CRR-COE IN-TOUCH

(MOBILE APP FOR COLLEGE (ANDROID AND IOS))

A PROJECT REPORT

Submitted by

G VAMSI SAI
RegNo:316177111036

B RAJA SEKHAR
RegNo:316177111012

K AKHIL
RegNo:316177111052

G ANUSHA
RegNo:316177111045

Under the Esteemed Guidance of

D SRINIVASA RAO

Assistant Professor

In fulfillment of the requirements for the award

of

BACHELOR OF TECHNOLOGY

in

INFORMATION TECHNOLOGY

DEPARTMENT OF INFORMATION TECHNOLOGY

SIR C R REDDY COLLEGE OF ENGINEERING

(Approved by AICTE, Affiliated to Andhra University)

2020

SIR C R REDDY COLLEGE OF ENGINEERING

ELURU

DEPARTMENT OF INFORMATION TECHNOLOGY

BONAFIDE CERTIFICATE

This to certify that this project report "**CRR-COE IN-TOUCH" (MOBILE APP FOR COLLEGE -ANDROID AND IOS)** being

Submitted by

G VAMSI SAI	316177111036
B RAJA SEKHAR	316177111012
K AKHIL	316177111052
G ANUSHA	316177111045

In partial fulfillment for the award of the Degree of Bachelor of Technology in Information Technology to the Andhra University is a record of bonafide work carried out under my guidance and supervision.

The results embodied in this project report have not been submitted to any other University or Institute for the award of any Degree.

D SRINIVASA RAO M.Tech
PROJECT GUIDE
DEPARTMENT OF IT
Sir C R Reddy College of Engineering

Dr. S. KRISHNA RAO M.Tech,PhD
HEAD OF THE DEPARTMENT
DEPARTMENT OF IT
Sir C R Reddy College of Engineering

TABLE OF CONTENTS

ACKNOWLEDGEMENT

We wish to express our sincere thanks to various personalities who were responsible for the successful completion of this project.

We thank our principal **Prof. Dr. G SAMBA SIVA RAO** for providing the necessary infrastructure required for our project.

We are grateful to **Dr. S KRISHNA RAO**, Head of the Information Technology department, for providing the necessary facilities for completing the project in specified time.

We express our deep felt gratitude to **Sri. D SRINIVASA RAO**, as his valuable guidance and unstinting encouragement enabled us to accomplish our project successfully in time.

Our special thanks to **Mr.D.S.R.K.V.Prasad**, librarian and to the entire library staff of Sir C R Reddy College of Engineering, for providing the necessary library facilities.

We express our earnest thanks to faculty members of IT for extending their valuable support.

PROJECT MEMBERS

G VAMSI SAI (316177111036)

B RAJA SEKHAR(316177111012)

K AKHIL(316177111052)

G ANUSHA(316177111045)

ABSTRACT

The main purpose of this Mobile Application is to maintain the Transparency between the Teaching Staff and the Students of Sir C R Reddy College of Engineering. This Mobile application is a type of College management system and this was developed by using Flutter -a cross platform app development SDK, and hence with single codebase using dart we developed application that will run on both android and IOS platforms

There are 2 different types of users for this application those are Teaching Staff and Students

This App has the following modules

Teaching Staff

- Posting important Announcements
- Setting daily quizzes for students at a particular time, and getting the results after students attempted
- Posting important articles on wall of the Application
- Fetching attendance of a total class on a particular input date
- Directly taking attendance of Students and posting Them

Students

- Fetching announcements made by Staff
- Attempting daily quizzes and getting key of exam after some time set by the staff
- Keeping Track of articles posted by Staff
- Fetching Monthly attendance percentage

1. **INTRODUCTION**

Now-a-days, the communication between students and faculty within an organisation is very much essential. At present the information shared by the faculty to the students through email and other communication platforms. Attendance is managed through paper-based system.

Here, in this application the communication between students and faculty is very effective regarding academic information.

Eliminating paper-based system, in this application Teachers can directly take attendance through app and students can easily check their attendance percentage in a month, and, teachers can fetch the attendance of a class on aparticular date

To increase standards of learning in college, In this application we developed a Quiz module by which Teachers can set some quizzes every day and the students can attempt them and results will be reched to teachers of the respective department after all students attempted the quiz, a student can attempt a quiz only once in the time slot fixed by the staff and later on after an hour of given time for quiz, Students can get their quiz key which will be displayed on key page for a day

All quizzes notifications and announcements made by teachers will be displayed of wall of app, and Teachers can share important articles on the wall of app with a photo and some descriptions which will be displayed to all users of application

2. PROBLEM SPECIFICATION

2.1EXISTING SYSTEM:

- The Existing system facilitates the students to view the academic related data i.e. posted by the concerned staff members, like Attendance, syllabus etc.via paper-based system
- Currently Taking attendance of a class and, knowing attendance percentage of a student will be conducted through paper-based system
- To improvise and revise the class-room learnt content is going through paper-based tests which are ineffective

2.2 PROPOSED SYSTEM:

- This application allows students to view the academic information and can keep track of important articles in digital form rather than Paper-based system
- By eliminating paper-based system, we introduced taking attendance directly through this application
- Paper-based system of checking skills through direct claas-room assignments, will be eliminated by quizzes conducted in this system, students can easily get the key and results to improve their skills

3. REQUIREMENT SPECIFICATION:

3.1 FUNCTIONAL REQUIREMENTS:

- The faculty should be able to login
- The faculty should be able to upload documents, assignments, and important articles
- The faculty should be able to post important notices.
- The faculty can take attendance of a class students
- The faculty can assign questions in form of quizzes for particular class students at particular time.
- The faculty can get results after students attempted the quizzes
- The student should be able to login.
- The student should be able to view the information.
- The students can attempt Quizzes.
- The students can get keys of quizzes, they were assigned to attempt
- The student can able to his/her monthly attendance percentage.

3.2 Non-Functional Requirements:

Usability:

- User can use the application effectively.
- Easy to learn and remember how to use.

Security:

- Application must provide unique id, password to prevent the system from Unauthorized access.
- Application backend firebase service provides best security rules to keep data safe

Performance:

- The application provides high performance for every instruction send by the user.
- The application response time for a post uploading is fast as it will take only few seconds

Availability:

- The application should always be available for access at 24 hours,7days a week

SYSTEM REQUIREMENTS:

3.3 SOFTWARE REQUIREMENTS:

Operating System	: Windows 7 or above
Programming Language	:Dart
Tools used	: Flutter SDK V1.12.13 (11/2019) with dart 2.7.2 Extension in Visual Studio code and Android Studio for Emulator
Database	: FIREBASE REALTIME DATABASE (NOSQL) PROVIDED BY Google cloud
Other Devices	:Android or IOS smartphone(optional)

3.4 HARDWARE REQUIREMENTS:

Processor used	: Intel xenon.
RAM required	: 8 GB or above
Hard Disk	: 160 GB.

4.ANALYSIS

Analysis is the phase where the system observes the feasibility of system. Development is the cost efficient based on the management approval, and then design, coding phases will be executed.

Analysis phase delivers requirement specification. The system specification serves as an interface between designer and developer as well as between developer and user. This describes external behavior of software without bothering the internal implementation.
Problem analysis is performed to getting a clear understanding of the needs of the clients and the users and what exactly desired from the software. Analysis leads to actual specification.
During the process of analysis, a massive amount of the information is collected in the form of interviews and information from documentation and so forth. One of the information can be effectively evaluated for completeness and consistency.

4.1MODELUES

After careful analysis the system has been identified to have the following modules:

4.1 .1FACULTY

4.2 .2STUDENTS

4.1 FACULTY: FACULTY module should be used by the faculty, they were able to send academic resources like Announcements, Notices, Assignments, Timetables, and other announcements. Faculty can also directly take attendance of a class, and can fetch the attendance of a class on particular date and Faculty can assign quizzes at any particular time to any patucular class studnets

4.2 STUDENTS: STUDENTS module must used by the students, they can keep track of information sent by Faculty, about Announcements, Notices, Assignments, Timetables, Placement related information, They can directly view their attendance percentage of a that month, Students can attempt quizzes assigned by the faculty and can get key of that quizzes after all students attempted

4.2 FEASIBILITY STUDY

4.2.1 TECHNICAL FEASIBILITY

In Technical Feasibility study, one has to test whether the proposed system can be developed using existing technology or not. It is planned to implement the proposed system using FLUTTER, FIREBASE(ON TEST MODE WITH LIMITED DATA REQUIRED FOR DEVELOPMENT). It is evident that the necessary hardware and software are available for development and implementation of the proposed system. Hence, the solution is technically feasible.

