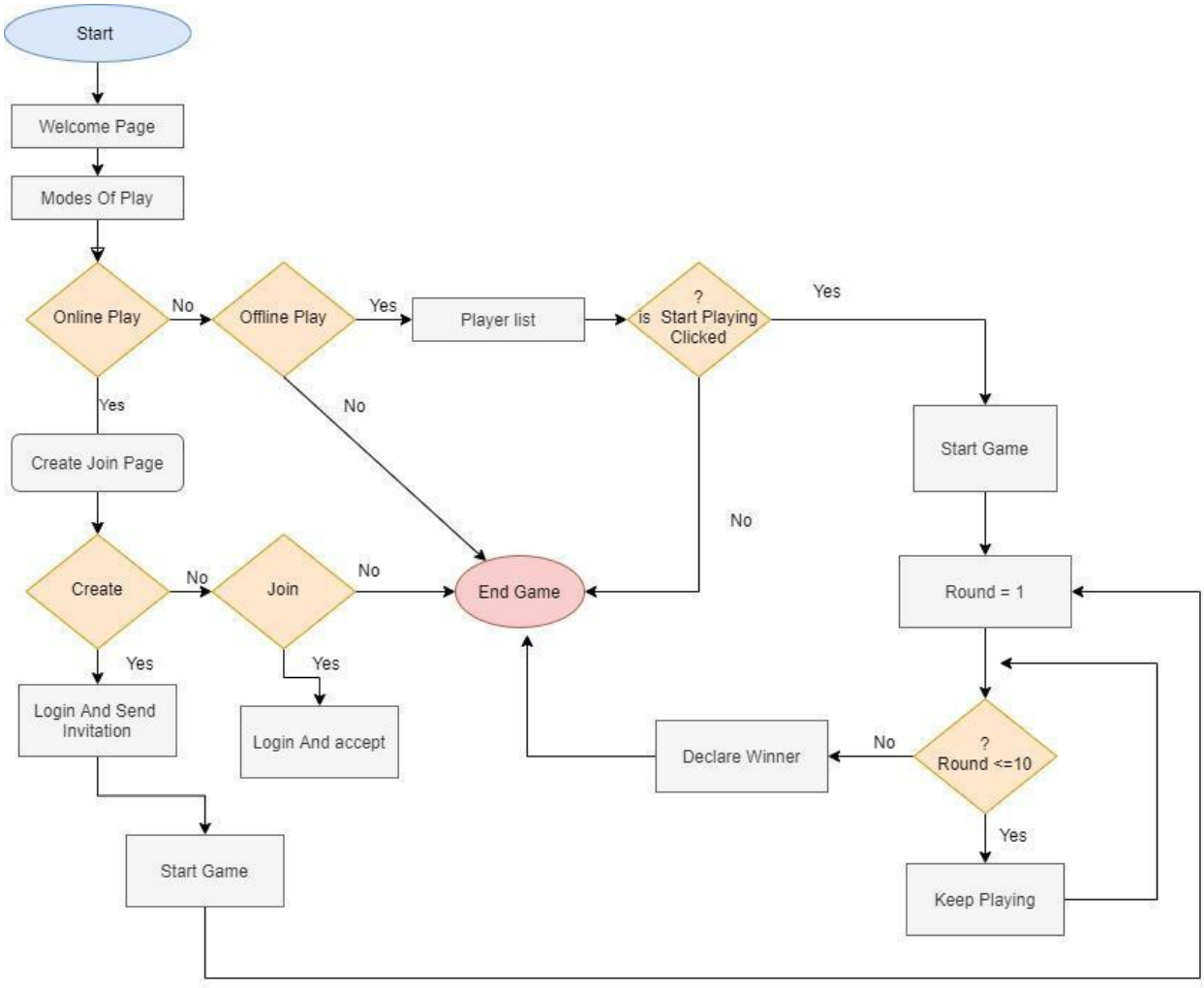
When the online mode is selected the user will be taken to the next intent activity where he either have to create a room or join a room.

i. If Create Room is selected

When the user will click on the create room button the he will be taken to the next intent Activity where he first have to login and then enter the email of the rest 3 players and respectively send invitation for join the room that creator have created for playing game.

ii. If Join Room is selected:

When creator of the room has send the request to the rest 3 players the rest three players after login only once can click on the accept button then the creator of the room on its create window enable the Start Playing Game and the game as explained above will start.



5.6 Objective

The main objective of the project on Chor Sipahi Gaming Application is to bind the digitally growing generation with the traditional game. It help to grow the younger and fast moving generation toward the digitalization. It is full of entertaining game and the application is developed in such a way that player will enjoy playing the game.

The main objective of this project are as follows:

- It enhance the cognitive knowledge of the people.
- It will act as a tool to make new friend and stay connected.
- It will help children to know the role of the different character used in the game.
- It will promote the primitive culture.
- It help in good decision making.
- It will help in enhancing the analytical knowledge of the people.
- It will help to sharpen the child's Memory and will enhance the concentration Power.

5.7 PURPOSE

The old system was suffering from a series of drawbacks. Since whole of the system was to be maintained with hands, the process of keeping, maintaining and retrieving the information was very tedious and lengthy. There would always be unnecessary consumption of time in making chits and updating score in the sheet . One more problem was that it was very difficult to check whether any of the friend helped him to guess. The reason behind it is to make the safe and memorable and fare game. For this reason, we have provided present system with fully featured and automated actually existing system is quite unsafe and helpless. The Proposed system will help in preventing the wastage of paper and protecting the tree as paper is made from the tree.

5.7.1 SCOPE

It may help collecting perfect details. In a very short time, collection will be obvious, simple and sensible. The main aim of this project are:

- To automate the game.
- It satisfies the user requirements.
- Easy to understand.
- Have a good user interface.
- Be expandable.
- Portable
- Free of Cost.

Chapter 6

Future Scope and Conclusion

6.1 Future Scope

We are working on the Online Modes and we will soon available with the Online Modes where the create room and join room will work perfectly from starting to declaration of the winner.

We will Launch the application by developing all the module in perfect manner on the Google Play store under Gaming Application Store.

We will Try to add more feature like checking which friend is online and available to play the game so that we can efficiently send invitation to them which highest probability of Joining the Game.

6.2 CONCLUSION

In this project we will provide application according to which the user will record any action within the Online Modes Play and save it in database. Chor Sipahi Gaming application will help user to play either Online or Offline Mode with same joy. So this proposed system will help to stay connect with the tradition as they are the jewels they have there own value. Its main motive is fair, safe and memorable play.

6.3 Declaration

I hereby declare that The Project entitled is an outcome of my own efforts under the guidance of **prof. Dr. Vipin Kumar**. The project is submitted to the department of MCA. For the partial fulfilment of Master of Computer Application 2019-22.

I also declare that project report is not submitted in any of the university previously.

Date: 14-August-2021

Place: Ghaziabad

BIBLIOGRAPHY :

- **GOOGLE**
- **AbhiAndroid**

Project Github Link :

https://github.com/niranjandarshan/Chor-sipahi-Gaming-Application