STATE UNIVERSITY OF BANGLADESH
join the trendsetter

Online Admission Examination System

This Project was submitted in Partial Fulfillment of The Requirement for The Degree of Bachelor Of Science In Computer Science Of State University Of Bangladesh.

Presented By

MD.Arif Hossen
ID:UG02-50-19-005
Afrin Sultana
ID:UG02-50-19-015
Sumaiya Rashmi
ID:UG02-52-19-012

Supervised By

Muhammad Masud Tarek
Associate Professor (Acting Head)
Department of Computer Science and Engineering.

Co-Supervised By

Muntasir Hasan Kanchan
Assistant Professor
Department of Computer Science.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING,
STATE UNIVERSITY OF BANGLADESH.

January 2023

DECLARATION

We, hereby, declare that the work presented in this Thesis is the outcome of the investigation performed by us under the supervision of **Muhammad Masud Tarek**, Head of The Department of Computer Science and Engineering, State University of Bangladesh. We also declare that no part of this project and thereof has been or is being submitted elsewhere for the award of any degree or Diploma.

Signed by

MD Arif Hossen
ID:UG02-50-19-005

Afrin Sultana
ID:UG05-15-19-015

Sumaiya Rashmi
ID:UG02-50-19-012

Counter signed by

Supervisor
Muhammad Masud Tarek
Associate Professor (Acting Head)
Department of Computer Science and
Engineering. State University of Bangladesh.

Co-Supervised By
Muntasir Hasan Kanchan
Assistant Professor
Department of Computer Science and
Engineering. State University of Bangladesh.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
STATE UNIVERSITY OF BANGLADESH

STATE UNIVERSITY OF BANGLADESH
join the trendsetter

CERTIFICATE OF APPROVAL

The foregoing Thesis is hereby accepted as a credible study of an engineering subject carried out and presented in a manner satisfactory to warrant its acceptance as a prerequisite to the degree for which it has been submitted. It is understood that by this approval the undersigned do not necessarily endorse or approve any statement made, opinion expressed or conclusion drawn therein, but approve the thesis only for the purpose for which it is submitted.

ACKNOWLEDGEMENT

First of all, we would like to thank the almighty ALLAH. We would like to thank our supervisor who guided us in doing the project. We would like to take the opportunity to express our gratitude to Muhammad Masud Tarek, our respected supervisor. He provided us with invaluable advice and helped us in different periods. His motivation and help contributed tremendous to the successful completion of the project.

Besides. We would like to thank all the faculty members who helped us by giving us advice and providing the equipment which we needed.

Also I would like to thank our family and friend for their support. Without that support we couldn’t have succeed in completing this project

At last but not the least, we would like to thank everyone who helped and motivated us to work on this project.

ABSTRACT

Online Admission Examination system is an app-based application for an organization or institution .it is easy to use and user-friendly. It gathers some people for an Exam which is controlled by an admin.This is an online tool where the system admin can call for an Exam, and the user can see the all details for exam purposes. This is useful for a small institution for that matter any group that is interested to organize it effectively.This project will be managed by an admin. It is the job of the admin to admit and monitor the whole process. When a user logs in to the system. He would only view the details of the applicants. He can't perform any changes. Applicant logging is to apply for the department by attending the MCQ and OPEN HAND QUESTION exam provided online. Applicants can view the result. Teachers can evaluate the answer script of the applicants. Finally, the results of this work show that the integration of an Online Admission Examination system can provide accurate and timely.

Such a system provides various features which include question management, applicant and faculty management

TABLE OF CONTENTS

CHAPTER 4 : REQUIREMENT ENGINEERING 16

CHAPTER 5 : SOFTWARE REQUIREMENT SPECIFICATION 18

CHAPTER 6 : DESIGN SPECIFICATION 21

CHAPTER 7 : SYSTEM IMPLEMENTATION 27

CHAPTER 8 : CODING 34

CHAPTER 9 : INTERFACE DESIGN 124

CHAPTER 10 : TESTING AND VALIDATION 137

CHAPTER 11 : CONCLUSION AND FUTURE IMPLEMENT 139

REFERENCES

Chapter 1
Introduction

1.Introduction

The increasing numbers of students seeking admission in the academic institutes (school, colleges, and universities) are causing tremendous pressure on the administrative body of the institutes to manage and arrange the admission process manually. It is difficult to conduct the process accurately and in timely manner. Hence, the need for online admission is inevitable.

Managing admissions can be a huge task for a university or institutions. There are the university/institution where entire admission process is handled manually, which is very slow and time-consuming. Now, it's high time to leave behind such traditional processes and go with computerized **automated student online admission system**. This system is an online system that can be accessed throughout the organization. The system has three modules. They are

Admin

Teacher

Applicant

1.2 Objectives of the proposed System

This system allows the Admin who can add the admin by using a Gmail and giving a feedback is it accept or decline. Admin can create a the exam and manage whole process and manage them and also accept the valid users request and manage them. A Teacher can evaluate the exam paper and manage the Exam.

The main objective of the project is to be used in any organization in a short range of time. It reduces the time consumption and workload that exist in the current manual system.

1.3 Overview

My Project paper tests about the Online Admission Examination Management System, advantages of using automated Online Admission Examination Management System. I will also discuss the continuing system and based on continuing system why will use the new automated system.My discussion will remain the used and required software and hardware for making and run Online Admission Examination Management System and the how users will use the system.

1.4 Purpose

The main purpose of this system is to schedule and save time for different types of commitments and Exams with people from different backgrounds and different penchants

- A sort list of purpose of the system has given below:
- This system shows that we make the best use of time.
- It will also be exposed to others that you value your own time.
- It is a time effective system.
- It is a cost-effective system.
- Users can expose their very simply
- It is more feasible and secure system

1.5 Continuing System

The Continuing system to evaluate is manual. Manually a date was set for the Exam of any organization with the consent of all. All members of the Exam are invited to the Exam by call or letter. These wastes a lot of time and many users of the Exam are unable to attend the Exam on time due to various reasons. I think it is a good process and time consuming.

1.5.1 Disadvantage of Continuing System

- This system is difficult to arrange Exam manually.
- The continuing system is very time-consuming.
- The manual Exam people may not attend time.
- It creates extirpates sure for the authority.

1.6 Objectives of the proposed System

This system allows the Admin who can add the admin by using a Gmail and giving a feedback is it accept or decline. Admin can create a exam process and manage them and also accept the valid users request and manage them. A Teacher can evaluate the paper and submitted marks to the admin an Exam and manage the Exam. The main objective of the project is to be used in any organization in a short range of time. It reduces the time consumption and workload that exist in the current manual system.

1.6.1 Advantages of Online Exam Management System

- Digital Exams are produced.
- Save time of the
- Flexibility of time.
- Secure because of authentication.
- Exam canceling
- Limitation of Online Exam Management System

1.7 Limitation

- It uses only a small organization.
- This system needs to be properly promoted to users.

1.8 Report

We discuss about my project documentation, initially put the project definition, introduction, purpose and objective as well as the design is followed by the implementation and testing phase. The project has been terminated successfully and the future augmentation of the project also given in this documentation.

Chapter 2

Background Study

2.1 Introduction

Before analyzing the problem and understand the context of this project, We have to analyze the requirement of the task to perform. Our going through some phase in this term. In the first phase we have to study the existing system, compare to the requirement of the new system. Both are equally important to make it better. First activity serves as a basis of giving the functional specifications. Then successful design of the proposed system. It is difficult to understand the properties and requirement of a new system. Also it is equally difficult to think of existing system. So we can go through feasibility study.

2.2 Feasibility Study

Feasibility is measure of how beneficial the development of the application will be to an organization. This is done by studying the existing system and get the ideas about new system.

There are some considerations are important in feasibility analysis. They are Economic, Technical and operational.

2.3 Economic Feasibility

Economic analysis is the most used method for the measurement of effectiveness of a proposed system. we can call it cost benefit analysis, which is a procedure to determine the benefits and saving that are expected from this system compare with costs.

The innovation of the new system has much influence on the economical side. Manual system is costly. And waste of paper.

If users are fluent with the new automated system, it will cut down the costs, which is economically feasible.

2.4 Technical Feasibility

In this new assignment system teacher will manage everything digitally. So technically it will more feasible than manual system.

2.5 Operational Feasibility

Operational feasibility aspects of the project are to be taken as an important part of the project implementation. Some of the important issues raised are to test the operational feasibility of this system or project. Some operations are used to check the system and solve it. The system will be used and work properly if it is being developed and implemented. User authentication are to be safe.

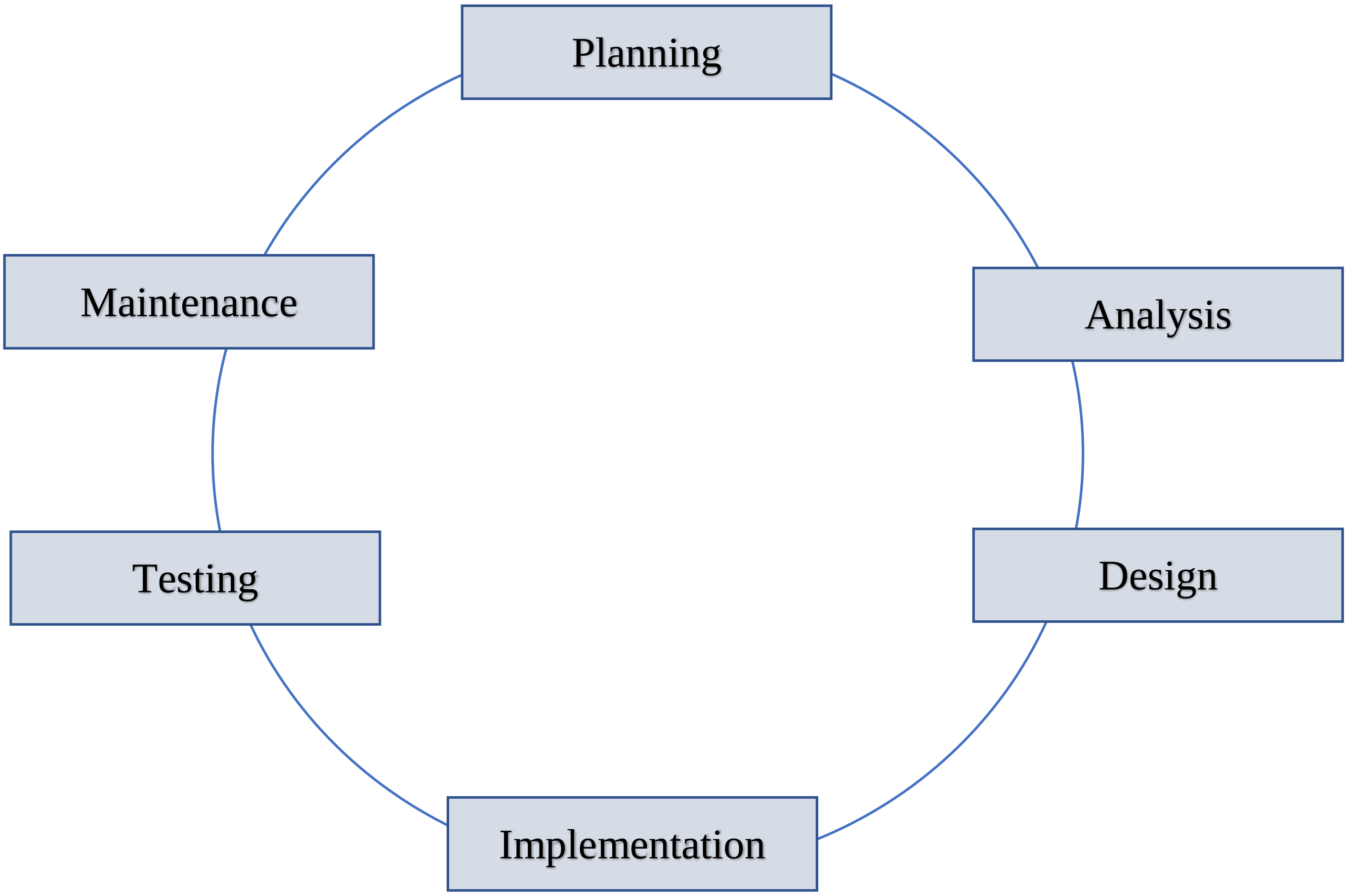
Chapter 3

Methodology and Model implementation

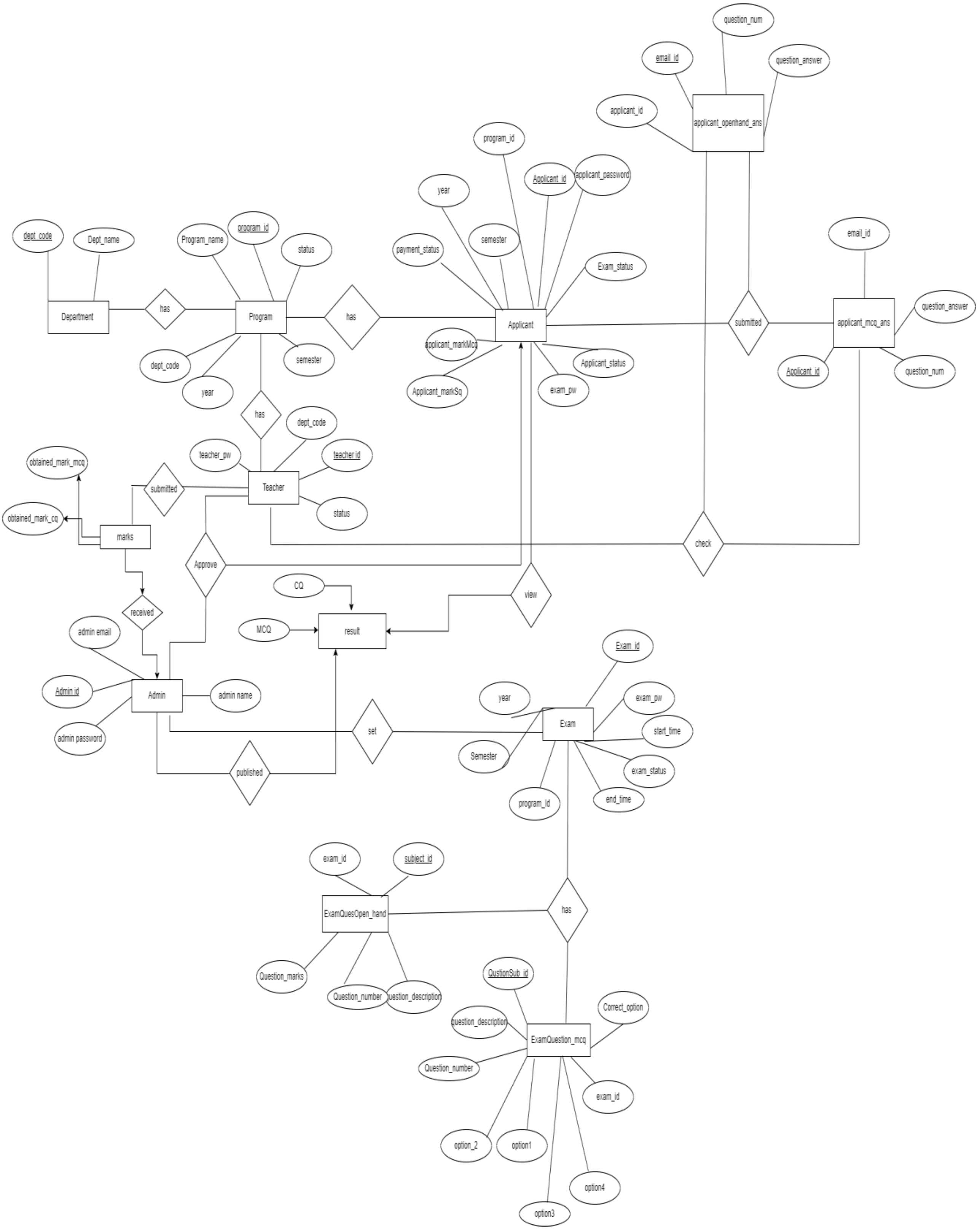
3. Methodology and Model Implementation

3.1 Methodology

To develop this project, we have to follow some methodology. Those methodology mentioned below:



3.2 Logical Data Model



3.3 Research Methodology

The main objective of the admission Exam management System is that it can be used in educational institutions to manage exam in more efficient way.

Online exams are a big-time saver. Not only is there less time between the setting of the “paper”, but it also saves students time by the time it takes to travel to-end from examination locations, then waiting’s for the papers to be out and collected, because most online examinations use auto-grading, teachers don’t have to extra amounts of time makings exam papers, and students get their results almost instantly.

3.4 User Characteristics

3.4.1 Admin.

Admin will do everything

Actually, the whole system is designed for all applicants, teachers.

1.Can Manage Profile

2.Can Manage Department (Add, Delete, Update)

3.Can Manage Programs (Add, Delete, Update)

4.Can Manage Teachers (Add, Delete, Update)

5.Can Manage Applicants (Add, Delete, Update)

3.5 Project Requirement

3.5.1 Admin.

- Signup
- Log in
- Change Email
- Change password
- Add department, manage
- Add program, manage
- Add teacher, manage
- Manage applicant
- Approved applicant
- Approved applicant for exam
- Set MCQ exam
- Set open hand exam
- Set teacher (judge)
- Received Marks
- View results
- Published Result

3.5.2 Teacher

- Log in
- Change password
- View Exam
- View MCQ Exam
- View Open hand Exam
- Submitted marks

3.5.3 Applicant

- Request to admin
- Log in
- Check Exam
- View Result

Chapter 4

Requirements Engineering

4.1 User Requirements

- Admin will approve user and active or inactive him/herself.
- Admin can create an Exam and update Exam information.
- Admin will be able to cancel the test.
- Users are able to join the Exam.

4.2 User Characteristics

There are three types of users. They are-

- Admin
- Teacher
- Student

➢ Admin

Admin can add or remove Teacher and Student and manage the whole process.

➢ Teacher

After exam Teacher will evaluate the papers and send the marks to the admin.

➢ Student

The applicant can login and see the created Exam and exam end result and update his/her profile

Chapter 5

Software Requirement Specification

5. Software Requirement Specification

5.1 Proposed

The main objective of Online Admission examination System is that it can be used to manage small organization. This reduces the time and workload existing in the current manual system. This helps to make Exam's members more punctual.

5.2 Scope

This System has many scopes. They are given below:

- The system operates all the operations and generates result as soon as the test is completed which saves the valuable time of the users.

- It can be used any kind of organization or institution.

- It is consuming time and cost effective.

- User can login through their devices with their email and Password.
- Users can see Exam information from anywhere.
- For being responsive users can easily access from any device and feel comfort

5.3 Specific Requirements

- Admin can log in and view details.
- Log in with validity.
- The Admin maintains the effectiveness of the process .
- Teacher can maintain the effectiveness of the Exam members.
- User can log in and participate in Exam.
- User can view Exam details.

5.4 Software Requirements

- Android Studio

5.5 Technology Requirements

5.5.1 Front End

- Java

5.5.2 Back End

- Java

5.5.3 Database

- Firebase

5.6 Hardware Requirements

This is a app-based application. Users must be needed electronic devices like Computer, Laptop, and Smart Phones that contain internet connection and support any browser like Chrome, Operating, and Firefox etc.

5.7 Conclusion

At this stage, we can understand the software requirements specification for the system. We have arranged all the necessary components for the development of the project in this stage. So, we will have a clear idea about the requirements before designing the project. Thus, our go to the design stage after the Implementation phase of the project.

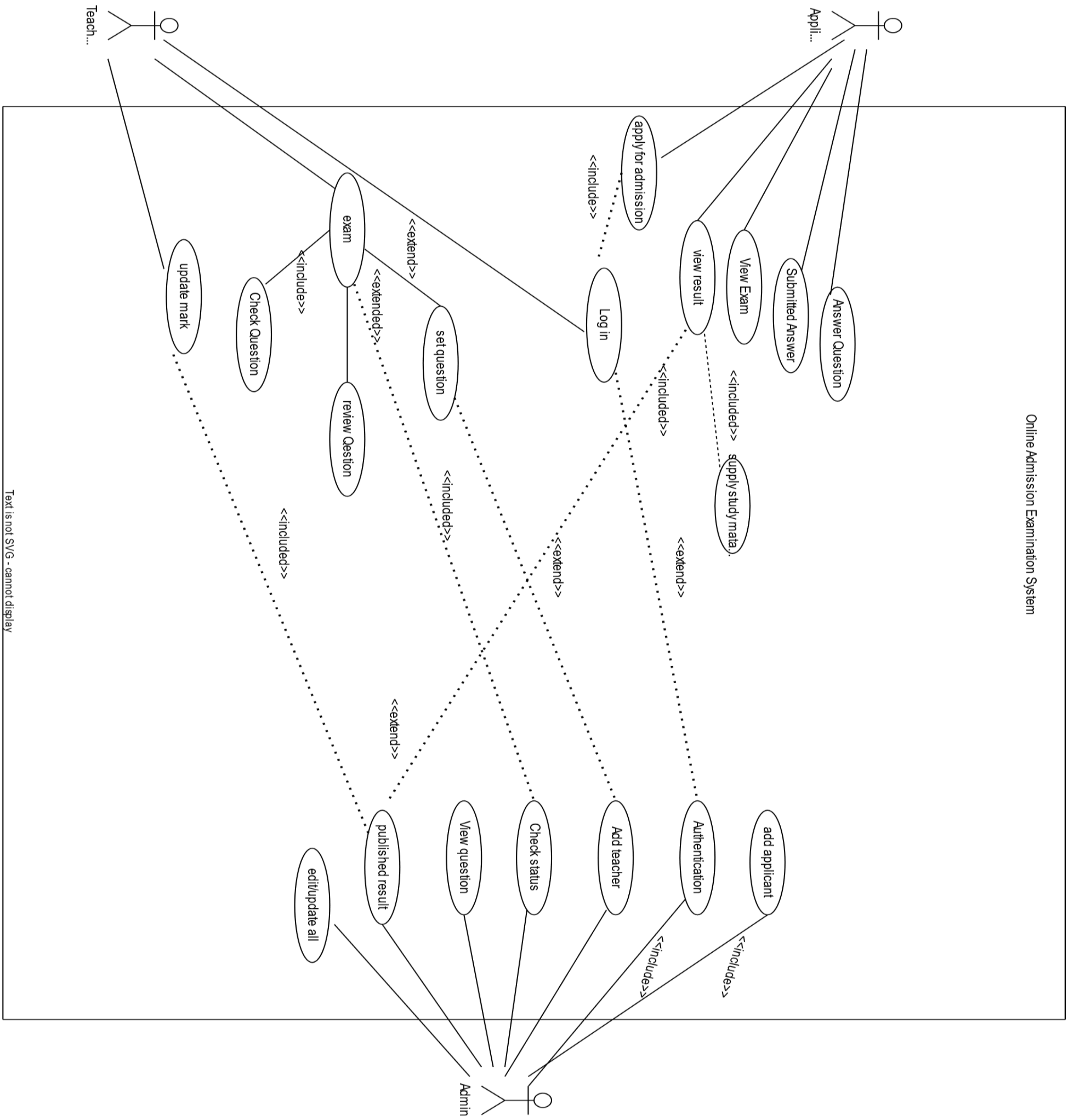
Chapter 6

Design Specification

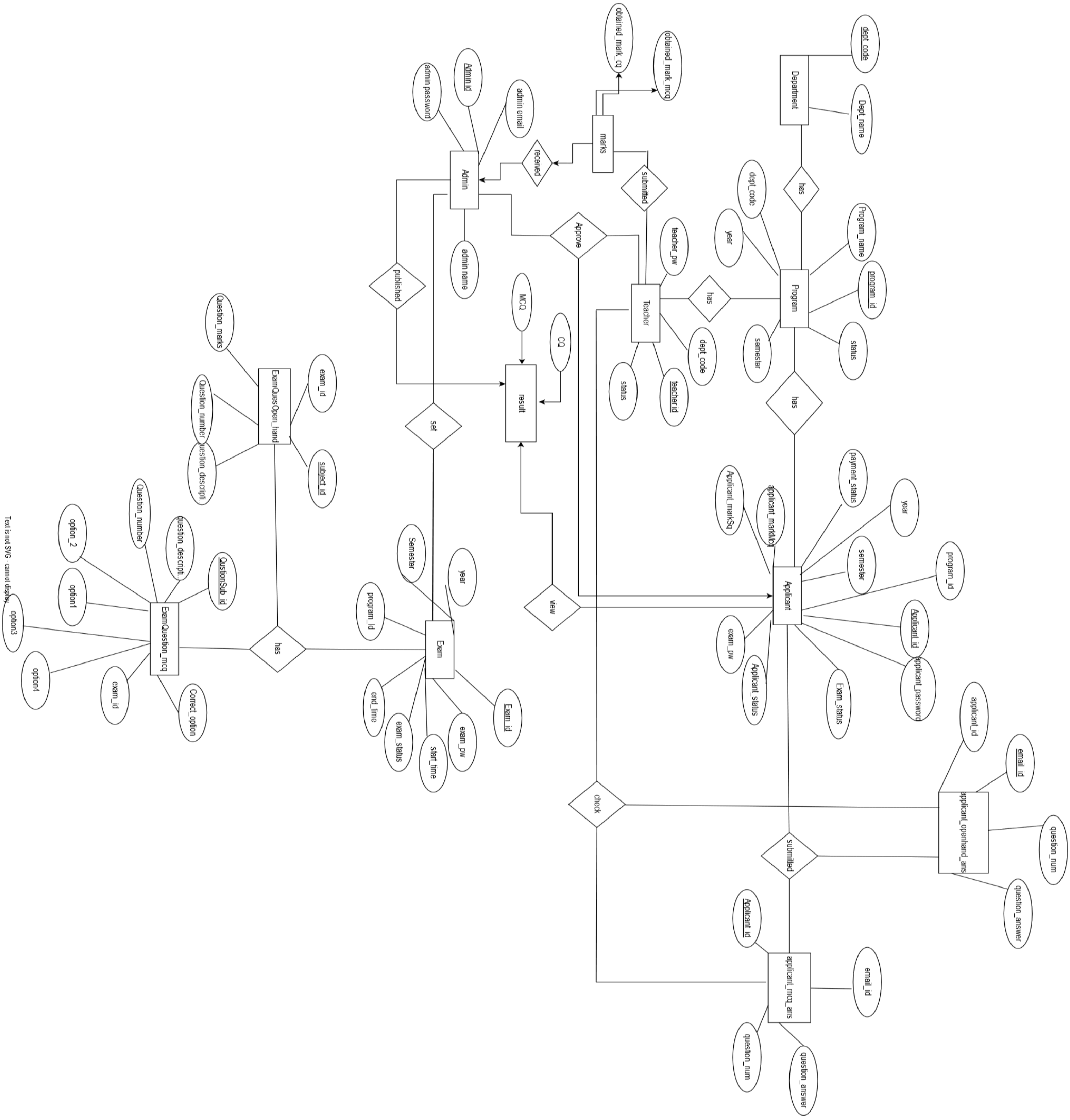
6. Design Specification

The main topics of the chapter is system design and design diagram. We will discuss about the system's design.

