TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

This is a simply and beautify android quiz application with SQLite database. Theandroid quiz applications test your knowledge and understanding in different areas. After each quiz, the user will see the final score and you have the ability togo through the quiz result analysis to see questions that you fail and the correct answers.

1.1 Overview

In today's world, Smart phones have changed our lives and have become an indispensable part of our lives because of its specialty to simplify our routine work and thereby saving our time. A Smartphone with an Android OS offers excellent functionality to the users offering a distinct experience. Android is a Linux based operating system and it was bought by Google in 2007.There are tons of application available and one of the prime reason for this vast number is android being an open source. On the other hand, android based device like mobile, tab are very user friendly. A survey has done by "Light Castle Partners" research wing which indicates that though other operating system mobile usersexist but the majority users are goes with android operating system. In this context, Project application is developed based on android platform. The name of application is defined as 'My Awesome Quiz'.

1.2 Purpose

This document provides a base to all the functionalities which should be carried out by the application, how that works the outputs available to the end user.

1.3 Motivation

Currently most of the Examination like organizational recruitment, University class test are paper based, which costs time and resources. Questionnaire is developed, printed, and then collect data, entry, editing, cleaning, which timeconsuming and costly. Proposed application is the starting for avoid those circumstances which are been currently faced by any organization.

CHAPTER 2
SYSTEM REQUIREMENT

2.1 Software & Hardware Requirements

Hardware Requirements:

- Hardware Specification: -Processor Intel Pentium V or higher
- Clock Speed: -1.7 GHz or more
- System Bus: -64 bits
- RAM: -16GB
- HDD: -2TB
- Monitor: -LCD Monitor
- Keyboard: -Standard keyboard

Software Requirements:

- Mouse: -Compatible mouse
- Operating System: -Windows 10
- Software: -Microsoft SQL Server
- Front End: -Java core/swings (NetBeans)
- Back End: -My SQL

2.2 Application and Technology used

Android Application Development is possible with a couple of software and development kits to support the software and execution, they are as follows

IntelliJ IDEA: IntelliJ IDEA is the official Integrated Development Environment (IDE) for designing, coding, debugging andexecuting applications for Google's Smartphone operating system, Android. It has all keywords inbuilt for ease in back end programming and also design statements, for ease in designing as well. The version of Android Studio used to design the application is 1.4.1.to 3.1 as we are working on IntelliJ IDEA.

Java Development kit (JDK): Since, Android applications requireJava programming for its backend programming; it needs a JAVA environment to support its functions, executions and syntax

CHAPTER 3
SYSTEM APPLICATION AND TECHNOLOGY USE

3.1 Description

Using "Android Quiz App" source code package you can create amazing android quiz app. "Android Quiz App" source code package is built with latest android studio and is easy to use and configure. With little or no coding knowledge you can easily create quiz app with the help of this source code package.

Main Features:

- Beautiful UI
- Multilingual support
- Easy to Customize/Re-skin
- Support all Screen size
- Set Unlimited Questions

This Quiz app package is easy to customize and use. Make most from this Quiz app source code package. Best code with lowest price on thismarketplace.

3.2 Entity-Relationship Diagram

An entity relationship diagram (ERD) shows the relationships of entity sets stored in a database. An entity in this context is an object, a component of data. An entity set is a collection of similar entities. These entities can have attributes that define its properties.

There are two reasons to create a database diagram. You're either designing a new schema or you need to document our existing structure. If you have an existing database you need to document, you create a database diagram using data directly from your database. You can export your data base structure as a CSV file (there are some scripts on how to do this here), then have a program generate the ERD automatically. An ER diagram is a means of visualizing how the information a system produces is related. There are five main components of an ERD:

- Entities, which are represented by rectangles. An entity is an object or concept about which you want to store information.
- A weak entity is an entity that must defined by a foreign key relationship with another entity as it cannot be uniquely identified by its own attributes alone.
- Actions, which are represented by diamond shapes, show how two entities share

information in the database.

- In some cases, entities can be self-linked. For example, employees can supervise other employees.
- Attributes, which are represented by ovals. A key attribute is the unique, distinguishing characteristic of the entity.
- A multivalued attribute can have more than one value. For example, an employee entity can have multiple skill values.
- A derived attribute is based on another attribute. For example, an employee'smonthly salary is based on the employee's annual salary.
- Connecting lines, solid lines that connect attributes to show the relationships of entities in the diagram.
- Cardinality specifies how many instances of an entity relate to one instance ofanother entity. Ordinality is also closely linked to cardinality.