4.2.2 ECONOMICAL FEASIBILITY

As part of this, the costs and benefits associated with the proposed system are compared, and the project is economically feasible since all the tools (FLUTTER, FIREBASE REALTIME DATABASE (ON TEST MODE WITH LIMITED DATA REQUIRED FOR DEVELOPMENT). used are of open source software. The system development costs will be significant. So, the proposed system is economically feasible.

4.2.3 OPERATIONAL FEASIBILITY

In Operational feasibility there are limitations between modules. Direct alerts with notifictions will not sent if any announcement made but as announcements are directly available on wall of the applcattion they can easily get them. So, the proposed system is operationally feasible.

5. DESIGN

UML DIAGRAMS

Unified Modeling Language (UML) is a method for describing the system architecture in detail using the blueprint. It represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems. It is a very important part of developing objects-oriented software and the software development process. It uses mostly graphical notations to express the design of software projects.

UML helps project teams communicate, explore potential designs, and validate the architectural design of the software.

UML is a general-purpose visual modeling language that is used to specify, visualize, construct, and document the artifacts of the software system.

USECASE DIAGRAM

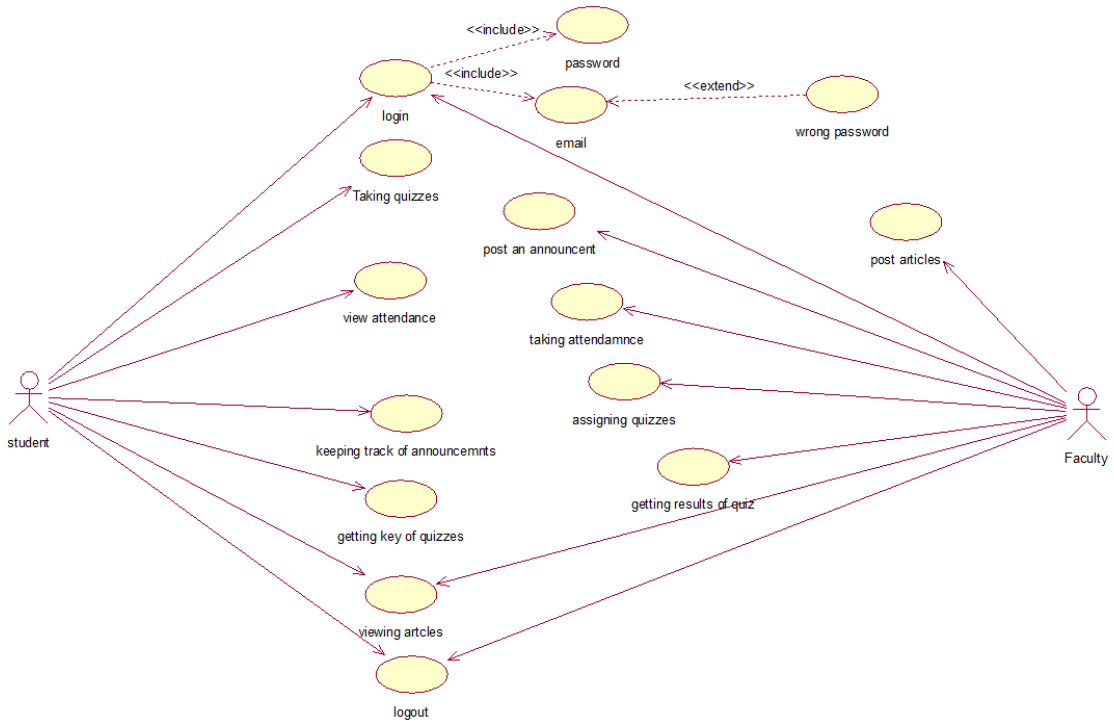


Fig 5.1

In this use case diagram, the actions performed by the users are clearly shown. The actions done by the students are login, taking quizzes, fetcing articles,etc are handled by Students . The actions done by the Faculties are post notices, assigning quizzes etc. as mentioned in use case diagram

Identified Actors and use cases:

Actors	use cases
Sudents	taking quizzes, fetching attendance, keeping track of articles etc…
Faculty	Getting results, taking attendance etc…

CLASS DIAGRAM:

The UML class diagram also refer to as object modeling, is the main static analysis diagram. Object modeling is the process by which the logical objects in the real world are represented by the actual objects in the program. These diagrams show a set of classes, interfaces, collaborations and their relationships.

Class diagrams commonly contain the following things:

- Classes
- Interfaces
- Collaborations
- Dependency, Generalization and association relationships.

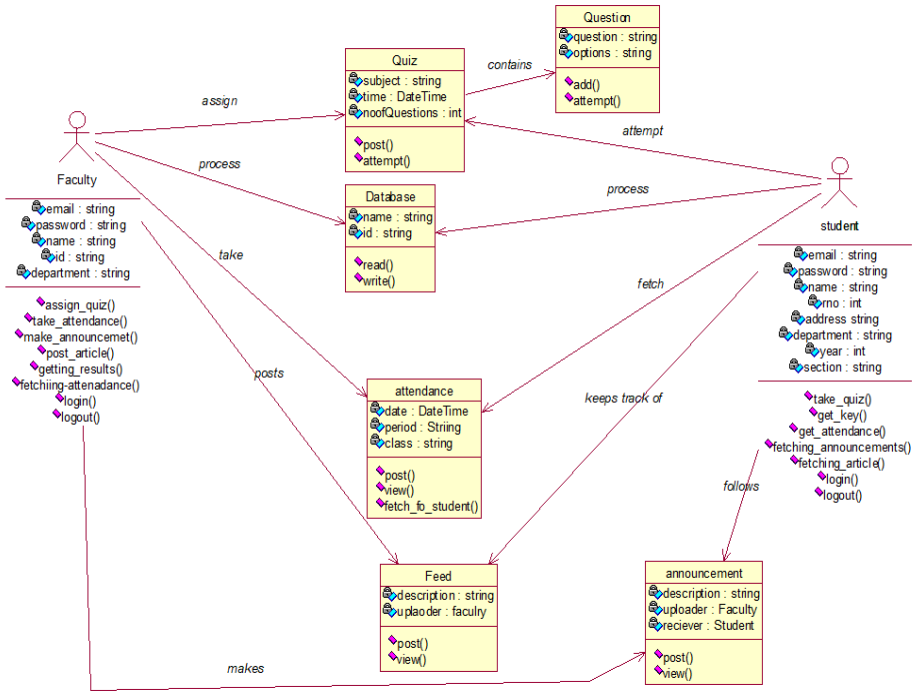


Fig 5.2

ACTIVITY DIAGRAM:

An activity model is similar to state chart diagram, where a token represents an operation. An activity is shown as a round box, containing the name of the operation. It is used mostly to show the internal state of an object, but external events may appear in them. The purpose of an activity diagram is to provide a view of flows and what is going on inside a use case or among several classes.

