HOSTEL MANAGEMENT SYSTEM

A Mini Project Report Submitted by

SWARAJ YJ (4NM16CS160)
TERRIL JOEL NAZARETH (4NM16CS162)

UNDER THE GUIDANCE OF

Mr. Ramesha Shettigar
Assistant Professor
Department of Computer Science and Engineering

in partial fulfillment of the requirements for the award of the Degree of

Bachelor of Engineering in
Computer Science & Engineering

from

Visvesvaraya Technological University, Belagavi

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

N.M.A.M. INSTITUTE OF TECHNOLOGY

(An Autonomous Institution under VTU, Belagavi) (AICTE approved, NBA Accredited, ISO 9001:2008 Certified) NITTE -574 110, Udupi District, KARNATAKA.

April 2019

NITTE EDUCATION TRUST | N.M.A.M. INSTITUTE OF TECHNOLOGY
(An Autonomous Institution affiliated to Visvesvaraya Technological University, Belagavi)
Nitte – 574 110, Karnataka, India
(ISO 9001:2015 Certified), Accredited with 'A' Grade by NAAC
☎: 08258 - 281039 – 281263, Fax: 08258 – 281265

Department of Computer Science and Engineering

B.E. CSE Program Accredited by NBA, New Delhi from 1-7-2018 to 30-6-2021

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

"Hostel Management System"

is a bonafide work carried out by

Swaraj YJ(4NM16CS160) Terril Joel Nazareth(4NM16CS162)

in partial fulfillment of the requirements for the award of Bachelor of Engineering Degree in Computer Science and Engineering prescribed by Visvesvaraya Technological University, Belagavi during the year 2018-2019.

It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report.

The Mini project report has been approved as it satisfies the academic requirements in respect of the project work prescribed for the Bachelor of Engineering Degree.

Signature of Guide Signature of HOD

ACKNOWLEDGEMENT

We believe that our project will be complete only after we thank the people who have contributed to make this project successful.

First and foremost, we express our deep sense of gratitude and indebtedness to our guide **Mr. Ramesha Shettigar**, Assistant Professor, Department of Computer Science and Engineering, for his inspiring guidance, constant encouragement, support and suggestions for improvement during the course of our project.

We sincerely thank **Dr. K.R. Udaya Kumar Reddy**, Head of Department of Computer Science and Engineering, Nitte Mahalinga Adyantaya Memorial Institute of Technology, Nitte.

our sincere thanks to our beloved principal, **Dr. Niranjan N. Chiplunkar** for giving us an opportunity to carry out our project work at our college and providing us with all the needed facilities.
We also thank all those who have supported us throughout the entire duration of our project.
Finally, we thank the staff members of the Department of Computer Science and Engineering and all our friends for their honest opinions and suggestions throughout the course of our project.

Swaraj YJ(4NM16CS160)
Terril Joel Nazareth (4NM16CS162)

ABSTRACT

For a past few years the number of educational institutions are increasing rapidly. There by the number of hostels are also increasing for the accommodation of the students studying in this institution. And hence there is lot of strain on the person who are running the hostel and software are not usually used in this context. This particular project deals with the problem on managing a hostel and avoids the problem which occur when carried out manually.

To provide a platform where information can be communicated to the user of the app regarding the activities that would come under the institute or even under the hostel. The meal schedule can be viewed by the user to have an understanding of the dishes that would be available. In the case of absence from the hostel on particular day “Movement register” can be used log that information. If a replacement is required or damage is incurred to the utilities in any person’s room this can brought to notice by the entry of the same into a “Maintenance register” that has been allocated for this purpose. It also has contact information of hostel in charge.

Identification of the drawbacks of the existing system leads to the designing of computerized system that will be compatible to the existing system with the system, which is more user friendly and more GUI oriented. We can improve the efficiency of the system, thus overcome the drawbacks of the existing system.

Hostel Management System

Table Of Contents:

CHAPTER 1

INTRODUCTION

1.1 Scope

Hostel management system is a flexible, easy to use, cost effective, secure and customizable Web- based tool designed to deliver conceivable benefit to fleet hostellers. It is used to digitize the circular systems, maintenance register, movement register and mess time table all in one single android mobile phone and manage them efficiently.

1.2 Importance

In a large hostel it is very difficult to keep track of all the information regarding students. With the help of hostel management system, the data is stored in the digitized form and hence reduces the paper work. The hostel management system makes use of one lone centralized database running from a central server. The data can be efficiently stored and retrieved which makes the application sophisticated and user friendly. The application is implemented on android platform which is linked to the Google firebase as a server for access of data.

1.3 Objective:

The objective of this project is a mobile application developed for managing various activities in the hostel. Identification of the drawbacks of the existing systems leads to the designing of computerized system that will be compatible to the existing system with the system which is more user friendly and more GUI oriented. We can improve the efficiency of the system, thus overcome the drawbacks of the existing system.

CHAPTER 2

LITERATURE SURVEY

2.1 Technical Background:

In Hostel Management System the information regarding mess timetable, circulars, movement registers and maintenance register are stored in the backend i.e., in Google Firebase Database. GUI of the application enables to access and modify data efficiently.

2.2 Existing system:

- In the hostel important circulars are put up on the notice board which doesn't guarantee that all the student will be aware of this.
- For the movement register student have to fill up record book manually.
- For the maintenance register student have to fill up the register manually in the book.
- Every student may not have contact numbers of warden, electrician and office.
- Mess timetable is made available only on the notice board

2.3 Proposed system

In this Hostel Management Application all the existing traditional system can be digitized and can be made available at the tip of the finger. All the above existing system is upgraded and manual work can be reduced for the record management. There will be one administrator who will manage the data at the back end.

- Circulars will be made available online through mobile application.
- Students can fill up and can modify movement registers anytime, anywhere through a mobile application.
- Maintenance registers made available online through mobile application.
- Contact numbers of concerned people of the hostel is made available in the application.
- And Mess time table can be viewed through an mobile application

CHAPTER 3

SYSTEM REQUIREMENT AND SPECIFICATION

3.1 Introduction

Requirements are during early stages of a system development as a specification of what should be implemented or as a constraint of some kind of on the system. They may be: a user level facility description, a detailed specification of expected system behavior, a general system property, a specific constraint on the system, and information on how to carry out some computation or a constraint on the development of the system. The end product of the requirement analysis phase is a requirement specification. The requirement specification is a reconstruction of the result of this analysis phase. Its purpose is to communicate this result to others. System requirements are more detailed descriptions of the user requirements. They may serve as the basis for a contract to the implementation of the system and should therefore be a complete and consistent specification of the whole system. They are used by software engineers as the starting point of system design. In principle, the system requirements should state what the system should do and not how it should be implemented. However, at the level of detail required to specify the system completely, it is virtually impossible to exclude all design information.

Natural language is often used to write system requirements specifications. Further problems with natural language can arise when it is used for more detailed specification:

1. Natural language understanding relies on the specification of the readers and writers using the same words for the same concept. This leads to misunderstandings because of the ambiguity of the natural language.
2. A natural language requirements specification is over-flexible. You can say the same thing in completely different ways. It is up to the reader to find out when requirements are same and when they are distinct.
3. There is no easy way to modularize natural language requirements. It may be difficult to find all the related requirements. To discover the consequence of a change, you may have to look at every requirement rather than just a group of related requirements.

3.2 Functional Requirements

The functional requirements are the statement of services the system should provide, how system reacts to particular inputs and how system should behave in particular situation. It describes the functionality that the system provides.
Our app requires:
I) Active internet connection.
II) A firebase console to store the data

3.3 User Requirements

The user requires active internet connection to use the app.