6.1 Use case Diagram

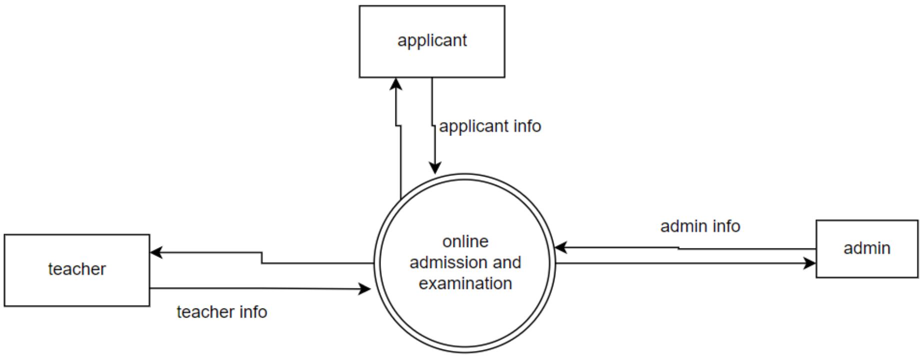


6.2 Entity Relation Diagram

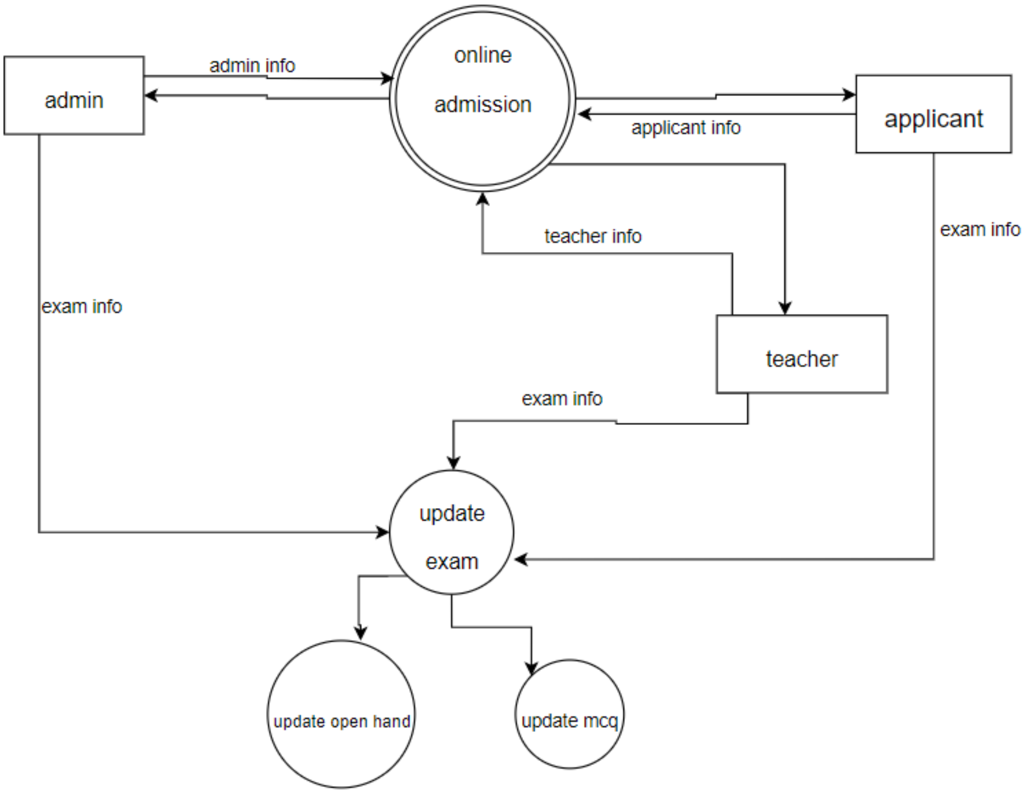

Department
dept_code
Dept_name
has
Program
Program_name
program_id
status
semester
year
dept_code
has
Teacher
teacher_pw
dept_code
teacher id
status
submitted
marks
obtained_mark_cq
obtained_mark_mcq
received
Admin
Admin id
admin password
admin email
admin name
Approve
published
MCQ
CQ
result
has
Applicant
payment_status
year
semester
program_id
Applicant_id
applicant_password
Exam_status
Applicant_status
exam_pw
Applicant_markSq
applicant_markMcq
view
set
Exam
Semester
year
program_id
Exam_id
end_time
exam_status
start_time
exam_pw
has
ExamQuesOpen_hand
exam_id
subject_id
Question_marks
Question_number
Question_descripti
ExamQuestion_mcq
Question_number
question_descripti
QustionSub_id
option_2
option1
option3
option4
exam_id
Correct_option
Text is not SVG - cannot display
check
applicant_openhand_ans
applicant_id
email_id
question_num
question_answer
submitted
applicant_mcq_ans
Applicant_id
email_id
question_num
question_answer

6.3Data Flow Diagram

context diagram/level 0

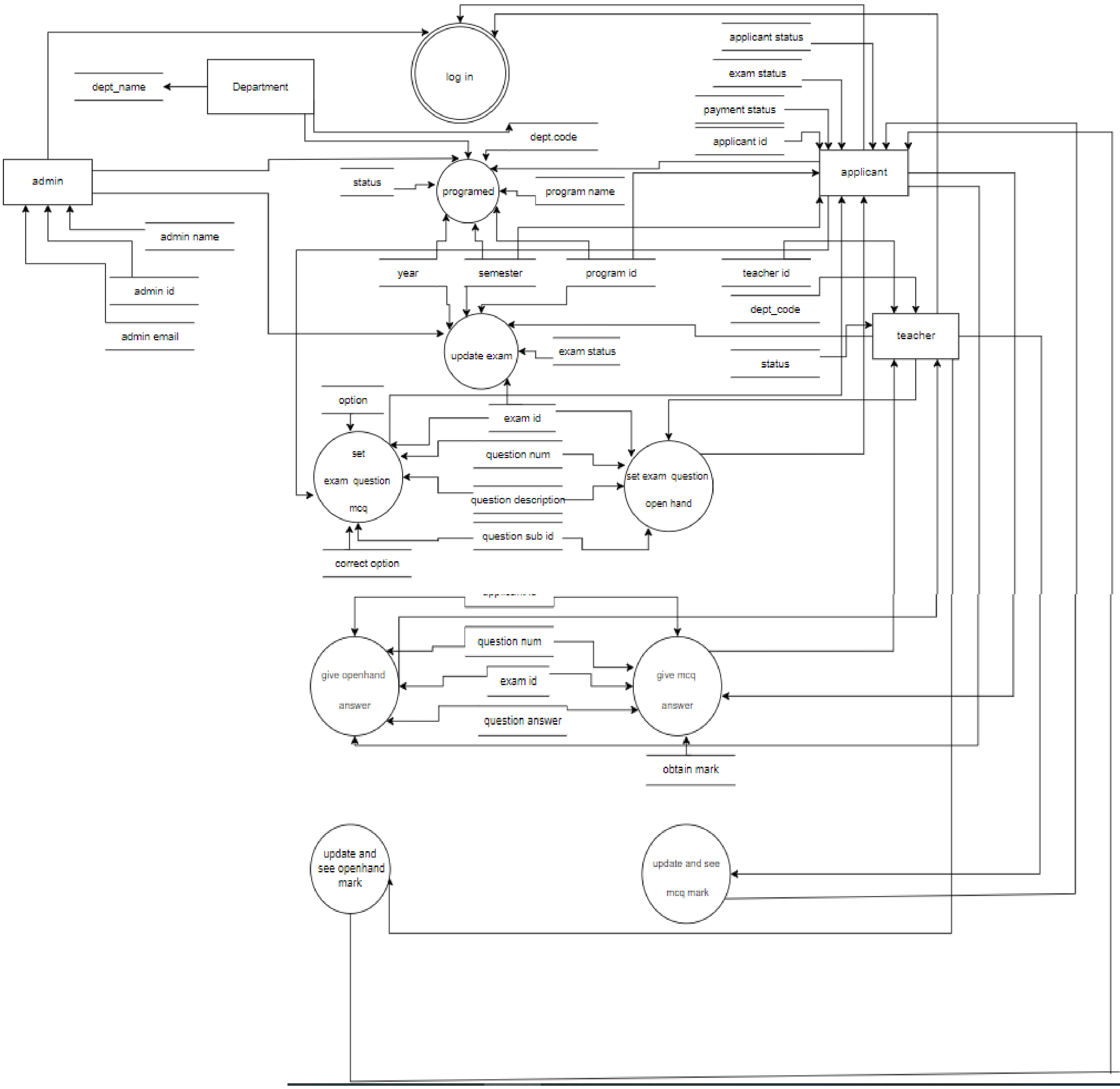


level 1

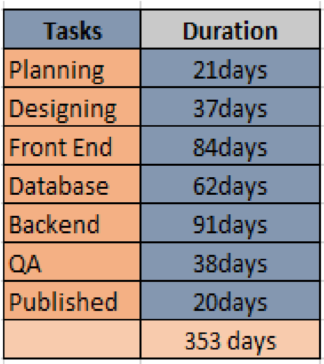


6.3 Data Flow Diagram

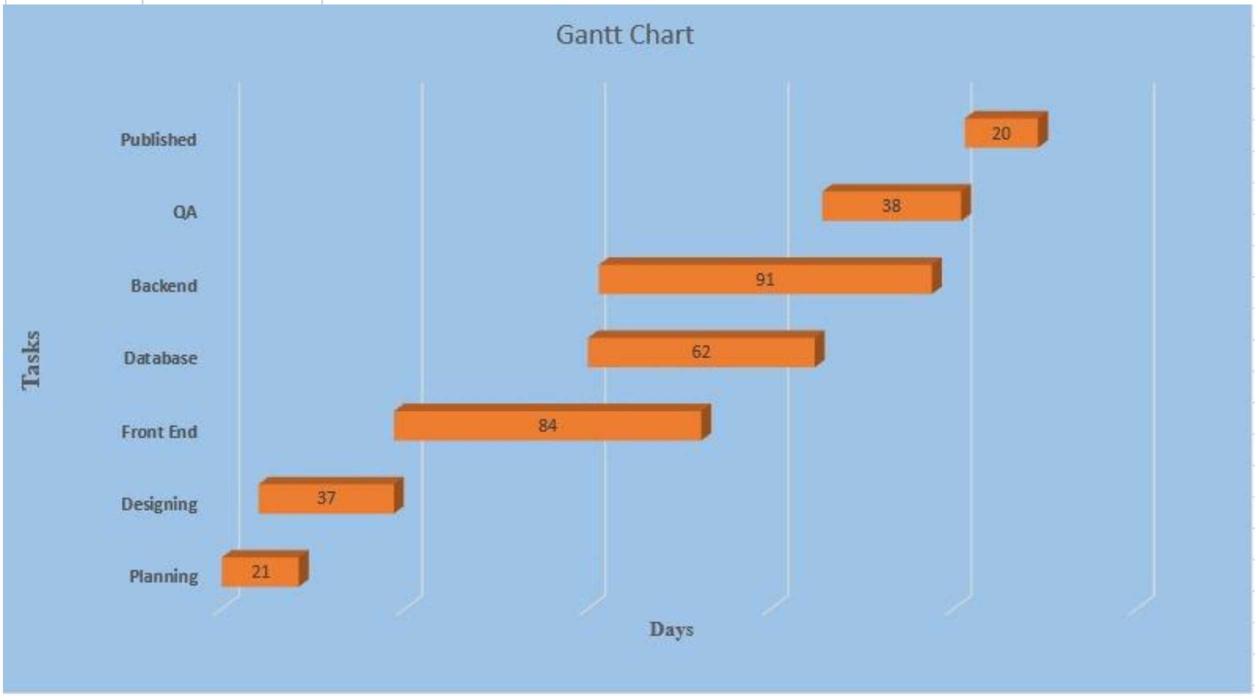
Level 2



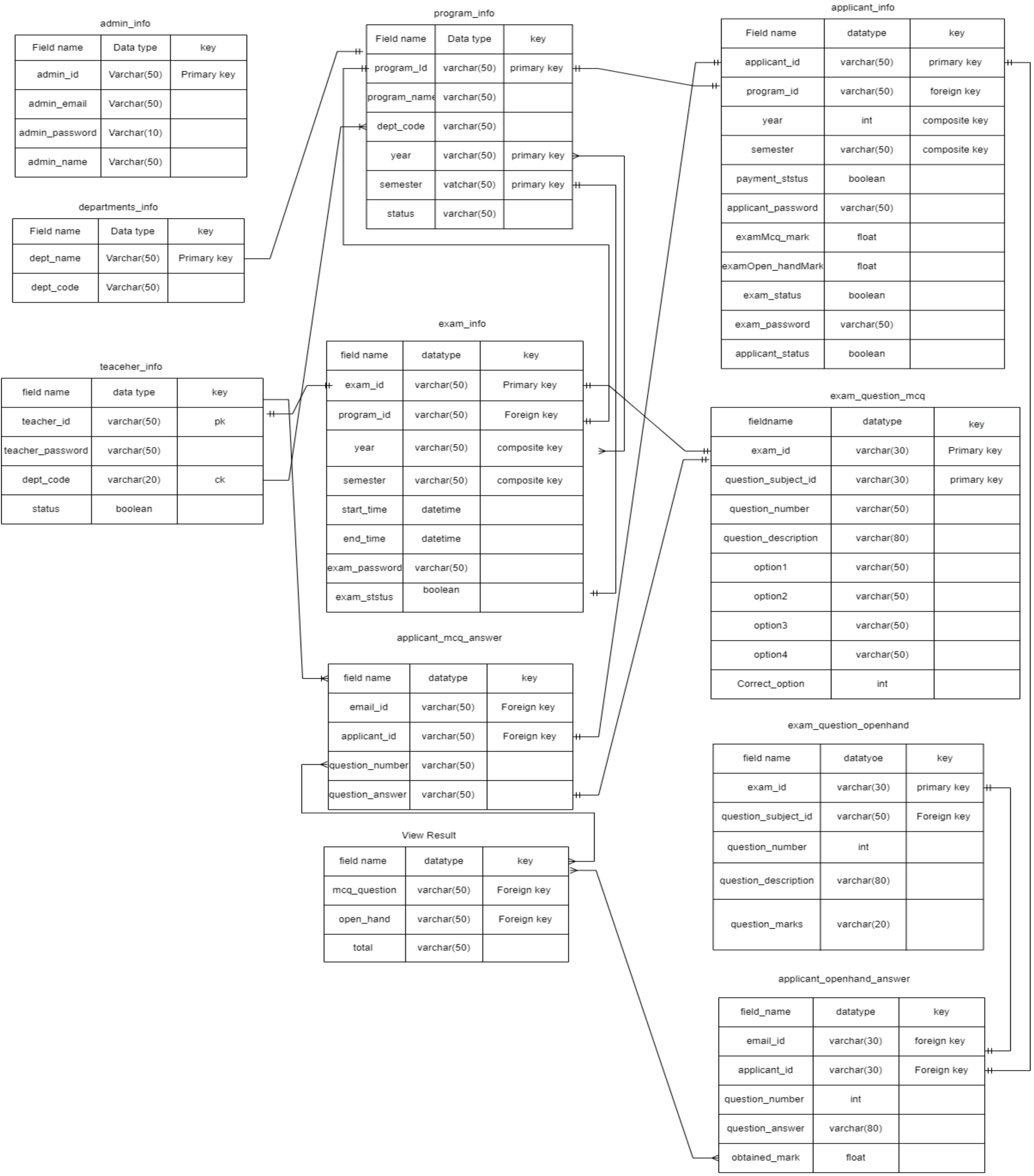
6.4 Gannt Chart



Tasks	Duration
Planning	21days
Designing	37days
Front End	84days
Database	62days
Backend	91days
QA	38days
Published	20days
	353 days



6. 5 Entity Relation Table



admin_info

Field name	Data type	key
admin_id	Varchar(50)	Primary key
admin_email	Varchar(50)	
admin_password	Varchar(10)	
admin_name	Varchar(50)	

departments_info

Field name	Data type	key
dept_name	Varchar(50)	Primary key
dept_code	Varchar(50)	

teaceher_info

field name	data type	key
teacher_id	varchar(50)	pk
teacher_password	varchar(50)	
dept_code	varchar(20)	ck
status	boolean	

program_info

Field name	Data type	key
program_Id	varchar(50)	primary key
program_name	varchar(50)	
dept_code	varchar(50)	
year	varchar(50)	primary key
semester	vatchar(50)	primary key
status	varchar(50)	

exam_info

field name	datatype	key
exam_id	varchar(50)	Primary key
program_id	varchar(50)	Foreign key
year	varchar(50)	composite key
semester	varchar(50)	composite key
start_time	datetime	
end_time	datetime	
exam_password	varchar(50)	
exam_ststus	boolean	

applicant_mcq_answer

field name	datatype	key
email_id	varchar(50)	Foreign key
applicant_id	varchar(50)	Foreign key
question_number	varchar(50)	
question_answer	varchar(50)	

View Result

field name	datatype	key
mcq_question	varchar(50)	Foreign key
open_hand	varchar(50)	Foreign key
total	varchar(50)	

applicant_info

Field name	datatype	key
applicant_id	varchar(50)	primary key
program_id	varchar(50)	foreign key
year	int	composite key
semester	varchar(50)	composite key
payment_ststus	boolean	
applicant_password	varchar(50)	
examMcq_mark	float	
examOpen_handMark	float	
exam_status	boolean	
exam_password	varchar(50)	
applicant_status	boolean	

exam_question_mcq

fieldname	datatype	key
exam_id	varchar(30)	Primary key
question_subject_id	varchar(30)	primary key
question_number	varchar(50)	
question_description	varchar(80)	
option1	varchar(50)	
option2	varchar(50)	
option3	varchar(50)	
option4	varchar(50)	
Correct_option	int	

exam_question_openhand

field name	datatyoe	key
exam_id	varchar(30)	primary key
question_subject_id	varchar(50)	Foreign key
question_number	int	
question_description	varchar(80)	
question_marks	varchar(20)	

applicant_openhand_answer

field_name	datatype	key
email_id	varchar(30)	foreign key
applicant_id	varchar(30)	Foreign key
question_number	int	
question_answer	varchar(80)	
obtained_mark	float	

Chapter 7

System Implementation

7. System Implementation

7.1 Database

First we need to create a data base. From the ER diagram we get the idea about the table and field name of the database. And now we have finalized the data type, default value. We have applied it to the database. And based on that we will maintain the system database. Below we will discuss the data base tables.

This System has tables primarily. They are given below:

1.admin_info

2.program_info

3.applicant _info

4.exam_info

5.exam_question_mcq

6.exam_question_openhand

7.applicant_openhand_answer

8.applicant_mcq_answer

9.teacher_info

10.Department_info

11.view_result _info

7.2 admin_info

Table name : admin_info		
Fieldname	Data type	Comments
admin_name	varchar(50)	Admin name
admin_id	varchar(50)	Primary key
admin_email	varchar(50)	Unique key
admin_password	varchar(100)	Admin's encrypted password

Remarks: Admin will manage the information with admin name, admin id (Primary Key), email and admin password (Encrypted).

7.3 program_info

Table name : program_info		
Fieldname	Data type	Comments
program_name	varchar(50)	Example : B.Sc in CSE
program_id	varchar(30)	Primary key. Example:UG02,PG02
dept_code	varchar(20)	Department's code CSE
year	Int	Current year 2022
semester	varchar(30)	Semester name Summer, Fall, Spring
status	boolean	1= active,0=inactive By default 1.

Remarks: Admin will set available current program and current semester for participating in the exam. Admin will check eligibility for the current program by the status option.

7.4 applicant_info

Table name : applicant _info		
Fieldname	Data type	Comments
applicant_id	varchar(30)	Primary key.
program_id	varchar(30)	Applicant program's id
year	Int	Current year
semester	varchar(30)	Semester name
payment_status	boolean	1= active,0=inactive By default 0.
applicant_password	varchar(100)	Applicant's encrypted password
examMcq_mark	float	Mcq marks
exam_openHand	float	Open hand question marks
exam_status	boolean	1= active,0=inactive By default 0.
applicant_password	varchar(100)	Exam password.
applicant_status	boolean	1= active,0=inactive By default1.

Remarks: Admin will set applicant id, program id, year and semester. Exam status will show the presence of the applicant. Applicant will choice the options.