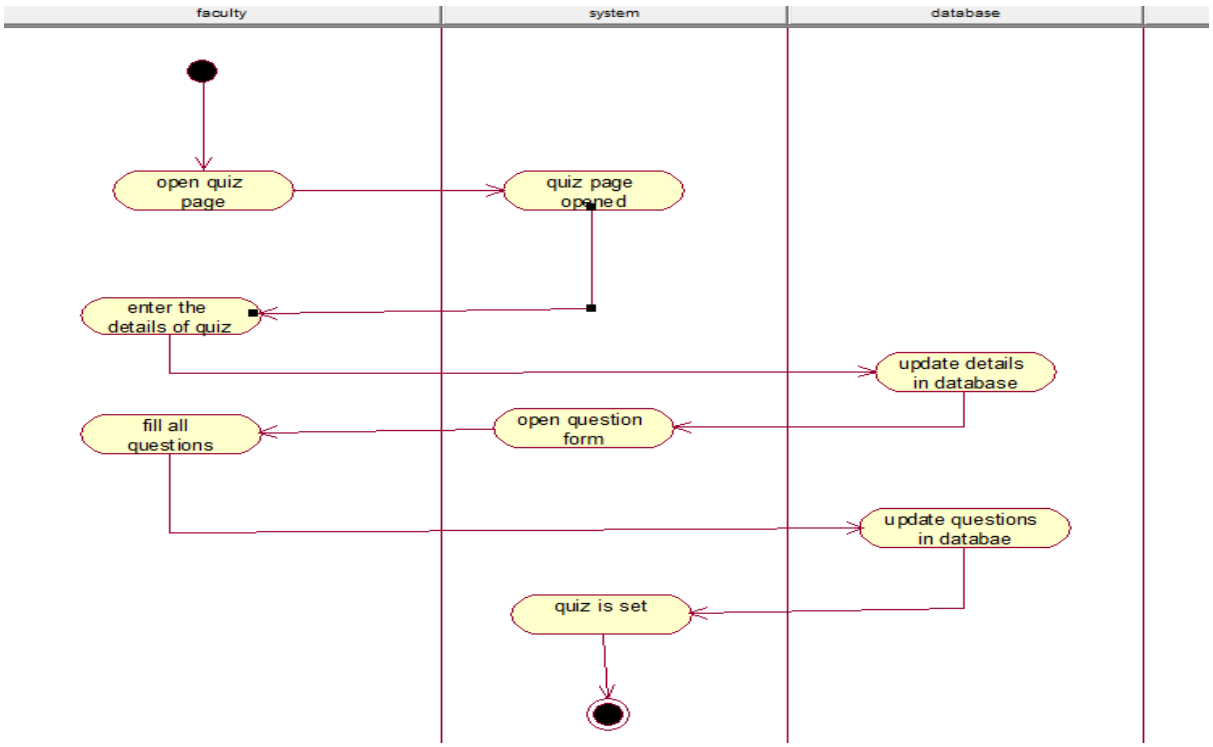


Fig 5.3 Activty Diagram for Assigning Quiz by faculty

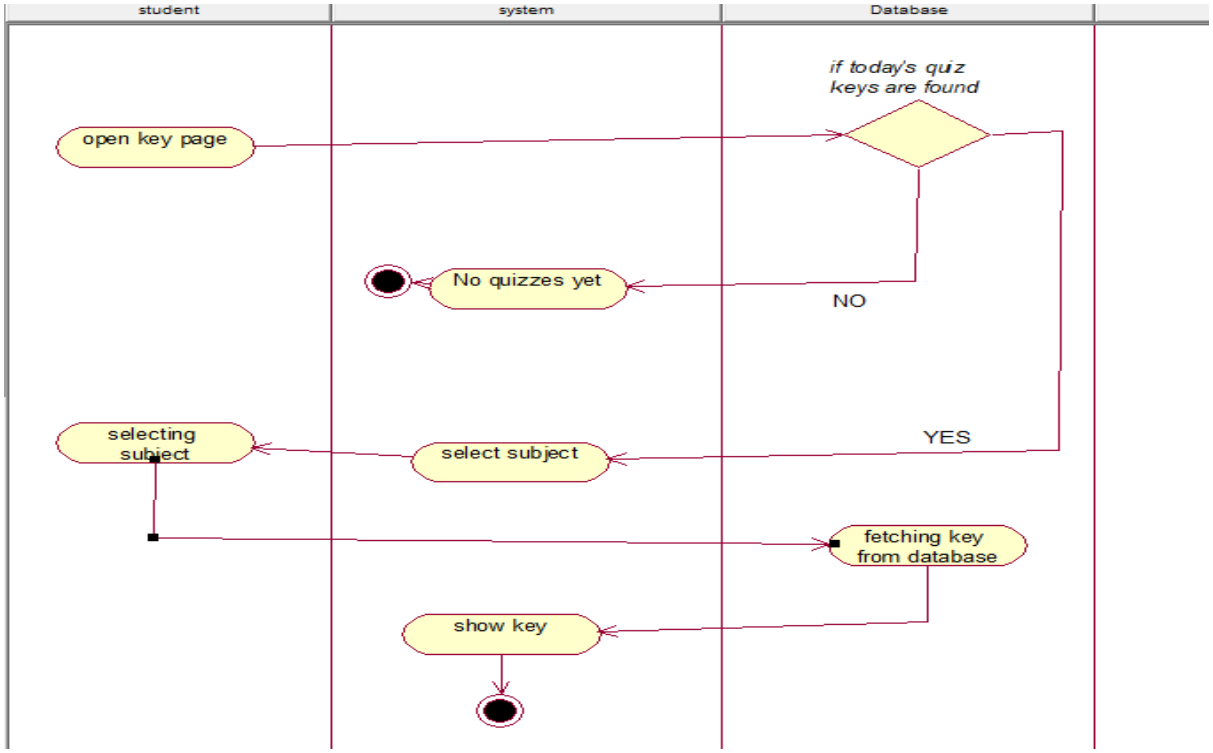


Fig 5.4 Activty Diagram for Getting key by students

SEQUENCE DIAGRAMS:

A sequence diagram simply depicts interaction between objects in a sequential order i.e. the order in which these interactions take place. We can also use the terms event diagrams or event scenarios to refer to a sequence diagram. Sequence diagrams describe how and in what order the objects in a system function. These diagrams are widely used by businessmen and software developers to document and understand requirements for new and existing systems.

Sequence Diagram for Student Login:

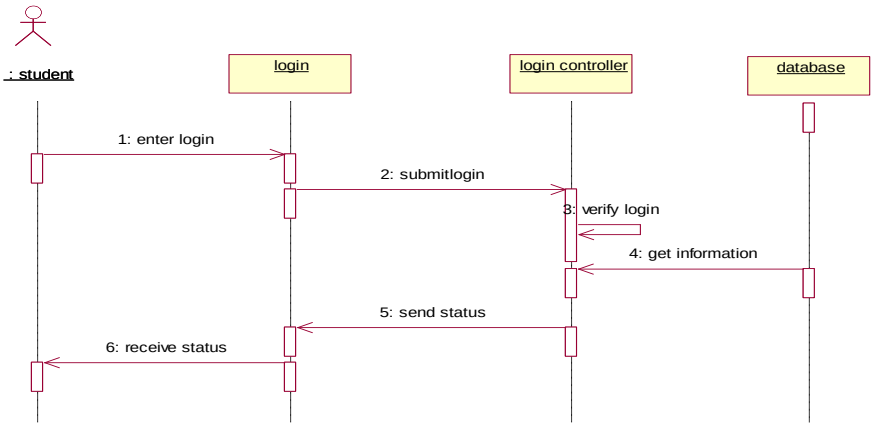


Fig 5.5

Student enters username and password and submits0 the form which reaches to login controller and form database details and retrieved and validated check details at login controller. After successful login administrator is intimated.

Sequence Diagram for upload data:

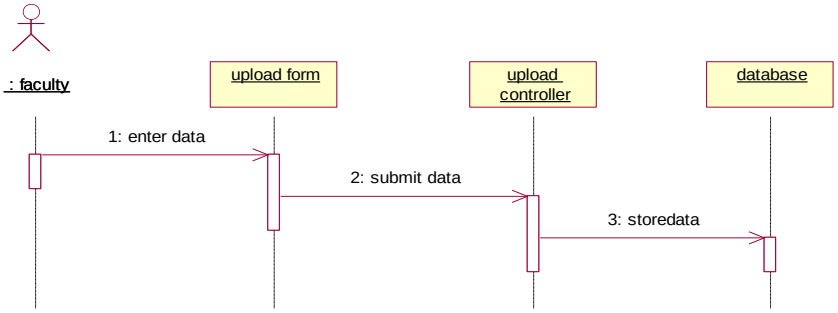


Fig 5.6

COLLABORATION DIAGRAMS:

Collaboration diagram for Student Login:

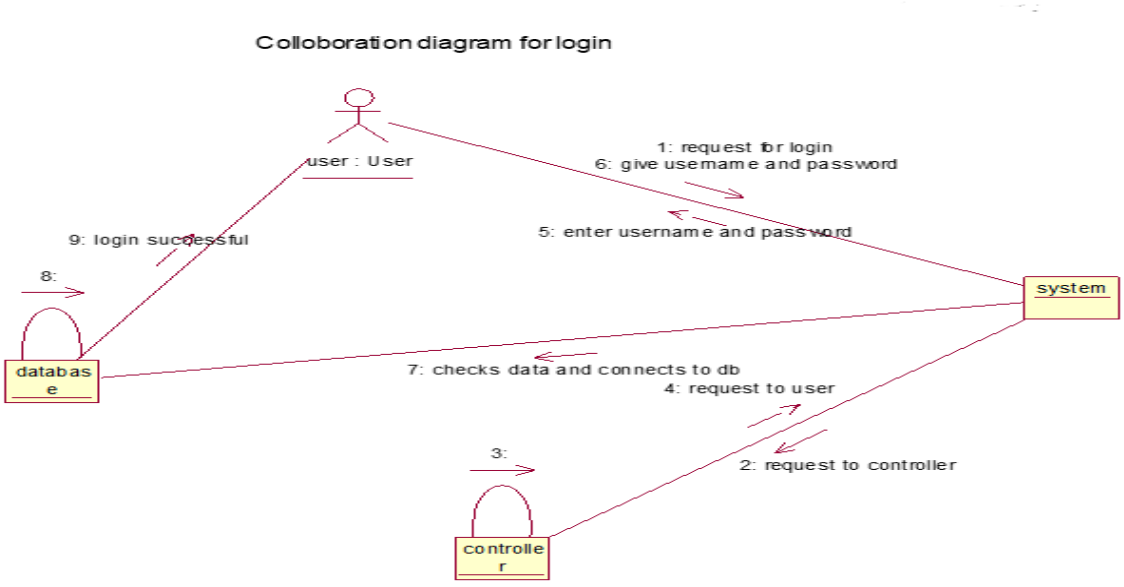


Fig 5.7

Collaboration diagram for upload data:

