

**Mobile System Engineering Laboratory**

**A mini project report**

**on**

**SET 4 – A Multiplayer Android Game**

**Submitted by**

**1PI11CS137, Rakshitha K Bhat**

**1PI11CS138, Rashmi Raghunandan**

**1PI11CS151, Sanjana S**

Department of Computer Science & Engineering

**PES INSTITUTE OF TECHNOLOGY**

*(An Autonomous Institute under VTU Belgaum)*

**100 Feet Road, BSK III Stage, Hosakerehalli, Bengaluru - 560 085**



**Mobile System Engineering Laboratory**

**A mini project report**

**on**

**SET 4 – A Multiplayer Android Game**

**Submitted by**

**1PI11CS137, Rakshitha K Bhat**

**1PI11CS138, Rashmi Raghunandan**

**1PI11CS151, Sanjana S**

Department of Computer Science & Engineering

**PES INSTITUTE OF TECHNOLOGY**

*(An Autonomous Institute under VTU Belgaum)*

**100 Feet Road, BSK III Stage, Hosakerehalli, Bengaluru - 560 085**



**PES INSTITUTE OF TECHNOLOGY**

*(An Autonomous Institute under VTU Belgaum)*

**100 Feet Road, BSK III Stage, Hosakerehalli, Bengaluru - 560 085**

Department of Computer Science & Engineering

CERTIFICATE

This is to certify that the mini project entitled “**SET 4 – A Multiplayer Android Game**

” has been carried out by

**1PI11CS137, Rakshitha K Bhat**

**1PI11CS138, Rashmi Raghunandan**

**1PI11CS151, Sanjana S**

in the partial fulfillment of fifth semester Mobile Systems Engineering Laboratory[11CS4]

Signature of the Signature of the

**Lab Incharge Head - CSE**

**Name & Signature of the Examiners:**

Examiner 1:

Examiner 2:

**ACKNOWLEDGEMENTS**

The satisfaction that accompanies the successful completion of any task would be incomplete without the mention of the people who made it possible, and whose constant guidance and encouragement helped us in completing the project successfully. We consider it a privilege to express gratitude and respect to all those who guided us throughout the course of the completion of the project.

We would like to express our heartfelt thanks to **Prof. M. R. Doreswamy**, PESIT founder, **Prof. D. Jawahar,** CEO and **Dr. K. N. Balasubramanya Murthy**, Principal, for providing us with a congenial environment for carrying out the project.

We express our gratitude to **Prof.Nitin V. Pujari**, Head of the Department, Computer Science, PESIT, whose guidance and support has been invaluable and for including the project as part of the course.

We extend our sincere thanks to our mentor **Prof. Nagegowda K S** for his constant guidance, encouragement, support and invaluable advice without which this project would not have been completed.

Last, but not the least, we would like to thank our friends whose invaluable feedback helped us to improve the software by leaps and bounces, and our parents for their unending encouragement and support.

**ABSTRACT**

**TITLE**: Set4 – Multiplayer Android Game

**AIM**: To implement a four player card game

**DESCRIPTION OF THE GAME:**

A “set” of cards, for the purpose of this game, is defined as four cards having the same value.

This version of the game of Set involves four players. There are four sets of cards (such as four Aces, four Kings, etc.). Hence there are 16 cards that form the deck being used in this game. The cards are shuffled and four cards are distributed randomly to each player i.e. every player holds a hand of four cards. Each player passes on one card that he does not want from his hand to the player on his left and accepts one card from the player on his right. Hence, at any time, each player must have exactly four cards. The passing and accepting of cards continues until one player's hand consists of four cards of the same value (for example, the player's hand consists of only four Aces, four Kings.).

**IMPLEMENTATION**:

The game is to be implemented using GUI and data structures in Java. The game can be played as a multiplayer game across phones connected over ADHOC through a server hosted by us. Internet Connection is required to play this game.

The focus of the project is to reduce the battery consumption and to make the game consistent across different screen resolutions and android versions.