7.5 exam_info

Table name : exam_info		
Fieldname	Data type	Comments
exam_id	varchar(30)	Applicant id number
program_id	varchar(30)	Primary key
year	int	Year
semester	varchar(30)	Semester
start_time	datetime	Contain both date and time (YYYY-MM-DD HH:MM:SS)
end_time	datetime	Contain both date and time (YYYY-MM-DD HH:MM:SS)
exam_password	varchar(100)	Exam’s encrypted password
exam_status	boolean	1= active,0=inactive By default 0.

Remarks: Admin will set the exam by using applicant exam id, program id, year, exam password and semester. Admin will enable the exam as per given schedule and student can perform the exam.

7.6 exam_question_mcq

Table name : exam_question_mcq		
Fieldname	Data type	Comments
exam_id	varchar(30)	Exam id number
question_subId	varchar(30)	Primary key
question_num	int	Question number
question_description	text	Question description
option1	varchar(50)	Mcq option 1,2,3,4
option2	varchar(50)	Mcq option 1,2,3,4
option3	varchar(50)	Mcq option 1,2,3,4
option4	varchar(50)	Mcq option 1,2,3,4
Correct_option	int	Correct answer

Remarks: Admin will set the MCQ as per question subject ID. It will show the MCQ options to the Applicants.

7.7 exam_question_openhand

Table name :exam_question_openhand		
Fieldname	Data type	Comments
exam_id	varchar(30)	Exam id number
question_subId	varchar(50)	Primary key
question_num	int	Question number
question_description	text	Question description for open-handed.
question_marks	varchar(20)	Marks for each question

Remarks: Admin will set the Openhanded as per question subject ID. It will show the Openhanded to the Applicants.

7.8 applicant_mcq_answer

Table name : applicant_mcq_answer		
Fieldname	Data type	Comments
exam_id	varchar(30)	Foreign key of exam info
applicant_id	varchar(30)	Primary key
question_num	int	Question number By default each mcq marks 1
question_answer	varchar(10)	Applicant mcq answer

Remarks: Applicant will submit response to the MCQ answers.

7.9 applicant_openhand_answer

Table name : applicant_openhand_answer		
Fieldname	Data type	Comments
exam_id	varchar(30)	Foreign key of exam info
applicant_id	varchar(30)	Primary key
question_num	int	Question Number
question_answer	text	Check the answer
obtained_mark	float	Obtain marks for open hand question

Remarks: Applicant will submit response to the openhand answers**.**

7.10 teacher_info

Table name : teacher_info		
Fieldname	Data type	Comments
teacher_id	varchar(30)	Primary key
teacher_password	varchar(100)	Teacher’s encrypted password
dept_code	varchar(20)	Teacher department’s code
status	boolean	1= active,0=inactive By default 1.

Remarks: Admin will set the teacher’s ID, teacher’s password, and department code. Admin will set the subject wise questions. Admin will set the status of the availability of the teacher.

7.11 Department_info

Table name : Department_info		
Fieldname	Data type	Comments
Dept_name	varchar(30)	Department name . Example-CSE
Dept_code	varchar(100)	Department code. Example=UG02

Remarks: Admin will set the department’s name and code.

7.12 view_result _info

Table name : view_result _info		
Fieldname	Data type	Comments
obtained_mark_MCQ	float	Obtain marks for mcq question
obtained_mark_open_hand	float	Obtain marks for open hand question

Remarks: Teacher will send the marks admin,then admin will published the obtained marks.

Chapter 8

Coding

8. Coding
8.1 admin log in activity

```
package com.example.exam_admin_app;

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

import android.content.Intent;
import android.os.Bundle;
import android.text.InputType;
import android.view.KeyEvent;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.OnFailureListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.auth.AuthResult;
import com.google.firebase.auth.FirebaseAuth;
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.Map;
import java.util.regex.Pattern;

public class LoginActivity extends AppCompatActivity implements View.OnClickListener {

    EditText name, email, password;

    Button enter;

    String Name, Email, Password;

    FirebaseAuth firebaseAuth;

    DatabaseReference databaseReference;
```

```
TextView signup, show_password;

int ok = 0;

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

    name = findViewById(R.id.username);
    email = findViewById(R.id.email);
    password = findViewById(R.id.password);
    show_password = findViewById(R.id.show_password);

    Name = name.getText().toString().trim();
    Email = email.getText().toString().trim();
    Password = password.getText().toString().trim();
    signup = findViewById(R.id.signup);

    signup.setOnClickListener(this);

    enter = (Button) findViewById(R.id.login);

    enter.setOnClickListener(this);

    show_password.setOnClickListener(this);

    name.setOnKeyListener(new View.OnKeyListener() {
        @Override
        public boolean onKey(View v, int keyCode, KeyEvent event) {

            if ((event.getAction() == KeyEvent.ACTION_DOWN) &&
                    (keyCode == KeyEvent.KEYCODE_ENTER)) {
                // Perform action on key press
                Name = name.getText().toString();

                return true;
            }

            return false;
        }
    });

    email.setOnKeyListener(new View.OnKeyListener() {
        @Override
        public boolean onKey(View v, int keyCode, KeyEvent event) {

            if ((event.getAction() == KeyEvent.ACTION_DOWN) &&
                    (keyCode == KeyEvent.KEYCODE_ENTER)) {
                // Perform action on key press
                Email = email.getText().toString();

                return true;
            }

            return false;
        }
    });

    password.setOnKeyListener(new View.OnKeyListener() {
        @Override
        public boolean onKey(View v, int keyCode, KeyEvent event) {

            if ((event.getAction() == KeyEvent.ACTION_DOWN) &&
                    (keyCode == KeyEvent.KEYCODE_ENTER)) {
                // Perform action on key press
                Password = password.getText().toString();

                return true;
            }

            return false;
        }
    });


    databaseReference = FirebaseDatabase.getInstance().getReference("Admins");
    firebaseAuth = FirebaseAuth.getInstance();

}

@Override
public void onClick(View view) {

    Email = email.getText().toString().trim();
    Name = name.getText().toString().trim();
    Password = password.getText().toString().trim();

    if(view.getId() == enter.getId()) {

        //name.setText("");
        //email.setText("");
        //password.setText("");

        Password = encripted(Password);

        //Toast.makeText(getApplicationContext(), "pass : " + Password,
Toast.LENGTH_SHORT).show();
        //Toast.makeText(getApplicationContext(), "Name : " + Name,
Toast.LENGTH_SHORT).show();
```

```
        //Toast.makeText(getApplicationContext(), "email : " + Email,
Toast.LENGTH_SHORT).show();

        if (!isEmailValid(Email)) {

            Toast.makeText(getApplicationContext(), "Email is not valid",
Toast.LENGTH_SHORT).show();
            return;

        }

        if (!isPasswordValid(Password)) {

            Toast.makeText(getApplicationContext(), "Password is not valid",
Toast.LENGTH_SHORT).show();
            return;

        }

        if (!isValidUserName(Name)) {

            Toast.makeText(getApplicationContext(), "Name is not valid",
Toast.LENGTH_SHORT).show();
            return;

        }

        firebaseAuth = FirebaseAuth.getInstance();

        firebaseAuth.signInWithEmailAndPassword(Email, Password).addOnCompleteListener(new
OnCompleteListener<AuthResult>() {
            @Override
            public void onComplete(@NonNull Task<AuthResult> task) {

                databaseReference.addValueEventListener(new ValueEventListener() {
                    @Override
                    public void onDataChange(@NonNull DataSnapshot snapshot) {

                        for (DataSnapshot dataSnapshot : snapshot.getChildren()) {

                            Map<String, String> map = (Map<String, String>) dataSnapshot.getValue();

                            String userName = map.get("name");
                            String userPassword = map.get("password");
                            //String UserSpetiality = map.get("subject");
                            String userEmail = map.get("email");
                            String ownId = map.get("id");
                            //String verfiedCode = map.get("verified_id");
                        if (userName.equalsIgnoreCase(Name) && Email.equalsIgnoreCase(userEmail) &&
userPassword.equalsIgnoreCase(Password)) {

                            Intent intent = new Intent(getApplicationContext(), Admin_Dashboard.class);

                            Bundle bundle = new Bundle();

                            bundle.putString("name", userName);
                            bundle.putString("password", userPassword);
                            //  bundle.putString("subject", UserSpetiality);
                            bundle.putString("email", userEmail);
                            bundle.putString("id", ownId);
                            //bundle.putString("verified_code", verfiedCode);

                            intent.putExtra("bundle", bundle);

                            startActivity(intent);

                            break;

                        }

                    }

                }

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

                }
            });

        }
    }).addOnFailureListener(new OnFailureListener() {
        @Override
        public void onFailure(@NonNull Exception e) {

            Toast.makeText(getApplicationContext(), "Log in failed", Toast.LENGTH_LONG).show();

        }
    });

    //
    // Toast.makeText(getApplicationContext(), "Log in is not sucessfull",
Toast.LENGTH_SHORT).show();

  } else if(view.getId() == signup.getId()) {

    Intent intent = new Intent(LoginActivity.this, Sign_Up.class);
    startActivity(intent);
```

```
        } else if(view.getId() == show_password.getId()) {

            if(ok == 0) {

                password.setInputType(InputType.TYPE_CLASS_TEXT);

                ok = 1;

            } else {

                password.setInputType(InputType.TYPE_TEXT_VARIATION_PASSWORD |
InputType.TYPE_CLASS_TEXT);

                ok = 0;

            }

        }

    }

    // A placeholder username validation check
    private boolean isEmailValid(String email) {
        if (email == null) {
            return false;
        }
        String emailRegex = "^[a-zA-Z0-9_+&*-]+(?:\\."+
                "[a-zA-Z0-9_+&*-]+)*@" +
                "(?:[a-zA-Z0-9-]+\\.)+[a-z" +
                "A-Z]{2,7}$";

        Pattern pat = Pattern.compile(emailRegex);
        if (email == null)
            return false;
        return pat.matcher(email).matches();
    }

    private boolean isValidUserName(String username) {

        return !username.isEmpty();

    }

    // A placeholder password validation check
    private boolean isPasswordValid(String password) {

        char a[] = password.toCharArray();

        int upper = 0, lower = 0, digit = 0, symbol = 0;
        for (char i : a) {
            if (Character.isUpperCase(i)) {
                ++upper;
            } else if (Character.isLowerCase(i)) {
                ++lower;
            } else if (Character.isDigit(i)) {
                ++digit;
            } else {
                ++symbol;
            }
        }
        return password != null && password.trim().length() >= 8 && upper >= 1 && lower >= 1 && digit >= 1 && symbol >= 1;
    }
    private String encripted(String pass) {
        char a[] = pass.toCharArray();
        int n = a.length;
        for(int i = 0; i < n; ++i) {
            if(Character.isLowerCase(a[i])) {
                a[i] = Character.toUpperCase(a[i]);
            } else if(Character.isUpperCase(a[i])) {
                a[i] = Character.toLowerCase(a[i]);
            } else if(Character.isDigit(a[i])) {
                int digit = (a[i] - '0') ;
                digit = Math.max(0, digit - 1);
```

```
              a[i] = (digit + "")
                      .charAt(0);

          }

      }

      return new String(a);

  }

  private String decripted(String pass) {

      char a[] = pass.toCharArray();

      int n = a.length;

      for(int i = 0; i < n; ++i) {

          if(Character.isLowerCase(a[i])) {

              a[i] = Character.toUpperCase(a[i]);

          } else if(Character.isUpperCase(a[i])) {

              a[i] = Character.toLowerCase(a[i]);

          } else if(Character.isDigit(a[i])) {

              int digit = (a[i] - '0') ;

              digit = Math.min(9, digit + 1);

              a[i] = (digit + "")
                      .charAt(0);

          }

      }

      return new String(a);
  }

  @Override
  public void onBackPressed() {

      onStart();

      super.onBackPressed();
}
```

8.2 Admin dashboard activity

```
package com.example.exam_admin_app;

import androidx.appcompat.app.AppCompatActivity;

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

import de.hdodenhof.circleimageview.CircleImageView;

public class Admin_Dashboard extends AppCompatActivity implements View.OnClickListener {

    String admin_name, admin_email, admin_id, admin_password;

    TextView change_email, change_password;
    CircleImageView programm_add, programm_details;
    CircleImageView teacher_add, teacher_details;
    CircleImageView department_add, department_details;

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

        Intent intent = getIntent();

        Bundle bundle = intent.getBundleExtra("bundle");

        admin_name = bundle.getString("name");
        admin_email = bundle.getString("email");
        admin_id = bundle.getString("id");
        admin_password = bundle.getString("password");

        change_email = findViewById(R.id.change_email);
        change_password = findViewById(R.id.change_password);

        programm_add = findViewById(R.id.programm_add);
        programm_details = findViewById(R.id.programm_details);

        teacher_add = findViewById(R.id.teacher_add);
        teacher_details = findViewById(R.id.teacher_details);

        department_add = findViewById(R.id.department_add);
        department_details = findViewById(R.id.department_details);
    change_password.setOnClickListener(this);
    change_email.setOnClickListener(this);
    change_email.setOnClickListener(this);
    change_password.setOnClickListener(this);
    programm_add.setOnClickListener(this);
    programm_details.setOnClickListener(this);
    teacher_add.setOnClickListener(this);
    teacher_details.setOnClickListener(this);
    department_details.setOnClickListener(this);
    department_add.setOnClickListener(this);

}

@Override
public void onClick(View view) {

    if(view.getId() == change_email.getId()) {

        Intent intent = new Intent(Admin_Dashboard.this, Change_Account.class);

        Bundle bundle = new Bundle();

        bundle.putString("admin_name", admin_name);
        bundle.putString("admin_email", admin_email);
        bundle.putString("admin_id", admin_id);
        bundle.putString("admin_password", admin_password);
        bundle.putString("task", "change");

        intent.putExtra("bundle", bundle);

        startActivity(intent);

    } else if(view.getId() == change_password.getId()) {

        Intent intent = new Intent(Admin_Dashboard.this, Change_Account.class);

        Bundle bundle = new Bundle();

        bundle.putString("admin_name", admin_name);
        bundle.putString("admin_email", admin_email);
        bundle.putString("admin_id", admin_id);
        bundle.putString("admin_password", admin_password);
        bundle.putString("task", "change");

        intent.putExtra("bundle", bundle);

        startActivity(intent);

    } else if(view.getId() == teacher_add.getId()) {
```

```
    Intent intent = new Intent(Admin_Dashboard.this, Teacher.class);

    Bundle bundle = new Bundle();

    bundle.putString("name", " ");
    //bundle.putString("semister", " ");
    //bundle.putString("year", " ");
    bundle.putString("task", "add");
    bundle.putString("course_id", " ");
    bundle.putString("dept_code", " ");
    bundle.putString("status", " ");
    bundle.putString("password", " ");
    bundle.putString("admin_name", admin_name);
    bundle.putString("admin_email", admin_email);
    bundle.putString("admin_id", admin_id);

    intent.putExtra("bundle", bundle);

    startActivity(intent);

} else if (view.getId() == teacher_details.getId()) {

    Intent intent = new Intent(Admin_Dashboard.this, Teacher_Seen.class);

    Bundle bundle = new Bundle();

    bundle.putString("admin_name", admin_name);
    bundle.putString("admin_email", admin_email);
    bundle.putString("admin_id", admin_id);

    intent.putExtra("bundle", bundle);

    startActivity(intent);

} else if(view.getId() == programm_add.getId()) {

    Intent intent = new Intent(Admin_Dashboard.this, Program.class);

    Bundle bundle = new Bundle();

    bundle.putString("name", " ");
    bundle.putString("semister", " ");
    bundle.putString("year", " ");
    bundle.putString("task", "add");
    bundle.putString("course_id", " ");
    bundle.putString("dept_code", " ");
    bundle.putString("status", " ");
    bundle.putString("admin_name", admin_name);
    bundle.putString("admin_email", admin_email);
    bundle.putString("admin_id", admin_id);
    intent.putExtra("bundle", bundle);

    startActivity(intent);

} else if (view.getId() == programm_details.getId()) {

    Intent intent = new Intent(Admin_Dashboard.this, Detalls_Activity.class);

    Bundle bundle = new Bundle();

    bundle.putString("admin_name", admin_name);
    bundle.putString("admin_email", admin_email);
    bundle.putString("admin_id", admin_id);
    bundle.putString("password", admin_password);

    intent.putExtra("bundle", bundle);

    startActivity(intent);

} else if(view.getId() == department_add.getId()) {

    Intent intent = new Intent(Admin_Dashboard.this, Department.class);

    Bundle bundle = new Bundle();

    bundle.putString("admin_name", admin_name);
    bundle.putString("admin_email", admin_email);
    bundle.putString("admin_id", admin_id);
    bundle.putString("task", "add");
    bundle.putString("dept_code", " ");
    bundle.putString("dept_name", " ");
    bundle.putString("status", " ");
    bundle.putString("manual_dept_code", " ");

    intent.putExtra("bundle", bundle);

    startActivity(intent);

} else if(view.getId() == department_details.getId()) {

    Intent intent = new Intent(Admin_Dashboard.this, Department_Details.class);

    Bundle bundle = new Bundle();

    bundle.putString("admin_name", admin_name);
    bundle.putString("admin_email", admin_email);
    bundle.putString("admin_id", admin_id);

    intent.putExtra("bundle", bundle);
```

```
            startActivity(intent);

        }

    }
```

