

A Project Report

on

**A GAME OF ANAGRAMS**

*Submitted for the partial fulfillment of the requirement*

*for the award of the Degree of*

**BACHELOR OF TECHNOLOGY**  
in  
**COMPUTER SCIENCE & ENGINEERING**

by

**Priyanshi Chamoli(Roll No-170102239)**  
**Vaibhav Gaur(Roll No-170102218)**

Under the Guidance of

**Ms. SANDHYA ADHIKARI**  
**Assistant Professor**  
**Department of Computer Science & Engineering**



**DIT UNIVERSITY, DEHRADUN, INDIA**  
**2020-21**



## DECLARATION

This is to certify that the Project entitled “**A GAME OF ANAGRAMS**” in partial fulfillment of the requirement for the award of the **Degree of Bachelor of technology in Computer Science & Engineering** ,submitted to **DIT University, Dehradun, Uttarakhand , India**, is an authentic record of bonafide work carried out by us, under the guidance of **Ms. Sandhya Adhikari**.

The matter embodied in this Project has not been submitted for the award of any other degree or diploma to any University/ Institution.

**Students Name & Signature:**

**Priyanshi Chamoli  
Vaibhav Gaur**

**Supervisor Name , Designation  
& Signature:**

**Ms. Sandhya Adhikari  
Assistant Professor**

**<Name>  
Head of Department  
Signature**

***Date:***

***Place: Dehradun***



## **CERTIFICATE**

---

This is to certify that the Project entitled “**A GAME OF ANAGRAMS**” in partial fulfillment of the requirement for the award of the **Degree of Bachelor of Technology** in **Computer Science & Engineering** , submitted to **DIT University, Dehradun, Uttarakhand, India**, is an authentic record of bonafide research work carried out by **Ms. Priyanshi Chamoli**(170102239) and **Mr. Vaibhav Gaur**(170102218) under my supervision/guidance.

**Signature and Name of Supervisor(s)/  
Guide**

**Ms. Sandhya Adhikari  
(Assistant Professor)**

***Date:***

***Place: Dehradun***

## ACKNOWLEDGMENT

I wish to acknowledge my deep gratitude to **Ms. Sandhya Adhikari**, Assistant Professor at Computer Science & Engineering Department, for her cooperation and guidance.

There is always a sense of gratitude which one express towards others for their help and supervision in achieving the goals. This formal piece of acknowledgement is an attempt to express the feeling of gratitude towards people who helpful me in successfully completing of my training. She was always there with her competent guidance and valuable suggestion throughout the pursuance of my first phase of project. I would also like to place of appreciation to all the respondents and group members whose responses and coordination were of utmost importance for the project.

Above all no words can express my feelings to my parents, friends all those persons who supported me during my project. I am also thankful to all the respondents whose cooperation & support has helped me a lot in collecting necessary information.

## **ABSTRACT**

In this project we are going to create a game in which the user is presented with a meaningful word and the user has to rearrange the letter of that word into another meaningful word in a fixed amount of time. The new word should have exact letters as in old one after the rearrangement.

The user can enter words in input field and submit it by clicking “GO” button. To clear your input and reset the letters without submitting the word, click the “Clear” button. If you want to give up and end the game, click the “Show Answers” button. Your game will also end if you run out of time.

If your time limit gets over or if user submit a wrong answer then a dialog box with a suitable pop up message will appear on the screen. In both the above cases correct answer will also get displayed on the screen along with scores.

About scoring and timing:

You score points for making answers. The number of points you gain per answer is determined automatically based on a number of different factors.

You start the game with a given amount of time, in seconds. If you run out of time, your game is over and you will be shown the answers you didn't get. You can also give up by pressing the “Show Answers” button to see the ones you missed.

When the game is over, you will gain a bonus based on your remaining time, if any. The bonus will be greatly increased if you have found all of the words.

We are going to learn a lot through this project.