3.4 Software Requirements

I. Operating System: Windows 7/8/10 (32-bit or 64-bit)
II. Android SDK
III. Android Studio
IV. Firebase account.

3.4.1 Android SDK

I. The Android SDK provides you the API libraries and developer tools necessary to build, test, and debug apps for Android. The ADT bundle includes the essential Android SDK components and a version of the Eclipse IDE with built-in Android Developer Tools to streamline the Android app development. ADT bundle consists of following components for developing the application
II. Eclipse + ADT plugin.
III. Android SDK Tools
IV. Android Platform-tools
V. The latest Android platform
VI. The latest Android system image for the emulator

3.4.2 Android Studio

Android Studio is the official integrated development environment (IDE) for Google's Android operating system, built on JetBrains' IntelliJ IDEA software and designed specifically for Android development. It is available for download on Windows, macOS and Linux based operating systems. It is a replacement for the Eclipse Android Development Tools (ADT) as the primary IDE for native Android application development.

Android Studio was announced on May 16, 2013 at the Google I/O conference. It was in early access preview stage starting from version 0.1 in May 2013, then entered beta stage starting from version 0.8 which was released in June 2014. The first stable build was released in December 2014, starting from version 1.0. The current stable version is 3.3, which was released in January 2019.

3.4.3 Firebase

Firebase is a mobile and web application development platform developed by Firebase, Inc. in 2011, then acquired by Google in 2014.As of October 2018, the Firebase platform has 18 products, which are used by 1.5 million apps. Firebase provides a real-time database and backend as a service. The service provides application developers an API that allows application data to be synchronized across clients and stored on Firebase's cloud. Firebase Storage provides secure file uploads and downloads for Firebase apps, regardless of network quality. The developer can use it to store images, audio, video, or other user-generated content. Firebase Storage is backed by Google Cloud Storage.

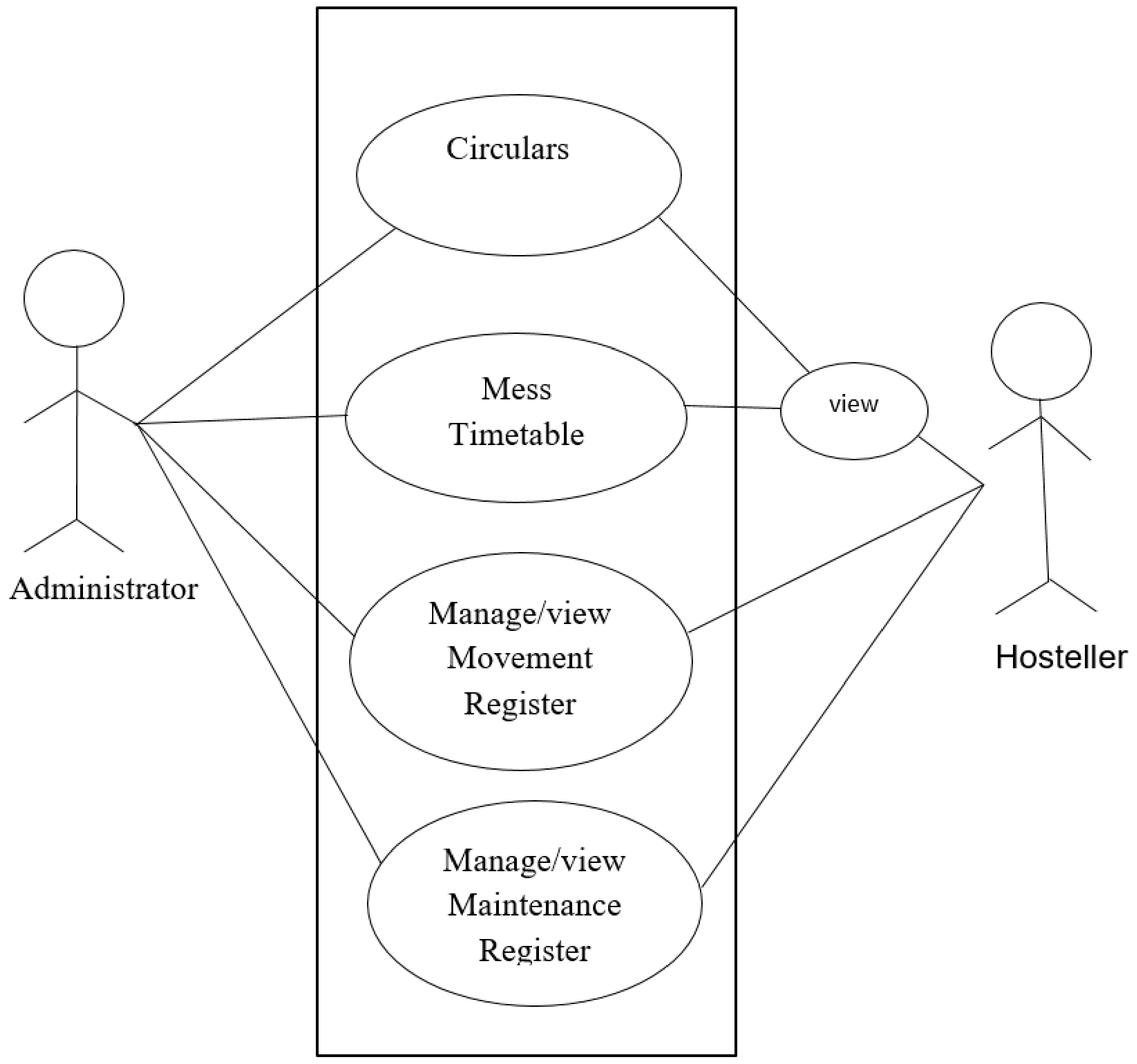
3.5 HARDWARE REQUIREMENTS

I. Intel Pentium Core 2 Duo processor.
II. Minimum 3 GB RAM (8GB recommended).
III. 2GB free disk space
IV. USB 2.0 or higher.
V. Android enabled devices required in the client side.