8.3 Admin details activity

```
package com.example.exam_admin_app;

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

import android.content.DialogInterface;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.Spinner;
import android.widget.TextView;
import android.widget.Toast;

import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.OnFailureListener;
import com.google.android.gms.tasks.Task;
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.List;
import java.util.Map;

import de.hdodenhof.circleimageview.CircleImageView;

public class Detalls_Activity extends AppCompatActivity implements View.OnClickListener {

    String admin_name, admin_email, admin_id, manual_dept_code;

    Spinner spinner;

    Programm_Adapter programm_adapter;
```

```
CircleImageView edit, delete;

String course_name, course_id, dept_code, semister, password;
int year;
boolean status;

List<Programm_info> list;

List<String> programms;

DatabaseReference databaseReference;

TextView mcq, view_result, manage_applicant, exam, open_hand;

Programm_info programm_info;

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

    Intent intent = getIntent();

    Bundle bundle = intent.getBundleExtra("bundle");

    admin_name = bundle.getString("admin_name");
    admin_email = bundle.getString("admin_email");
    admin_id = bundle.getString("admin_id");
    password = bundle.getString("password");

    programms = new ArrayList<>();

    spinner = findViewById(R.id.list);

    mcq = findViewById(R.id.mcq);
    view_result = findViewById(R.id.view_result);
    exam = findViewById(R.id.manage_exam);
    manage_applicant = findViewById(R.id.manage_applicant);
    open_hand = findViewById(R.id.open_hand_manage_exam);
    edit = findViewById(R.id.edit);
    delete = findViewById(R.id.delete);

    mcq.setOnClickListener(this);
    view_result.setOnClickListener(this);
    exam.setOnClickListener(this);
    manage_applicant.setOnClickListener(this);
    open_hand.setOnClickListener(this);
    delete.setOnClickListener(this);
    edit.setOnClickListener(this);
```

```
        databaseReference =
FirebaseDatabase.getInstance().getReference("Programm").child(admin_name);

        list = new ArrayList<>();

        programm_adapter = new Programm_Adapter(Detalls_Activity.this, list);


//programm_adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);

        spinner.setOnItemSelectedListener(new AdapterView.OnItemSelectedListener() {
            @Override
            public void onItemSelected(AdapterView<?> adapterView, View view, int i, long l) {

                try {

                    programm_info = list.get(i - 1);

                    course_id = programm_info.course_id;
                    year = programm_info.year;
                    course_name = programm_info.course_name;
                    semister = programm_info.semister;
                    status = programm_info.status;
                    dept_code = programm_info.dept_code;

                    //Toast.makeText(getApplicationContext(), "name : " + course_name,
Toast.LENGTH_SHORT).show();

                    /*Intent intent = new Intent(Detalls_Activity.this, Program.class);

                    intent.putExtra("task", "update");
                    intent.putExtra("name", programm_info.course_name);
                    intent.putExtra("year", programm_info.year + "");
                    intent.putExtra("course_id", programm_info.course_id);
                    intent.putExtra("dept_code", programm_info.dept_code);
                    intent.putExtra("semister", programm_info.semister);
                    intent.putExtra("status", programm_info.status + "");
                    intent.putExtra("admin_name", admin_name);
                    intent.putExtra("admin_email", admin_email);
                    intent.putExtra("admin_id", admin_id);

                    startActivity(intent);*/

                } catch (Exception e) {

                }

            }

            @Override
            public void onNothingSelected(AdapterView<?> adapterView) {

            }
        });

    }

    ArrayAdapter aa;

    @Override
    public void onClick(View view) {

        if(mcq.getId() == view.getId()) {

            /*

            Intent intent = this.getIntent();

        Bundle bundle = intent.getBundleExtra("bundle");

        Name = bundle.getString("name");//admin name
        Password = bundle.getString("password");// admin password
        Id = bundle.getString("id");// admin id
        Email = bundle.getString("email");// admin email
        Subject = bundle.getString("subject");//programm name
        Verified_Id = bundle.getString("verified_code");// programm code

             */

            if(course_id == null || course_name == null) {

                Toast.makeText(getApplicationContext(), "please select an programm",
Toast.LENGTH_LONG).show();
                return;

            }

            Intent intent = new Intent(Detalls_Activity.this, MCQ.class);

            Bundle bundle = new Bundle();

            bundle.putString("name", admin_name);
            bundle.putString("id", admin_id);
            bundle.putString("email", admin_email);
            bundle.putString("subject", course_name);
            bundle.putString("verified_code", course_id);
            bundle.putString("password", password);

            intent.putExtra("bundle", bundle);
            startActivity(intent);

        } else if(view.getId() == view_result.getId()) {

            if(programm_info == null) {

                Toast.makeText(getApplicationContext(), "please select a subject",
Toast.LENGTH_LONG).show();
                return;

            }

            Intent intent = new Intent(Detalls_Activity.this, Exam_Chooser.class);

            Bundle bundle = new Bundle();

            bundle.putString("admin_name", admin_name);
            bundle.putString("dept_code", programm_info.dept_code);
            bundle.putString("programm", programm_info.course_name);

            if(!programm_info.status) {

                Toast.makeText(getApplicationContext(), "programm is not active now",
Toast.LENGTH_LONG).show();

                return;

            }

            bundle.putString("status", programm_info.status + "");

            intent.putExtra("bundle", bundle);

            startActivity(intent);

        } else if(view.getId() == exam.getId()) {

            AlertDialog.Builder  builder = new AlertDialog.Builder(Detalls_Activity.this);

            builder.setNegativeButton("MCQ", new DialogInterface.OnClickListener() {
                @Override
                public void onClick(DialogInterface dialog, int which) {

                    /*

            Intent intent = this.getIntent();

        Bundle bundle = intent.getBundleExtra("bundle");

        Name = bundle.getString("name");//admin name
        Password = bundle.getString("password");// admin password
        Id = bundle.getString("id");// admin id
        Email = bundle.getString("email");// admin email
        Subject = bundle.getString("subject");//programm name
        Verified_Id = bundle.getString("verified_code");// programm code

            */

                if(course_id == null || course_name == null) {

                    Toast.makeText(getApplicationContext(), "please select an programm",
Toast.LENGTH_LONG).show();
                    return;

                }

                Intent intent = new Intent(Detalls_Activity.this, MCQ_Exam_Code.class);

                Bundle bundle = new Bundle();

                bundle.putString("name", admin_name);
                bundle.putString("id", admin_id);
                bundle.putString("email", admin_email);
                bundle.putString("subject", course_name);
                bundle.putString("verified_code", course_id);
                bundle.putString("password", password);

                intent.putExtra("bundle", bundle);

                startActivity(intent);

            }
        });

        builder.setPositiveButton("Open Hand", new DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int which) {

                /*

            Intent intent = this.getIntent();

        Bundle bundle = intent.getBundleExtra("bundle");

        Name = bundle.getString("name");// admin name
        Password = bundle.getString("password");//admin password
        Id = bundle.getString("id");//admin id
        Email = bundle.getString("email");//admin email
        Subject = bundle.getString("subject");//subject/ programm
        Verified_Id = bundle.getString("verified_code");//programm id
```

```
    */

        if(course_name == null) {

           Toast.makeText(getApplicationContext(), "please select an subject",
Toast.LENGTH_LONG).show();

           return;

        }

        Intent intent = new Intent(Detalls_Activity.this, CQ_Exam_Code.class);

        Bundle bundle = new Bundle();

        bundle.putString("name", admin_name);
        bundle.putString("password", password);
        bundle.putString("id", admin_id);
        bundle.putString("email", admin_email);
        bundle.putString("subject", course_name);
        bundle.putString("verified_code", course_id);

        intent.putExtra("bundle", bundle);

        startActivity(intent);

      }
    });

    builder.setCancelable(true);

    builder.setMessage("What type of exam you want to update?");

    builder.setTitle("Exam choose");

    builder.create();

    builder.show();

  } else if(view.getId() == manage_applicant.getId()) {

    Intent intent = new Intent(Detalls_Activity.this, request_Student_Department.class);

    Bundle bundle = new Bundle();

    bundle.putString("admin_name", admin_name);
    bundle.putString("admin_email", admin_email);
    bundle.putString("admin_id", admin_id);
    bundle.putString("password", password);
```

```
            intent.putExtra("bundle", bundle);

            startActivity(intent);

        } else if(open_hand.getId() == view.getId()) {

            /*

            Intent intent = this.getIntent();

        Bundle bundle = intent.getBundleExtra("bundle");

        Name = bundle.getString("name");// admin name
        Password = bundle.getString("password");//admin password
        Id = bundle.getString("id");//admin id
        Email = bundle.getString("email");//admin email
        Subject = bundle.getString("subject");//subject/ programm
        Verified_Id = bundle.getString("verified_code");//programm id

             */

            if(course_name == null) {

                Toast.makeText(getApplicationContext(), "please select an subject",
Toast.LENGTH_LONG).show();

                return;

            }

            Intent intent = new Intent(Detalls_Activity.this, CQ.class);

            Bundle bundle = new Bundle();

            bundle.putString("name", admin_name);
            bundle.putString("password", password);
            bundle.putString("id", admin_id);
            bundle.putString("email", admin_email);
            bundle.putString("subject", course_name);
            bundle.putString("verified_code", course_id);

            intent.putExtra("bundle", bundle);

            startActivity(intent);

        } else if(view.getId() == edit.getId()) {

            if(course_id == null || course_name == null || dept_code == null || semister == null) {
                Toast.makeText(getApplicationContext(), "please select an item in the list",
Toast.LENGTH_SHORT).show();
                return;

            }

            //Toast.makeText(getApplicationContext(), "course_name : " + course_name,
Toast.LENGTH_SHORT).show();

            //Programm_info programm_info = new Programm_info(course_name,  course_id, dept_code,
semister,  year,  status);

            Intent intent = new Intent(Detalls_Activity.this, Program.class);

            Bundle bundle = new Bundle();

            /*intent.putExtra("task", "update");
            intent.putExtra("name", course_name);
            intent.putExtra("year", year + "");
            intent.putExtra("course_id", course_id+"");
            intent.putExtra("dept_code", dept_code+"");
            intent.putExtra("semister", semister + "");
            intent.putExtra("status", status+"");
            intent.putExtra("admin_name", admin_name);
            intent.putExtra("admin_email", admin_email);
            intent.putExtra("admin_id", admin_id);*/

            bundle.putString("task", "update");
            bundle.putString("name", course_name);
            bundle.putString("year", year + "");
            bundle.putString("course_id", course_id+"");
            bundle.putString("dept_code", dept_code+"");
            bundle.putString("semister", semister + "");
            bundle.putString("status", status+"");
            bundle.putString("admin_name", admin_name);
            bundle.putString("admin_email", admin_email);
            bundle.putString("admin_id", admin_id);

            intent.putExtra("bundle", bundle);

            startActivity(intent);

        } else if(view.getId() == delete.getId()) {

            try {

                databaseReference.child(course_id).removeValue().addOnCompleteListener(new
OnCompleteListener<Void>() {
                    @Override
                    public void onComplete(@NonNull Task<Void> task) {
```

```
                    Toast.makeText(getApplicationContext(), "Data is deleted",
Toast.LENGTH_LONG).show();

                    onStart();

                    /*programms.remove((course_name));

                    String s[] = new String[programms.size()];

                    int index = 0;

                    for(String i : programms) {

                        s[index++] = i;

                    }

                    ArrayAdapter aa = new
ArrayAdapter(Detalls_Activity.this,android.R.layout.simple_spinner_item,s);
                    aa.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);

                    spinner.setAdapter(null);

                    spinner.setAdapter(aa);*/

                }
            }).addOnFailureListener(new OnFailureListener() {
                @Override
                public void onFailure(@NonNull Exception e) {

                    Toast.makeText(getApplicationContext(), "Data is not deleted",
Toast.LENGTH_LONG).show();

                }
            });

        } catch (Exception e) {


        }

    }

}


@Override
protected void onStart() {
    list.clear();
    programms.clear();

    programms.add("List");

    spinner = findViewById(R.id.list);

    //programms.add("List");

    databaseReference.addValueEventListener(new ValueEventListener() {
        @Override
        public void onDataChange(@NonNull DataSnapshot snapshot) {

            for(DataSnapshot dataSnapshot : snapshot.getChildren()) {

                Map<String, String> map = (Map<String, String>) dataSnapshot.getValue();

                String course_name = map.get("course_name");
                String course_id = map.get("course_id");
                String semlster = map.get("semister");
                String dept_code = map.get("dept_code");
                boolean status = Boolean.parseBoolean(String.valueOf(map.get("status")) );
                int year = Integer.parseInt( String.valueOf(map.get("year")) );

                programms.add(semlster + " " + year + " " + course_name);

                Programm_info programm_info = new Programm_info(course_name, course_id,
dept_code,  semlster,  year,  status);

                list.add(programm_info);

            }

            try {

                String s[] = new String[programms.size()];

                int index = 0;

                for(String i : programms) {

                    s[index++] = i;

                }

                aa = new ArrayAdapter(Detalls_Activity.this,android.R.layout.simple_spinner_item,s);
                aa.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);

                //spinner = findViewById(R.id.list);
                spinner.setAdapter(aa);

            } catch (Exception e) {

            }

        }

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

            Toast.makeText(getApplicationContext(), "failed to load data",
Toast.LENGTH_SHORT).show();

        }
    });

    super.onStart();
}

}
```

8.4 Exam approved

```
package com.example.exam_admin_app;

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

import android.content.DialogInterface;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.view.ViewStub;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.Spinner;
import android.widget.Toast;

import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.OnFailureListener;
import com.google.android.gms.tasks.Task;
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.List;
import java.util.Map;

public class Exam_Approved extends AppCompatActivity implements View.OnClickListener{

    String admin_name, department;
    String admin_email, admin_id, password, dept_name;

    DatabaseReference databaseReference, databaseReference1;

    Map<String , Boolean> maping;

    List<student_info> list;

    List<String> student_names;

    boolean overall = false;

    Spinner spinner;

    ArrayAdapter<String> aa;

    Button enter, remove;

    student_info student_info;

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

        Intent intent = getIntent();

        Bundle bundle = intent.getBundleExtra("bundle");

        admin_name = bundle.getString("admin_name");
        department = bundle.getString("department");

        databaseReference =
FirebaseDatabase.getInstance().getReference("Approved_Student").child(admin_name);

        databaseReference1 =
FirebaseDatabase.getInstance().getReference("Exam_Permission").child(admin_name).child(department
);

        maping = new HashMap<>();
        list = new ArrayList<>();
        student_names = new ArrayList<>();

        aa = new
ArrayAdapter<>(Exam_Approved.this,android.R.layout.simple_spinner_item,student_names);
        aa.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);

        spinner = findViewById(R.id.spinner);
        remove = findViewById(R.id.remove);
        enter = findViewById(R.id.enter);

        try {

            spinner.setOnItemSelectedListener(new AdapterView.OnItemSelectedListener() {
                @Override
                public void onItemSelected(AdapterView<?> parent, View view, int position, long id) {

                    if(position - 1 >= 0) {

                        student_info = list.get(position - 1);

                        AlertDialog.Builder builder = new AlertDialog.Builder(Exam_Approved.this);

                        builder.setTitle("approved_surity");

                        builder.setCancelable(true);

                        builder.setMessage("active?");

                        builder.setPositiveButton("yes", new DialogInterface.OnClickListener() {
                            @Override
                            public void onClick(DialogInterface dialog, int which) {

                                Toast.makeText(getApplicationContext(), "please enter the button to finalize it",
Toast.LENGTH_LONG).show();

                                overall = true;

                            }
                        });

                        builder.setNegativeButton("no", new DialogInterface.OnClickListener() {
                            @Override
                            public void onClick(DialogInterface dialog, int which) {

                                Toast.makeText(getApplicationContext(), "please enter the button to finalize it",
Toast.LENGTH_LONG).show();

                                overall = false;
                            }
                        });

                        builder.create();
                        builder.show();

                    }

                }

                @Override
                public void onNothingSelected(AdapterView<?> parent) {

                }
            });

        } catch (Exception e) {


        }

        enter.setOnClickListener(this);
        remove.setOnClickListener(this);

    }

    @Override
    protected void onStart() {

        list.clear();
        student_names.clear();

        maping.clear();

        student_names.add("List");

        databaseReference1.addValueEventListener(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot snapshot) {

                for(DataSnapshot dataSnapshot : snapshot.getChildren()) {

                    Map<String, String> map = (Map<String, String>) dataSnapshot.getValue();

                    String approved_id = map.get("approved_id");
                    String department_name = map.get("department_name");
                    String status = String.valueOf(map.get("status"));
                    String student_name = map.get("student_name");
                    String student_email = map.get("student_email");
                    String student_request_id = map.get("student_request_id");

                    student_info student_info = new student_info(student_name,  department_name,
student_request_id, student_email, approved_id, Boolean.parseBoolean(status));

                    maping.put(student_info.student_name, student_info.status);

                }

            }

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

            }
        });

        databaseReference.addValueEventListener(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot snapshot) {

                for(DataSnapshot dataSnapshot : snapshot.getChildren()) {

                    Map<String, String> map = (Map<String, String>) dataSnapshot.getValue();

                    String approved_id = map.get("approved_id");
                    String department_name = map.get("department_name");
                    String status = String.valueOf(map.get("status"));
                    String student_name = map.get("student_name");
                    String student_email = map.get("student_email");
                    String student_request_id = map.get("student_request_id");

                    student_info student_info = new student_info(student_name,  department_name,
student_request_id, student_email, approved_id, Boolean.parseBoolean(status));

                    list.add(student_info);

                    student_names.add(student_info.student_name + " " +
((maping.containsKey(student_info.student_name) ?  (maping.get(student_info.student_name) ? "active"
: "de active") : "de active" )));

                }

                spinner.setAdapter(aa);

            }

            @Override
            public void onCancelled(@NonNull DatabaseError error) {
            }
        });

        super.onStart();
    }

    @Override
    public void onClick(View v) {

        if(v.getId() == enter.getId()) {

            if(student_info != null) {

                student_info.status = overall;


databaseReference1.child(student_info.approved_id).setValue(student_info).addOnCompleteListener(ne
w OnCompleteListener<Void>() {
                    @Override
                    public void onComplete(@NonNull Task<Void> task) {

                        Toast.makeText(getApplicationContext(), "data is seted",
Toast.LENGTH_LONG).show();

                        onStart();

                    }
                }).addOnFailureListener(new OnFailureListener() {
                    @Override
                    public void onFailure(@NonNull Exception e) {

                        Toast.makeText(getApplicationContext(), "failed to update",
Toast.LENGTH_LONG).show();

                    }
                });

            } else {

                Toast.makeText(getApplicationContext(), "please select an student",
Toast.LENGTH_LONG).show();

            }

        } else if(v.getId() == remove.getId()) {

            if(student_info != null) {
```

```
databaseReference.child(student_info.approved_id).removeValue().addOnCompleteListener(new
OnCompleteListener<Void>() {
                    @Override
                    public void onComplete(@NonNull Task<Void> task) {

                        Toast.makeText(getApplicationContext(), "data is removed successfully",
Toast.LENGTH_LONG).show();
                        onStart();

                    }
                }).addOnFailureListener(new OnFailureListener() {
                    @Override
                    public void onFailure(@NonNull Exception e) {

                        Toast.makeText(getApplicationContext(), "data is not removed",
Toast.LENGTH_LONG).show();

                    }
                });

            } else {

                Toast.makeText(getApplicationContext(), "please select a student",
Toast.LENGTH_LONG).show();

            }

        }

    }

}
```

8.5 Result

```
package com.example.exam_admin_app;

public class Result {

    boolean publish;
    String programm, admin_name, exam_code;
    int set;

    public Result(boolean publish, String programm, String admin_name, int set) {
        this.publish = publish;
        this.programm = programm;
        this.admin_name = admin_name;
        this.set = set;
}

public Result(boolean publish, String programm, String admin_name, String exam_code, int set) {
   this.publish = publish;
   this.programm = programm;
   this.admin_name = admin_name;
   this.exam_code = exam_code;
   this.set = set;

}

public String getExam_code() {
   return exam_code;
}

public void setExam_code(String exam_code) {
   this.exam_code = exam_code;
}

public Result() {
}

public boolean isPublish() {
   return publish;
}

public void setPublish(boolean publish) {
   this.publish = publish;
}

public String getProgramm() {
   return programm;
}

public void setProgramm(String programm) {
   this.programm = programm;
}

public String getAdmin_name() {
   return admin_name;
}

public void setAdmin_name(String admin_name) {
   this.admin_name = admin_name;
}

public int getSet() {
   return set;
}
```

```
    public void setSet(int set) {
        this.set = set;
    }
}
```

8.6 Question

```
package com.example.exam_admin_app;

public class Question {

    String question, option1, option2, option3, option4, answer, id, exam_set, link;

    int number;

    public Question(String question, String option1, String option2, String option3, String option4, String
answer, String id, String exam_set) {


        try {

            /*question.replaceAll("//.", " ");
            question.replaceAll("\\$", "");
            question.replaceAll("//#", "");
            String s = question.replaceAll("\\[", "(");
            question = s;
            question.replaceAll("\\]", ")");*/

            char a[] = question.toCharArray();

            StringBuilder sb = new StringBuilder();

            for(char i : a) {

                if(i == '.') {

                    sb.append(" ");

                } else if(i == '$') {

                    sb.append("");

                } else if(i == '[') {

                    sb.append("(");

                } else if(i == ']') {
                    sb.append(")");

                } else {

                    sb.append(i);

                }

            }

            question = sb.toString();

            System.out.println(question);

        } catch (Exception e) {

        }

        this.question = question;
        this.option1 = option1;
        this.option2 = option2;
        this.option3 = option3;
        this.option4 = option4;
        this.answer = answer;
        this.id = id;
        this.exam_set = exam_set;
    }

    public Question(String question, String option1, String option2, String option3, String option4, String
answer, String id, String exam_set, String link) {


        try {

            /*question.replaceAll("//.", " ");
            question.replaceAll("\\$", "");
            question.replaceAll("//#", "");
            String s = question.replaceAll("\\[", "(");
            question = s;
            question.replaceAll("\\]", ")");*/

            char a[] = question.toCharArray();

            StringBuilder sb = new StringBuilder();

            for(char i : a) {

                if(i == '.') {

                    sb.append(" ");
                } else if(i == '$') {

                    sb.append("");

                } else if(i == '[') {

                    sb.append("(");

                } else if(i == ']') {

                    sb.append(")");

                } else {

                    sb.append(i);

                }

            }

            question = sb.toString();

            System.out.println(question);

        } catch (Exception e) {


        }

        this.question = question;
        this.option1 = option1;
        this.option2 = option2;
        this.option3 = option3;
        this.option4 = option4;
        this.answer = answer;
        this.id = id;
        this.exam_set = exam_set;
        this.link = link;
    }

    public String getLink() {
        return link;
    }

    public int getNumber() {
        return number;
    }

    public void setNumber(int number) {
        this.number = number;
    }

    public void setLink(String link) {
        this.link = link;
    }

    public Question(String exam_set) {
        this.exam_set = exam_set;
    }

    public Question() {
    }

    public String getQuestion() {
        return question;
    }

    public void setQuestion(String question) {


        try {

            /*question.replaceAll("//.", " ");
            question.replaceAll("\\$", "");
            question.replaceAll("//#", "");
            String s = question.replaceAll("\\[", "(");
            question = s;
            question.replaceAll("\\]", ")");*/

            char a[] = question.toCharArray();

            StringBuilder sb = new StringBuilder();

            for(char i : a) {

                if(i == '.') {

                    sb.append(" ");

                } else if(i == '$') {

                    sb.append("");

                } else if(i == '[') {

                    sb.append("(");

                } else if(i == ']') {
```

```
                sb.append(")");

            } else {

                sb.append(i);

            }

        }

        question = sb.toString();

        System.out.println(question);

    } catch (Exception e) {

    }

    this.question = question;
}

public String getOption1() {
    return option1;
}

public void setOption1(String option1) {
    this.option1 = option1;
}

public String getOption2() {
    return option2;
}

public void setOption2(String option2) {
    this.option2 = option2;
}

public String getOption3() {
    return option3;
}

public void setOption3(String option3) {
    this.option3 = option3;
}

public String getOption4() {
    return option4;
}

public void setOption4(String option4) {
        this.option4 = option4;
    }

    public String getAnswer() {
        return answer;
    }

    public void setAnswer(String answer) {
        this.answer = answer;
    }

    public String getId() {
        return id;
    }

    public void setId(String id) {
        this.id = id;
    }

    public String getExam_set() {
        return exam_set;
    }

    public void setExam_set(String exam_set) {
        this.exam_set = exam_set;
    }
}
```

8.7 Teacher home

```
package com.example.exam_teacher_app;

import androidx.appcompat.app.AppCompatActivity;

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

public class Home extends AppCompatActivity implements View.OnClickListener {

    String Name, Password, admin_name, dept_code, teacher_id, teacher_password, status, 
teacher_name;

    Button change_pass, view_exam;

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

        Intent intent = getIntent();

        Bundle bundle = intent.getBundleExtra("bundle");

        //Name = bundle.getString("name");
        //Password = bundle.getString("password");
        admin_name = bundle.getString("admin_name");
        dept_code = bundle.getString("dept_code");

        teacher_id = bundle.getString("teacher_id");
        teacher_password = bundle.getString("password");
        status = bundle.getString("status");
        teacher_name = bundle.getString("name");

        change_pass = findViewById(R.id.change_pass);
        view_exam = findViewById(R.id.view_exam);

        change_pass.setOnClickListener(this);
        view_exam.setOnClickListener(this);

    }

    @Override
    public void onClick(View v) {

        if(v.getId() == change_pass.getId()) {

            Intent intent = new Intent(Home.this, LoginActivity.class);

            Bundle bundle = new Bundle();

            bundle.putString("name", teacher_name);
            bundle.putString("password", teacher_password);
            bundle.putString("admin_name", admin_name);
            bundle.putString("dept_code", dept_code);
            bundle.putString("status", status);
            bundle.putString("teacher_id", teacher_id);

            intent.putExtra("bundle", bundle);

            startActivity(intent);

        } else if(v.getId() == view_exam.getId()) {

            Intent intent = new Intent(Home.this, Exam_Chooser.class);

            Bundle bundle = new Bundle();
            bundle.putString("name", teacher_name);
            bundle.putString("password", teacher_password);
            bundle.putString("admin_name", admin_name);
            bundle.putString("dept_code", dept_code);
            bundle.putString("status", status);
            bundle.putString("teacher_id", teacher_id);

            intent.putExtra("bundle", bundle);

            startActivity(intent);

        }

    }
}
```

8.8 Teacher log in

```
package com.example.exam_teacher_app;

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

import android.content.Intent;
import android.os.Bundle;
import android.text.InputType;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.OnFailureListener;
import com.google.android.gms.tasks.Task;
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.List;
import java.util.Map;
import java.util.regex.Pattern;
```

```
public class LoginActivity extends AppCompatActivity implements View.OnClickListener {

    EditText name, password;

    Button enter;

    TextView signup, show_password;

    int ok = 0;

    String Name, Password, admin_name, dept_code, teacher_id, teacher_password, status,
teacher_name;

    DatabaseReference databaseReference;

    boolean new_entry = false;

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

        Intent intent = getIntent();

        Bundle bundle = intent.getBundleExtra("bundle");

        //Name = bundle.getString("name");
        //Password = bundle.getString("password");
        admin_name = bundle.getString("admin_name");
        teacher_id = bundle.getString("teacher_id");
        teacher_password = bundle.getString("password");
        status = bundle.getString("status");
        teacher_name = bundle.getString("name");
        dept_code = bundle.getString("dept_code");

        name = findViewById(R.id.name);
        password = findViewById(R.id.password);
        show_password = findViewById(R.id.show_password);

        show_password.setOnClickListener(this);

        if(teacher_name != null && !teacher_name.equals(" ")) {

            name.setText(teacher_name);

            new_entry = true;

        }

        if(teacher_password != null && !teacher_password.equalsIgnoreCase(" ")) {
            password.setText(teacher_password);

            new_entry = true;

        }

        Name = name.getText().toString().trim();
        Password = password.getText().toString().trim();

        databaseReference = FirebaseDatabase.getInstance().getReference("Teacher_Info");

        enter = findViewById(R.id.enter);

        enter.setOnClickListener(this);

    }

    @Override
    public void onClick(View v) {

        Name = name.getText().toString().trim();
        Password = password.getText().toString().trim();

        if(v.getId() == enter.getId()) {

            if(isPasswordValid(Password) && isValidUserName(Name)) {

                if(new_entry) {

                    //Name = teacher_name;

                    teacher_info teacher_info = new teacher_info(Name, admin_name, teacher_id, dept_code,
teacher_password, status);


databaseReference.child(admin_name).child(teacher_id).setValue(teacher_info).addOnCompleteListene
r(new OnCompleteListener<Void>() {
                        @Override
                        public void onComplete(@NonNull Task<Void> task) {

                            Toast.makeText(getApplicationContext(), "data changes successfully",
Toast.LENGTH_LONG).show();

                        }
                    }).addOnFailureListener(new OnFailureListener() {
                        @Override
                        public void onFailure(@NonNull Exception e) {
                            Toast.makeText(getApplicationContext(), "failed to store",
Toast.LENGTH_LONG).show();

                        }
                    });

                    return;

                }

                databaseReference.addValueEventListener(new ValueEventListener() {
                    @Override
                    public void onDataChange(@NonNull DataSnapshot snapshot) {

                        for (DataSnapshot dataSnapshot : snapshot.getChildren()) {

                            Map<String, Map<String, String>> list = (Map<String, Map<String, String>>)
dataSnapshot.getValue();

                            boolean find = false;

                            //for (List<Map<String, String>> j : list.values()) {

                                for (Map<String, String> i : list.values()) {

                                    admin_name = i.get("admin_name");
                                    dept_code = i.get("dept_code");
                                    teacher_name = i.get("name");
                                    status = i.get("status") + "";
                                    teacher_id = i.get("teacher_id");
                                    teacher_password = i.get("teacher_password");

                                    //Toast.makeText(getApplicationContext(), "teacher : " + teacher_name,
Toast.LENGTH_LONG).show();

                                    if (!status.equalsIgnoreCase("active")) {

                                        continue;

                                    }

                                    if (teacher_name.equals(Name) && teacher_password.equals(Password)) {

                                        find = true;

                                        Intent intent = new Intent(LoginActivity.this, Home.class);

                                        Bundle bundle = new Bundle();

                                        bundle.putString("name", teacher_name);
                                        bundle.putString("password", teacher_password);
                                        bundle.putString("admin_name", admin_name);
                                        bundle.putString("dept_code", dept_code);
                                        bundle.putString("status", status);
                                        bundle.putString("teacher_id", teacher_id);

                                        intent.putExtra("bundle", bundle);

                                        startActivity(intent);

                                        break;

                                    }

                                }

                            }



                        //}
                    }
                    @Override
                    public void onCancelled(@NonNull DatabaseError error) {

                    }
                });

            } else {

                Toast.makeText(getApplicationContext(), "info you entered is not valid. Your password must
have more than or equal to 8 length and have 1 upper and lower character one symbol one digit",
Toast.LENGTH_LONG).show();

            }

        } else if(v.getId() == show_password.getId()) {

            if(ok == 0) {

                password.setInputType(InputType.TYPE_CLASS_TEXT);

                ok = 1;

            } else {

                password.setInputType(InputType.TYPE_TEXT_VARIATION_PASSWORD |
InputType.TYPE_CLASS_TEXT);

                ok = 0;
            }

        }

    }

    // A placeholder username validation check
    private boolean isEmailValid(String email) {
        if (email == null) {
            return false;
        }
        String emailRegex = "^[a-zA-Z0-9_+&*-]+(?:\\."+
                "[a-zA-Z0-9_+&*-]+)*@" +
                "(?:[a-zA-Z0-9-]+\\.)+[a-z" +
                "A-Z]{2,7}$";

        Pattern pat = Pattern.compile(emailRegex);
        if (email == null)
            return false;
        return pat.matcher(email).matches();
    }

    private boolean isValidUserName(String username) {

        return !username.isEmpty();

    }

    // A placeholder password validation check
    private boolean isPasswordValid(String password) {

        char a[] = password.toCharArray();

        int upper = 0, lower = 0, digit = 0, symbol = 0;

        for (char i : a) {

            if (Character.isUpperCase(i)) {

                ++upper;

            } else if (Character.isLowerCase(i)) {

                ++lower;

            } else if (Character.isDigit(i)) {

                ++digit;
            } else {

                ++symbol;

            }

        }

        return password != null && password.trim().length() >= 8 && upper >= 1 && lower >= 1 &&
digit >= 1 && symbol >= 1;

    }

    private String encripted(String pass) {

        char a[] = pass.toCharArray();

        int n = a.length;

        for(int i = 0; i < n; ++i) {

            if(Character.isLowerCase(a[i])) {

                a[i] = Character.toUpperCase(a[i]);

            } else if(Character.isUpperCase(a[i])) {

                a[i] = Character.toLowerCase(a[i]);

            } else if(Character.isDigit(a[i])) {

                int digit = (a[i] - '0') ;

                digit = Math.max(0, digit - 1);

                a[i] = (digit + "")
                        .charAt(0);

            }

        }

        return new String(a);

    }

    private String decripted(String pass) {

        char a[] = pass.toCharArray();
        int n = a.length;

        for(int i = 0; i < n; ++i) {

            if(Character.isLowerCase(a[i])) {

                a[i] = Character.toUpperCase(a[i]);

            } else if(Character.isUpperCase(a[i])) {

                a[i] = Character.toLowerCase(a[i]);

            } else if(Character.isDigit(a[i])) {

                int digit = (a[i] - '0') ;

                digit = Math.min(9, digit + 1);

                a[i] = (digit + "")
                        .charAt(0);

            }

        }

        return new String(a);

    }

    @Override
    public void onBackPressed() {

        onStart();

        super.onBackPressed();

    }


}
```