## TABLE OFCONTENTS

Title	Page No.
<b>CHAPTER 1      INTRODUCTION</b>	
1.1    Purpose .....	2
1.2    Objective.....	2
1.3    Motivation.....	2
1.4    Definition and Overview.....	3
<b>CHAPTER 2      OVERALL DESCRIPTION</b>	
2.1    Project Perspective .....	4
2.2    Project Functions.....	4
2.3    E-R, DFD Diagrams, Class Diagram, Use Case .	5
2.4    Constraints and Assumptions.....	7
<b>CHAPTER3      EXTERNAL INTERFACE REQUIREMENTS</b>	
3.1    User Interfaces.....	8
3.2    Hardware Interfaces.....	8
3.3    Software Interfaces.....	9
3.4    Communications Interfaces.....	9
<b>CHAPTER4      SYSTEM FEATURES</b>	
4.1    System Feature.....	11
4.2    System Feature 2 (and so on).....	11
<b>CHAPTER 5      OTHER NONFUNCTIONAL REQUIREMENTS</b>	
5.1    Performance Requirements.....	13
5.2    Safety Requirements.....	13
5.3    Security Requirements.....	13
5.4    Software Quality Attributes.....	13

## **CHAPTER 6            CONCLUSIONANDFUTUREWORK**

6.1	Conclusions .....	15
6.2	Scope for Future Work .....	15

<b>REFERENCES</b> .....	16
-------------------------	----

APPENDIX.....	
---------------	--

PLAGIARISM REPORT

# **CHAPTER 1**

## **INTRODUCTION**

### **1.1 Purpose**

The purpose of this project is to make a word game which can be played by people of all age groups. This game will help in enhancing the IQ of all type of people. Word games are beneficial for our brain as it make the part of brain more active by increasing thinking power of brain. Player will be able to learn more and more words and with this it will improve the vocabulary of the individual. All this will help in increasing the response time of our brain.

### **1.2 Objective**

The motive of this project is to create a game in Java in which the user is presented with a meaningful word and the user has to rearrange the letter of that word into another meaningful word in a fixed amount of time. The new word should have exact letters as in old one after the rearrangement.

Basically, the main objective of this game will be on focusing to increase the thinking power of the player by giving them boundation of time.

### **1.3 Motivation**

The motivation for doing this project was primarily an interest for working in the android studio project. We get this opportunity in college to work in android studio by developing a game which will enhance the brain power and IQ of people of all age groups , to make an app which will be beneficial for people in one way or other and to make a project which includes different technologies in the single platform . So thus having a little good knowledge of Java and advance java we started to work on this project named as “A GAME OF ANAGRAMS”.



## **1.4 Definition and Overview**

“A Game of Anagram”: In this project we are going to create a game in which the user is presented with a meaningful word and the user has to rearrange the letter of that word into another meaningful word in a fixed amount of time. The new word should have exact letters as in old one after the rearrangement.

We have made an android game application which will let the user to register and after player logins, difficulty level can be selected and the anagrammed word will be generated according to the level. At last user score and rating will be displayed.

The user can enter words in input field and submit it by clicking “GO” button.

To clear your input and reset the letters without submitting the word, click the “Clear” button. If you want to give up and end the game, click the “Show Answers” button.

Your game will also end if you run out of time.

## **CHAPTER 2**

### **OVERALL DESCRIPTION**

#### **2.1 Project Perspective**

In this project we are going to create a game in which the user is presented with a meaningful word and the user has to rearrange the letter of that word into another meaningful word in a fixed amount of time. The new word should have exact letters as in old one after the rearrangement.

The user can enter words in input field and submit it by clicking “GO” button. To clear your input and reset the letters without submitting the word, click the “Clear” button. If you want to give up and end the game, click the “Show Answers” button. Your game will also end if you run out of time.

If your time limit gets over or if user submit a wrong answer then a dialog box with a suitable pop up message will appear on the screen. In both the above cases correct answer will also get displayed on the screen along with scores.

About scoring and timing:

You score points for making answers. The number of points you gain per answer is determined automatically based on a number of different factors.

You start the game with a given amount of time, in seconds. If you run out of time, your game is over and you will be shown the answers you didn't get. You can also give up by pressing the

“Show Answers” button to see the ones you missed.

When the game is over, you will gain a bonus based on your remaining time, if any. The bonus will be greatly increased if you have found all of the words.

We are planning to develop this android game further, that multiple player will be able to play with each other . And we will make this game compatible with IOS devices . This game will be free for all users.