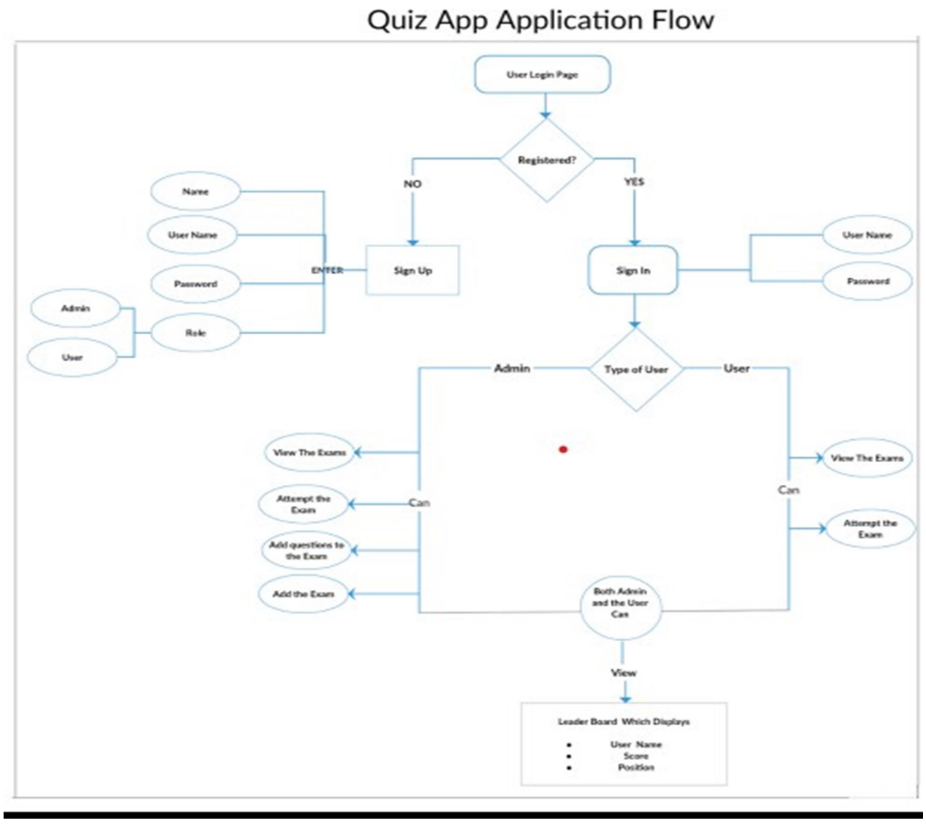

Fig1: Flow chart

CHAPTER 4
DESCRIPTION

This project is done on the concept of quiz system which automatically shows the answer for the better interaction with the participant. This application is done on java swing which stores the admin and user (Registered credentials) on the MySQL backed and also stores the questions and answers.

A Quiz App is a software application designed to facilitate quizzes, assessments, or trivia games for users. It typically presents a series of questions to the user, allows them to input their answers, and provides feedback on their performance. The app may cover various topics, ranging from educational subjects to general knowledge or entertainment.

Features:

User Registration/Login: Users can create accounts or log in to track their progress and scores. Personalization features may include user profiles and preferences.

Quiz Creation: Admins or authorized users can create quizzes. Each quiz consists of multiple-choice questions, true/false statements, or other question types. Questions may have multimedia elements such as images or videos.

Question Types:

- Multiple Choices
- Users select the correct answer from a list of options.
- True/False: Users choose whether a statement is true or false.
- Fill in the Blank: Users provide the missing word or phrase.
- Matching: Users match items from two columns.

Scoring and Feedback:

- Instant feedback on each question, indicating correctness.
- Cumulative scoring throughout the quiz.
- Final score and performance summary at the end of the quiz.

Timed Quizzes:

- Optional time limits for each question or the entire quiz.
- Display a countdown timer to create a sense of urgency.

Leader boards:

- Display high scores and rankings.

- Encourage competition among users.

Progress Tracking:

- Users can review their quiz history and track improvements.
- View detailed results for each attempt.

Social Sharing:

- Share quiz results on social media.
- Challenge friends to take the same quiz.

Categories and Tags:

- Organize quizzes by categories or tags (e.g., science, history, general knowledge).
- Allow users to filter quizzes based on their interests.

Accessibility:

- Ensure the app is user-friendly and accessible to individuals with disabilities.
- Provide options for adjusting text size and colour contrast.

Security:

- Implement secure user authentication to protect user data.
- Safeguard against cheating or unauthorized access to quiz content.

Offline Mode:

- Allow users to download quizzes for offline use.

Analytics:

- Collect data on user performance and preferences.
- Use analytics to enhance and personalize the user experience.

Remember, the specific features and functionalities can vary based on the intended purpose of the Quiz App, whether it's for educational purposes, entertainment, or corporate training. Additionally, the user interface and user experience design play a crucial role in the app's success and engagement.

All the mentioned systems primarily focus on day-to-day employee project and delivery monitoring. In contrast, Enterprise Quiz App primarily focuses on evaluating the employee’s calibre for upcoming projects and emerging technologies. Most of the systems available in the market will gauge the employee’s performance in the current project. Hence there is very much need for an application which primarily focuses on developing the employee’s capabilities on emerging technologies and upcoming project requirements, Enterprise Quiz App is developed to address such issues and will be very helpful in the constant development of the employee

I came up with a web application, where the user can access the application through the user login method and can attempt the exam. Since the exams are assigned only for a short period, the user can skip questions and proceed further to avoid negative marking. After each attempt, the user can get access to check the results through a leader board, where the user can know about his/her position out of several people. Quiz App has a very granulated permit system. Every action is performed with a privilege. The Administrator can add the Exam, Questions and can see the Leader board, whereas a user can only attempt an exam and view the Leader board.