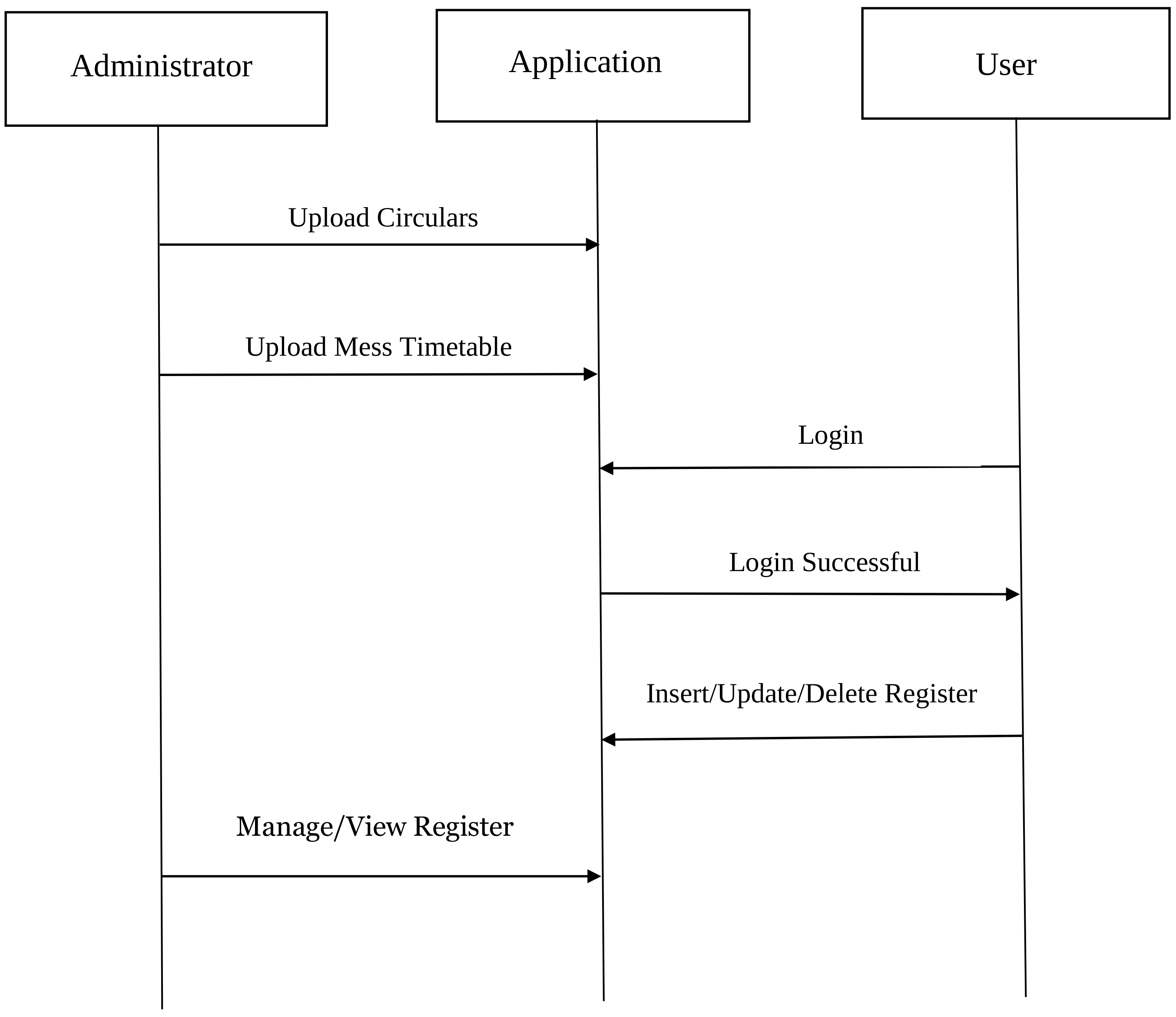
CHAPTER 4

SYSTEM DESIGN

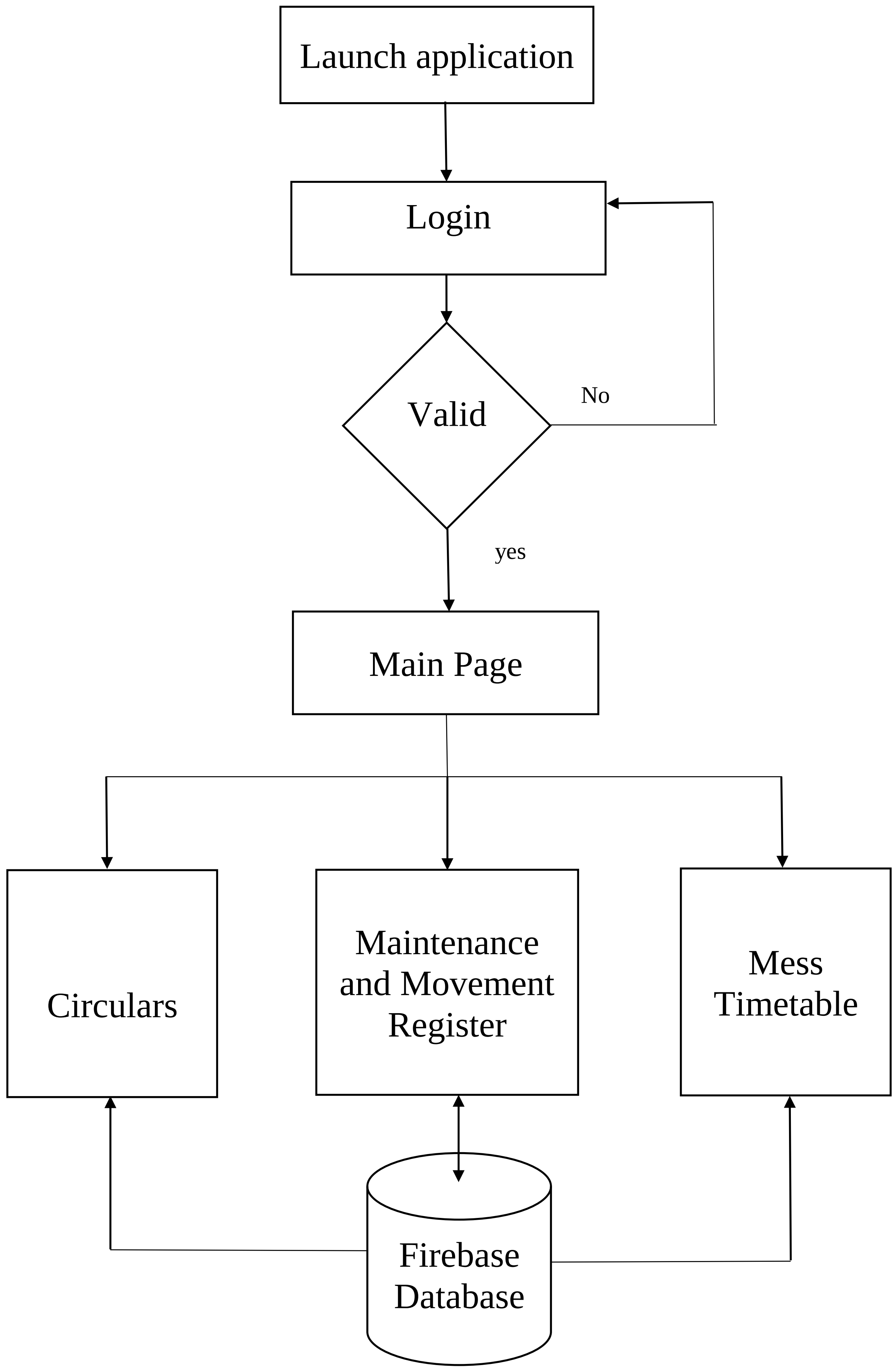
4.1 Use case model for Hostel Management


Circulars
Mess Timetable
view
Administrator
Manage/view Movement Register
Hosteller
Manage/view Maintenance Register

4.2 Sequence diagram for Hostel Management System



4.3 Data Flow Diagram

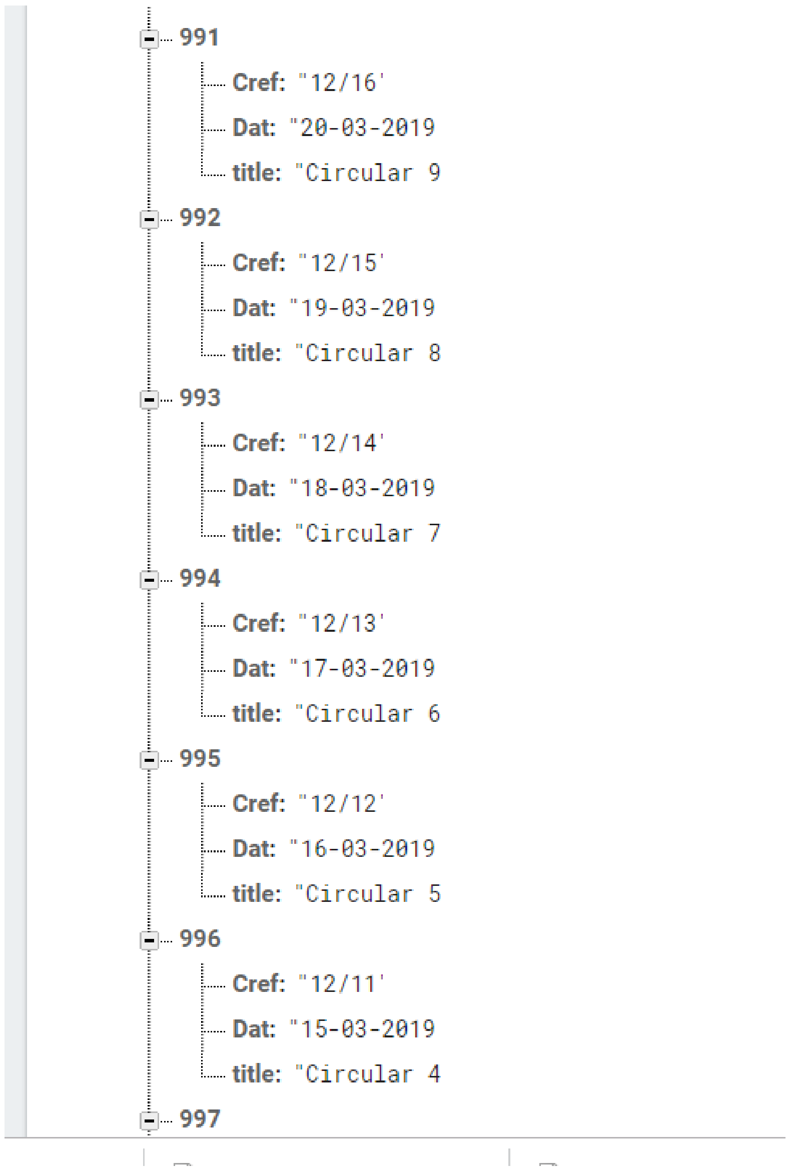


1. The administrator logs into google firebase account, uploads circulars, mess timetables and manages movement registers and maintenance register
2. The user at the front end will be able to view circulars, timetables and can insert update delete the movement and maintenance registers.
3. Each user is given with a unique username and password.