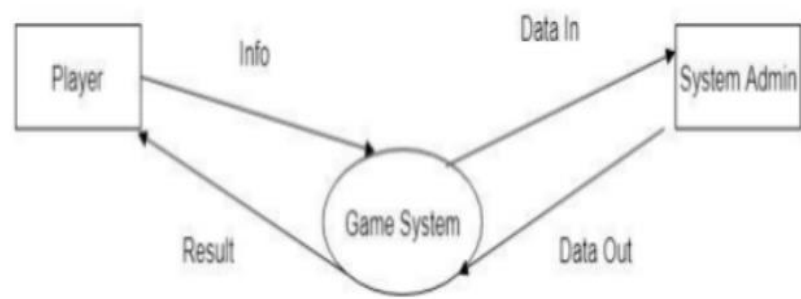
## **2.2 Project Function**

Anagrams lets the user play multiple rounds of the anagram game. In a round, the user is offered an anagram of a random word from a list, and the user can repeatedly try to guess the word from which the anagram is derived. Each guess is evaluated as either correct or incorrect. No limits are imposed on the number of rounds, the number of guesses per round, and the time taken for making a guess. A game score is maintained: how many rounds were eventually correctly guessed and how many not.

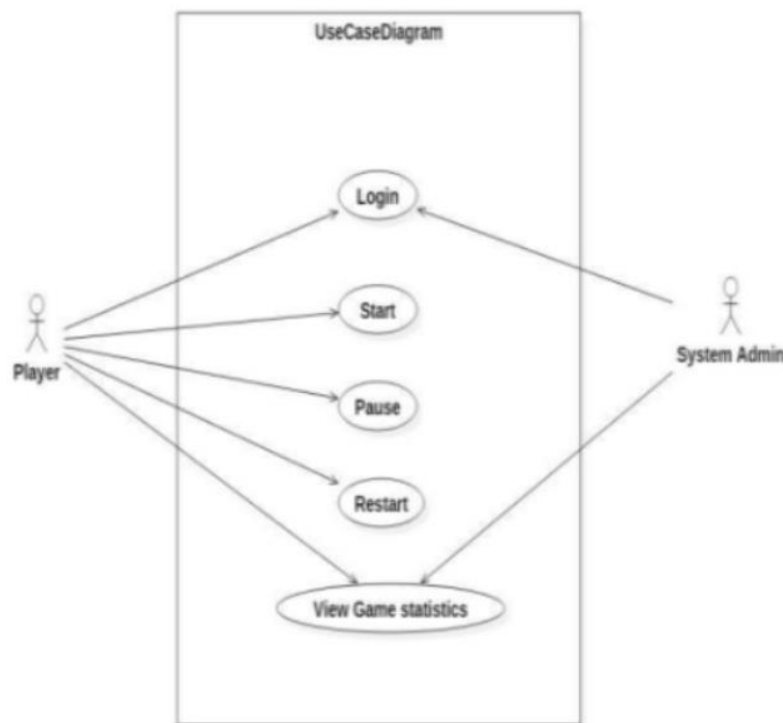
The functioning of our project till now is something like this:

1. Opening the game, it will show splash screen.
2. Then login screen will be displayed where user will be need to register first.
3. Now, user will choose the difficulty level.
4. According to the level selected user will get 10 jumbled word one by one .
5. For the new anagrammed word the user will have to click on the button NEXT WORD.
6. After completion of the game rating will be showed and the message will be displayed according to the performance.
7. If the user wish to play the game again he/she can click the button PLAY AGAIN, or can click on CLOSE PROGRAM.