Apart from widely used applications for gauging pupil's capabilities on a specific topic, I am more interested in developing a forum where employees of an enterprise will be engaged to audit their capabilities on upcoming project technologies. It will help the employer to choose the best team to fulfil the project requirements easily. For gauging employee's capabilities, there are very few applications such as "Basecamp" which primarily focuses on employee engagement. Most of the Software Companies are still relying on face to face interviews to select the team to take up the new project. It will take valuable time and will be very tedious for a project manager. Also, there will be a very confined set of employees to gauge. The enterprise Quiz app will decrease manual efforts in selecting the right set of employees for the upcoming project. The Employer can conduct quizzes on the topics based on the requirements of the new project, and all the employees in the organization can attempt the test. The Employer can go through the Leader board console and select the best team that can fulfil the project's requirement. This application can have a huge impact on larger scale organizations. Let's take an example of an Organization which has 500 employees. It will be very tedious to choose the best team for an upcoming project by interviewing each employee in person. Hence the project manager can conduct a few tests on the technologies used in the 4 upcoming project and can filter top 10 of the Leader board. On the other hand, even the employee will know his rank is seeing the Leader board and try focusing on the future endures. The gaming like the design of the system, especially with the Leader board will be fun for an employee to attempt the tests.

CHAPTER 5
ICON OF APPLICATION

Icon:

This is the icon of the project which will display on the app drawer of our android phone. After clicking this icon the app will run. This icon is created by using Adobe Photoshop

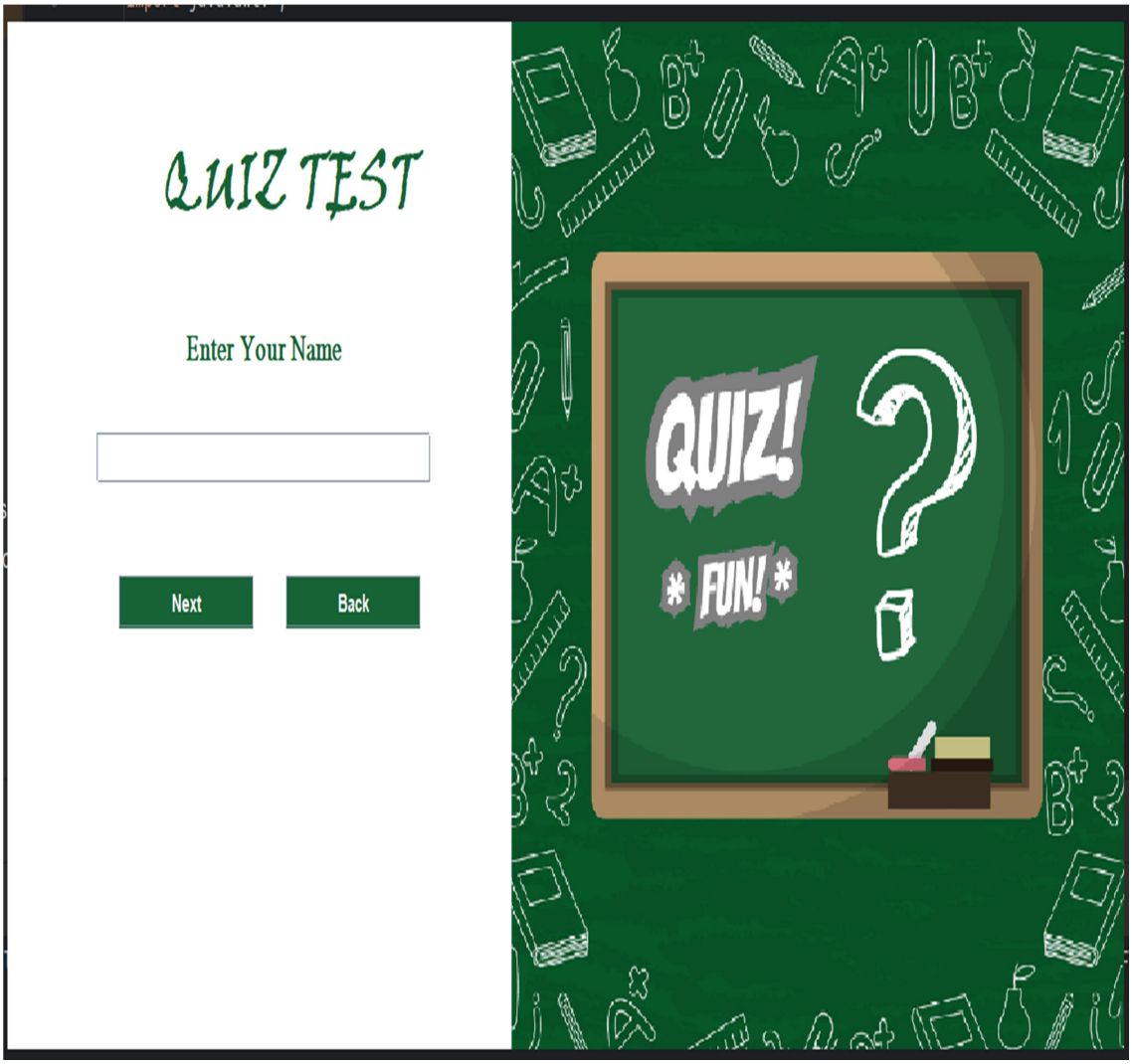

Fig 2: Image icon of project

CHAPTER 6

PROCESS AND VALIDATION

I am working on a quiz app where i am retrieving my data from my model class. Everything is working fine for now. However i want to implement button validation for my answers. For instance, when the user answers a question i want the correct answer (button) to blink green color and the wrong one to blink red color in case the user gets it wrong.