4.4 Database Design

The database is designed using Google Firebase Console in which data is stored in popular data structure known as JSON tree (JavaScript Object Notation). Every time when the data transfer happens from client end, the information given to the UI is converted into JSON tree structure which is efficient and faster way to retrieve and store data.

A snapshot of the data of circular list in a firebase



```
991
    Cref: "12/16'
    Dat: "20-03-2019
    title: "Circular 9
992
    Cref: "12/15'
    Dat: "19-03-2019
    title: "Circular 8
993
    Cref: "12/14'
    Dat: "18-03-2019
    title: "Circular 7
994
    Cref: "12/13'
    Dat: "17-03-2019
    title: "Circular 6
995
    Cref: "12/12'
    Dat: "16-03-2019
    title: "Circular 5
996
    Cref: "12/11'
    Dat: "15-03-2019
    title: "Circular 4
997
```

CHAPTER 5

IMPLEMENTATION

We are here designing hostel Management system to automate and manage all the activities of the hostel. We are designing circular management system so that user remain update with all the events occurring in hostel. Movement register form is also provided if student wants to move out of hostel. Maintenance register keep track of any damage to the utilities. It also provides the information about the mess time table.

5.1 SOLUTION APPROACH/METHODOLOGY

We are here using xml and java for the front end and firebase for the backend as a server.

5.1.1 Firebase

Firebase is considered as web application platform. It helps query for inserting, updating, deleting or adding data to it. It is the backend of a system that is used as a database for storing data

Firebase real-time database feature is very easy to use. Once the Firebase and database dependency is added to the app, unstructured data can be added to database by the following
code [7]:

```
//Write a message to the database
FirebaseDatabase database= FirebaseDatabase.getInstance;
DatabaseReference myref=database.getReference("node");
myRef.setValue ("Hello, World");
```

5.1.2 Storage

The files like images, audio, video etc can be stored in the app. The data stored is highly secured and is robust in nature means it resumes from the last point if any network error occurs. The steps below are to be followed to use storage feature in Android application:
added to the application, create instance of
FirebaseStorage storageobject =FirebaseStorage.getInstance();
StorageReference FileRef = storageRef.child("filePath");
putFile(), putData() or putStream() method which
returns to UploadTask.

5.1.3 FIREBASE AND ANDROID APP

An Android application has been developed for the demonstration of Firebase. In this app images along with strings are loaded to Firebase and retrieved from Firebase similar to Instagram. For the development of an Android app to demonstrate the use of Firebase, prototyping model has been followed.

Steps for connecting App to Firebase:

Step1: An account in the Firebase Login has to be created at https://www.firebase.com/login/ using the Google account.

Step2. Creating a new application on Firebase. Firebase creates a new application when one logs in for the first time. Also, at the bottom left corner, one can find an option to create a new application on the Firebase server. The app url has to be unique among all applications deployed on Firebase.

Step3. Next step is to add Firebase as a project dependency. Make changes to the following lines to the build.gradle file, which is located in the app"s project folder, and not the root folder.
After adding any dependency one has to make sure to sync the
application.
If there is any build error complaint about duplicate files then one can choose to exclude those files by adding the packaging Options directive to the build.gradle file:
android

Step4. Next, add permissions to Android application, Add network
permission to the app, the same way it has been done for parse
earlier. Now add the following line to the
AndroidManifest.xml file:
<uses-permission android:name="android.permission.INTERNET" />

Firebase is a Backend-as-a-Service—BaaS—that started as a YC11 startup and grew up into a next-generation app-development platform on Google Cloud Platform.

It's a Realtime Database

Real-time data is the way of the future. Nothing compares to it. Most databases require you to make HTTP calls to get and sync your data. Most databases give you data only when you ask for it.When you connect your app to Firebase, you're not connecting through normal HTTP. You're connecting through a WebSocket. WebSockets are much, much faster than HTTP. You don't have to make individual WebSocket calls, because one socket connection is plenty. All of your data syncs automagically through that single WebSocket as fast as your client's network can carry it. Firebase sends you new data as soon as it's updated. When your client saves a change to the data, all connected clients receive the updated data almost instantly.

It's File Storage

Firebase Storage provides a simple way to save binary files—most often images, but it could be anything—to Google Cloud Storage directly from the client!!!

5.1.4 Java

There are a number of ways to create apps for Android devices, but the recommended method for most developers is to write native apps using Java and the Android SDK. Java for Android apps is both similar and quite different from other types of Java applications.

If you have experience with Java (or a similar language) then you'll probably feel pretty comfortable diving right into the code and learning how to use the Android SDK to make your app run. But if you're new to programming or object-oriented languages then you'll probably want to get familiar with the syntax of the Java language and how to accomplish basic programming tasks before learning how to use the Android SDK.

5.2 IMPLEMENTATION CODE

5.2.1 The following code is used to insert data into firebase:

```
FirebaseApp.initializeApp(this);
reff = FirebaseDatabase.getInstance().getReference("Maintenance Register");
add.setOnClickListener(new View.OnClickListener() {
    @RequiresApi(api = Build.VERSION_CODES.N)
    @Override
    public void onClick(View v) {
        String bc = MainPageActivity.b;
        String i = issue.getText().toString().trim();
        String bn = blockno.getText().toString().trim();
        String rn = roomno.getText().toString().trim();

        if(i.equalsIgnoreCase(""))
        {
            issue.setError("Fields cannot be empty");
            return ;
        }
        if(bn.equalsIgnoreCase(""))
        {
            blockno.setError("Fields cannot be empty");
            return ;
        }
        if(rn.equalsIgnoreCase(""))
        {
            roomno.setError("Fields cannot be empty");
            return ;
        }
        int intbn=Integer.parseInt(bn);
        if(intbn<=0||intbn>=6)
        {
            blockno.setError("Invalid block number");
            return;
        }
        int intrn=Integer.parseInt(rn);
        if(intrn<=0)
        {
            roomno.setError("Invalid room number");
            return;
        }
        String date=new SimpleDateFormat("dd-MM-yyyy",
Locale.getDefault()).format(new Date());


        mrc2.setDate(date);

        mrc2.setBlockno(bn);
        mrc2.setIssue(i);
        mrc2.setRoomno(rn);
        mrc2.setStatus("0");

        String id = reff.push().getKey();

        reff.child(bc).child(id).setValue(mrc2);
        Toast.makeText(getApplicationContext(),"Registered
Sucessfully",Toast.LENGTH_LONG).show();
```