8.9 Teacher check paper

```
package com.example.exam_teacher_app;

import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.os.Bundle;
import android.os.CountDownTimer;
import android.widget.TextView;

//import com.example.exam_admin_app.ui.login.LoginActivity;

import de.hdodenhof.circleimageview.CircleImageView;

public class MainActivity extends AppCompatActivity {

    CircleImageView circleImageView;
    TextView textView;

    String Name = " ", Password = " ", admin_name = " ", dept_code = " ", teacher_id = " ",
teacher_password = " ", status = " ", teacher_name = " ";

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

        circleImageView = findViewById(R.id.logo_image);
        textView = findViewById(R.id.message);

        CountDownTimer countDownTimer = new CountDownTimer(5000, 1000) {
            @Override
            public void onTick(long l) {

            }

            @Override
            public void onFinish() {

                Intent intent = new Intent(MainActivity.this, LoginActivity.class);

                Bundle bundle = new Bundle();

                bundle.putString("name", teacher_name);
                bundle.putString("password", teacher_password);
                bundle.putString("admin_name", admin_name);
                bundle.putString("dept_code", dept_code);
                bundle.putString("status", status);
                bundle.putString("teacher_id", teacher_id);

                intent.putExtra("bundle", bundle);

                startActivity(intent);
            }
        };

        countDownTimer.start();

    }

}
```

8.10 Leaderboard

```
package com.example.exam_teacher_app;

public class Leaderboard {

    String student_name, student_password, student_email, student_url, student_verified_code,
student_Spetiality;
    String Name, Set_Code, Subject, Verified_Id, answer_script_download_url, admin_name;
    long marks;

    String exam_id;

    String answer_script_uri;

    public Leaderboard(String student_name, String student_password, String student_email, String
student_url, String student_verified_code, String student_Spetiality, String name, String set_Code,
String subject, String verified_Id, long marks, String admin_name, String exam_id) {
        this.student_name = student_name;
        this.exam_id = exam_id;
        this.student_password = student_password;
        this.student_email = student_email;
        this.student_url = student_url;
        this.admin_name = admin_name;
        this.student_verified_code = student_verified_code;
        this.student_Spetiality = student_Spetiality;
        Name = name;
        Set_Code = set_Code;
        Subject = subject;
        Verified_Id = verified_Id;
        this.marks = marks;
    }

    public Leaderboard(String student_name, String student_password, String student_email, String
student_url, String student_verified_code, String student_Spetiality, String name, String set_Code,
String subject, String verified_Id, String answer_script_uri) {
        this.student_name = student_name;
        this.student_password = student_password;
        this.student_email = student_email;
        this.student_url = student_url;
        this.student_verified_code = student_verified_code;
        this.student_Spetiality = student_Spetiality;
        Name = name;
        Set_Code = set_Code;
        Subject = subject;
        Verified_Id = verified_Id;
        //this.marks = marks;
        this.answer_script_uri = answer_script_uri;
    }

    public Leaderboard() {
    }

    public void setStudent_name(String student_name) {
        this.student_name = student_name;
    }

    public void setStudent_password(String student_password) {
        this.student_password = student_password;
    }

    public void setStudent_email(String student_email) {
        this.student_email = student_email;
    }

    public void setStudent_url(String student_url) {
        this.student_url = student_url;
    }

    public void setStudent_verified_code(String student_verified_code) {
        this.student_verified_code = student_verified_code;
    }

    public void setStudent_Spetiality(String student_Spetiality) {
        this.student_Spetiality = student_Spetiality;
    }

    public void setName(String name) {
        Name = name;
    }

    public void setSet_Code(String set_Code) {
        Set_Code = set_Code;
    }

    public void setSubject(String subject) {
        Subject = subject;
    }
```

```
public void setVerified_Id(String verified_Id) {
   Verified_Id = verified_Id;
}

public String getAnswer_script_download_url() {
   return answer_script_download_url;
}

public void setAnswer_script_download_url(String answer_script_download_url) {
   this.answer_script_download_url = answer_script_download_url;
}

public void setMarks(long marks) {
   this.marks = marks;
}

public String getAnswer_script_uri() {
   return answer_script_uri;
}

public void setAnswer_script_uri(String answer_script_uri) {
   this.answer_script_uri = answer_script_uri;
}

public String getStudent_name() {
   return student_name;
}

public String getStudent_password() {
   return student_password;
}

public String getStudent_email() {
   return student_email;
}

public String getStudent_url() {
   return student_url;
}

public String getStudent_verified_code() {
   return student_verified_code;
}

public String getStudent_Spetiality() {
   return student_Spetiality;
}

public String getName() {
   return Name;
}

public String getSet_Code() {
   return Set_Code;
}

public String getSubject() {
   return Subject;
}

public String getVerified_Id() {
   return Verified_Id;
}

public long getMarks() {
   return marks;
}

@Override
public String toString() {
   return "Leaderboard{" +
         "student_name='" + student_name + '\'' +
         ", student_password='" + student_password + '\'' +
         ", student_email='" + student_email + '\'' +
         ", student_url='" + student_url + '\'' +
         ", student_verified_code='" + student_verified_code + '\'' +
         ", student_Spetiality='" + student_Spetiality + '\'' +
         ", Name='" + Name + '\'' +
         ", Set_Code='" + Set_Code + '\'' +
         ", Subject='" + Subject + '\'' +
         ", Verified_Id='" + Verified_Id + '\'' +
         ", marks=" + marks +
         '}';
}

public String getAdmin_name() {
   return admin_name;
}

public void setAdmin_name(String admin_name) {
   this.admin_name = admin_name;
}

public String getExam_id() {
   return exam_id;
}

public void setExam_id(String exam_id) {
   this.exam_id = exam_id;
}
```

```
}
```

8.11 student info

```
package com.example.eschool_student;

public class student_info {

    String student_name, department_name, student_request_id, student_email;
    boolean status;

    String  approved_id, student_id, password;
    boolean exam_permission;

    public student_info(String student_name, String department_name, String student_request_id,
boolean status) {
        this.student_name = student_name;
        this.department_name = department_name;
        this.student_request_id = student_request_id;
        this.status = status;
    }

    public student_info(String student_name, String department_name, String student_request_id, String
student_email, boolean status) {
        this.student_name = student_name;
        this.department_name = department_name;
        this.student_request_id = student_request_id;
        this.student_email = student_email;
        this.status = status;
    }

    public student_info(String student_name, String department_name, String student_request_id, String
student_email, String approved_id, boolean status) {
        this.student_name = student_name;
        this.department_name = department_name;
        this.student_request_id = student_request_id;
        this.student_email = student_email;
        this.approved_id = approved_id;
        this.status = status;
    }

    public student_info() {
    }

    public String getStudent_name() {
        return student_name;
    }

    public void setStudent_name(String student_name) {
```

```
    this.student_name = student_name;
}

public String getDepartment_name() {
    return department_name;
}

public void setDepartment_name(String department_name) {
    this.department_name = department_name;
}

public String getApproved_id() {
    return approved_id;
}

public void setApproved_id(String approved_id) {
    this.approved_id = approved_id;
}

public String getStudent_id() {
    return student_id;
}

public void setStudent_id(String student_id) {
    this.student_id = student_id;
}

public String getPassword() {
    return password;
}

public void setPassword(String password) {
    this.password = password;
}

public boolean isExam_permission() {
    return exam_permission;
}

public void setExam_permission(boolean exam_permission) {
    this.exam_permission = exam_permission;
}

public String getStudent_request_id() {
    return student_request_id;
}

public void setStudent_request_id(String student_request_id) {
    this.student_request_id = student_request_id;
}
```

```
    public boolean isStatus() {
        return status;
    }

    public void setStatus(boolean status) {
        this.status = status;
    }

    public String getStudent_email() {
        return student_email;
    }

    public void setStudent_email(String student_email) {
        this.student_email = student_email;
    }
}
```