Code:

```
/ This file contains questions from QuestionBank
public class QuestionLibrary{
   // array of questions
   private String mQuestions [] = {
// my questions
 };
   // array of multiple choices for each question
   private String mChoices [][] = {
// array of choices appear here
            };
   // array of correct answers – in the same order as array of questions
private String mCorrectAnswers[] = {
         // correct answers appear here
};
   // method returns number of questions
   public int getLength(){
      return mQuestions.length;
   }
   // method returns question from array textQuestions[] based on array index
   public String getQuestion(int a) {
      String question = mQuestions[a];
      return question;
   }
   // method return a single multiple choice item for question based on array index,
   // based on number of multiple choice item in the list – 1, 2, 3 or 4 as an argument
```

```
    public String getChoice(int index, int num) {
        String choice0 = mChoices[index][num-1];
        return choice0;
    }
    //  method returns correct answer for the question based on array index
    public String getCorrectAnswer(int a) {
        String answer = mCorrectAnswers[a];
        return answer;
    }
}
public class QuizActivity extends AppCompatActivity {
    private QuestionLibrary mQuestionLibrary = new QuestionLibrary();
    Button button1;
    Button button2;
    Button button3;
    Button button4;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_begineer);
        button1 = (Button) findViewById(R.id.firstOption);
        button2 = (Button) findViewById(R.id.secondOption);
        button3 = (Button) findViewById(R.id.thirdOption);
        button4 = (Button) findViewById(R.id.fourthOption);
        updateQuestion(); //update question
        updateQuizNumber(mquizNumber);
    }
    private void updateQuizNumber(int mquizNumber) {
        msingleQuestion.setText("" + mquizNumber+"/"+mQuestionLibrary.getLength());
    }
    private void updateQuestion() {
        // check if we are not outside array bounds for questions
        if(mQuestionNumber<mQuestionLibrary.getLength() ){
            // set the text for new question, and new 4 alternative to answer on four buttons
```

```
            mQuestion.setText(mQuestionLibrary.getQuestion(mQuestionNumber));
            button1.setText(mQuestionLibrary.getChoice(mQuestionNumber, 1));
             button2.setText(mQuestionLibrary.getChoice(mQuestionNumber, 2));
             button3.setText(mQuestionLibrary.getChoice(mQuestionNumber, 3));
            button4.setText(mQuestionLibrary.getChoice(mQuestionNumber,4));
             mAnswer = mQuestionLibrary.getCorrectAnswer(mQuestionNumber);
             mQuestionNumber++;
        }
        else {
            Intent intent = new Intent(this, MenuOptions.class);
//              intent.putExtra("score", mScore); // pass the current score to the second screen
             startActivity(intent);
        }
    }
    public void onClick(View view) {
        //all logic for all answers buttons in one method
        Button answer = (Button) view;
        // if the answer is correct, increase the score
        if (answer.getText() == mAnswer){
            Toast.makeText(BegineerActivity.this, "Correct!", Toast.LENGTH_SHORT).show();
// i need to validate correct answer here by making the button blink green
        }else   {
            Toast.makeText(BegineerActivity.this, "Wrong!", Toast.LENGTH_SHORT).show();
// i need to validate wrong answer here by making the button blink red
        }
        if (mQuestionNumber < mQuestionLibrary.getLength()) {
            // once user answer the question, we move on to the next one, if any
            updateQuestion();
            updateQuizNumber(mquizNumber);
        } else {
//
        }  } }}
```

CHAPTER 7

JAVA CODES

```
package quiz.app;
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
public class Rules extends Jframe implements ActionListener {
    Jbutton start, back;
    String name;
    Rules(String name){
        this.name = name;
        Jlabel heading = new Jlabel("Welcome"+ name +"to QUIZ TEST");
        heading.setBounds(150,100,700,30);
        heading.setFont( new Font("Viner Hand ITC", Font.BOLD, 28));
        heading.setForeground(new Color(22,99,54));
        add(heading);
        Jlabel rules = new Jlabel();
        rules.setBounds(70,150,700,350);
        rules.setFont( new Font("Tahoma", Font.PLAIN, 16));
        rules.setForeground(new Color(22,99,54));
        rules.setText(
                "<html>"+
                "1. Participation in the quiz is free and open to all
persons above 18 years old." + "<br><br>" +
                "2. There are a total 10 questions. " + "<br><br>" +
                "3. You only have 15 seconds to answer the question." +
"<br><br>" +
                "4. No cell phones or other secondary devices in the room
or test area." + "<br><br>" +
                "5. No talking." + "<br><br>" +
                "6. No one else can be in the room with you." + "<br><br>"
+
                "<html>"        );
        add(rules);
        back = new Jbutton("Back");
        back.setBounds(300,500,100,30);
        back.setBackground(new Color(22,99,54));
        back.setForeground(Color.WHITE);
        back.addActionListener(this);
        add(back);
        start = new Jbutton("Start");
        start.setBounds(450,500,100,30);
        start.setBackground(new Color(22,99,54));
        start.setForeground(Color.WHITE);
        start.addActionListener(this);
        add(start);
        ImageIcon i1 = new
ImageIcon(ClassLoader.getSystemResource("icons/back.png"));
        Image i =
i1.getImage().getScaledInstance(800,650,Image.SCALE_DEFAULT);
        ImageIcon i2 = new ImageIcon(i);
        Jlabel image = new Jlabel(i2);
        image.setBounds(0,0,800,650);
        add(image);
        setSize(800,650);
```

```
        setLocation(350,100);
        setLayout(null);
        setUndecorated(true);
        setVisible(true);
    }
    @Override
    public void actionPerformed(ActionEvent e) {
        if (e.getSource() == start){
            setVisible(false);
            new Quiz(name);
        } else {
            setVisible(false);
            new Login();
        }
    }
    public static void main(String[] args) {

        new Rules("User");
    }
}
```

Fig 3: java codes

CHAPTER 8
TESTING

This chapter gives the outline of all the testing methods that are carried out to get a bug free application.