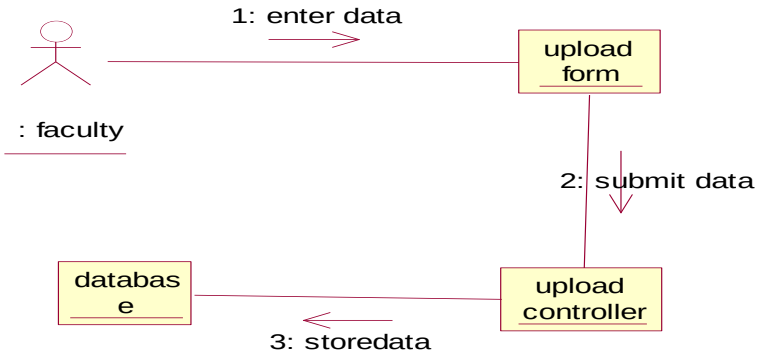


Fig 5.8

STATE CHART DIAGRAM:

A **state diagram** is used to represent the condition of the system or part of the system at finite instances of time. It's a **behavioral** diagram and it represents the behavior using finite state transitions.

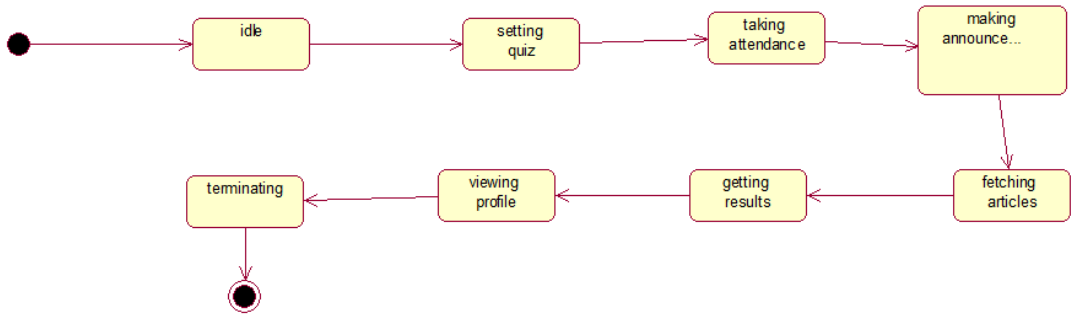


Fig 5.9 State Chart Diagram for Faculty

Fig 5.10 State Chart Diagram for Faculty

COMPONENT DIAGRAM:

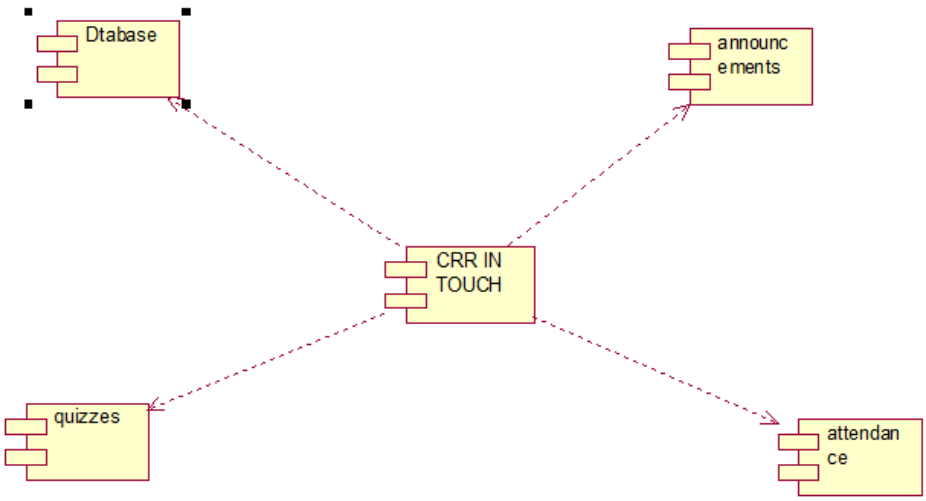


Fig 5.11

DEPLOYMENT DIAGRAM:

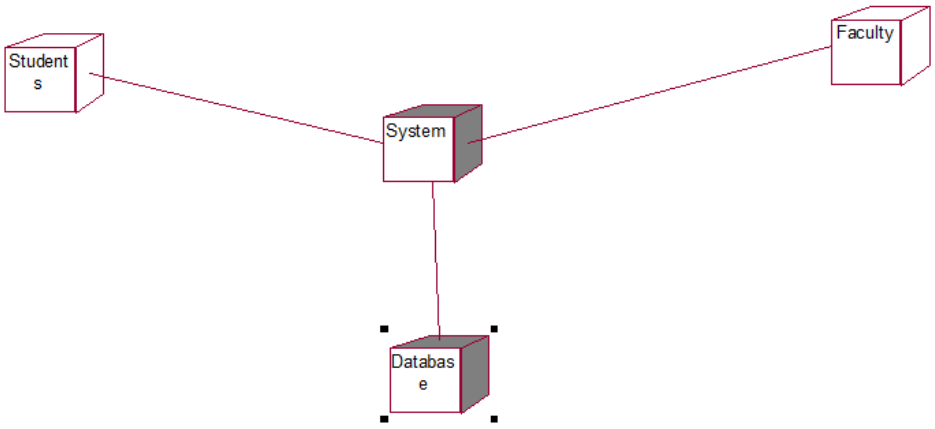


Fig 5.12

6.CODING

6.1 Modals

announcement.dart

```
import 'package:flutter/cupertino.dart';
import '../modals/staff_class.dart';

class Announcement {
  final String announcement;
  final int toStudentYear;
  final String toStudentSection;
  final Staff uploader;
  final DateTime time;
  Announcement(
      {@required this.announcement,
      @required this.toStudentYear,
      @required this.toStudentSection,
      @required this.uploader,
      @required this.time});
}
```

6.2 Providers

auth.dart

```
import 'dart:convert';
import 'dart:async';

import 'package:flutter/widgets.dart';
import 'package:http/http.dart' as http;
import 'package:shared_preferences/shared_preferences.dart';

class Auth with ChangeNotifier {
```

```
  String _token;
  DateTime _expiryDate;
  String _userId;
  Timer _authTimer;
  bool isTeacher;

  bool get isAuth {
    return token != null;
  }

  String get token {
    if (_expiryDate != null &&
        _expiryDate.isAfter(DateTime.now()) &&
        _token != null) {
      return _token;
    }
    return null;
  }

  String get userId {
    return _userId;
  }

  Future<void> _authenticate(
      String email, String password, String urlSegment) async {
    final url =

'https://www.googleapis.com/identitytoolkit/v3/relyingparty/$urlSegment?key=AIzaSyAqjdPj13sI35w1YuRVWSnntyzbI4N0Fg4';
    try {
      print('okokokok');
      final response = await http.post(
        url,
        body: json.encode(
```

```
        {
          'email': email,
          'password': password,
          'returnSecureToken': true,
        },
      ),
    );
    final responseData = json.decode(response.body);
```

6.3 Drawer files

drawer_theme.dart

```
import 'package:flutter/material.dart';

TextStyle listTitleDefaultTextStyle = TextStyle(color: Colors.white70, fontSize: 20.0,
fontWeight: FontWeight.w600);
TextStyle listTitleSelectedTextStyle = TextStyle(color: Colors.white, fontSize: 20.0,
fontWeight: FontWeight.w600);

Color selectedColor = Colors.amberAccent;
Color drawerBackgroundColor = Color(0xFF272D34);


];
```

6.4 Quiz

quizpage.dart

```
import 'dart:async';

import 'package:flutter/material.dart';
import 'package:flutter/services.dart';
```

```
import './resultpage.dart';


class GetJson extends StatelessWidget {
  // accept the langname as a parameter
List questions;
  final String subName;
  GetJson(this.subName,this.questions);


  // a function
  // sets the asset to a particular JSON file
  // and opens the JSON
  // setasset() {
  //   if (langname == "Python") {
  //     assettoload = "assets/python.json";
  //   } else if (langname == "Java") {
  //     assettoload = "assets/java.json";
  //   } else if (langname == "Javascript") {
  //     assettoload = "assets/js.json";
  //   } else if (langname == "C++") {
  //     assettoload = "assets/cpp.json";
  //   } else {
  //     assettoload = "assets/linux.json";
  //   }
  // }

}
```