5.2.2 The following code is used to update the data in the firebase:

```
reff=FirebaseDatabase.getInstance().getReference().child("Maintenance Register");
        reff.addListenerForSingleValueEvent(new ValueEventListener() {
            @RequiresApi(api = Build.VERSION_CODES.N)
            @Override
            public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
                if (dataSnapshot.child(bc).child(id).exists()) {
                    String y =
dataSnapshot.child(bc).child(id).child("status").getValue().toString().trim();
                    if (y.equalsIgnoreCase("1")) {
                        Toast.makeText(getApplicationContext(), "Updatation not allowed",
Toast.LENGTH_LONG).show();
                        return;
                    }

                    String i = e1.getText().toString().trim();
                    String bn = e2.getText().toString().trim();
                    String rn= e3.getText().toString().trim();



                    if(i.equalsIgnoreCase(""))
                    {
                        e1.setError("Fields cannot be empty");
                        return ;
                    }
                    if(bn.equalsIgnoreCase(""))
                    {
                        e2.setError("Fields cannot be empty");
                        return ;
                    }
                    if(rn.equalsIgnoreCase(""))
                    {
                        e3.setError("Fields cannot be empty");
                        return ;
                    }
                    int intbn=Integer.parseInt(bn);
                    if(intbn<=0||intbn>=6)
                    {
                        e2.setError("Invalid block number");
                        return;
                    }
                    int intrn=Integer.parseInt(rn);
                    if(intrn<=0)
                    {
                        e3.setError("Invalid room number");
                        return;
                    }
                    String date=new SimpleDateFormat("dd-MM-yyyy",
Locale.getDefault()).format(new Date());



                    mrc2.setIssue(i);
                    mrc2.setBlockno(bn);
                    mrc2.setRoomno(rn);
                    mrc2.setDate(date);


                    reff.child(bc).child(id).setValue(mrc2);
                    Toast.makeText(getApplicationContext(), "Updated Sucessfully",
Toast.LENGTH_LONG).show();

                }
            }
            @Override
            public void onCancelled(@NonNull DatabaseError databaseError) {

            }
        });
    }
});
```

5.2.3 The following code is used to delete the data in the firebase:

```
reff=FirebaseDatabase.getInstance().getReference().child("Maintenance Register");
reff.addListenerForSingleValueEvent(new ValueEventListener() {
    @Override
    public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
        String y =
dataSnapshot.child(bc).child(id).child("status").getValue().toString().trim();
        if (y.equalsIgnoreCase("1")) {
            Toast.makeText(getApplicationContext(), "Deletion not allowed",
Toast.LENGTH_LONG).show();
```

```
            return;
        }
        if (dataSnapshot.child(bc).child(id).exists()) {

            reff.child(bc).child(id).removeValue();


            Toast.makeText(getApplicationContext(), "Deletion successfull",
Toast.LENGTH_LONG).show();
            finish();
        }
    }

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

    }
});
```

The following code is used to display the data after fetching from the firebase:

```
myreff= FirebaseDatabase.getInstance().getReference("Maintenance Register");
myreff.addListenerForSingleValueEvent(new ValueEventListener() {
    @Override
    public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
        for(DataSnapshot ds:dataSnapshot.child(MainPageActivity.b).getChildren())
        {

            String r = ds.getKey();
            String s = "Ref No   :" + ds.getKey() + "\nIssue    :" +
ds.child("issue").getValue() + "\nBlock No :" + ds.child("blockno").getValue() +
"\nRoom No :" + ds.child("roomno").getValue();

            temp.add(s);
            reffno1.add(r);


        }
        int n= (int) dataSnapshot.child(MainPageActivity.b).getChildrenCount();
        for(int i=n-1;i>=0;i--)
        {
            list.add(temp.get(i));
            reffno2.add(reffno1.get(i));
        }
        listView.setAdapter(adapter);
    }
    @Override
    public void onCancelled(@NonNull DatabaseError databaseError) {

    }
})
```

CHAPTER 6

SYSTEM TESTING

6.1 INTRODUCTION

System testing of software or hardware is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirement. System testing falls within the scope of black box testing, and as such, requires no knowledge of the inner design of the code or logic.

System testing is usually required before and after a system is put in place. A series of systematic procedures are referred to while testing is being performed. These procedures tell the tester how the system should perform and where common mistakes may be found. Testers usually try to "break the system" by entering data that may cause the system to malfunction or return incorrect information. System testing is performed on the entire system in the context of System Requirement Specification (SRS). System testing is an investigatory testing phase, where the focus is to have almost a destructive attitude and tests not only the design, but also the behaviour and even the believed expectation of the user. It is also intended to test up to and beyond the bound on the Software Requirement Specification The testing phase is performed after the coding to detect all the errors and provide quality assurance and ensure reliability of the software. Testing is vital to the success of the system. During testing, the software to be tested and executed with a set of test cases, and the behaviour of the system for the test cases is evaluated to determine if the system is performing as expected. Clearly, the success of the testing in revealing errors depends critically on the test cases.

The different testing strategies employed in this project are explained in this chapter.

6.2 UNIT TESTING

In computer programming, unit testing is a software verification and validation method in which a programmer tests if individual units of source code are fit for use. A unit is a smallest testable part of an application. It is also called as module testing. The goal of unit testing is to isolate each part of the program first and then testing the sum of its parts, integration testing becomes much easier.
In our project, we apply this by testing the various modules of the application and also each feature individually.

6.3 INTEGRATION TESTING

Integration testing is thee phase in software testing in which individual software modules are combined and tested as a group. It occurs after unit testing and before system testing. Integration testing takes as its input modules that have been unit tested, groups them in larger aggregates, applies tests defined in an integration test plan to those aggregates, and delivers as its output the integrated system ready for the system testing.

The purpose of the integration testing is to verify functional, performance and reliability requirement placed on major design items. All the different modules of the project are combined and tested.

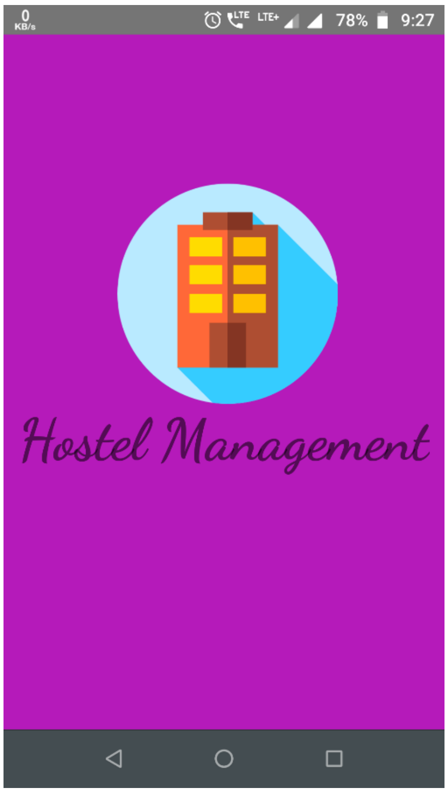
Test case for Hostel Management System:

Sl.no	TEST CASES	OUTPUT EXPECTED	ACTUAL OUTPUT
1	Click on ‘Login’ button	When the user clicks on the login button, the username and password is taken form the textbox and sent to server for validation	Same as expected
2	Click on ‘Circulars’ button	When the user clicks on this button, all the circulars are fetched from the firebase database and displayed in the form of a list.	Same as expected
3	Click on ‘Mess Timetable’ button	When the user clicks on this button, it shows the dishes served on particular day of week based on the user’s choice.	Same as expected
4	Click on ‘Add Record’ button	When the user clicks on this button, form is displayed for the entry of record either for Movement Register or Maintenance Register, based on user’s choice.	Same as expected
5	Click on ‘Display Record’ button	When the user clicks on this button, list of all the existing records of a user is displayed and when particular item is selected, update and delete option is shown.	Same as expected
6	Click on ‘Contact Info’ button	When the user clicks on this button, list of all officers is shown, which when clicked is redirected to call.	Same as expected