8.1 Testing Process

Testing is an integral part of software development. Testing process, in a waycertifies, whether the product, that is developed, compiles with the standards, that itwas designed to. Testing process involves building of test cases, against which, theproduct has to be tested. In some cases, test cases are done based on the system requirements specified for the product/software, which is to be developed.

8.2 Testing Objectives

The main objectives of testing process are as follows:

- Testing is a process of executing a program with the intent of finding an error.
- A good test case is one that has high probability of finding an as yet undiscovered error.
- A successful test is one that uncovers an as yet undiscovered error.

8.3 Levels of Testing

Different levels of testing are used in the testing process; each level of testing aims to test different aspects of the system. The basic levels are unit testing, integration testing, system testing and acceptance testing.

8.4 Unit Testing

Unit testing focuses verification effort on the smallest unit of softwaredesign the module. The software built, is a collection of individual modules. In this kind of testing exact flow of control for each module was verified. With detailed design consideration used as a guide, important control paths are tested to uncover errors within the boundary of the module.

8.5 Integration testing

The second level of testing is called integration testing. In this, many class-tested modules are combined into subsystems, which are then tested. The goal here is to see if all the modules can be integrated properly. We havebeen identified and debugged.

8.6 System testing

Here the entire application is tested. The reference document for this process is the requirement document, and the goal is to see IF the applicationmeets its requirements. Each module and component of ethereal was thoroughly tested to remove bugs through a system testing strategy. Test caseswere generated for all possible input sequences and the output was verified for its correctness.