6.5 Screens

staff_home.dart

```
import 'package:flutter/material.dart';
```

```
import 'package:provider/provider.dart';
import '../modals/feed_class.dart';
import '../providers/feed_provider.dart';
import 'package:intl/intl.dart';
import '../drawer/collapsing_navigation_drawer_widget.dart';
import './feed_upload.dart';
import './making_announcement.dart';

class StaffHome extends StatefulWidget {
  static const routeName = '/staff-home';
  @override
  _StaffHomeState createState() => _StaffHomeState();
}

class _StaffHomeState extends State<StaffHome> {
  List<Feed> _totalFeed = [];

  Future<void> _refreshFeed(BuildContext context) async {
    await Provider.of<FeedProvider>(context,listen: false).fetchNewsFeed().then((_) {
      _totalFeed = Provider.of<FeedProvider>(context,listen: false).items;
    });
  }




  Widget buildEachFeed(Feed data) {
    return Card(
      shape: RoundedRectangleBorder(
        borderRadius: BorderRadius.circular(15),
      ),
      elevation: 6,
      margin: EdgeInsets.all(10),
      child: Column(
```

```
children: <Widget>[
 Padding(
  padding: EdgeInsets.all(15),
  child: ListTile(
   title: Text(data.uploader.name),
   subtitle: Text(data.uploader.department),
   trailing: Text(
    DateFormat.yMEd().format(data.time) +
      '\n' +
      'at ' +
      DateFormat.Hm().format(data.time),
    style: TextStyle(fontSize: 16, fontWeight: FontWeight.w300),
   ),
  ),
 ),
 Padding(
  padding: EdgeInsets.all(15),
  child: Container(
    alignment: Alignment.topLeft,
    child: Text(
     data.description,
     style: TextStyle(fontSize: 18, fontWeight: FontWeight.normal),
    )),
 ),
 Stack(
  children: <Widget>[
   ClipRRect(
    borderRadius: BorderRadius.only(
     topLeft: Radius.circular(15),
     topRight: Radius.circular(15),
    ),
    child: FadeInImage(
     placeholder: AssetImage('assets/images/feed_placeholder.jpg'),
     image:NetworkImage( data.imageUrl),
```

```
                height: 250,
                width: double.infinity,
                fit: BoxFit.cover,
              ),
            ),
          ],
        ),
      ],
    ),              children: <Widget>[
              FloatingActionButton(
                heroTag: '+',
                onPressed: () {
                  Navigator.of(context).pushNamed(MakeAnAnnouncementScreen.routeName);
                },
                child: Icon(Icons.add_alert),
              ),
              FloatingActionButton(
                heroTag: '++',
                onPressed: () {
                  Navigator.of(context).pushNamed(FeedUploadScreen.routeName);
                },
                child: Icon(Icons.add),
              )
            ],
          ),
        )
      );
  }
}
```

main.dart

```
import 'package:flutter/material.dart';
import 'package:firebase_messaging/firebase_messaging.dart';
```

```
import './providers/feed_provider.dart';
import 'package:flutter_complete_guide/providers/quiz_entry_provider.dart';
import './screens/attendance_for_student.dart';
import './providers/quiz_provider.dart';
import './quiz/quizhome.dart';
void main() => runApp(MyApp());

class MyApp extends StatefulWidget {
  @override
  _MyAppState createState() => _MyAppState();
}

class _MyAppState extends State<MyApp> {
  final FirebaseMessaging _messaging = FirebaseMessaging();
  @override
  void initState() {
    print('haiii');
    _messaging.getToken().then((token) {
      print(token);
    });
    super.initState();
  }

  @override
  Widget build(BuildContext context) {
    return MultiProvider(
      providers: [
        ChangeNotifierProvider.value(
          value: Auth(),
        ),
```

pubspec.yaml

```
name: flutter_complete_guide
description: A new Flutter project.
```

```
version: 1.0.0+1

environment:
  sdk: ">=2.3.0 <3.0.0"

dependencies:
  flutter:
    sdk: flutter
  shared_preferences: ^0.5.6+2
  http: ^0.12.0+4
  provider: 3.0.0
  intl: ^0.16.1
  carousel_slider: ^1.3.0
  # The following adds the Cupertino Icons font to your application.
  # Use with the CupertinoIcons class for iOS style icons.
  cupertino_icons: ^0.1.2
  image_picker:
  cloud_firestore:
  firebase_core:
  firebase_database:
  firebase_messaging:
  firebase_storage:
  firebase_auth:
```

7. TESTING

INTRODUCTION:

After finishing the development of any computer based system the next complicated time consuming process is system testing. During the time of testing only the development company can know that, how far the user requirements have been met out, and so on.

Following are the some of the testing methods applied to this effective project:

SOURCE CODE TESTING:

This examines the logic of the system. If we are getting the output that is required by the user, then we can say that the logic is perfect.

SPECIFICATION TESTING:

We can set with, what program should do and how it should perform under various condition. This testing is a comparative study of evolution of system performance and system requirements.

MODULE LEVEL TESTING:

In this the error will be found at each individual module, it encourages the programmer to find and rectify the errors without affecting the other modules.

UNIT TESTING:

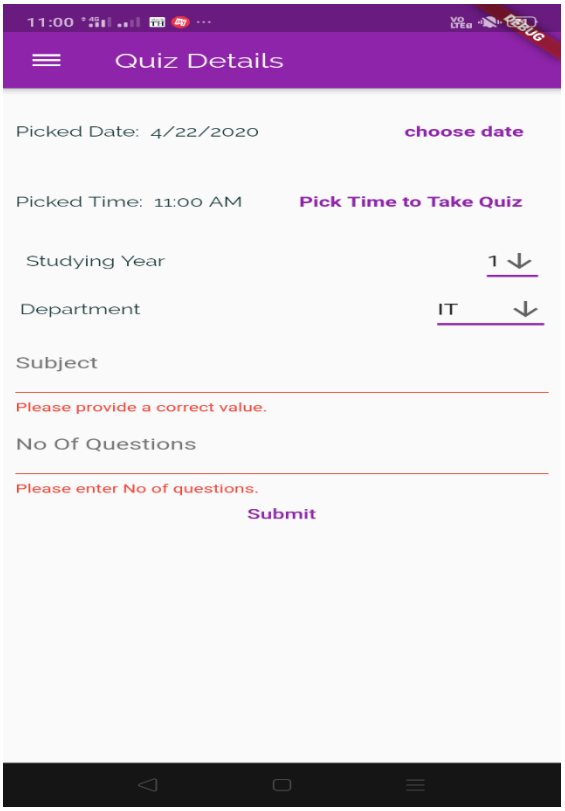
Unit testing focuses on verifying the effort on the smallest unit of software-module. The local data structure is examined to ensure that the date stored temporarily maintains its integrity during all steps in the algorithm’s execution. Boundary conditions are tested to ensure that the module operates properly at boundaries established to limit or restrict processing.

INTEGRATION TESTING:

Data can be tested across an interface. One module can have an inadvertent, adverse effect on the other. Integration testing is a systematic technique for constructing a program structure while conducting tests to uncover errors associated with interring.

VALIDATION TESTING:

It begins after the integration testing is successfully assembled. Validation succeeds when the software functions in a manner that can be reasonably accepted by the client. In this the majority of the validation is done during the data entry operation where there is a maximum possibility of entering wrong data. Other validation will be performed in all process where correct details and data should be entered to get the required results.