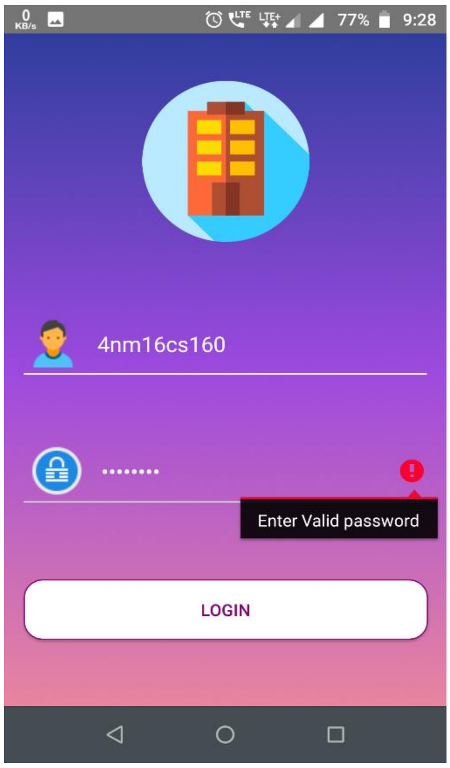
CHAPTER 7

SCREENSHOTS

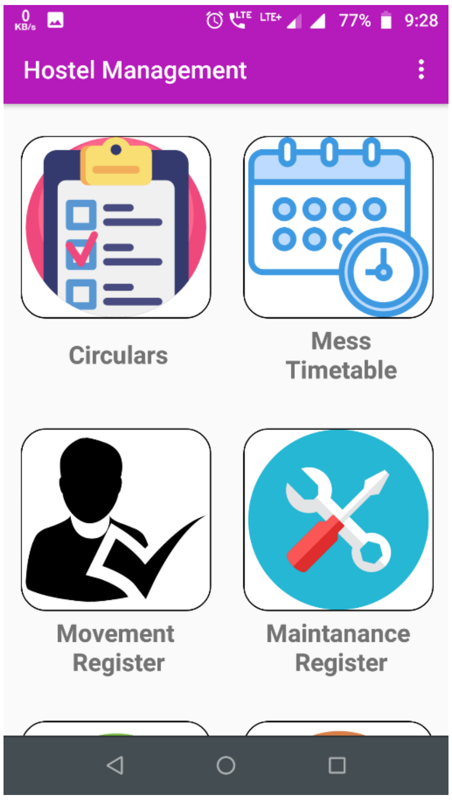
I. First Loading page

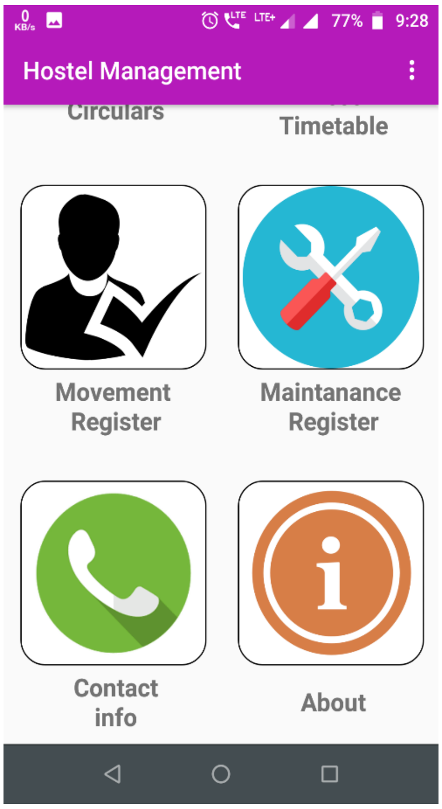


II. Login Page

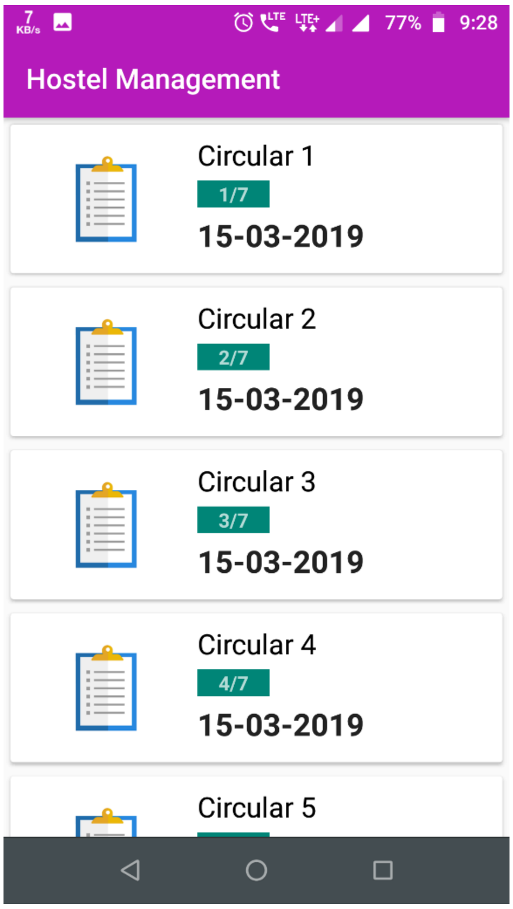


III.Main Page

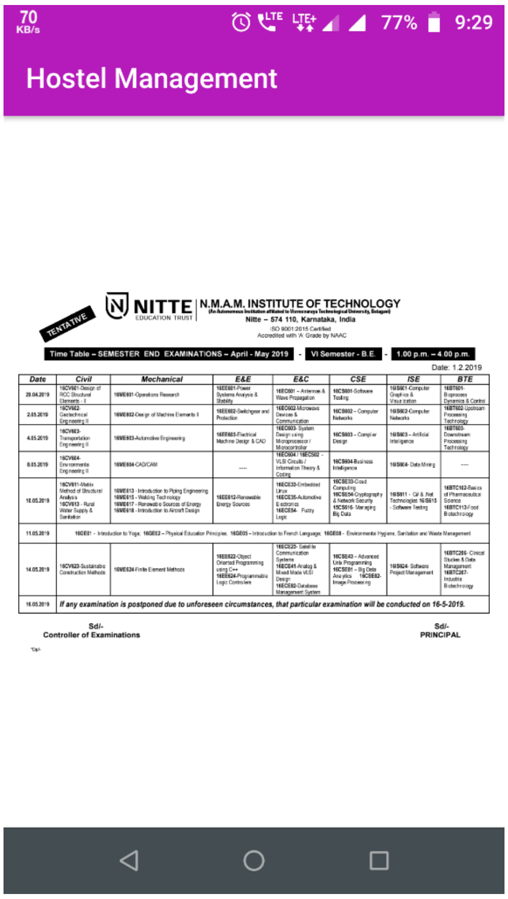




IV. Circular Recycler view



V. Circular PDF view



TENTATIVE

NITTE EDUCATION TRUST | N.M.A.M. INSTITUTE OF TECHNOLOGY
(An Autonomous Institution affiliated to Visvesvaraya Technological University, Belagavi)
Nitte – 574 110, Karnataka, India
ISO 9001:2015 Certified
Accredited with 'A' Grade by NAAC

Time Table – SEMESTER END EXAMINATIONS – April - May 2019 - VI Semester - B.E. - 1.00 p.m. – 4.00 p.m.