CHAPTER 9
CREATING LAYOUT

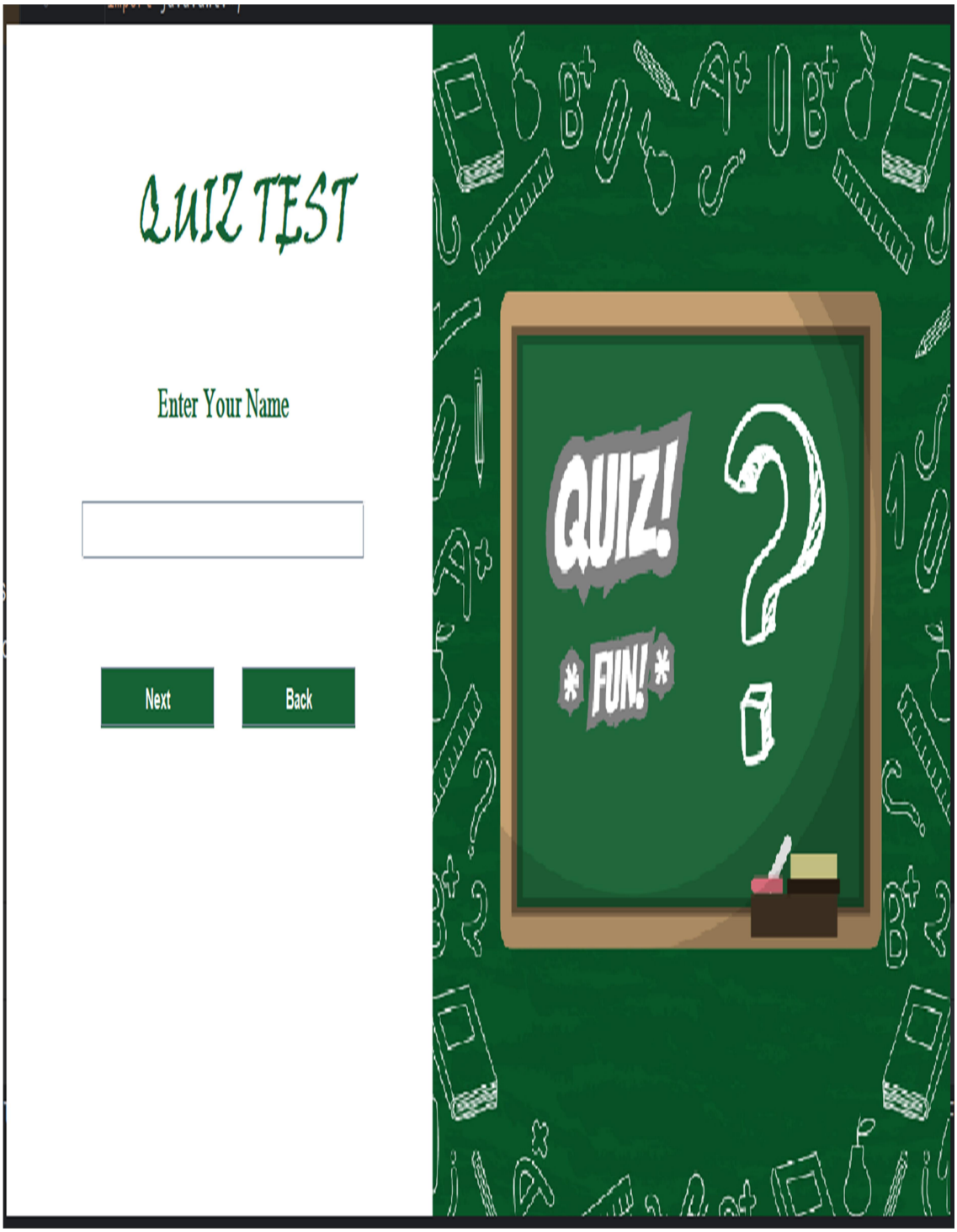

Fig 4: Layout of the project

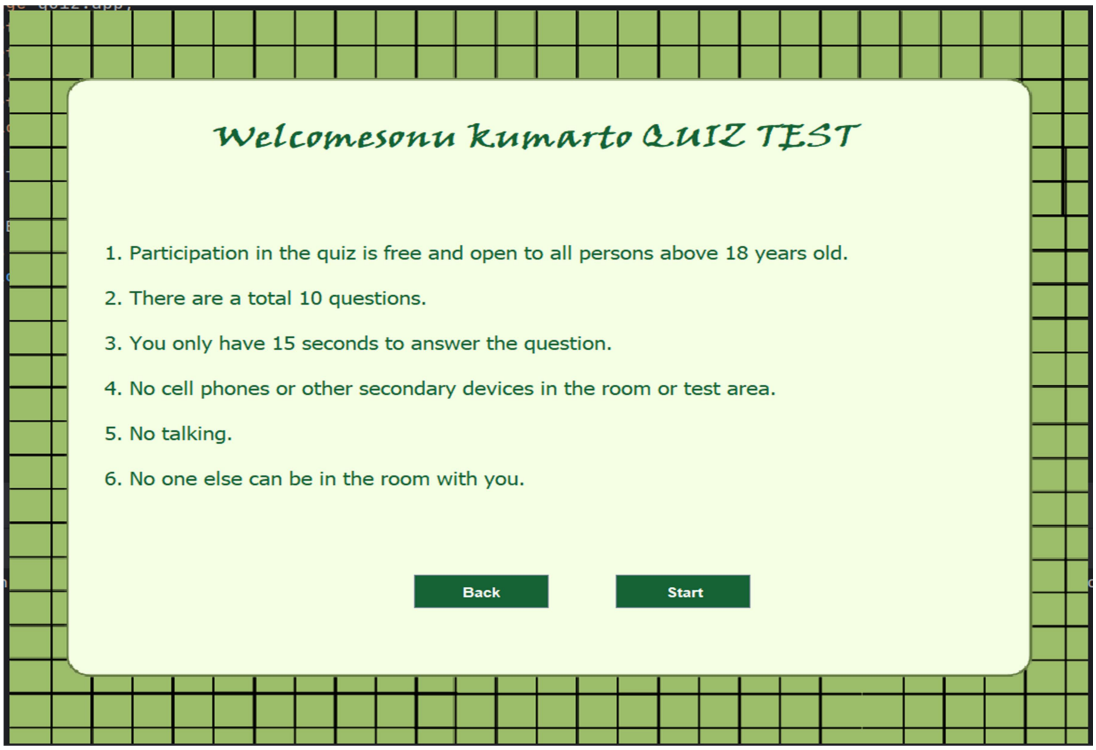

Fig 5: Rule for the test

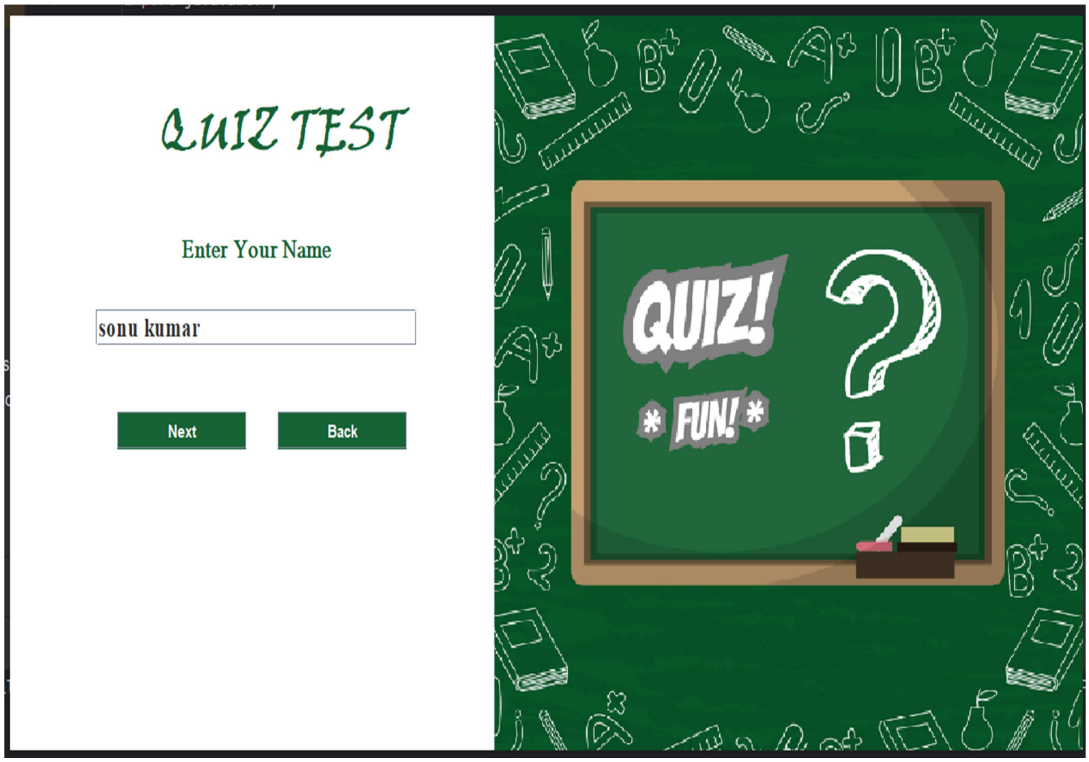

Fig 6: Enter user name

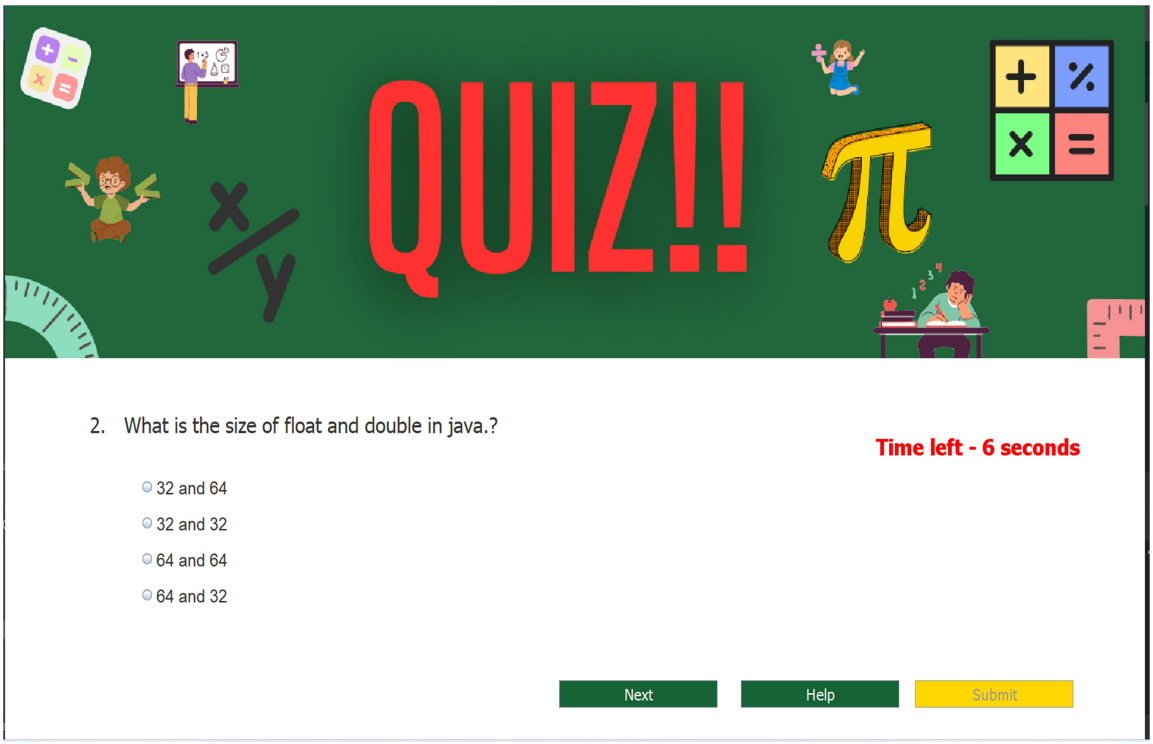