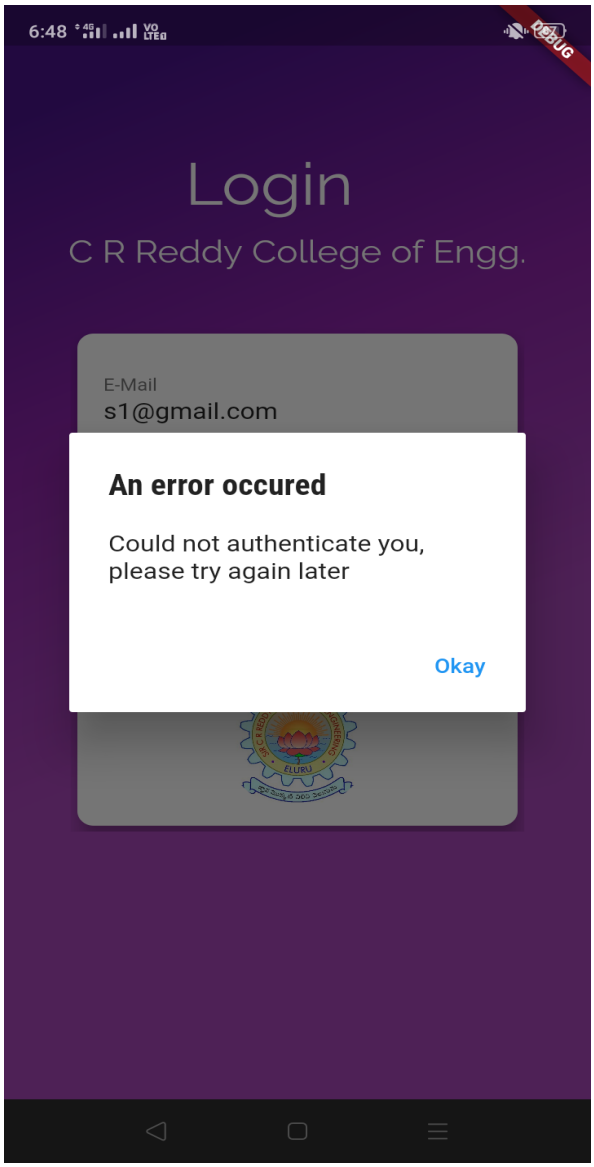
Validated forms

RECOVERY TESTING:

Recovery Testing is a system that forces the software to fail in variety of ways and verifies that the recovery is properly performed. If recovery is automatic, re-initialization, and data recovery are each evaluated for correctness.

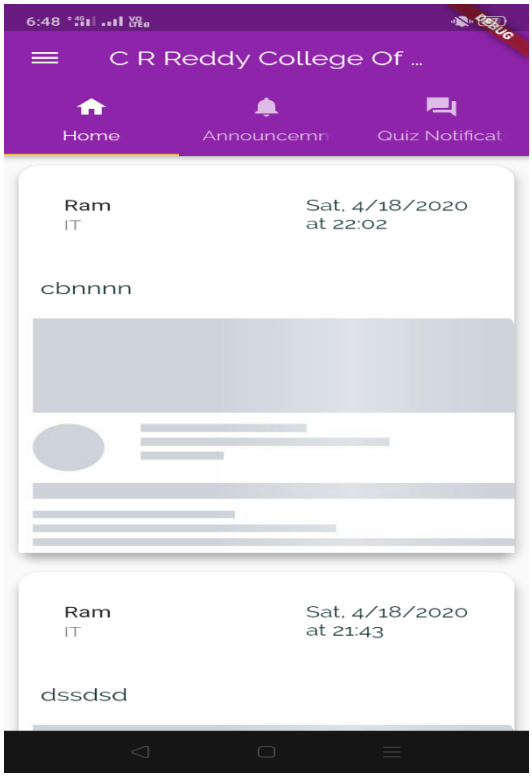
SECURITY TESTING:

Security testing attempts to verify that protection mechanism built into system will in fact protect it from improper penetration. The tester may attempt to acquire password through external clerical means, may attack the system with custom software design to break down any defenses to others, and may purposely cause errors.



PERFORMANCE TESTING:

Performance Testing is used to test runtime performance of software within the context of an integrated system. Performance test are often coupled with stress testing and require both software instrumentation. Boundary conditions are tested to ensure that the module operates properly at boundaries established to limit or restrict processing.



It achieves better performance by engaging the users when data is loading from server

BLACKBOX TESTING:

Black- box testing focuses on functional requirement of software. It enables to derive ets of input conditions that will fully exercise all functional requirements for a program.

Black box testing attempts to find error in the following category:

- Incorrect or missing function
- Interface errors
- Errors in data structures or external database access and performance errors.

OUTPUT TESTING:

After performing the validation testing, the next step is output testing of the proposed system since no system would be termed as useful until it does produce the required output in the specified format. Output format is considered in two ways, the screen format and the printer.

As The is built by using flutter, which is best knouwn for building best user Interfaces, The application performs well in output testing by giving required output in beuitiful interfaces

8. OUTPUT SCREENS

Login Screen

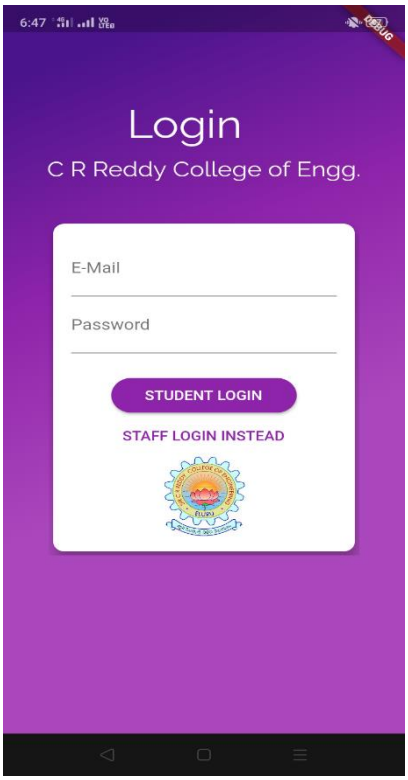


Fig.8.1

Profile Screen

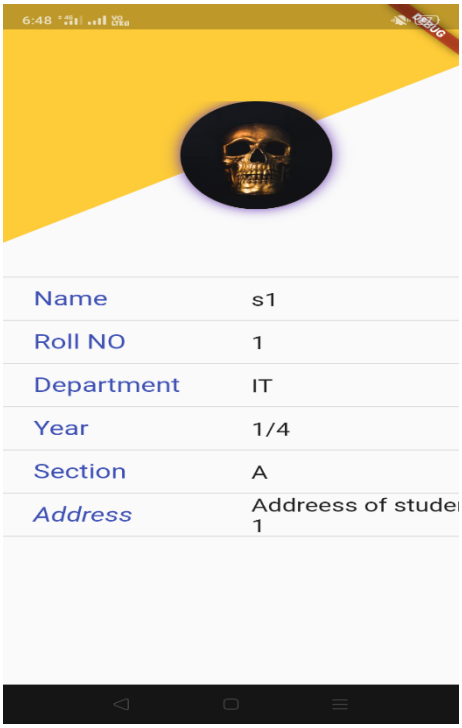


Fig.8.2

Drawer

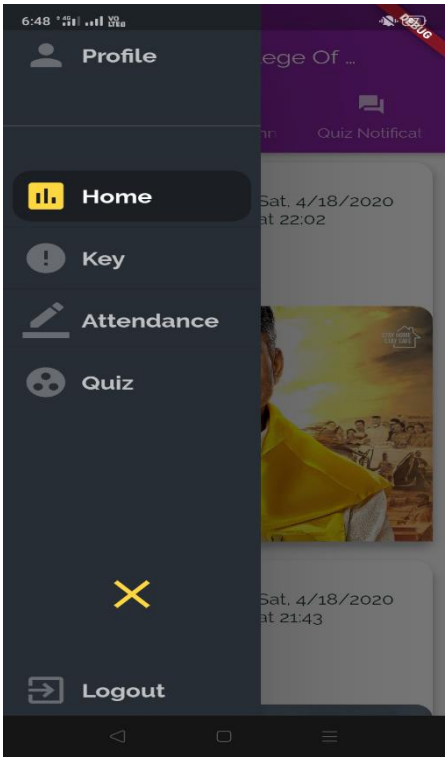


Fig.8.3

Uploading a post

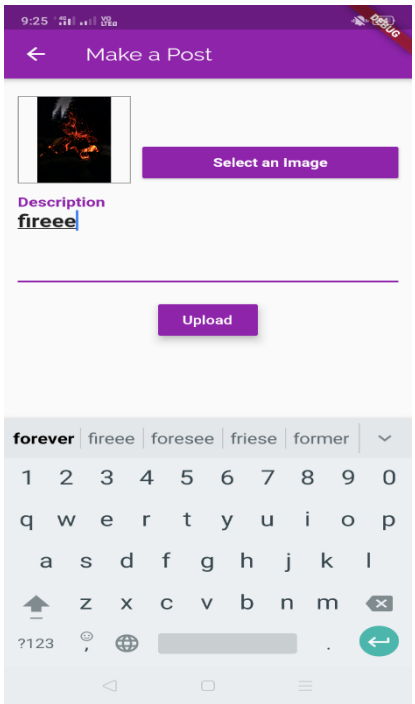


Fig.8.4

Wall of App (articles published by staff)