8.12 Give Mcq Exam

```
package com.example.eschool_student;

import static android.Manifest.permission.FOREGROUND_SERVICE;

import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.content.IntentFilter;
import android.content.pm.PackageManager;
import android.graphics.Color;
import android.net.Uri;
import android.os.Build;
import android.os.Bundle;
import android.os.Handler;
import android.view.View;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;

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

import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.OnFailureListener;
import com.google.android.gms.tasks.Task;
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 com.squareup.picasso.Picasso;

import java.time.LocalDate;
import java.time.LocalDateTime;
import java.time.LocalTime;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;

public class Give_Mcq_Exam extends AppCompatActivity implements View.OnClickListener {

    String exam_id, Name, Email, Id, Subject, Verified_Id, myday, myMonth, myYear, myHour,
myMinute, myday_end, myMonth_end, myYear_end, myHour_end, myMinute_end;

    DatabaseReference databaseReference, databaseReference1, databaseReference2;

    String department_name;

    private Intent intent;

    int day, month, year, hour, minute;
    int myday1, myMonth1, myYear1, myHour1, myMinute1;

    //String Name, Email, Id, Subject, Verified_Id, myday, myMonth, myYear, myHour, myMinute,
myday_end, myMonth_end, myYear_end, myHour_end, myMinute_end;;

    String userName, userPassword, UserSpetiality, userEmail, ownId, verfiedCode, url, Password,
Set_Code, type;

    TextView textView, question, Hour, Min, Sec;

    String student_request_id;

    Button option1, option2, option3, option4, next, prev, submit;

    public static long stop_time = System.nanoTime();

    public static Handler sHandler;
    private final int playPause = 0;
    private final int reset = 1;
    private int secs = 0;
    private int mins = 0;
    private int millis = 0;
    private long currentTime = 0L;
    private boolean isBound = false;
    private MyIntentService myService;

    Map<Integer, Integer> map = new HashMap<>();

    List<MCQ> list;

    int selected_question = 0, total_question = 0;

    public static long total_time = 0L, score_gained = 0L;

    ImageView imageView;

    //long secs, mins, millis;

    @RequiresApi(api = Build.VERSION_CODES.O)
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_give_mcq_exam);

        ActivityCompat.requestPermissions(this, new String[]{FOREGROUND_SERVICE},
PackageManager.PERMISSION_GRANTED);

        Intent intent = this.getIntent();

        Bundle bundle = intent.getBundleExtra("bundle");

        Name = bundle.getString("Name");
        //Password = bundle.getString("Password");
        Email = bundle.getString("Email");
        Id = bundle.getString("Id");
        Set_Code = bundle.getString("set_code");
        Verified_Id = bundle.getString("verified_id");
        exam_id = bundle.getString("exam_id");
        myYear = bundle.getString("myYear");
        myMonth = bundle.getString("myMonth");
        myday = bundle.getString("myday");
        myYear_end = bundle.getString("myYear_end");
        myMonth_end = bundle.getString("myMonth_end");
        myday_end = bundle.getString("myday_end");
        myHour = bundle.getString("myHour");
        myMinute = bundle.getString("myMinuite");
        myHour_end = bundle.getString("myHour_end");
        myMinute_end = bundle.getString("myMinuite_end");
        Subject = bundle.getString("Subject");
        department_name = bundle.getString("department_name");
        student_request_id = bundle.getString("student_request_id");
        userName = bundle.getString("username");
        //userPassword = bundle.getString("userpassword");
        //UserSpetiality = bundle.getString("student_spetiality_subject");
        userEmail = bundle.getString("useremail");
        ownId = bundle.getString("ownid");
        verfiedCode = bundle.getString("verified_code");
        //url = bundle.getString("useruri");

        type = bundle.getString("type");

        //Toast.makeText(getApplicationContext(), "start time : " + myMinute + " " + myMinute_end,
Toast.LENGTH_SHORT).show();

        databaseReference =
FirebaseDatabase.getInstance().getReference("MCQ_Question").child(Name).child(Subject).child(Nam
e).child(Set_Code);

        LocalDate dateObj = LocalDate.now();
        String currentDate = dateObj.toString();

        String Date_details[] = currentDate.trim().split("-");

        // 0 -> year 1 -> month 2 -> date

        LocalTime timeObj = LocalTime.now();
        String currentTime = timeObj.toString();

        String time_details[] = currentTime.split(":");

        // 0 -> hour 1 -> min

        if(type!= null && type.equalsIgnoreCase("practice_past_questions")) {

            //DateTime dateTime = new DateTime(myYear, myMonth, myday, myHour, myMinute);

            LocalDateTime date1 = LocalDateTime.now();
            LocalDateTime date2 = LocalDateTime.of(Integer.parseInt(myYear),
Integer.parseInt(myMonth), Integer.parseInt(myday), Integer.parseInt(myHour) ,
Integer.parseInt(myMinute));
            LocalDateTime date3 = LocalDateTime.of(Integer.parseInt(myYear_end),
Integer.parseInt(myMonth_end), Integer.parseInt(myday_end), Integer.parseInt(myHour_end) ,
Integer.parseInt(myMinute_end));

            System.out.println("date : " + date1.toString());
            System.out.println("date2 : " + date2.toString());
            System.out.println("date3 : " + date3.toString());

            if(date1.isAfter(date2) && date1.isAfter(date3)) {
            } else {

                Toast.makeText(getApplicationContext(), "Exam is not prepare for practice",
Toast.LENGTH_LONG).show();

                onBackPressed();

            }

            databaseReference1 =
FirebaseDatabase.getInstance().getReference("Practice_MCQ_Leaderboard");

            databaseReference2 =
FirebaseDatabase.getInstance().getReference("Practice_MCQ_Report_Of_Student").child(Name).child(
exam_id).child(Subject).child(Set_Code).child(ownId);

        } else {

            LocalDateTime date1 = LocalDateTime.now();
            LocalDateTime date2 = LocalDateTime.of(Integer.parseInt(myYear),
Integer.parseInt(myMonth), Integer.parseInt(myday), Integer.parseInt(myHour) ,
Integer.parseInt(myMinute));
            LocalDateTime date3 = LocalDateTime.of(Integer.parseInt(myYear_end),
Integer.parseInt(myMonth_end), Integer.parseInt(myday_end), Integer.parseInt(myHour_end) ,
Integer.parseInt(myMinute_end));

            System.out.println("date : " + date1.toString());
            System.out.println("date2 : " + date2.toString());
            System.out.println("date3 : " + date3.toString());

            if(date1.isAfter(date2) && date1.isBefore(date3)) {


            } else {

                Toast.makeText(getApplicationContext(), "Exam is not prepare for practice",
Toast.LENGTH_LONG).show();

                if(date1.isAfter(date3)) {

                    Intent intent1 = new Intent(Give_Mcq_Exam.this, MCQ_Exam_Time.class);

                    Bundle bundle1 = intent.getBundleExtra("bundle");

                    bundle1.putString("student_name", userName);
                    bundle1.putString("student_email", userEmail);
                    bundle1.putString("aproved_id", ownId);
                    bundle1.putString("department_name", department_name);
                    bundle1.putString("student_request_id", student_request_id);
                    bundle1.getString("admin_name", Name);
                    bundle1.getString("subject", Subject);

                    intent1.putExtra("bundle", bundle1);

                    startActivity(intent1);

                }

                //onBackPressed();

            }

            databaseReference1 =
FirebaseDatabase.getInstance().getReference("Leaderboard_Of_MCQ_Exam").child(Name);

            databaseReference2 =
FirebaseDatabase.getInstance().getReference("MCQ_Report_Of_Student").child(Name).child(exam_id)
.child(Subject).child(Set_Code).child(ownId);

        }

        list = new ArrayList<>();

        imageView = findViewById(R.id.question_image);

        //textView = findViewById(R.id.team_poles_of_power);

        /*

        year month day hour minuite

        2022 10   21  10   30
        2022 10   21  11   30
        -----------------------
        0    0    0   1    0
                      60
                      3600

         */

        total_time = Math.abs(Long.parseLong(myYear_end.trim()) - Long.parseLong(myYear.trim())) *
31536000L +
                Math.abs(Long.parseLong(myMonth_end.trim()) - Long.parseLong(myMonth.trim())) *
2628000L +
                Math.abs(Long.parseLong(myday_end.trim()) - Long.parseLong(myday.trim())) * 86400L +
                Math.abs(Long.parseLong(myHour.trim()) - Long.parseLong(myHour_end.trim())) * 3600L +
                Math.abs(Long.parseLong(myMinute_end.trim()) - Long.parseLong(myMinute.trim())) *
60L;
```

```
        total_time *= 1000L;

        //Toast.makeText(getApplicationContext(), "time : " + total_time,
Toast.LENGTH_SHORT).show();

        option1 = findViewById(R.id.option1);
        option2 = findViewById(R.id.option2);
        option3 = findViewById(R.id.option3);
        option4 = findViewById(R.id.option4);
        next = findViewById(R.id.next);
        prev = findViewById(R.id.prev);
        question = findViewById(R.id.question);
        Hour = findViewById(R.id.Hour);
        Min = findViewById(R.id.Min);
        Sec = findViewById(R.id.Sec);
        submit = findViewById(R.id.submit);

        option4.setOnClickListener(this);
        option3.setOnClickListener(this);
        option2.setOnClickListener(this);
        option1.setOnClickListener(this);
        next.setOnClickListener(this);
        prev.setOnClickListener(this);
        submit.setOnClickListener(this);

        BroadcastReceiver broadcastReceiver = new BroadcastReceiver() {
            @Override
            public void onReceive(Context context, Intent intent) {

                try {
                    // Do some background work

                    //String time = intent.getStringExtra("time");

                    //currentTime = Long.parseLong(time);

                    secs = Integer.parseInt(intent.getStringExtra("sec"));
                    mins = Integer.parseInt(intent.getStringExtra("min"));

                    millis = Integer.parseInt(intent.getStringExtra("msec"));
                    setTime();

                    long now = millis + secs * 1000L + mins * 60 * 1000L;

                    if(now == total_time) {

                        long score = getScore_gained();

                        Toast.makeText(getApplicationContext(), "score gained : " + score,
Toast.LENGTH_LONG).show();
```

```
                getApplicationContext().stopService(intent);

                store_in_database(score);

                /*Intent intent1 = new Intent(Give_Mcq_Exam.this, See_Score.class);

                Bundle bundle = new Bundle();

                bundle.putString("name", userName);
                bundle.putString("password", userPassword);
                bundle.putString("subject", UserSpetiality);
                bundle.putString("email", userEmail);
                bundle.putString("id", ownId);
                bundle.putString("verified_code", verfiedCode);
                bundle.putString("uri", url);
                bundle.putString("score", score + "");

                intent1.putExtra("bundle", bundle);

                startActivity(intent1);*/

            }

            //setTime();

            //System.out.println("I am here " + time);

        } finally {
            // Must call finish() so the BroadcastReceiver can be recycled

        }

    /*String time = intent.getStringExtra("time");

    currentTime = Long.parseLong(time);

    secs = (int) (currentTime / 1000);
    mins = secs / 60;
    secs = secs % 60;
    millis = (int) (currentTime % 1000);
    setTime();

    System.out.println("I am here " + time);*/

    }
};

IntentFilter filter = new IntentFilter();
filter.addAction("time");
```

```
    registerReceiver(broadcastReceiver, filter );

    try {

        question.setText(list.get(selected_question).getQuestion());
        option1.setText(list.get(selected_question).getOption1());
        option2.setText(list.get(selected_question).getOption2());
        option3.setText(list.get(selected_question).getOption3());
        option4.setText(list.get(selected_question).getOption4());

    } catch (Exception exception) {

    }

}

@Override
protected void onDestroy() {
    super.onDestroy();

    try {

        getApplicationContext().stopService(intent);
        finishAffinity();

    } catch (Exception exception) {

    }

}

@Override
public void onClick(View v) {

    if(v.getId() == option1.getId()) {

        map.put(selected_question, 1);

        option1.setTextColor(Color.GREEN);

        option2.setTextColor(Color.BLACK);
        option3.setTextColor(Color.BLACK);
        option4.setTextColor(Color.BLACK);

        if(list.get(selected_question).getAnswer() == 1) {

            if(list.get(selected_question).marks != 1) {

                list.get(selected_question).marks = (list.get(selected_question).marks + 1) % 2;
                }

            } else {

                list.get(selected_question).marks = 0;

            }

            student_mcq student_mcq = new student_mcq(list.get(selected_question).Question,
list.get(selected_question).Option1, list.get(selected_question).Option2,
list.get(selected_question).Option3, list.get(selected_question).Option4,
list.get(selected_question).Answer, list.get(selected_question).marks, list.get(selected_question).url, 1);


databaseReference2.child(list.get(selected_question).Question).setValue(student_mcq).addOnComplete
Listener(new OnCompleteListener<Void>() {
                @Override
                public void onComplete(@NonNull Task<Void> task) {

                    Toast.makeText(getApplicationContext(), "your marked noted",
Toast.LENGTH_LONG).show();

                }
            }).addOnFailureListener(new OnFailureListener() {
                @Override
                public void onFailure(@NonNull Exception e) {

                    Toast.makeText(getApplicationContext(), "failed to store your opinion",
Toast.LENGTH_LONG).show();

                }
            });

        } else if(v.getId() == option2.getId()) {

            map.put(selected_question, 2);

            option2.setTextColor(Color.GREEN);

            option1.setTextColor(Color.BLACK);
            option3.setTextColor(Color.BLACK);
            option4.setTextColor(Color.BLACK);

            if(list.get(selected_question).getAnswer() == 2) {

                if(list.get(selected_question).marks != 1) {

                    list.get(selected_question).marks = (list.get(selected_question).marks + 1) % 2;

                }
            } else {

                list.get(selected_question).marks = 0;

            }

            student_mcq student_mcq = new student_mcq(list.get(selected_question).Question,
list.get(selected_question).Option1, list.get(selected_question).Option2,
list.get(selected_question).Option3, list.get(selected_question).Option4,
list.get(selected_question).Answer, list.get(selected_question).marks, list.get(selected_question).url, 2);


databaseReference2.child(list.get(selected_question).Question).setValue(student_mcq).addOnComplete
Listener(new OnCompleteListener<Void>() {
                @Override
                public void onComplete(@NonNull Task<Void> task) {

                    Toast.makeText(getApplicationContext(), "your marked noted",
Toast.LENGTH_LONG).show();

                }
            }).addOnFailureListener(new OnFailureListener() {
                @Override
                public void onFailure(@NonNull Exception e) {

                    Toast.makeText(getApplicationContext(), "failed to store your opinion",
Toast.LENGTH_LONG).show();

                }
            });

        } else if(v.getId() == option3.getId()) {

            map.put(selected_question, 3);

            option3.setTextColor(Color.GREEN);

            option2.setTextColor(Color.BLACK);
            option1.setTextColor(Color.BLACK);
            option4.setTextColor(Color.BLACK);

            if(list.get(selected_question).getAnswer() == 3) {

                if(list.get(selected_question).marks != 1) {

                    list.get(selected_question).marks = (list.get(selected_question).marks + 1) % 2;

                }
            } else {

                list.get(selected_question).marks = 0;

            }

            student_mcq student_mcq = new student_mcq(list.get(selected_question).Question,
list.get(selected_question).Option1, list.get(selected_question).Option2,
list.get(selected_question).Option3, list.get(selected_question).Option4,
list.get(selected_question).Answer, list.get(selected_question).marks, list.get(selected_question).url, 3);


databaseReference2.child(list.get(selected_question).Question).setValue(student_mcq).addOnComplete
Listener(new OnCompleteListener<Void>() {
                @Override
                public void onComplete(@NonNull Task<Void> task) {

                    Toast.makeText(getApplicationContext(), "your marked noted",
Toast.LENGTH_LONG).show();

                }
            }).addOnFailureListener(new OnFailureListener() {
                @Override
                public void onFailure(@NonNull Exception e) {

                    Toast.makeText(getApplicationContext(), "failed to store your opinion",
Toast.LENGTH_LONG).show();

                }
            });

        } else if(v.getId() == option4.getId()) {

            map.put(selected_question, 4);

            option4.setTextColor(Color.GREEN);

            option2.setTextColor(Color.BLACK);
            option3.setTextColor(Color.BLACK);
            option1.setTextColor(Color.BLACK);

            if(list.get(selected_question).getAnswer() == 4) {

                if(list.get(selected_question).marks != 1) {

                    list.get(selected_question).marks = (list.get(selected_question).marks + 1) % 2;

                }

            } else {
                list.get(selected_question).marks = 0;

            }

            student_mcq student_mcq = new student_mcq(list.get(selected_question).Question,
list.get(selected_question).Option1, list.get(selected_question).Option2,
list.get(selected_question).Option3, list.get(selected_question).Option4,
list.get(selected_question).Answer, list.get(selected_question).marks, list.get(selected_question).url, 4);


databaseReference2.child(list.get(selected_question).Question).setValue(student_mcq).addOnComplete
Listener(new OnCompleteListener<Void>() {
                @Override
                public void onComplete(@NonNull Task<Void> task) {

                    Toast.makeText(getApplicationContext(), "your marked noted",
Toast.LENGTH_LONG).show();

                }
            }).addOnFailureListener(new OnFailureListener() {
                @Override
                public void onFailure(@NonNull Exception e) {

                    Toast.makeText(getApplicationContext(), "failed to store your opinion",
Toast.LENGTH_LONG).show();

                }
            });

        } else if(v.getId() == next.getId()) {

            selected_question = (selected_question + 1) % list.size();

            try {

                question.setText(list.get(selected_question).getQuestion());
                option1.setText(list.get(selected_question).getOption1());
                option2.setText(list.get(selected_question).getOption2());
                option3.setText(list.get(selected_question).getOption3());
                option4.setText(list.get(selected_question).getOption4());

                if(!list.get(selected_question).url.equals("no image")) {

                    Uri uri = Uri.parse(list.get(selected_question).url);

                    Picasso.get().load(uri).fit().into(imageView);

                }
```

```
    } catch (Exception exception) {

    }

    try {

        Picasso.get().load(Uri.parse(list.get(selected_question).url)).fit().into(imageView);

    } catch (Exception e) {


    }

    option1.setTextColor(Color.BLACK);

    option2.setTextColor(Color.BLACK);
    option3.setTextColor(Color.BLACK);
    option4.setTextColor(Color.BLACK);

    if(map.containsKey(selected_question)) {

        if(map.get(selected_question) == 1) {

            option1.setTextColor(Color.GREEN);

        } else if(map.get(selected_question) == 2) {

            option2.setTextColor(Color.GREEN);

        } else if(map.get(selected_question) == 3) {

            option3.setTextColor(Color.GREEN);

        } if(map.get(selected_question) == 4) {

            option4.setTextColor(Color.GREEN);

        }

    }

} else if(v.getId() == prev.getId()) {

    selected_question = ((selected_question - 1) <= 0 ? 1 : selected_question - 1) % list.size();

    try {

        question.setText(list.get(selected_question).getQuestion());
        option1.setText(list.get(selected_question).getOption1());
        option2.setText(list.get(selected_question).getOption2());
        option3.setText(list.get(selected_question).getOption3());
        option4.setText(list.get(selected_question).getOption4());

        if(!list.get(selected_question).url.equals("no image")) {

            Uri uri = Uri.parse(list.get(selected_question).url);

            Picasso.get().load(uri).fit().into(imageView);

        }

        try {

            Picasso.get().load(Uri.parse(list.get(selected_question).url)).fit().into(imageView);

        } catch (Exception e) {


        }

    } catch (Exception exception) {

    }

    option1.setTextColor(Color.BLACK);
    option2.setTextColor(Color.BLACK);
    option3.setTextColor(Color.BLACK);
    option1.setTextColor(Color.BLACK);

    if(map.containsKey(selected_question)) {

        if(map.get(selected_question) == 1) {

            option1.setTextColor(Color.GREEN);

        } else if(map.get(selected_question) == 2) {

            option2.setTextColor(Color.GREEN);

        } else if(map.get(selected_question) == 3) {

            option3.setTextColor(Color.GREEN);

        } if(map.get(selected_question) == 4) {

            option4.setTextColor(Color.GREEN);

        }

    }
```

```
        } else if(v.getId() == submit.getId()) {

            long score = getScore_gained();

            Toast.makeText(getApplicationContext(), "score : " + score, Toast.LENGTH_LONG).show();
            getApplicationContext().stopService(intent);

            store_in_database(score);

            /*Intent intent = new Intent(Give_Mcq_Exam.this, See_Score.class);

            Bundle bundle = new Bundle();

            bundle.putString("name", userName);
            bundle.putString("password", userPassword);
            bundle.putString("subject", UserSpetiality);
            bundle.putString("email", userEmail);
            bundle.putString("id", ownId);
            bundle.putString("verified_code", verfiedCode);
            bundle.putString("uri", url);
            bundle.putString("score", score + "");

            intent.putExtra("bundle", bundle);

            startActivity(intent);*/

        }

    }

    @Override
    protected void onStart() {

        databaseReference1.child(Subject).child(Set_Code).child(userName).addValueEventListener(new
ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot snapshot) {

                if(snapshot.exists()) {

                    Toast.makeText(getApplicationContext(), "your mark is not noted as you already give the
exam", Toast.LENGTH_SHORT).show();

                    databaseReference1 =
FirebaseDatabase.getInstance().getReference("Practice_MCQ_Leaderboard");

                    databaseReference2 =
FirebaseDatabase.getInstance().getReference("Practice_MCQ_Report_Of_Student").child(Name).child(
exam_id).child(Subject).child(Set_Code).child(ownId);
```

```
            } else {



            }

        }

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

        }
    });

    list.clear();

    intent = new Intent(Give_Mcq_Exam.this, MyIntentService.class);

    intent.putExtra("time", total_time + "");

    getApplicationContext().startService(intent);

    databaseReference.addValueEventListener(new ValueEventListener() {
        @Override
        public void onDataChange(@NonNull DataSnapshot snapshot) {

            for(DataSnapshot dataSnapshot : snapshot.getChildren()) {

                Map<String, String> map = (Map<String, String>) dataSnapshot.getValue();

                String Question = map.get("question");
                String Option1 = map.get("option1");
                String Option2 = map.get("option2");
                String Option3 = map.get("option3");
                String Option4 = map.get("option4");
                int Answer = Integer.parseInt(map.get("answer").trim());
                String link = map.get("link");

                MCQ mcq = new MCQ(Question, Option1, Option2, Option3, Option4, Answer, 0, link);

                //Toast.makeText(Give_Mcq_Exam.this, "" + mcq.toString(),
Toast.LENGTH_SHORT).show();

                list.add(mcq);

            }

            try {
```

```
          question.setText(list.get(selected_question).getQuestion());
          option1.setText(list.get(selected_question).getOption1());
          option2.setText(list.get(selected_question).getOption2());
          option3.setText(list.get(selected_question).getOption3());
          option4.setText(list.get(selected_question).getOption4());

        } catch (Exception exception) {

        }

      }

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

        Toast.makeText(getApplicationContext(), "error : " + error,
Toast.LENGTH_SHORT).show();

      }
    });

    total_question = list.size();

    //Toast.makeText(getApplicationContext(), "size : " + total_question,
Toast.LENGTH_SHORT).show();

    try {

      question.setText(list.get(selected_question).getQuestion());
      option1.setText(list.get(selected_question).getOption1());
      option2.setText(list.get(selected_question).getOption2());
      option3.setText(list.get(selected_question).getOption3());
      option4.setText(list.get(selected_question).getOption4());

    } catch (Exception exception) {

    }

    try {

      super.onStart();

    } catch (Exception exception) {

    }

  }

  public void setTime() {
    Hour.setText("" + mins);
```

```
        Min.setText(secs + "");
        Sec.setText(millis + "");
    }

    long getScore_gained() {

        long score = 0L;

        for(MCQ mcq : list) {

            score += mcq.marks;

        }

        return score;

    }

    void store_in_database(long score) {

        Leaderboard leaderboard = new Leaderboard(userName, userPassword, userEmail, url, ownId,
UserSpetiality, Name, Set_Code, Subject, Verified_Id, score, Name, exam_id);


databaseReference1.child(Subject).child(Set_Code).child(userName).setValue(leaderboard).addOnCom
pleteListener(new OnCompleteListener<Void>() {
            @Override
            public void onComplete(@NonNull Task<Void> task) {

                Toast.makeText(getApplicationContext(), "your score noted",
Toast.LENGTH_LONG).show();

                Intent intent = new Intent(Give_Mcq_Exam.this, Marker.class);

                Bundle bundle = new Bundle();

                bundle.putString("mark", getScore_gained() + "");

                intent.putExtra("bundle", bundle);

                startActivity(intent);

            }
        }).addOnFailureListener(new OnFailureListener() {
            @Override
            public void onFailure(@NonNull Exception e) {

                Toast.makeText(getApplicationContext(), "your score is not stored",
Toast.LENGTH_LONG).show();
```

```
        }
    });

}

@Override
public void onBackPressed() {

    long score = getScore_gained();

    Toast.makeText(getApplicationContext(), "score : " + score, Toast.LENGTH_LONG).show();
    getApplicationContext().stopService(intent);

    store_in_database(score);

    Intent intent = new Intent(Give_Mcq_Exam.this, Marker.class);

    Bundle bundle = new Bundle();

    bundle.putString("mark", score + "");

    intent.putExtra("bundle", bundle);

    startActivity(intent);

    /*Intent intent = new Intent(Give_Mcq_Exam.this, See_Score.class);

    Bundle bundle = new Bundle();

    bundle.putString("name", userName);
    bundle.putString("password", userPassword);
    bundle.putString("subject", UserSpetiality);
    bundle.putString("email", userEmail);
    bundle.putString("id", ownId);
    bundle.putString("verified_code", verfiedCode);
    bundle.putString("uri", url);
    bundle.putString("score", score + "");

    intent.putExtra("bundle", bundle);

    startActivity(intent);*/

    super.onBackPressed();

}

boolean isValidTime(String start_time_selected, String end_time_selected) {

    if(start_time_selected != null && end_time_selected != null) {
```

```
String a[] = start_time_selected.split(" / ");
String b[] = end_time_selected.split(" / ");

if(Long.parseLong(a[0].trim()) < Long.parseLong(b[0].trim())) {

    return true;

} if(Long.parseLong(a[0].trim()) == Long.parseLong(b[0].trim())) {

    if(Long.parseLong(a[1].trim()) < Long.parseLong(b[1].trim())) return true;

    if(Long.parseLong(a[1].trim()) == Long.parseLong(b[1].trim())) {

        if(Long.parseLong(a[2].trim()) < Long.parseLong(b[2].trim())) return true;

        if(Long.parseLong(a[2].trim()) <= Long.parseLong(b[2].trim())) {

            if(Long.parseLong(a[3].trim()) < Long.parseLong(b[3].trim())) return true;

            if(Long.parseLong(a[3]) <= Long.parseLong(b[3])) {

                if(Long.parseLong(a[4]) <= Long.parseLong(b[4])) {

                    return true;

                } else {

                    return false;

                }

            } else {

                return false;

            }

        } else {

            return false;

        }

    } else {

        return false;

    }

} else {
```

```
                return true;
            }
        }
        return false;
    }
```