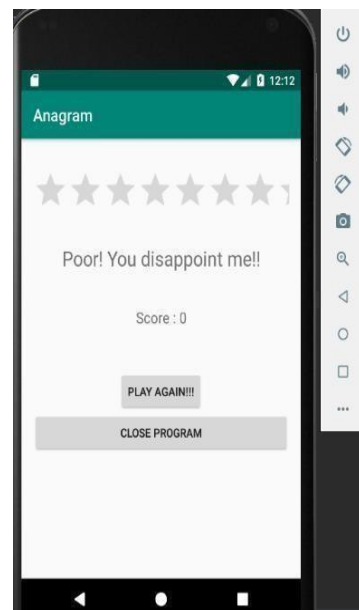
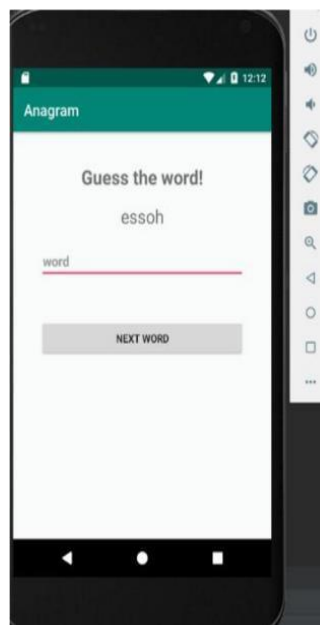
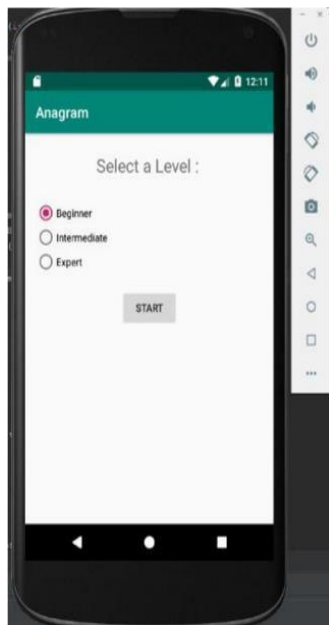
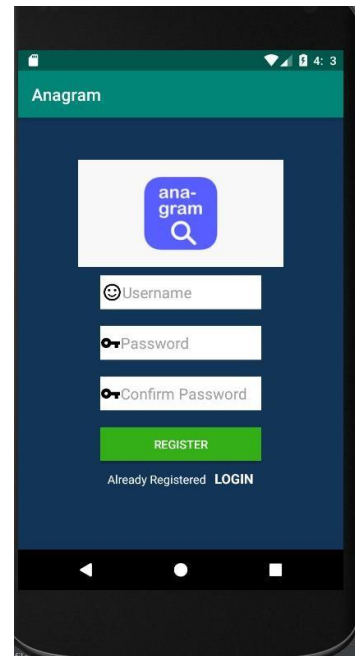
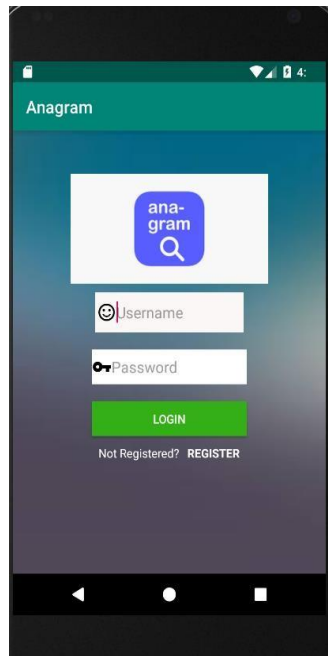
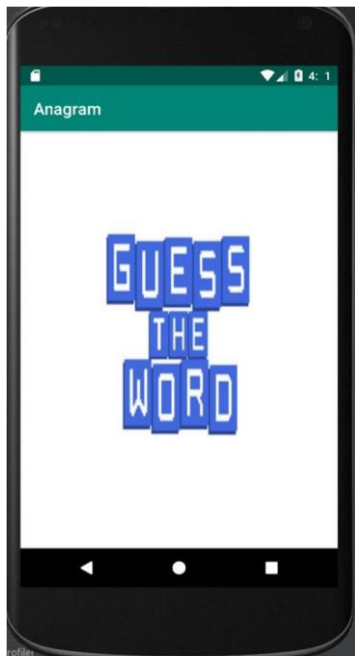
**2.3. Context Level Diagram**



**2.4 Use Case Diagram**



## 2.2.1 Snapshots



## 2.2.2. Code Snapshot

### 1. Splash screen code

```
1 package com.example.anagram;
2
3 import ...
4
5
6
7
8
9 public class splash extends AppCompatActivity {
10
11     @Override
12     protected void onCreate(Bundle savedInstanceState) {
13         super.onCreate(savedInstanceState);
14         setContentView(R.layout.activity_splash);
15         Handler hd = new Handler();
16         hd.postDelayed(() -> {
17             Intent i=new Intent( packageContext: splash.this,LogInActivity.class);
18             startActivity(i);
19         }, delayMillis: 5000);
20     }
21 }
22
23
24
25
26
27
```