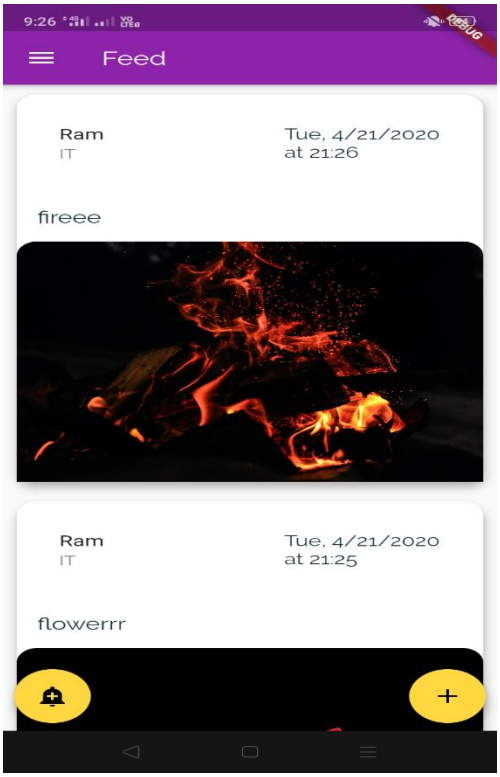


Fig.10.5

Picking date while assigning Quiz

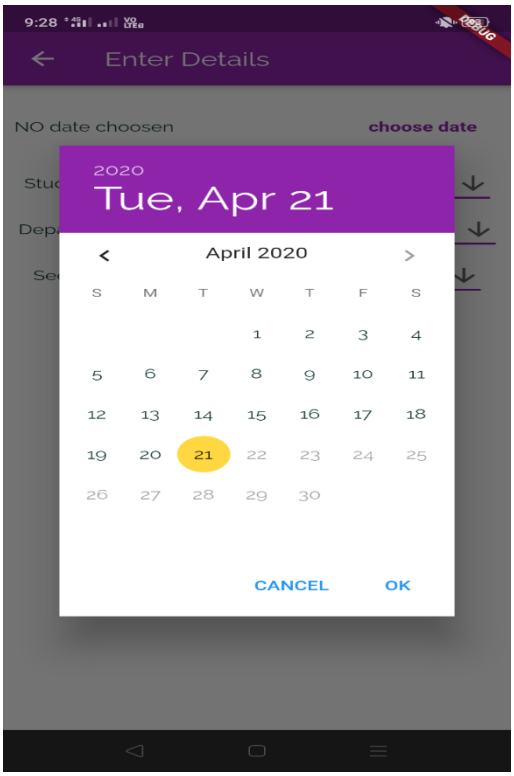


Fig.8.6

Taking attendance of a class

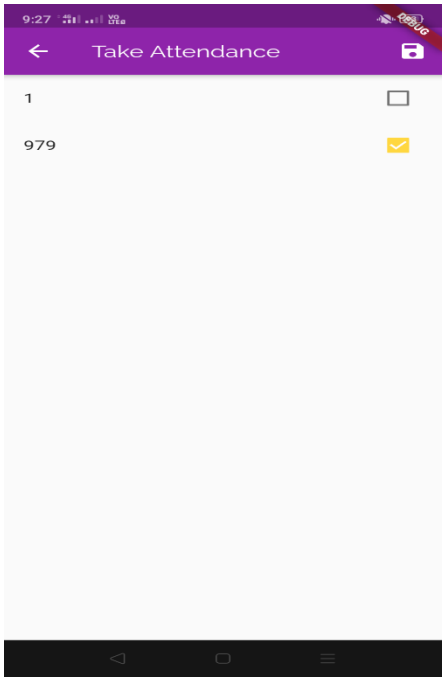


Fig.8.7

Attenance of a class on a date

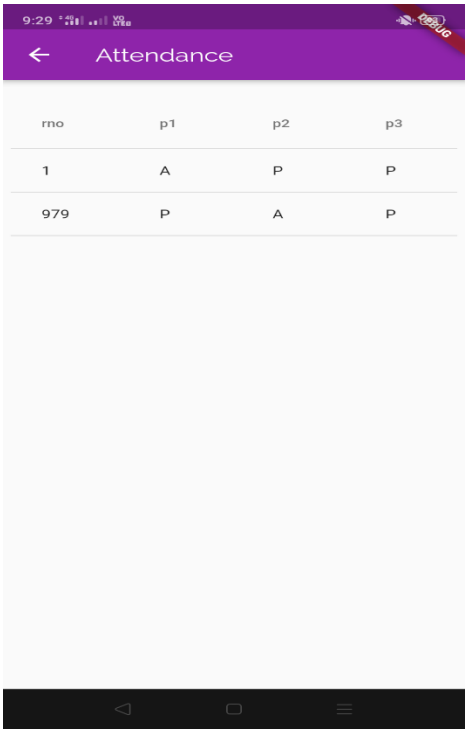


Fig.8.8

Question entering form

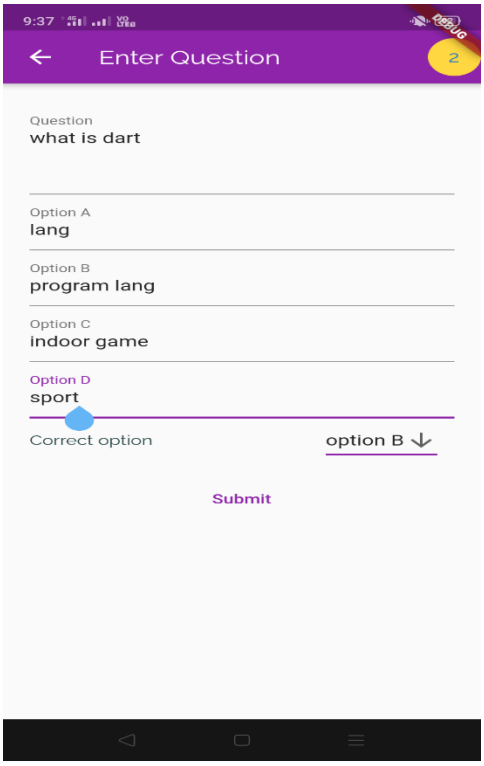
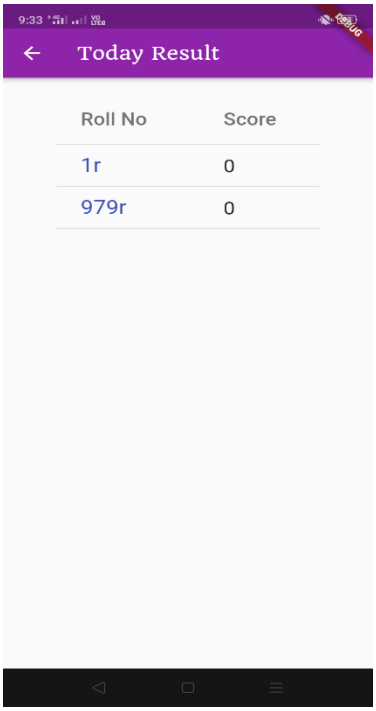