Date: 1.2.2019

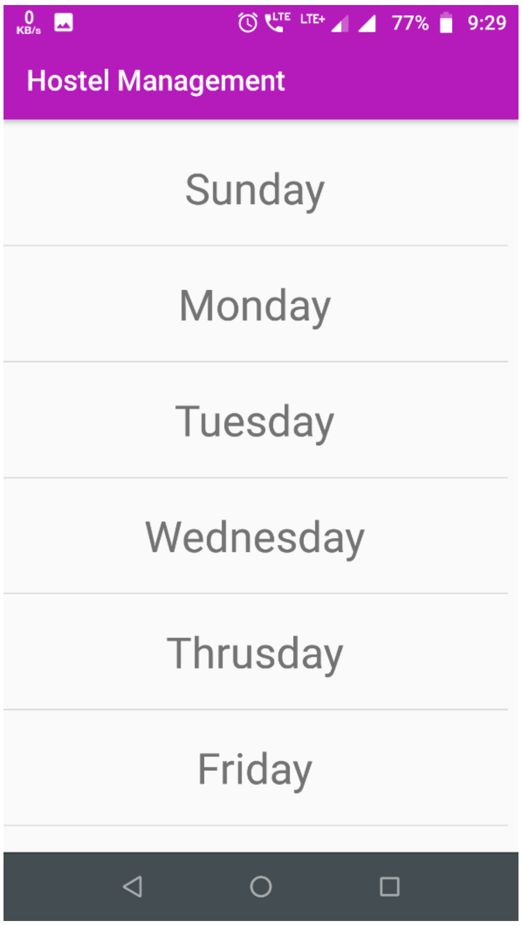
Date	Civil	Mechanical	E&E	E&C	CSE	ISE	BTE
29.04.2019	16CV601-Design of RCC Structural Elements - II	16ME601-Operations Research	16EE601-Power Systems Analysis & Stability	16EC601 – Antennas & Wave Propagation	16CS601-Software Testing	16IS601-Computer Graphics & Visualization	16BT601-Bioprocess Dynamics & Control
2.05.2019	16CV602-Geotechnical Engineering II	16ME602-Design of Machine Elements II	16EE602-Switchgear and Protection	16EC602-Microwave Devices & Communication	16CS602 – Computer Networks	16IS602-Computer Networks	16BT602-Upstream Processing Technology
4.05.2019	16CV603-Transportation Engineering II	16ME603-Automotive Engineering	16EE603-Electrical Machine Design & CAD	16EC603- System Design using Microprocessor / Microcontroller	16CS603 – Compiler Design	16IS603 – Artificial Intelligence	16BT603-Downstream Processing Technology
8.05.2019	16CV604-Environmental Engineering II	16ME604-CAD/CAM	----	16EC604 / 16EC502 - VLSI Circuits / Information Theory & Coding	16CS604-Business Intelligence	16IS604- Data Mining	----
10.05.2019	16CV611-Matrix Method of Structural Analysis 16CV613 - Rural Water Supply & Sanitation	16ME613 - Introduction to Piping Engineering 16ME615 - Welding Technology 16ME617 - Renewable Sources of Energy 16ME618 - Introduction to Aircraft Design	16EE612-Renewable Energy Sources	16ECE33-Embedded Linux 16ECE35-Automotive Electronics 16ECE54- Fuzzy Logic	16CSE33-Cloud Computing 16CSE54-Cryptography & Network Security 15CS616- Managing Big Data	16IS611 - C# & .Net Technologies 16IS615 - Software Testing	16BTC102-Basics of Pharmaceutical Science 16BTC113-Food Biotechnology
11.05.2019	16GE01 - Introduction to Yoga; 16GE02 – Physical Education Principles; 16GE05 – Introduction to French Language; 16GE08 - Environmental Hygiene, Sanitation and Waste Management						
14.05.2019	16CV623-Sustainable Construction Methods	16ME624-Finite Element Methods	16EE622-Object Oriented Programming using C++ 16EE624-Programmable Logic Controllers	16ECE23- Satellite Communication Systems 16ECE41-Analog & Mixed Mode VLSI Design 16ECE62-Database Management System	16CSE43 – Advanced Unix Programming 16CSE61 – Big Data Analytics 16CSE62-Image Processing	16IS624- Software Project Management	16BTC206- Clinical Studies & Data Management 16BTC207-Industrial Biotechnology
16.05.2019	*If any examination is postponed due to unforeseen circumstances, that particular examination will be conducted on 16-5-2019.*						