**SOFTWARE REQUIREMENTS:**

Min SDK: 2.2

Target SDK: 4.4

**CONTENTS**

**Acknowledgements………………………………………………………….. 1**

**Abstract……………………………………………………………………… 2**

**Introduction…………………………………………………………………. 4**

**Design Details…………………………………………………………………5**

**Implementation………………………………………………………………7**

**Test Cases and Results...……………………………………………………..12**

**Conclusion…………………………………………………………………… 14**

**Bibliography…………………………………………………………………..15**

**INTRODUCTION**

Set4 is an android game that can be played in both single layer as well as muti player mode. The multiplayer games of android use existing engines like AndEngine and ApWrap Servers. We chose to create the server rather than using existing hosting sites.

Android games are very popular and our game,Set4 has potential to be popular as it involves challenging your friends and is also a popular offline game.

**DESIGN DETAILS**

The Game can be played in both single player and multi player mode.

In single Player mode, the robot (i.e our server) plays as if it were a real person, serving the right cards and trying to make a set of 4.

In multiplayer mode, we have a server that maintains the game across users. It can create a room as well as host existing rooms so new users can join.

When all four players have joined the room, the game begins. Each player can click on a card he wishes to send/give away. This card is then passed through the server to the next player. The player who passed the card cannot perform an action during this stage.

When one of the players has four of the same card, he sends a message to the server indicating that he has won.

A player sends a header message to the server continuously to check whether it his turn. If it is, he sends a GET request to server to obtain the card passed to him.

The player also polls the server continuously to see if there was any winner. If there was, then each of the applications re draws their screen to show that game was finished.

The game is battery optimised as it requires only internet connection and no other feature.

The game’s APK is 2.04 Mb in size. This makes it easy to download and use.

**IMPLEMENTATION**

Client Side:

The activity screens were designed in XML using Relative Layout.

We had five activity screens:

1. Game Home Page
2. Join/Create Room Page
3. Waiting for players to join
4. Game Page
5. Winners Page

We used buttons and Image Buttons to navigate between Activities.

The multiplayer game was in constant contact with the server. Android can connect to local server through the public IP of the computer hosting the server.

Connection to a server is a Asynchronous task. It runs on a worker thread. The UI thread must not be contacted at this stage.

Thread t=**new** Thread()

{

**public** **void** run()

{

HttpClient myClient=**new** DefaultHttpClient();

**Try**

{

String data=URLEncoder.*encode*(gameId,"UTF-8");

HttpGet get=**new** HttpGet("http://192.168.1.6/mse/createroom.php?RoomId="+data);

HttpResponse response;

response = myClient.execute(get);

res=EntityUtils.*toString*(response.getEntity());

}

**catch** (Exception e)

{

e.printStackTrace();

}

}};

The server pages contacted are:

1. CreateRoom.php- This is used to initialise the room, assign cards to players when they join the room.
2. playGame.php- This helps the user pas cards between players
3. Winner and Turn.php are used to monitor a global variable to find the winner and the player whose turn it is.

We send a header message from client to server periodically using poling technique. This requires us to periodically call a method. We use Runnable of Java to achieve this.

Runnable mStatusChecker = new Runnable() {

@Override

public void run() {

polling(); //this function can change value of mInterval.

mHandler.postDelayed(mStatusChecker, mInterval);

}

};

We have modified the manifest file and added this line:

<uses-permission android:name="android.permission.INTERNET"></uses-permission>

**TEST CASES AND RESULTS**

The game was played in multiple rooms by multiple players and was found to be successful.

\*PLEASE INSERT SCREENSHOTS\*

1. First Activity
2. Game Page
3. Game in progress(card missing)
4. Results
5. Select Room Page

**CONCLUSION**

The Register Allocation was implemented successfully. The use of Code Optimization and Code generation is evident in the application.

**BIBLIOGRAPHY**

1. GNU Makefile

2. ANSI C by Ritchie and Kernighan

3. MAN pages