Fig.8.9

Taking Quiz by students

Fig.8.10

Results of a class



Quiz announcements

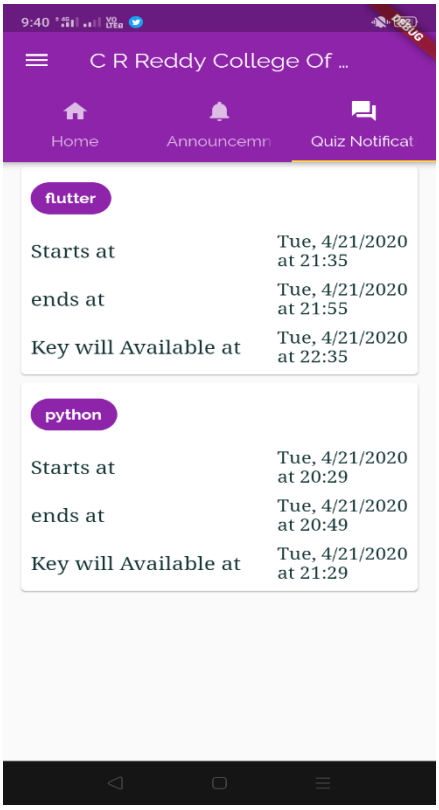


Fig.8.13

Regular announcements

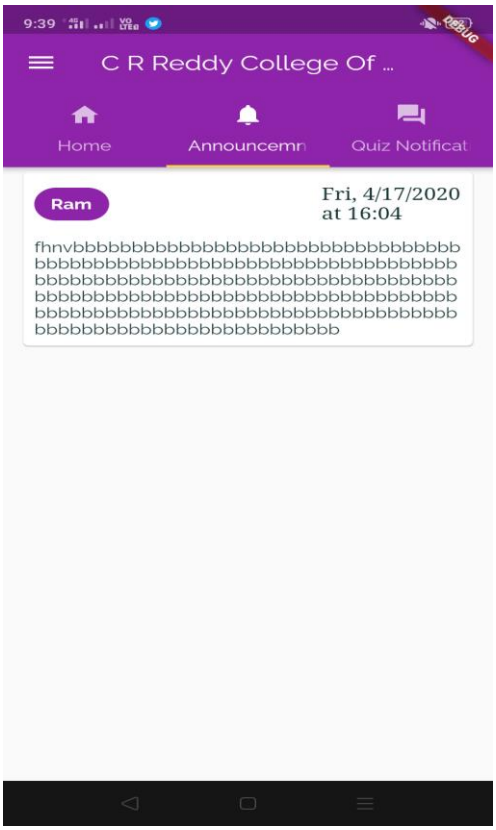


Fig 8.14

No data handling

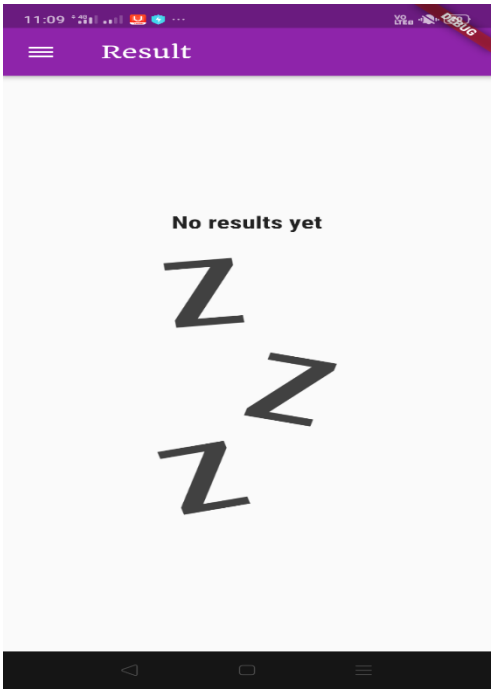


Fig 8.14

Monthly percentage of student

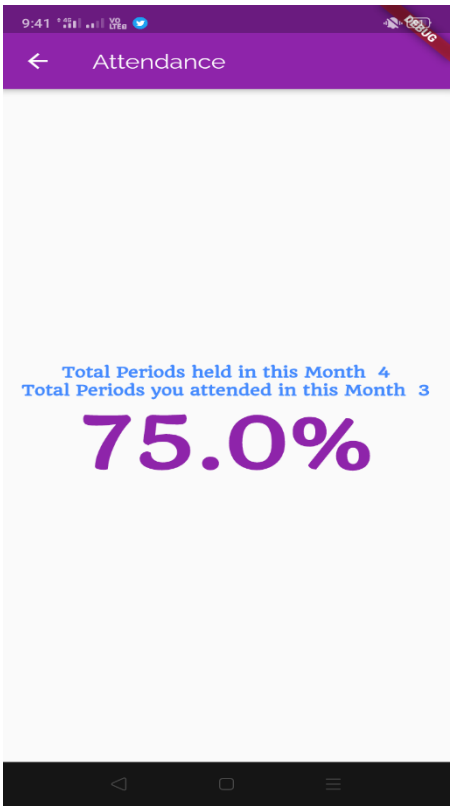


Fig 8.16

Quizzes key

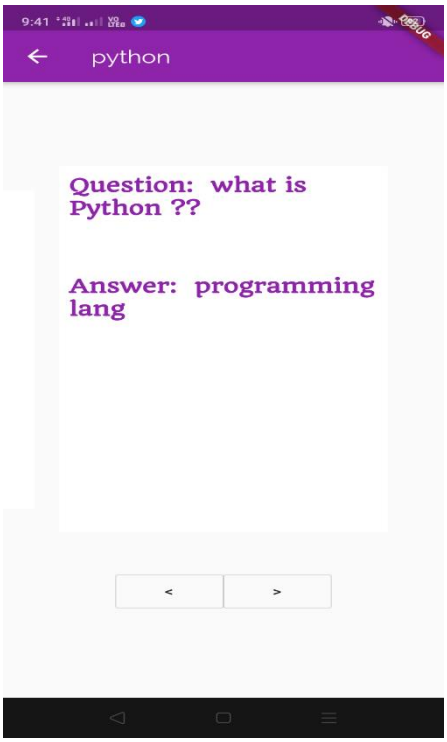


Fig 8.17

Selecting Page

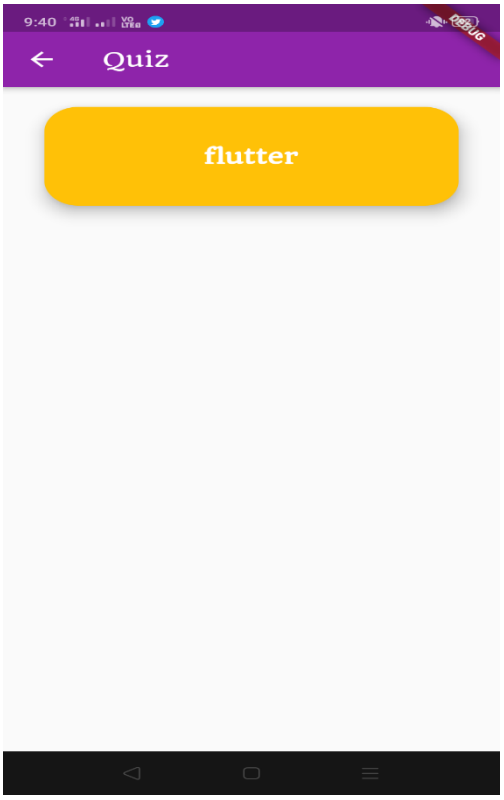


Fig 8.18

Splash page and loading pages

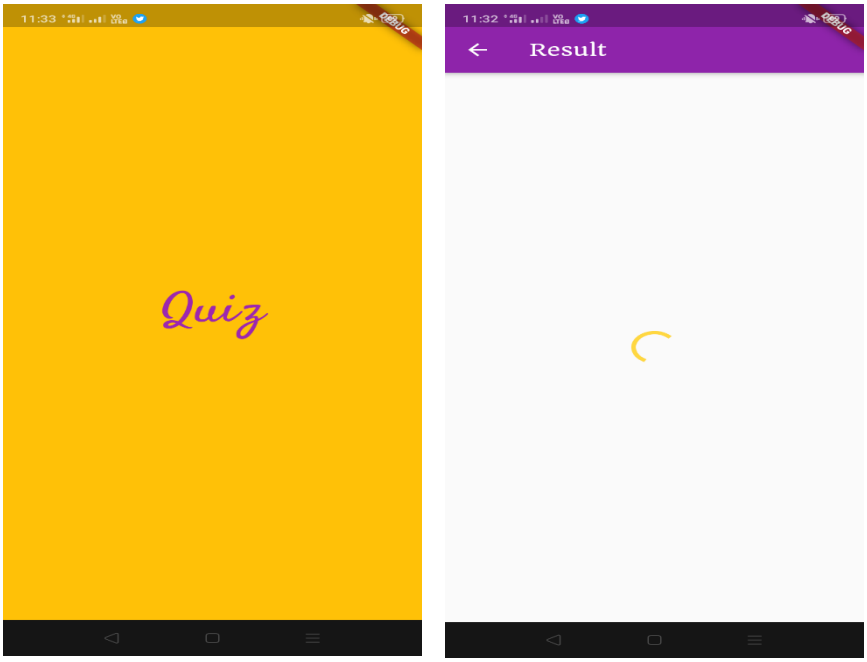


Fig 8.19

9.CONCLUSION

CRR-IN_TOUCH is an mobile application which helps the students, favulty to communicate with each other easily, in this application students can not only communicate with faculty they can analyze themselves by attempting tests that are assigned by staff.

Using this application Faculty can take attendance of a class and Students can view their monthly attendance percentage

As this application was developed using Flutter which is known for Best User Interface building technology,This application has best user Interface and provide with best user Interface

The project was successfully completed with necessary modules and functionality was matched as expected. Every effort has been made to present the system in more user-friendly manner.

10. FUTURE ENHANCEMENT

- Direct notifications will be snd to students If any article or announcement made by staff
- Admin work, to avoid anobymous registreations be planned to do in another web application (which is currently in stage of mobile application) will be upgraded using another way
- All other Educational Institutional works like fee payment, video lectures etc… may be fulfilled by usimg this application
- Ofcourse this application provides better customistion for users and, It will be improved in future to provide best

11. REFERENCES

1. https://flutter.dev/ (Flutter Official Site)
2. Flutter Youtube channel
3. https://www.udemy.com/course/learn-flutter-dart-to-build-ios-android-apps/
4. https://dart.dev/tutorials
5. https://stackoverflow.com/
6. https://github.com/
7. https://firebase.google.com/docs