8.13 Give Open Hand Question

```
package com.example.eschool_student;

import android.app.DownloadManager;
import android.content.BroadcastReceiver;
import android.content.ContentResolver;
import android.content.Context;
import android.content.Intent;
import android.content.IntentFilter;
import android.net.Uri;
import android.os.Build;
import android.os.Bundle;
import android.os.Handler;
import android.view.View;
import android.webkit.MimeTypeMap;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;

import androidx.activity.result.ActivityResultCallback;
import androidx.activity.result.ActivityResultLauncher;
import androidx.activity.result.contract.ActivityResultContracts;
import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.annotation.RequiresApi;
import androidx.appcompat.app.AppCompatActivity;

import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.OnFailureListener;
import com.google.android.gms.tasks.OnSuccessListener;
import com.google.android.gms.tasks.Task;
import com.google.android.material.snackbar.Snackbar;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.storage.FirebaseStorage;
import com.google.firebase.storage.StorageReference;
import com.google.firebase.storage.UploadTask;

import java.time.LocalDate;
import java.time.LocalDateTime;
import java.time.LocalTime;

public class Give_CQ_Exam extends AppCompatActivity implements View.OnClickListener {

    String userName, userPassword, UserSpetiality, userEmail, ownId, verfiedCode, url;

    String Name, Email, Password, Id, Subject, Verified_Id, question_name, exam_set, question_id, prev_file, start_time_selected, end_time_selected, choice;

    String question_key, exam_id, uri, type;

    String present_start_time, present_end_time, Year, Month, Day, Year_end, Month_end, Day_end, Hour, Minuite, Hour_end, Minuite_end;

    TextView Minuite_TextView, Second_Text_View, Milli_Second_TextView, Uploaded_File_Name;

    Button Upload, Submit;

    public static long total_time = 0;

    public static long stop_time = System.nanoTime();

    Uri answer_script;

    private static final int GALLERY_REQUEST = 1;

    public static Handler sHandler;
    private final int playPause = 0;
    private final int reset = 1;
    private int secs = 0;
    private int mins = 0;
    private int millis = 0;
    private long currentTime = 0L;
    private boolean isBound = false;
    private My_Another_Intent_Service my_another_intent_service;
    private Intent intent1;

    DatabaseReference databaseReference, databaseReference1;

    FirebaseStorage firebaseStorage;

    StorageReference storageReference;

    @RequiresApi(api = Build.VERSION_CODES.GINGERBREAD)
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_give_cq_exam);

        Intent intent = this.getIntent();

        Bundle bundle = intent.getBundleExtra("bundle");

        //Intent intent = this.getIntent();

        //Bundle bundle = intent.getBundleExtra("bundle");

        userEmail = bundle.getString("email");
        userName = bundle.getString("name");
        verfiedCode = bundle.getString("verified_code");
        ownId = bundle.getString("id");
        //url = bundle.getString("uri_student");
        UserSpetiality = bundle.getString("subject");
        //userPassword = bundle.getString("password");

        Year = bundle.getString("myYear");
        Month = bundle.getString("myMonth");
        Hour = bundle.getString("myHour");
        Minuite = bundle.getString("myMinute");
        Year_end = bundle.getString("myYear_end");
        Month_end = bundle.getString("myMonth_end");
        Day = bundle.getString("day");
        Day_end = bundle.getString("myday_end");
        Hour_end = bundle.getString("myHour_end");
        Minuite_end = bundle.getString("myMinute_end");
        Name = bundle.getString("Name");
        Subject = bundle.getString("subject");
        question_key = bundle.getString("question_key");//question_key
        exam_id = bundle.getString("exam_id");
        prev_file = bundle.getString("file_name");
        exam_set = bundle.getString("set");
        uri = bundle.getString("uri");
        type = bundle.getString("type");

        //Toast.makeText(getApplicationContext(), "uri : " + prev_file, Toast.LENGTH_LONG).show();
        //Toast.makeText(getApplicationContext(), "exam id : " + exam_id,
Toast.LENGTH_LONG).show();

        LocalDate dateObj = null;
        if (android.os.Build.VERSION.SDK_INT >= android.os.Build.VERSION_CODES.O) {
            dateObj = LocalDate.now();
        }
        String currentDate = dateObj.toString();

        String Date_details[] = currentDate.trim().split("-");
        // 0 -> year 1 -> month 2 -> date

        LocalTime timeObj = null;
        if (android.os.Build.VERSION.SDK_INT >= android.os.Build.VERSION_CODES.O) {
            timeObj = LocalTime.now();
        }
        String currentTime = timeObj.toString();

        String time_details[] = currentTime.split(":");

        if(type != null && type.equalsIgnoreCase("practice_past_questions")) {

            LocalDateTime date1 = null;
            if (android.os.Build.VERSION.SDK_INT >= android.os.Build.VERSION_CODES.O) {
                date1 = LocalDateTime.now();
            }
            LocalDateTime date2 = null;
            if (android.os.Build.VERSION.SDK_INT >= android.os.Build.VERSION_CODES.O) {
                date2 = LocalDateTime.of(Integer.parseInt(Year), Integer.parseInt(Month),
Integer.parseInt(Day), Integer.parseInt(Hour) , Integer.parseInt(Minuite));
            }
            LocalDateTime date3 = null;
            if (android.os.Build.VERSION.SDK_INT >= android.os.Build.VERSION_CODES.O) {
                date3 = LocalDateTime.of(Integer.parseInt(Year_end), Integer.parseInt(Month_end),
Integer.parseInt(Day_end), Integer.parseInt(Hour_end) , Integer.parseInt(Minuite_end));
            }

            System.out.println("date : " + date1.toString());
            System.out.println("date2 : " + date2.toString());
            System.out.println("date3 : " + date3.toString());

            if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
                if(date1.isAfter(date2) && date1.isAfter(date3)) {


                } else {

                    Toast.makeText(getApplicationContext(), "Exam is not prepare for practice",
Toast.LENGTH_LONG).show();

                    finishAffinity();

                }
            }

            databaseReference =
FirebaseDatabase.getInstance().getReference("Practice_CQ_Exam_Details_Database");

            firebaseStorage = FirebaseStorage.getInstance();
            storageReference =
firebaseStorage.getReference("UPLOADED_FILE").child("Students_practice_Answer_Scripts");

        } else {

            LocalDateTime date1 = null;
            if (android.os.Build.VERSION.SDK_INT >= android.os.Build.VERSION_CODES.O) {
                date1 = LocalDateTime.now();
            }
            LocalDateTime date2 = null;
            if (android.os.Build.VERSION.SDK_INT >= android.os.Build.VERSION_CODES.O) {
                date2 = LocalDateTime.of(Integer.parseInt(Year), Integer.parseInt(Month),
Integer.parseInt(Day), Integer.parseInt(Hour) , Integer.parseInt(Minuite));
            }
            LocalDateTime date3 = null;
            if (android.os.Build.VERSION.SDK_INT >= android.os.Build.VERSION_CODES.O) {
                date3 = LocalDateTime.of(Integer.parseInt(Year_end), Integer.parseInt(Month_end),
Integer.parseInt(Day_end), Integer.parseInt(Hour_end) , Integer.parseInt(Minuite_end));
            }

            System.out.println("date : " + date1.toString());
            System.out.println("date2 : " + date2.toString());
            System.out.println("date3 : " + date3.toString());

            if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
                if(date1.isAfter(date2) && date1.isBefore(date3)) {


                } else {

                    Toast.makeText(getApplicationContext(), "Exam is not prepare for practice",
Toast.LENGTH_LONG).show();

                    onBackPressed();

                }
            }

            databaseReference =
FirebaseDatabase.getInstance().getReference("CQ_Exam_Database_Answer_Script");

            firebaseStorage = FirebaseStorage.getInstance();
            storageReference =
firebaseStorage.getReference("UPLOADED_FILE").child("Students_Answer_Scripts");

        }

        /*databaseReference1 = FirebaseDatabase.getInstance().getReference("CQ_Exam").child(Subject);

        String question[] = new String[1];
        databaseReference1.addValueEventListener(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot snapshot) {

                for(DataSnapshot dataSnapshot : snapshot.getChildren()) {

                    Map<String, String> map = (Map<String, String>) dataSnapshot.getValue();

                    for(String i : map.keySet()) {

                        if(i.equalsIgnoreCase(question_id)) {

                            question[0] = (String) map.get("uri");
                            prev_file = question[0];

                            try {

                                Intent intent2 = new Intent(Intent.ACTION_VIEW);

                                //URI uri1 = new URI(question[0]);

                                intent2.setDataAndType(Uri.parse((question[0])), "application/*");
                                intent2.setFlags(Intent.FLAG_ACTIVITY_NO_HISTORY);
                                startActivity(intent2);

                                Toast.makeText(getApplicationContext(), "prev file is uploading...",
Toast.LENGTH_SHORT).show();

                            } catch (Exception e) {

                                Toast.makeText(getApplicationContext(), "exception : " + e + "\n" + question[0],
Toast.LENGTH_SHORT).show();

                            }

                            Toast.makeText(getApplicationContext(), "question here : " + question[0],
Toast.LENGTH_LONG).show();
                            //break;

                        }

                    }

                }

            }

            @Override
            public void onCancelled(@NonNull DatabaseError error) {
            }
        });*/

        //prev_file = question[0];

        try {

            //Uri.parse(uri);

            url = uri;

            DownloadManager downloadManager = (DownloadManager)
getApplicationContext().getSystemService(Context.DOWNLOAD_SERVICE);
            Uri uri = Uri.parse(url);
            DownloadManager.Request request = new DownloadManager.Request(uri);

request.setNotificationVisibility(DownloadManager.Request.VISIBILITY_VISIBLE_NOTIFY_COMP
LETED);
            request.setDestinationInExternalFilesDir(getApplicationContext(), "Downloads", "exam file");
            assert downloadManager != null;
            downloadManager.enqueue(request);
            Snackbar snackbar = (Snackbar) Snackbar
                    .make(findViewById(android.R.id.content), "Downloading...",
Snackbar.LENGTH_LONG);
            snackbar.show();
            Toast.makeText(getApplicationContext(), "prev file is uploading...",
Toast.LENGTH_SHORT).show();

        } catch (Exception e) {

            Toast.makeText(getApplicationContext(), "exception : " + e + "\n" + uri,
Toast.LENGTH_SHORT).show();

        }

        my_another_intent_service = new My_Another_Intent_Service();

        intent1 = new Intent(Give_CQ_Exam.this, My_Another_Intent_Service.class);

        getApplicationContext().startService(intent1);

        total_time = Math.abs(Long.valueOf(Year_end.trim()) - Long.valueOf(Year.trim())) * 31536000L
+
                Math.abs(Long.valueOf(Month_end.trim()) - Long.valueOf(Month.trim())) * 2628000L +
                Math.abs(Long.valueOf(Day_end.trim()) - Long.valueOf(Day.trim())) * 86400L +
                Math.abs(Long.valueOf(Hour_end.trim()) - Long.valueOf(Hour.trim())) * 3600L +
                Math.abs(Long.valueOf(Minuite_end.trim()) - Long.valueOf(Minuite.trim())) * 60L;

        total_time *= 1000L;
        present_start_time = Year + " / " + Month + " / " + Day + ", " + Hour + " : " + Minuite;
        present_end_time = Year_end + " / " + Month_end + " / " + Day_end + ", " + Hour_end + " : " +
Minuite_end;

        Minuite_TextView = findViewById(R.id.Hour);
        Second_Text_View = findViewById(R.id.Min);
        Milli_Second_TextView = findViewById(R.id.Sec);
        Uploaded_File_Name = findViewById(R.id.uploaded_file_name);

        Upload = findViewById(R.id.upload);
        Submit = findViewById(R.id.Submit);

        Upload.setOnClickListener(this);
        Submit.setOnClickListener(this);

        BroadcastReceiver broadcastReceiver = new BroadcastReceiver() {
            @Override
            public void onReceive(Context context, Intent intent) {

                try {
                    // Do some background work

                    //String time = intent.getStringExtra("time");

                    //currentTime = Long.parseLong(time);

                    secs = Integer.parseInt(intent.getStringExtra("sec"));
                    mins = Integer.parseInt(intent.getStringExtra("min"));

                    millis = Integer.parseInt(intent.getStringExtra("msec"));
                    setTime1();

                    long now = millis + secs * 1000L + mins * 60 * 1000L;

                    if(now == total_time) {

                        getApplicationContext().stopService(intent1);

                        Submit.setOnClickListener(Give_CQ_Exam.this);

                        /*long score = getScore_gained();

                        Toast.makeText(getApplicationContext(), "score gained : " + score,
Toast.LENGTH_LONG).show();

                        getApplicationContext().stopService(intent);

                        store_in_database(score);
                        Intent intent1 = new Intent(Give_Mcq_Exam.this, See_Score.class);

                        Bundle bundle = new Bundle();

                        bundle.putString("name", userName);
                        bundle.putString("password", userPassword);
                        bundle.putString("subject", UserSpetiality);
                        bundle.putString("email", userEmail);
                        bundle.putString("id", ownId);
                        bundle.putString("verified_code", verfiedCode);
                        bundle.putString("uri", url);
                        bundle.putString("score", score + "");

                        intent1.putExtra("bundle", bundle);

                        startActivity(intent1);*/

                    }

                    //setTime();

                    //System.out.println("I am here " + time);

                } finally {
                    // Must call finish() so the BroadcastReceiver can be recycled

                }

            /*String time = intent.getStringExtra("time");

            currentTime = Long.parseLong(time);

            secs = (int) (currentTime / 1000);
            mins = secs / 60;
            secs = secs % 60;
            millis = (int) (currentTime % 1000);
            setTime();

            System.out.println("I am here " + time);*/

            }
        };

        IntentFilter filter = new IntentFilter();
        filter.addAction("time");
        registerReceiver(broadcastReceiver, filter );

    }

    public void setTime1() {
        Minuite_TextView.setText("" + mins);
        Second_Text_View.setText(secs + "");
        Milli_Second_TextView.setText(millis + "");
    }

    @Override
    protected void onDestroy() {
        super.onDestroy();

        try {

            getApplicationContext().stopService(intent1);
            finishAffinity();

        } catch (Exception exception) {

        }

    }

    ActivityResultLauncher<String> mGetContent = registerForActivityResult(new
                    ActivityResultContracts.GetContent(),
            new ActivityResultCallback<Uri>() {
                @Override
                public void onActivityResult(Uri uri) {

                    answer_script = uri;

                    Uploaded_File_Name.setText((String.valueOf(answer_script) +
getfileExtension(answer_script)) );

                }
            });

    @Override
    public void onClick(View v) {

        if(Upload.getId() == v.getId()) {

            /*Intent photoPickerIntent = new Intent(Intent.ACTION_GET_CONTENT);
            photoPickerIntent.setType("application/*");
            startActivityForResult(photoPickerIntent, GALLERY_REQUEST);*/

            mGetContent.launch("application/*");

        } else if(Submit.getId() == v.getId()) {

            if(answer_script != null) {

                getApplicationContext().stopService(intent1);
                CQ_Leaderboard();

            } else {

                Toast.makeText(getApplicationContext(), "You have to upload your answer script",
Toast.LENGTH_LONG).show();

            }

        }

    }

    @Override
    public void onBackPressed() {

        Submit.setOnClickListener(Give_CQ_Exam.this);

        super.onBackPressed();
    }

    private String getfileExtension(Uri uri)
    {
        String extension;
        ContentResolver contentResolver = getContentResolver();
        MimeTypeMap mimeTypeMap = MimeTypeMap.getSingleton();
        extension= mimeTypeMap.getExtensionFromMimeType(contentResolver.getType(uri));
        return extension;
    }

    @Override
    protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {

        super.onActivityResult(requestCode, resultCode, data);

        if(requestCode == GALLERY_REQUEST && resultCode==RESULT_OK && data != null){

            answer_script = (data.getData());

            Uploaded_File_Name.setText((String.valueOf(answer_script) + getfileExtension(answer_script))
);

            //enter.setVisibility(View.VISIBLE);

        }

    }

    void CQ_Leaderboard() {
```

```
storageReference.child(Subject).child(Name).child(exam_set).child(userName).child(exam_id).putFile(
answer_script).addOnSuccessListener(new OnSuccessListener<UploadTask.TaskSnapshot>() {
            @Override
            public void onSuccess(UploadTask.TaskSnapshot taskSnapshot) {

                Task<Uri> uri1 = taskSnapshot.getStorage().getDownloadUrl();

                while (!uri1.isComplete());

                Uri answer_scripts_download_url = (uri1.getResult());

                Leaderboard leaderboard = new Leaderboard(userName, userPassword, userEmail, url, ownId,
UserSpetiality, Name, exam_set, Subject, Verified_Id, String.valueOf(answer_scripts_download_url));


databaseReference.child(Subject).child(Name).child(exam_set).child(exam_id).child(userName).setVal
ue(leaderboard).addOnCompleteListener(new OnCompleteListener<Void>() {
                    @Override
                    public void onComplete(@NonNull Task<Void> task) {

                        Toast.makeText(getApplicationContext(), "Data stored successfully",
Toast.LENGTH_SHORT).show();

                    }
                }).addOnFailureListener(new OnFailureListener() {
                    @Override
                    public void onFailure(@NonNull Exception e) {

                        Toast.makeText(getApplicationContext(),"Data is not uploaded successfully",
Toast.LENGTH_LONG).show();

                    }
                });

            }
        }).addOnFailureListener(new OnFailureListener() {
            @Override
            public void onFailure(@NonNull Exception e) {

                Toast.makeText(getApplicationContext(), " failed to upload your answer script",
Toast.LENGTH_LONG).show();

            }
        });

    }

    @Override
    protected void onStart() {
        /*try {

            /*Intent intent2 = new Intent(Intent.ACTION_VIEW);
            intent2.setDataAndType(Uri.parse(uri), "application/*");
            intent2.setFlags(Intent.FLAG_ACTIVITY_NO_HISTORY);
            startActivity(intent2);

            url = uri;

            DownloadManager downloadManager = (DownloadManager)
getApplicationContext().getSystemService(Context.DOWNLOAD_SERVICE);
            Uri uri = Uri.parse(url);
            DownloadManager.Request request = new DownloadManager.Request(uri);

request.setNotificationVisibility(DownloadManager.Request.VISIBILITY_VISIBLE_NOTIFY_COMP
LETED);
            request.setDestinationInExternalFilesDir(getApplicationContext(), "Downloads", "exam file");
            assert downloadManager != null;
            downloadManager.enqueue(request);
            Snackbar snackbar = (Snackbar) Snackbar
                    .make(findViewById(android.R.id.content), "Downloading...",
Snackbar.LENGTH_LONG);
            snackbar.show();
            Toast.makeText(getApplicationContext(), "prev file is uploading...",
Toast.LENGTH_SHORT).show();

        } catch (Exception e) {
            // Error...
        }*/

        try {

            super.onStart();

        } catch (Exception exception) {

        }

    }

    boolean isValidTime(String start_time_selected, String end_time_selected) {

        if(start_time_selected != null && end_time_selected != null) {

            String a[] = start_time_selected.split(" / ");
            String b[] = end_time_selected.split(" / ");

            if(Long.parseLong(a[0].trim()) < Long.parseLong(b[0].trim())) {
```

```
    return true;
} if(Long.parseLong(a[0].trim()) == Long.parseLong(b[0].trim())) {
    if(Long.parseLong(a[1].trim()) < Long.parseLong(b[1].trim())) return true;
    if(Long.parseLong(a[1].trim()) == Long.parseLong(b[1].trim())) {
        if(Long.parseLong(a[2].trim()) < Long.parseLong(b[2].trim())) return true;
        if(Long.parseLong(a[2].trim()) <= Long.parseLong(b[2].trim())) {
            if(Long.parseLong(a[3].trim()) < Long.parseLong(b[3].trim())) return true;
            if(Long.parseLong(a[3]) <= Long.parseLong(b[3])) {
                if(Long.parseLong(a[4]) <= Long.parseLong(b[4])) {
                    return true;
                } else {
                    return false;
                }
            } else {
                return false;
            }
        } else {
            return false;
        }
    } else {
        return false;
    }
} else {
    return true;
}
```

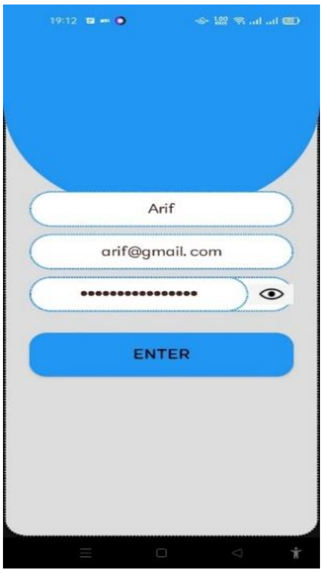
```
        }

        return false;

    }
}
```

CHAPTER 9
Interface Design

9. Interface Design

9.1 Admin dashboard

Admin firstly sign up by using their Name, Email and Password and then admin can log in to the system.



19:12

Arif

arif@gmail. com

••••••••••••••••

Fig 9.1.1: Admin Sign Up

This is the log dashboard of the Admin. Admin will use his Name, Email and Password for login.

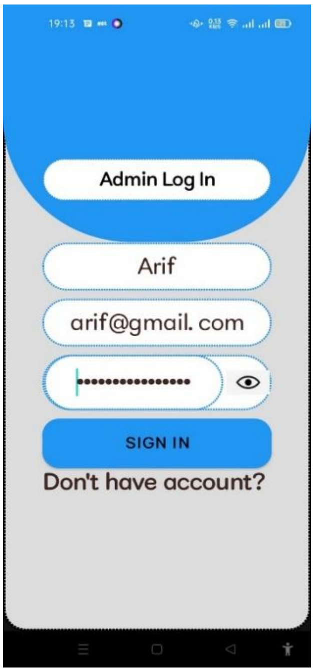


Fig 9.1.2: Admin Log in

After successfully log in this dashboard will appear first.

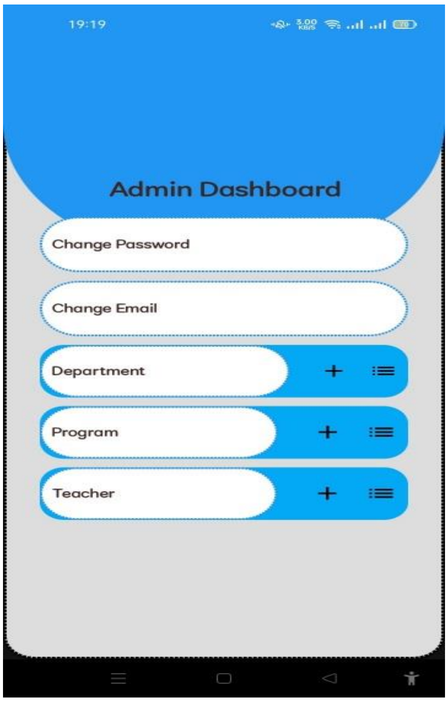
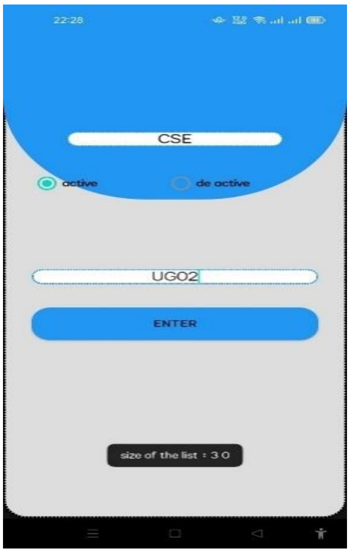


Fig 9.1.3: Admin Dashboard

Admin can add department and see the department list



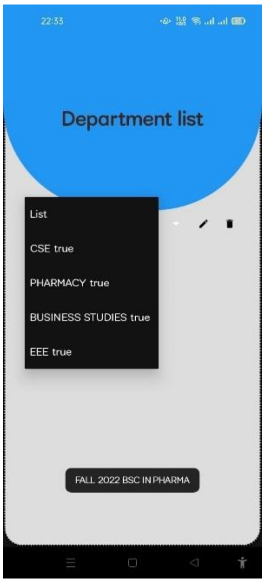
CSE

UG02

size of the list : 3 0

Fig 9.1.4: Add Department

If admin wants he can edit departments name and delete the departments.



List

CSE true

PHARMACY true

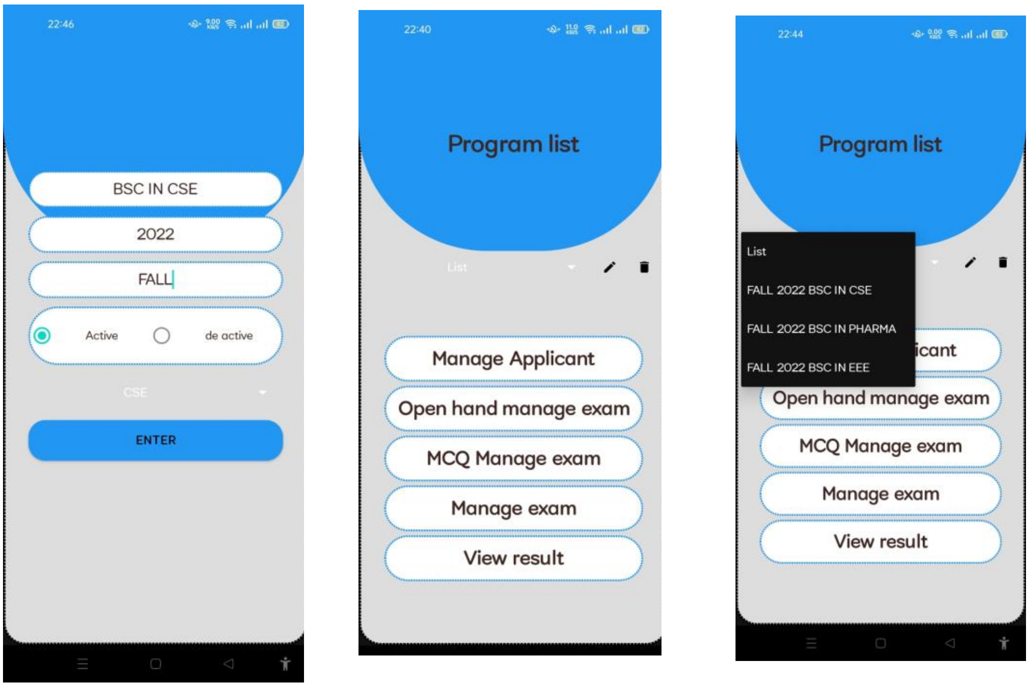
BUSINESS STUDIES true

EEE true

FALL 2022 BSC IN PHARMA

Fig 9.1.5: Edit/Delete Department

Here admin can add program under the departments and also see the program dashboard.



BSC IN CSE

2022

FALL

● Active ○ de active

CSE

Fig 9.1.6: Add Program and See List

In this portion admin can manage applicant, set open hand exam of MCQ and CQ and manage exam.

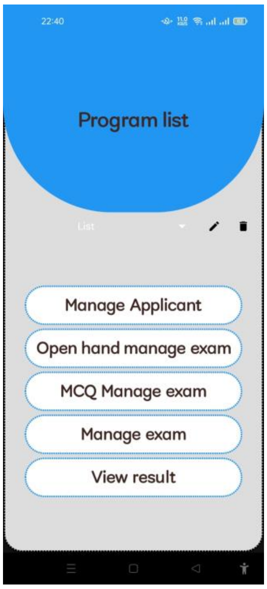
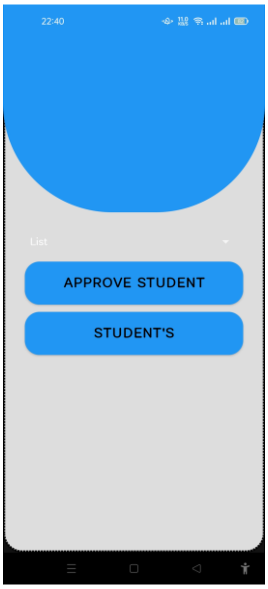
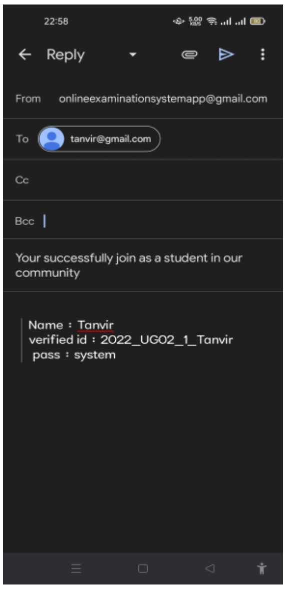


Fig 9.1.7: Manage Applicant

Admin will receive the request of the applicant then admin will send an email to the applicant for the password and id.





From onlineexaminationsystemapp@gmail.com

To tanvir@gmail.com

Cc

Bcc

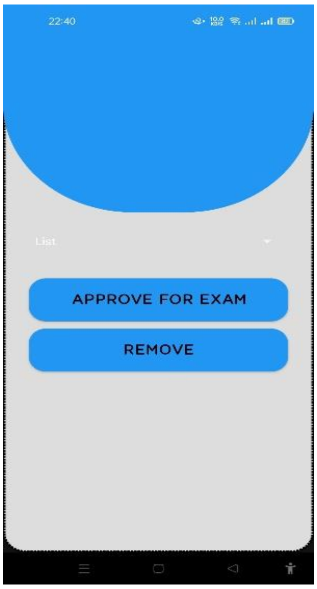
Your successfully join as a student in our community

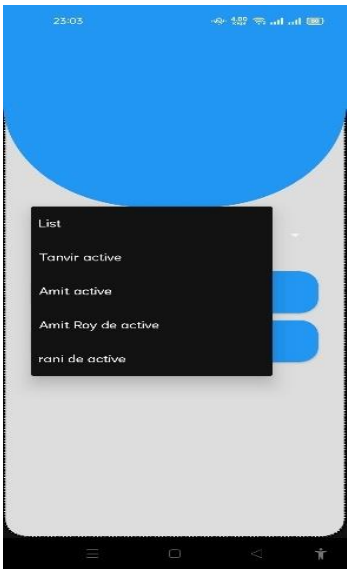
Name : Tanvir
verified id : 2022_UG02_1_Tanvir
pass : system

Fig 9.1.8: Approved Applicant

Fig 9.1.9: Send Password and ID For Applicant

Admin will approve the applicant for participating the examination.



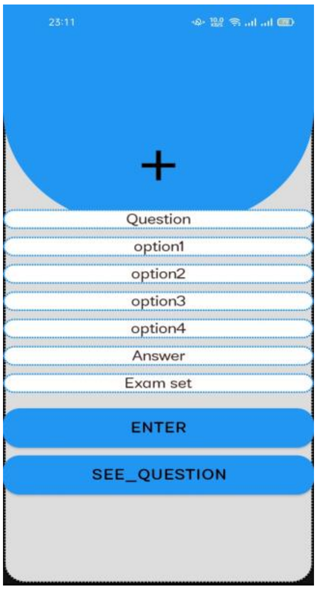


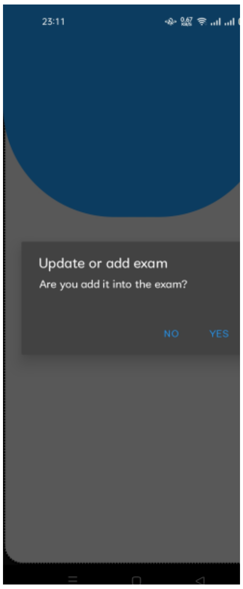
List
Tanvir active
Amit active
Amit Roy de active
rani de active

Fig 9.1.10: Approve Applicant

Fig 9.1.11: Student list

Admin will set the mcq questions here. Admin can update, delete or see questions. Admin can active and deactivate the exam here.





Update or add exam

Are you add it into the exam?

Fig 9.1.12: Set MCQ **Fig 9.1.13: Update/Delete Exam**

After making the mcq questions, admin will set the starting and ending time as well as set the date.

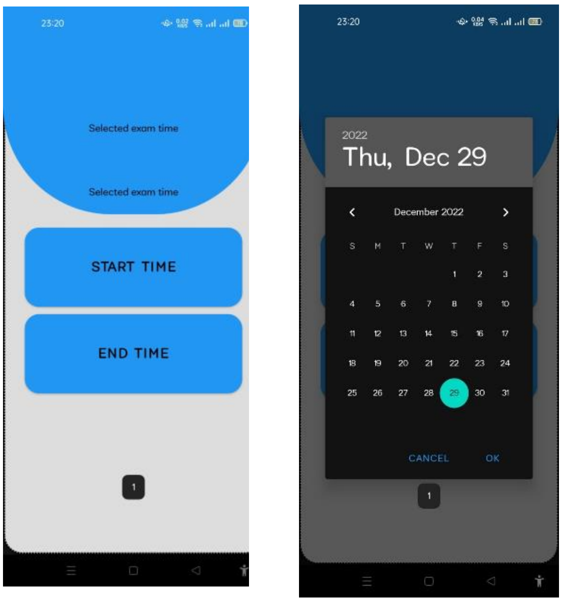


Fig 9.1.14: Set Judge and Time

Here admin can upload the open hand questions for the exam.