### 2. Code to check whether the user is registered or not

```
Boolean res = db.checkUser(user,pwd);
if(res==true)
{
    Intent j = new Intent( packageContext: LogInActivity.this,FirstActivity.class);
    startActivity(j);
}
else
{
    Toast.makeText( context: LogInActivity.this, text: "LogIn Error",Toast.LENGTH_SHORT).show();
}
```

### 3. Code to do registration

```
if (pwd.equals(chr_pwd)) {
    long val = db.addUser(user, pwd);
    if (val > 0) {
        Toast.makeText( context: RegisterActivity.this, text: "You have registered", Toast.LENGTH_SHORT).show();
        Intent moveToLogin = new Intent( packageContext: RegisterActivity.this, LogInActivity.class);
        startActivity(moveToLogin);
    } else {
        Toast.makeText( context: RegisterActivity.this, text: "Registration Error", Toast.LENGTH_SHORT).show();
    }
} else {
    Toast.makeText( context: RegisterActivity.this, text: "Password is not matching", Toast.LENGTH_SHORT).show();
}
```

#### 4. Code to select levels of game

```
public void showDetails(View view) {
    int level = 0;
    switch (rg.getCheckedRadioButtonId()) {
        case R.id.radioButton1:
            level = 1;
            break;
        case R.id.radioButton2:
            level = 2;
            break;
        case R.id.radioButton3:
            level = 3;
            break;
    }
    if (level != 0) {
        Intent intent = new Intent( packageContext: FirstActivity.this, MainActivity.class);
        intent.putExtra( name: "level", level);
        startActivity(intent);
        finish();
    }
}
```

#### 5. Code to load words from dictionary according to levels

```
private void initializeDictionary() {
    dictionary = new ArrayList<>();
    try {
        InputStream is = getResources().openRawResource(R.raw.diot);
        BufferedReader reader = new BufferedReader(new InputStreamReader(is));
        String str = reader.readLine();
        while (str != null) {
            if (level == 1 && (str.length() == 5 || str.length() == 6)) {
                dictionary.add(str);
            } else if (level == 2 && (str.length() == 7 || str.length() == 8)) {
                dictionary.add(str);
            } else if (level == 3 && str.length() > 8) {
                dictionary.add(str);
            }
            str = reader.readLine();
        }
    } catch (Exception ex) {
        Log.e( tag: "MainActivity", ex.toString());
    }
    Collections.shuffle(dictionary);
    while (dictionary.size() > 10) {
        dictionary.remove( index: 0);
    }
    for (String word : dictionary) {
        Log.e( tag: "MainActivity", word);
    }
}
```

## 6. Code to display score and message

```
private void showScore() {
    String message;
    if(score>=8){
        message = "Awesome! You did a great job!!";
    }
    else if(score>=5){
        message = "Good! but Practice More!!";
    }
    else{
        message = "Poor! You disappoint me!!";
    }
    tvMessage.setText(message);
    tvScore.setText("Score : "+score);
    bar.setRating(score);
}
```

## 7. Code for saving data in database

```
public long addUser(String user, String password){
    SQLiteDatabase db = this.getWritableDatabase();
    ContentValues contentValues = new ContentValues();
    contentValues.put("username",user);
    contentValues.put("password",password);
    long res = db.insert( table: "registeruser", nullColumnHack: null,contentValues);
    db.close();
    return res;
}

public boolean checkUser(String username, String password){
    String[] columns = { COL_1 };
    SQLiteDatabase db = getReadableDatabase();
    String selection = COL_2 + "=? " + " and " + COL_3 + "=?";
    String[] selectionArgs = { username, password };
    Cursor cursor = db.query(TABLE_NAME,columns,selection,selectionArgs,groupBy: null, having: null, orderBy: null);
    int count = cursor.getCount();
    cursor.close();
    db.close();

    if(count>0)
        return true;
    else
        return false;
}
```



## **2.3 Constraints and assumption**

### **Constraints:**

1. The jumbled words which are going to be presented to the player is already built in the application there is no way that the application should get same jumbled word to guess or same word everytime to guess . This not even require any kind of network support to run the game till now.
2. To solve the anagram in limited period of time.
3. After learning hint from the computer the score will be deducted.
4. Player will get limited chances to skip the anagram.