Fig 7: Question Formats

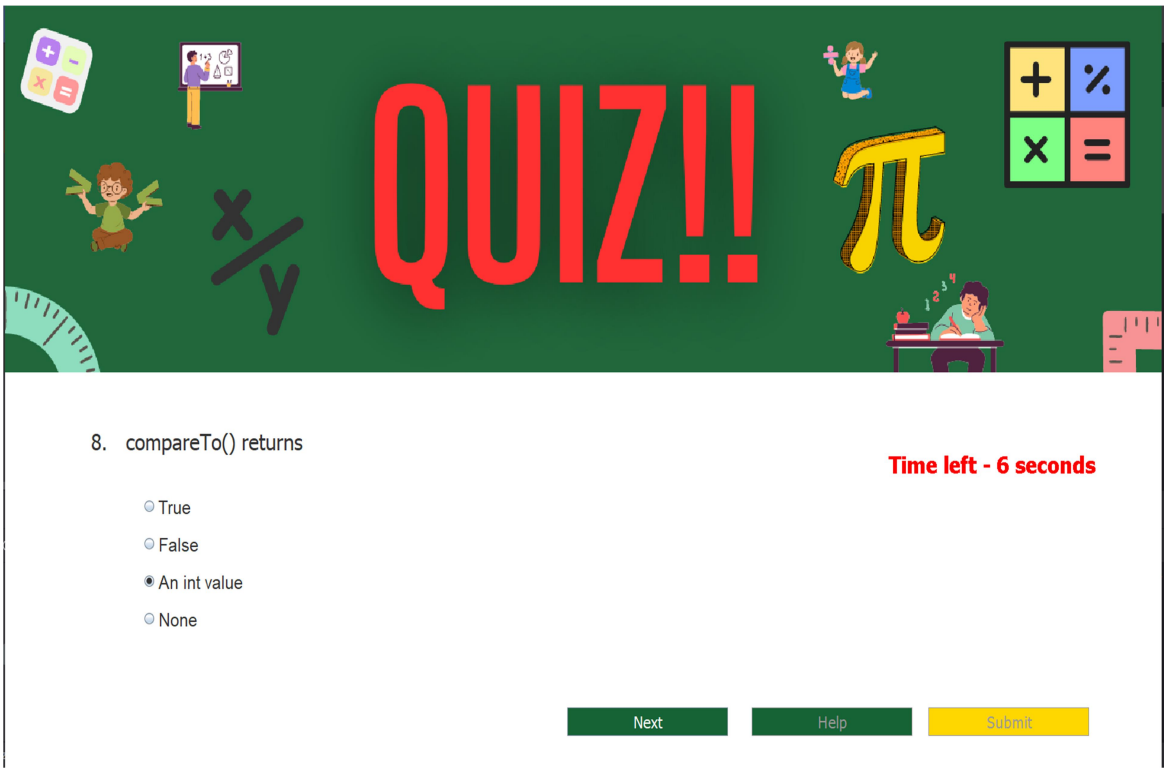

Fig 8: Question Formats

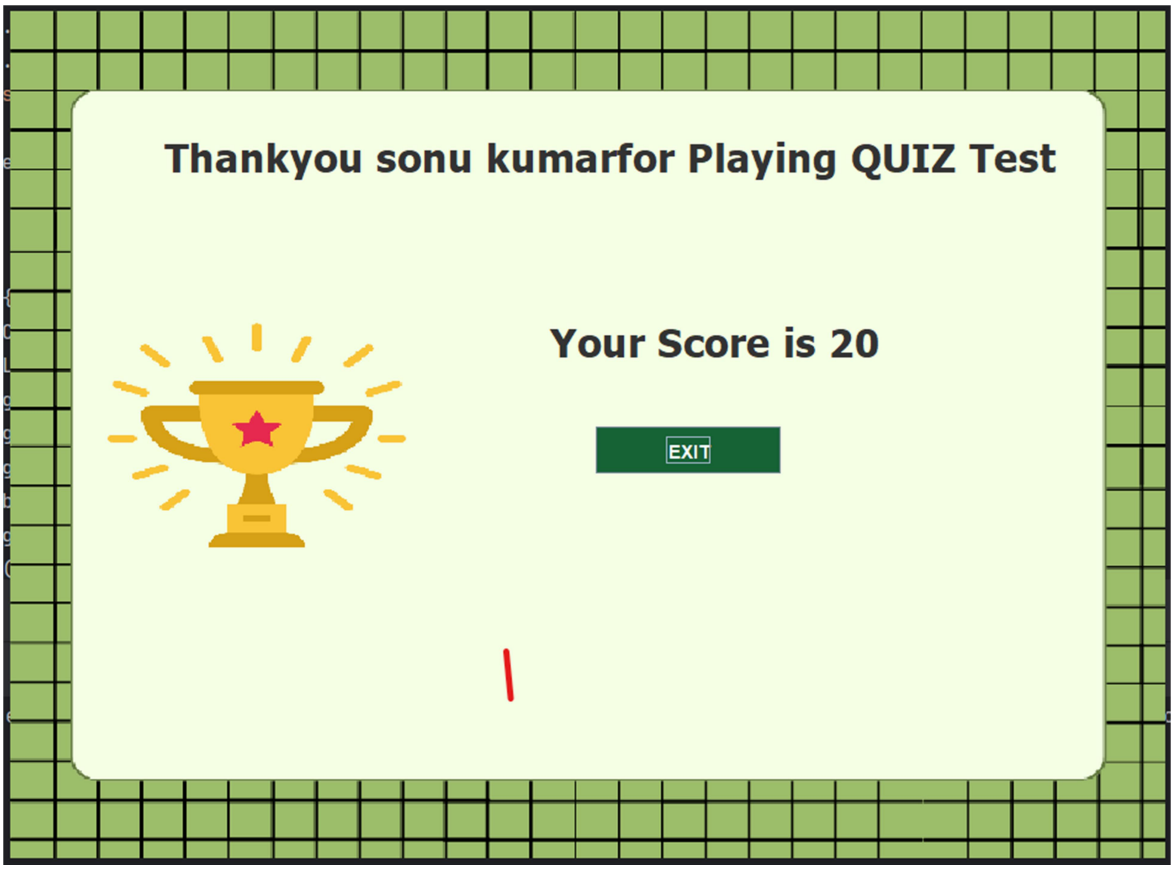

Fig 9: Score of user

CONCLUSION

The project has been successfully completed by having established the users with the help of Android Studio tool. It consists of design plots, layouts plots, java codes, on create plots and computing plots. At the same time there is some scope for improvement in the future. It can be possible to make it more users friendly by adding more variety of functions to it.