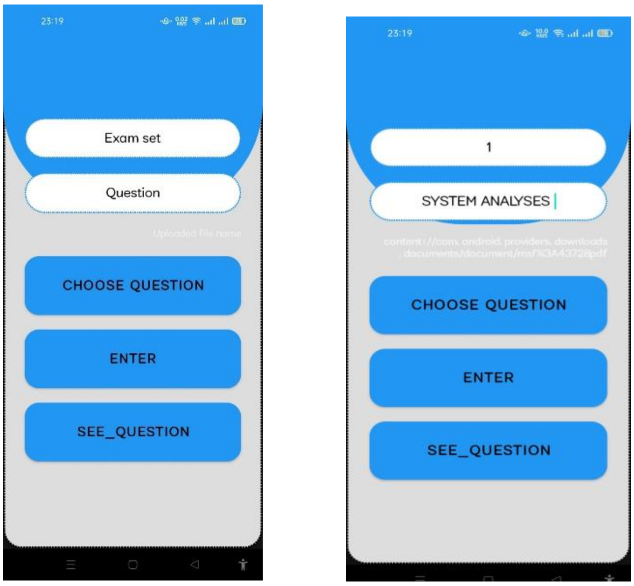
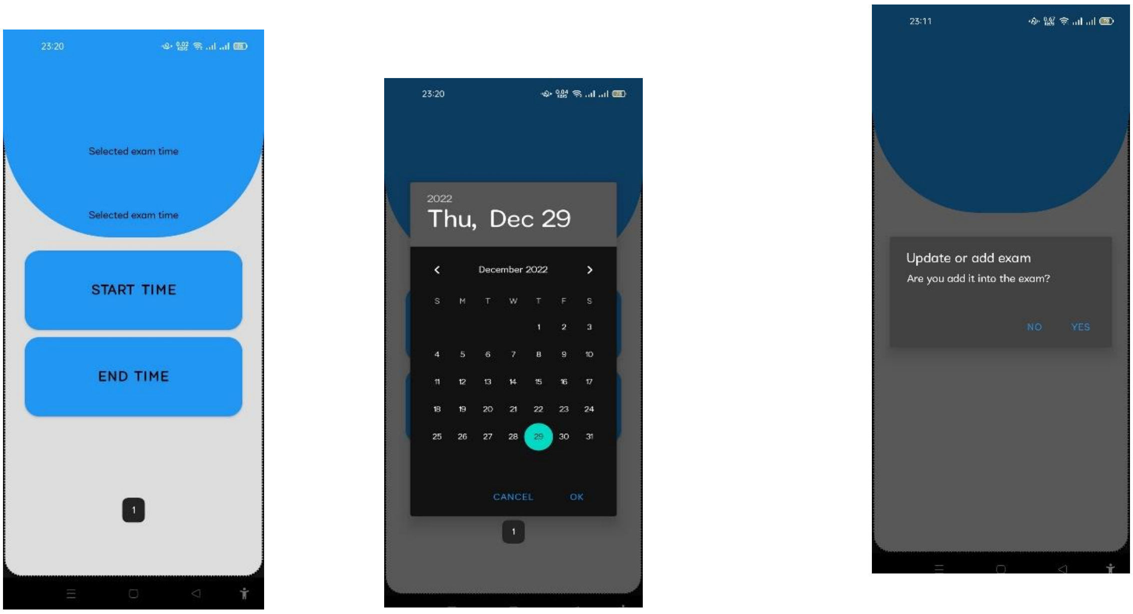


Fig 9.1.15: Manage Open Hand

Admin will set the judge and set the exam time

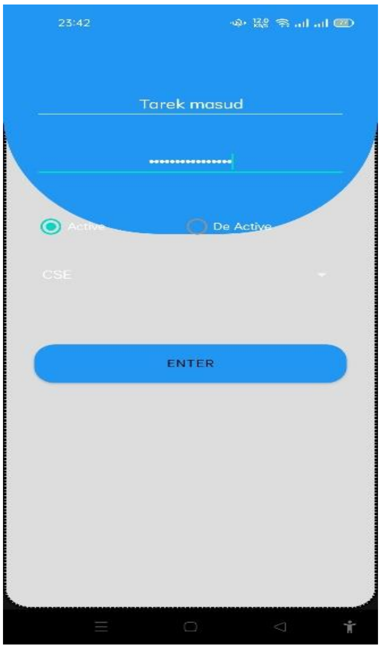


Update or add exam

Are you add it into the exam?

Fig 9.1.16: Set Judge and Time

Admin add the current active teacher in this dashboard



Tarek masud

●●●●●●●●●●●●●●●●

● Active ○ De Active

CSE

Fig 9.1.17: Add teacher

Admin can update teacher information and delete in this dashboard

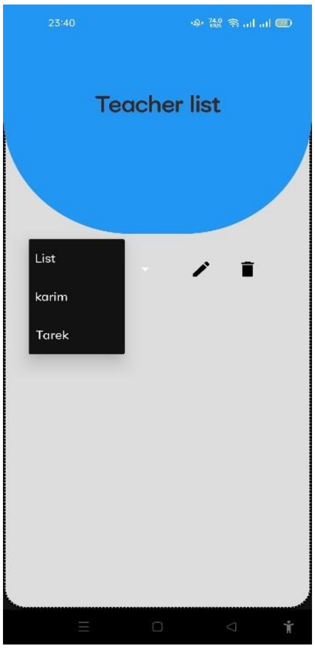
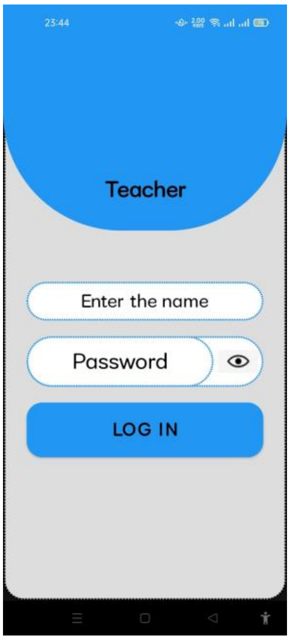


Fig 9.1.18: Update/Delete Teacher Info

9.2 Teacher Dashboard

Teacher received the password and then he can log into the apps



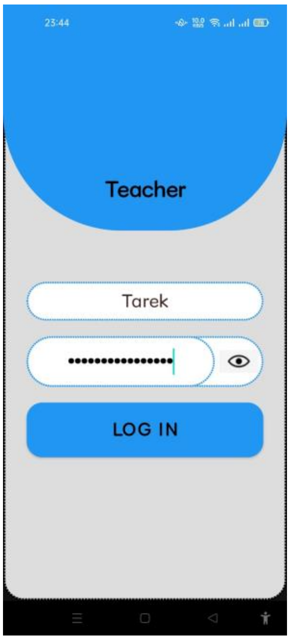


Fig 9.2.19: Received password **Fig 9.2.20: Teacher Login**

This is the dashboard of the teacher. Teacher can change his password and view the examination.

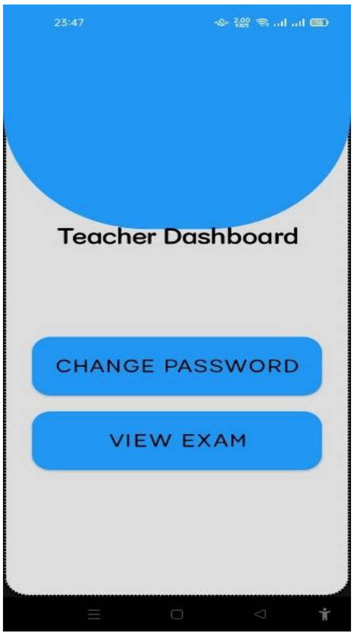


Fig 9.2.21: Teacher Dashboard

Teacher will check the exam and wait for the end the exam.

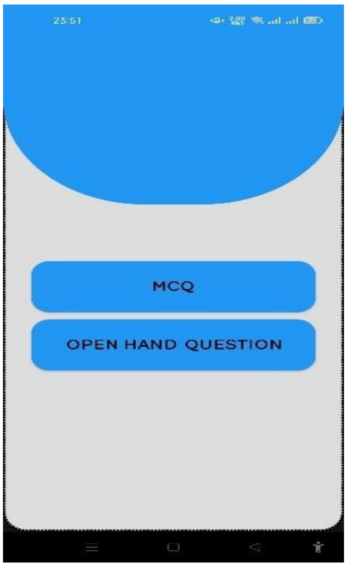


Fig 9.2.22: View Exam

Then teacher can evaluate the exam paper and submitted the exam marks to admin

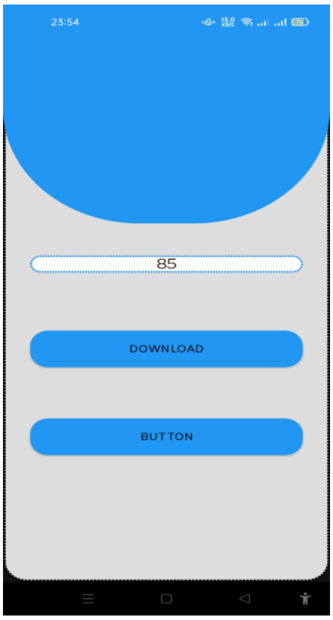
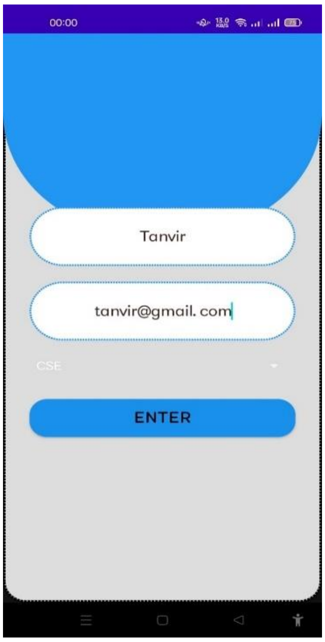


Figure 9.2.23: Submit Marks

9.3 Applicant Dashboard

Applicant send a request to the admin for getting their id and password



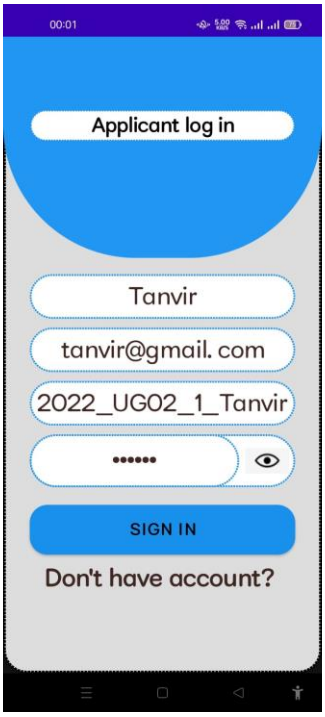
Tanvir

tanvir@gmail.com

CSE

Fig 9.3.1: Applicant Request

Applicant received the ID and password from admin. Now applicant can login



Tanvir

tanvir@gmail. com

2022_UG02_1_Tanvir

••••••

Fig 9.3.2: Applicant gets permission for login

After log of the applicant they can check the exam date for Attend the exam.

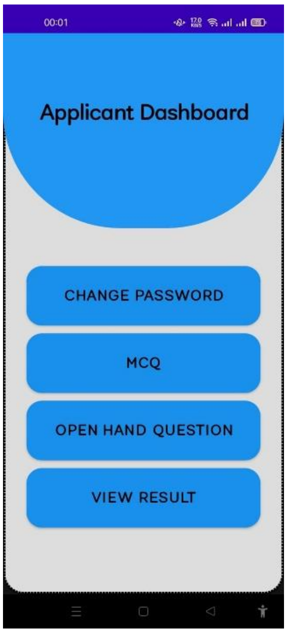
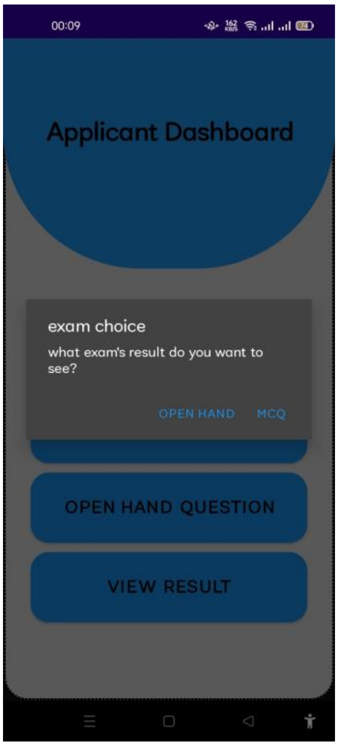


Fig 9.3.3: Applicant Dashboard

If the result published, applicants can see the exam result.

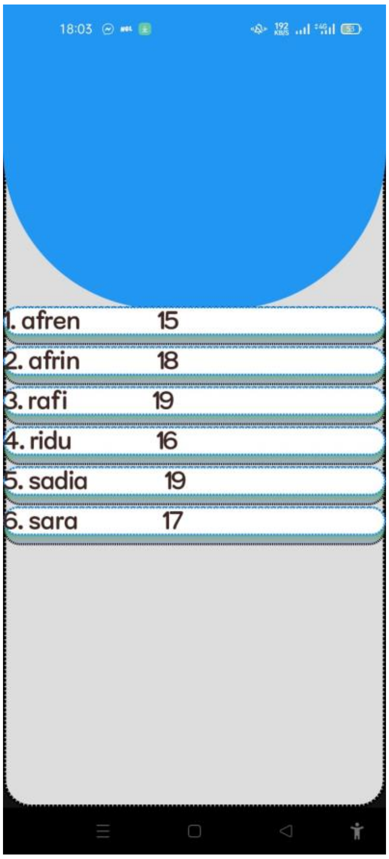


exam choice

what exam's result do you want to see?

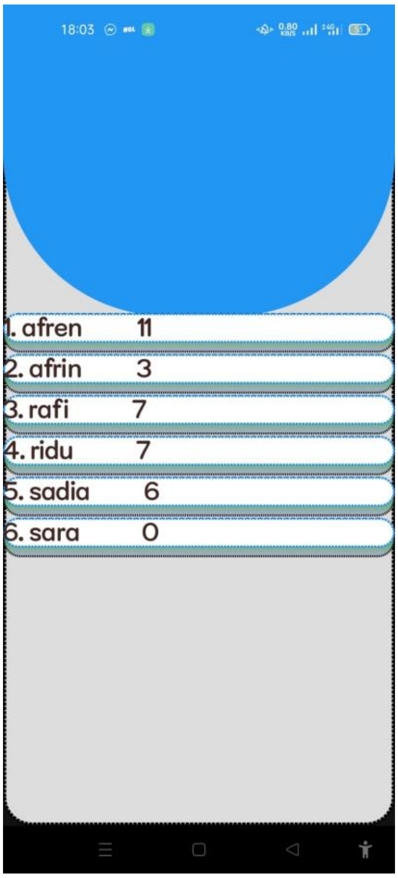
Fig 9.3.4: MCQ/CQ Result

Here is the result of MCQ and OPEN HAND marks of FOOD Deparment



1. afren	15
2. afrin	18
3. rafi	19
4. ridu	16
5. sadia	19
6. sara	17

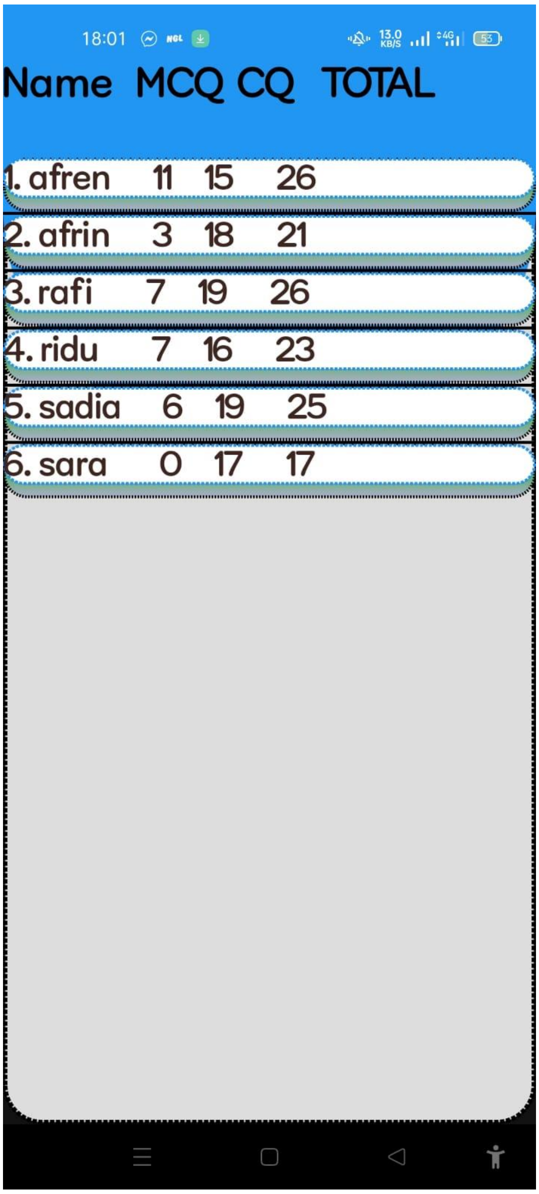
Fig 9.3.5 :open hand food department marks



1. afren	11
2. afrin	3
3. rafi	7
4. ridu	7
5. sadia	6
6. sara	0

Fig9.3.6:mcq marks of food department

Here is the Total marks of FOOD Deparment

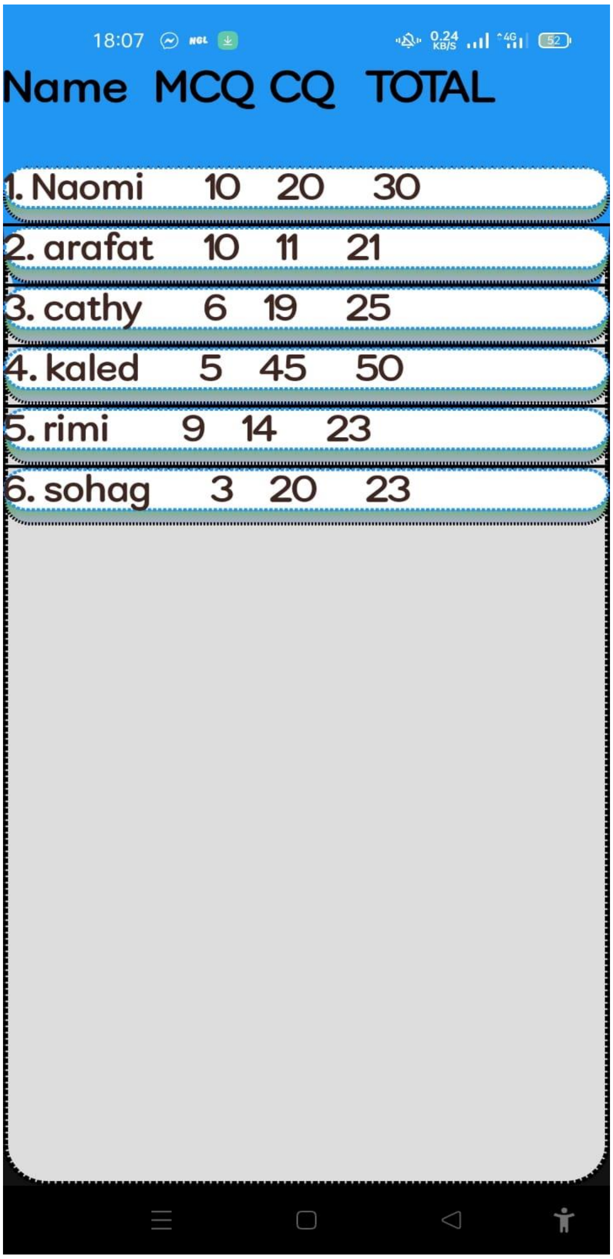


18:01

Name	MCQ	CQ	TOTAL
1. afren	11	15	26
2. afrin	3	18	21
3. rafi	7	19	26
4. ridu	7	16	23
5. sadia	6	19	25
6. sara	0	17	17

Fig 9.3.7 : Total marks of food department

Here is the total marks of CSE Deparment



Name	MCQ	CQ	TOTAL
1. Naomi	10	20	30
2. arafat	10	11	21
3. cathy	6	19	25
4. kaled	5	45	50
5. rimi	9	14	23
6. sohag	3	20	23

Fig 9.3.8: Total marks of CSE department

CHAPTER 10
Testing and Validation

10. Testing and Validation

10.1 Introduction

System testing is a level of testing that verifies a complete and fully integrated software product. The purpose of the system test is to evaluate the specification of the end-to-end system. System testing is actually a series of different tests whose has sole purpose is to exercise a complete computer-based system.

Testing is the process of running a program with the intention of finding an error. Like unit testing. Integration testing etc.

10.2 Unit Testing

Unit testing is defined as an individual testing system. The unit system to design each unit of the system. System testing is a fully integrated system test that evaluates compliance with specific system requirements. System testing. As its input, accepts all the integrated components that pass the integration testing.

10.3 Integration Testing

Integration testing is the stage of software testing where individual modules are assembled and tested as a group. System integration testing is performed to verify the overall software system performance.

All test portions of this code are successful.

10.4 Validation

The goal of verification is that the system works in a way that can be responsibly expected by the users. A test has been performed to verify compatibility for user requirements regarding ID and password which should be valid through the server and the ID and password should be matched.we have verified the validity of both the front end and back end from the data.

Chapter 11

Conclusion and Future Implementation

11.1 Conclusion:

We developed a system that online exam system. The main goal of this System is to decrease the workload of a teacher. In this modern world, the teacher has to do a lot of work to maintain quality education. There is no time to waste or extra pressure. At the same, we have to do this work accurately also.

In the system, a teacher has to take an extra workload for the Online exam system. That's why we claimed to do something that will decrease the work and pressure. Also, increase the accuracy rate of work. tried to give my best to build this system. we maintained the full work process of a project. At first, did a feasibility analysis to show the difference between the manual system and the present system, then have done the UML Diagram to clarify the user. Then we made a Data Flow Diagram to maintain the process and database more clearly to maintain good design. Then we applied these to the code. Then tested with two testing methods. After successfully testing, we declared it valid. After all the work and testing have been done, we assure you that this system will decrease the workload and also the system's output will be accurate But still, we need some improvements in the future. Though some processes and options have to be implemented in the future, those are described in the next sub-chapter.

have grateful to our supervisor **(Muhammad Masud Tarek),** who has assisted in completing our project work in every way possible.

11.2 Future Developments:

After advanced technology in every aspect of life and to comfort man's life these small electronic machines like computers have become an indispensable part of our lives. This system can be used on a large scale. Interface design can be more user-friendly in further.

REFERENCES

[1] https://stackoverflow.com/

[2] https://app.diagrams.net/

[3] https://developer.android.com/ml?gclid=EAIaIQobChMIs7SL4NWl_AIVwQ0rCh3JWAnUEAAYASABEgL6IPD_BwE&gclsrc=aw.ds

[4] https://www.w3schools.blog/android-tutorial

[5] https://www.w3schools.com/java/

[6] https://developer.android.com/design

[7] https://www.google.com/

[8] https://www.youtube.com/

[9] https://stackoverflow.com/questions/12947620/email-address-validation-in-android-on-edittext

[10] https://www.gantt.com/creating-gantt-charts