### **Assumption:**

1. Anagram will not get repeated.
2. Anagram will be generated according to selected difficulty.
3. There will always be one winner.
4. Final assumption of this project is to be played by multiple player.

## **CHAPTER 3**

### **SYSTEM REQUIREMENTS**

#### **3.1. External Interface Requirements**

##### **User Interface Requirement**

The system must support following user interfaces:

- i.** web-based
- ii.** stand-alone
- iii.** cellular phone

In this project we are going to create a game in which the user is presented with a meaningful word and the user has to rearrange the letter of that word into another meaningful word in a fixed amount of time. The new word should have exact letters as in old one after the rearrangement.

The user can enter words in input field and submit it by clicking “GO” button. To clear your input and reset the letters without submitting the word, click the “Clear” button. If you want to give up and end the game, click the “Show Answers” button. Your game will also end if you run out of time.

If your time limit gets over or if user submit a wrong answer then a dialog box with a suitable pop up message will appear on the screen. In both the above cases correct answer will also get displayed on the screen along with scores.

About scoring and timing:

You score points for making answers. The number of points you gain per answer is determined automatically based on a number of different factors.

You start the game with a given amount of time, in seconds. If you run out of time, your game is over and you will be shown the answers you didn't get. You can also give up by pressing the

“Show Answers” button to see the ones you missed.

When the game is over, you will gain a bonus based on your remaining time, if any. The bonus will be greatly increased if you have found all of the words.

We are going to learn a lot through this project.

### **Hardware interface Requirement**

“A Game of Anagram”:In this project we are going to create a game in which the user is presented with a meaningful word and the user has to rearrange the letter of that word into another meaningful word in a fixed amount of time. The new word should have exact letters as in old one after the rearrangement.

We have made an android game application which will let the user to register and after player logins, difficulty level can be selected and the anagrammed word will be generated according to the level. At last user score and rating will be displayed.

Requirements of Android emulator or Android mobile device:

- i. Nexus 4 API 25
- ii. 1 Ghz processor(recommended)
- iii. 1GB of RAM(recommended)

### **Software Interface Requirement**

“A Game of Anagrams” has been developed till now using android game development tool named as android studio. It is an official integrated development kit or we can say that tool in human terms which is used for developing applications which will be based on android and can easily run on any android platform. This tool plays a great role in the advancement of the technology field in the world by making things efficient and handy for the people. The basic language for working in android is Java . One should have the good knowledge of Java language to easily get to the points of android studio.

#### **Requirements:**

- i. Java Development Kit
- ii. Windows 10 or Windows 8
- iii. SQL( Local Database)
- iv. Android SDK

## **Communications Interfaces**

Currently the APK will communicate through offline databases and after making the game online it will communicate through online databases

### **3.2 Functional Requirement**

#### **Minimum Requirement:**

1. CPU: Qualcomm Snapdragon S4 Pro
2. OS: Android 7.1.1 (Google APIs))

### **3.3 Non-functional Requirement**

Dictionary.txt is used to read the words.

### **3.4 Hardware and Performance Requirement**

#### **Minimum Requirement:**

1. SCREEN: 4.7" diagonal; 768 x 1280:xhdpi pixel resolution ,WXGA  
IPS; API 25
2. CPU: Qualcomm Snapdragon(TM) S4 Pro
3. MEMORY: 16 GB internal storage (actual formatted capacity will be less);  
2 GB RAM  
OS: Android 7.1.1 (Google APIs))

## CHAPTER 4

### SYSTEM FEATURE

This system contain following features

1. Authentication
2. Notification through toast messages
3. Buttons for navigation

#### 1. Authentication

##### **Description and Priority**

This will give the player a descriptive splash screen for 5 second as well as login screen followed by registration screen. The login is enabled for all the newly or already logged in users. All the entries gets saved to local database for easy further log in process.

##### **Stimulus/Response Sequences**

It will consist of two required fields, naming Username and Password followed by registration button. A descriptive line is boldly displayed at the screen defining the login surface at the top and for registration at the bottom. There is a login button. On successful login user is directed to level selection screen and on unsuccessful login the user should go to registration screen and an error message get displayed.

##### **Functional Requirements**

If the user is already registered in database then he should directly go to level selection screen otherwise first he should go to registration screen then login screen and after that to level selection screen.

## **2. Notification through toast messages**

### **Description and Priority**

This android application will provide upto date information regarding the current processes going on the application and no matter how many times player plays the game it will always displays the latest and the best results everytime.

Whenever an unwanted work is done a toast message gets displayed accordingly.

### **Stimulus/Response Sequences**

Non-registered user can register in the database . No same ID will get saved again and again in local database.

### **Functional Requirements**

The function of the notification is to provide toast messages at correct timing so that the player can take action accordingly .

## **3. Buttons for navigation**

### **Description and priority**

The player will be able to navigate to different screens of the game and this feature will also help the player by making the game efficient and everything accessible to the player easily .

### **Stimulus/Response Sequences**

Buttons like next word move us to next screen with new jumbled word, login button move user to level selection screen, play again button move user to level selection screen again and close program will close the game in respective emulator or android device.

### **Functional Requirements**

Navigation to correct screen.

## **CHAPTER 5**

### **OTHER NON-FUNCTIONAL REQUIREMENTS**

#### **5.1. Performance Requirements**

##### **Real Time**

This android application will provide upto date information regarding the current processes going on the application and no matter how many times player plays the game it will always displays the latest and the best results Everytime.

##### **System Resource Consumption**

As when the application runs on android emulator it should not reach that amount of time that it leads to do cold boosting of emulator as at that time the emulator freezes due to more number of attempt.

The application should not consume more power.

#### **5.2. Safety Requirement**

- i. Player need to register for the game and make their details get saved.
- ii. Player should not play this game during riding or doing any such kind of work.

#### **5.3. Security Requirement**

The security system features include making every user get register for the game and saving there login and password details in database at background which is not able to get accessed by anyone besides the administrators.

- 1. LogIn Requirements
- 2. Password Requirements

#### **5.4 Software Quality Attributes**

##### **5.4.1. Reliability**

The application will meet all of the functional requirements without any unexpected behavior.

#### **5.4.2. Availability**

The application will be available at all times on the user's Android device, as long as the device is in proper working order. Soon the functionality of the application will depend on any external services such as internet access that are required. If those services are unavailable, the user should be alerted.

#### **5.4.3. Security**

The software should never disclose any personal information of any users, and should collect no personal information from its own users.

#### **5.4.4. Maintainability**

The application can be maintained easily.

#### **5.4.5. Portability**

This software will be designed to run on any Android operating system version 5.4.1 or higher. The software will be forward compatible for all currently released Android operating system versions.



## **CHAPTER 6**

### **CONCLUSION AND FUTURE WORK**

#### **6.1 Conclusions**

In this project we are going to create a game in which the user is presented with a meaningful word and the user has to rearrange the letter of that word into another meaningful word in a fixed amount of time. The new word should have exact letters as in old one after the rearrangement.

We have made an android game application which will let the user to register and after player logins, difficulty level can be selected and the anagrammed word will be generated according to the level. At last user score and rating will be displayed.

In this journey of our project we have developed the android application software named as 'Anagram.apk' till now and through this journey we get a vast knowledge of android studio . We have learned a lot through this project which we believe will surely help us in future projects.

Currently we are working on further extending our project by putting more enhanced graphical design and by making the project multiplayer through online databases.

#### **6.2 Scope for Future Work**

1. Put Graphical representation: Enhancement of graphical display.
2. Our future work also include making this game multiplayer by using online database such as firebase.
3. Introduce new game features: Multiplayer mode will be made available.
4. More user friendly.
5. Process of displaying scores will be enhanced.

## REFERENCES

1. [www.geeksforgeeks.org](http://www.geeksforgeeks.org)
2. [www.playwordplay.com](http://www.playwordplay.com)
3. <https://en.wikipedia.org>
4. <https://www.thewordfinder.com>