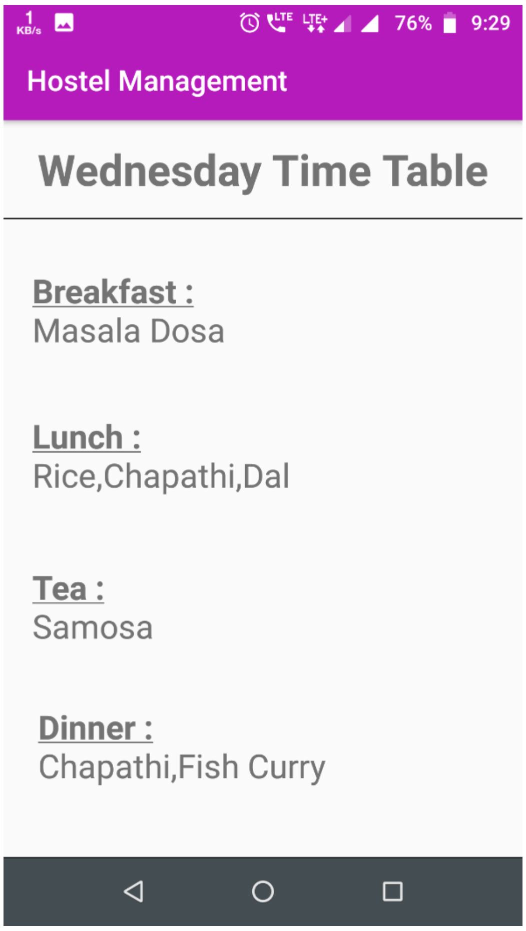
Sd/-
Controller of Examinations

Sd/-
PRINCIPAL

"Cip/-

VI. Mess Timetable





Wednesday Time Table

Breakfast :
Masala Dosa

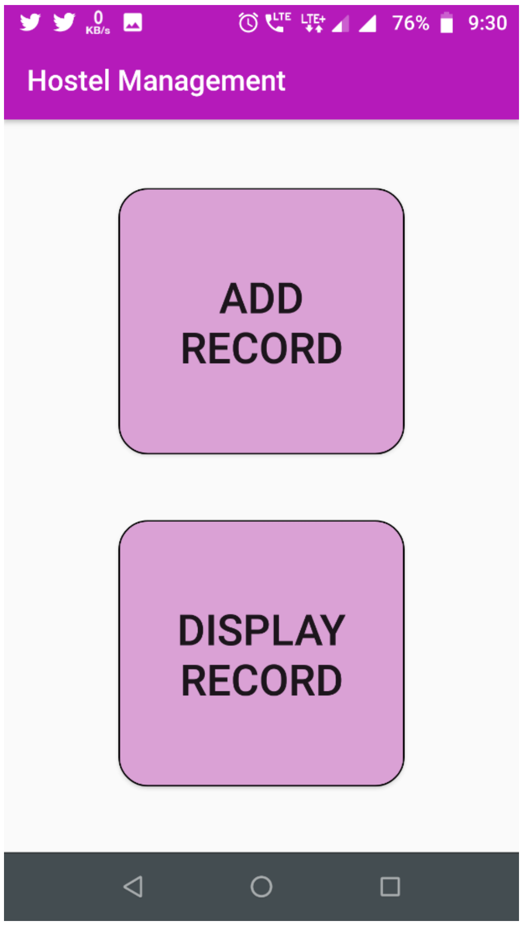
Lunch :
Rice,Chapathi,Dal

Tea :
Samosa

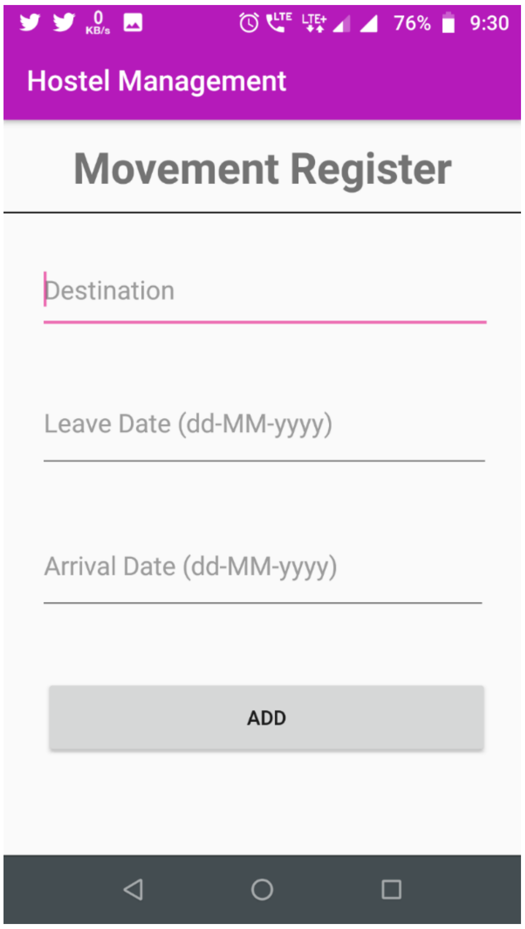
Dinner :
Chapathi,Fish Curry

VII. Movement Register

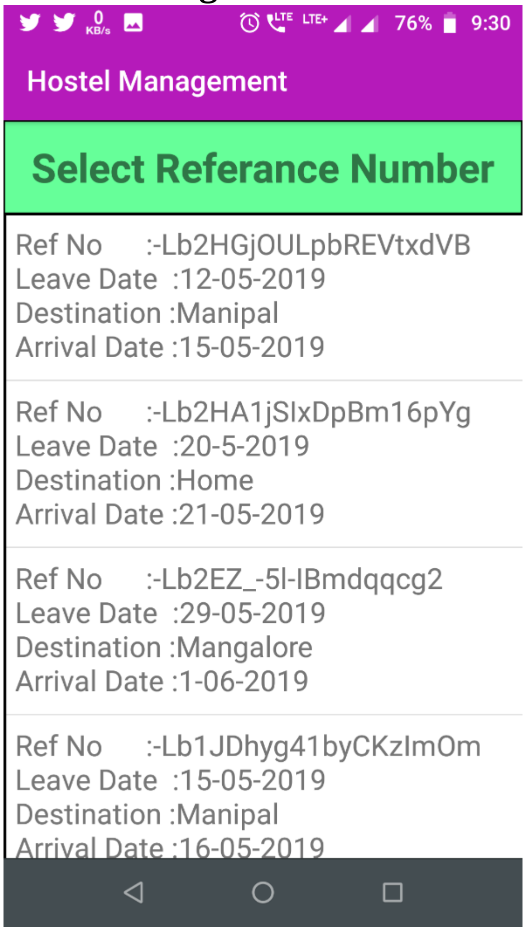
i. Options with the register



ii. Add register form



iii. List of registers added



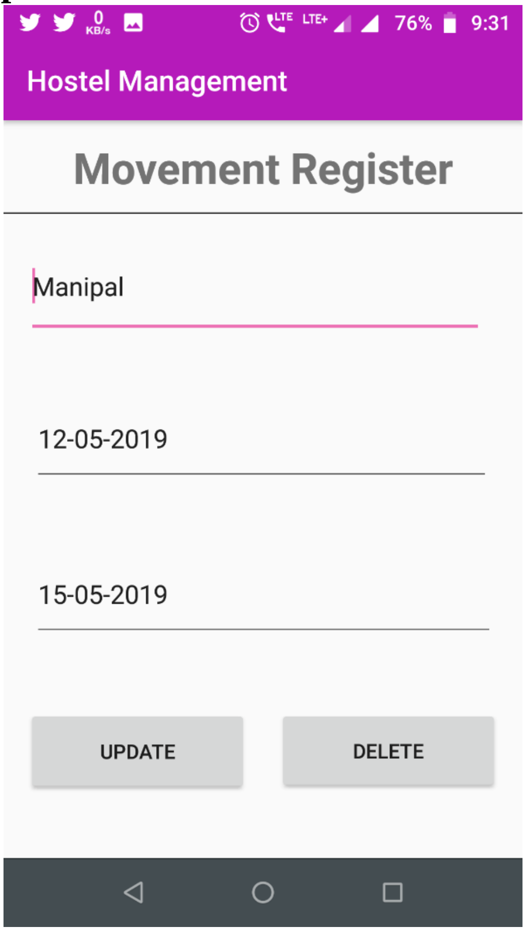
Ref No :-Lb2HGjOULpbREVtxdVB
Leave Date :12-05-2019
Destination :Manipal
Arrival Date :15-05-2019

Ref No :-Lb2HA1jSlxDpBm16pYg
Leave Date :20-5-2019
Destination :Home
Arrival Date :21-05-2019

Ref No :-Lb2EZ_-5l-IBmdqqcg2
Leave Date :29-05-2019
Destination :Mangalore
Arrival Date :1-06-2019

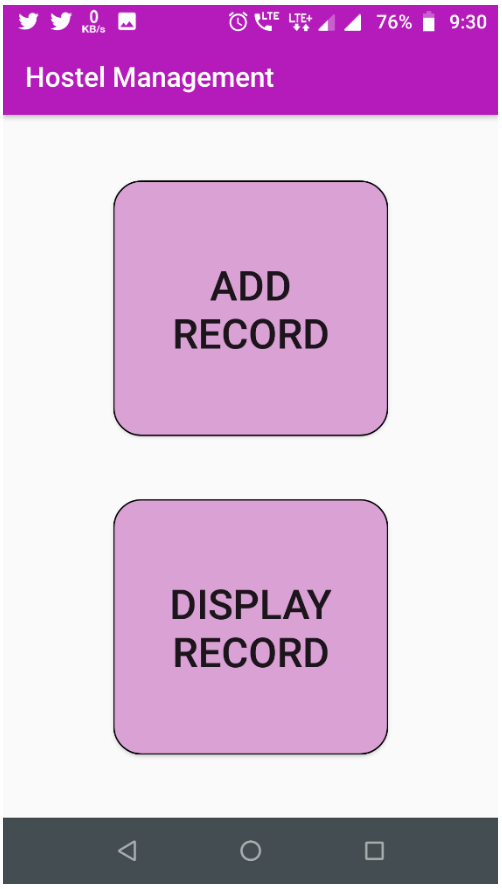
Ref No :-Lb1JDhyg41byCKzImOm
Leave Date :15-05-2019
Destination :Manipal
Arrival Date :16-05-2019

iv. Update form with a delete button

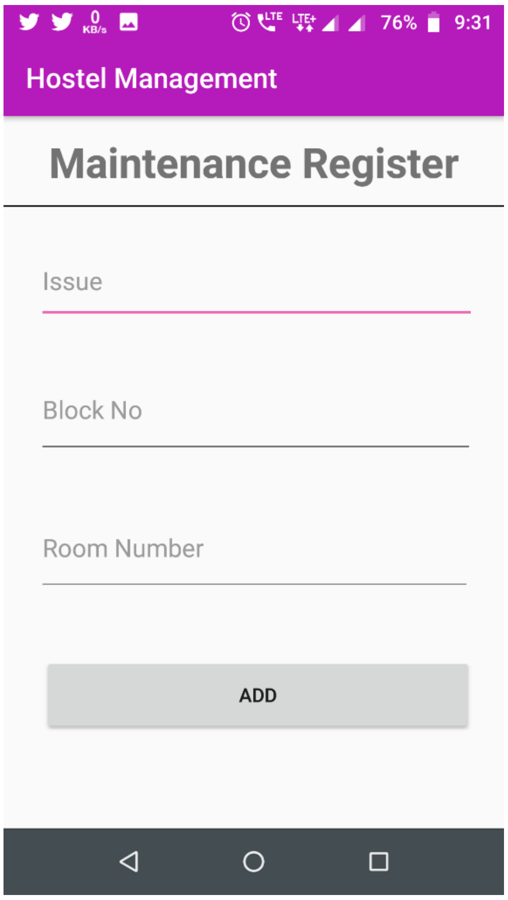


VIII. Maintenance Register

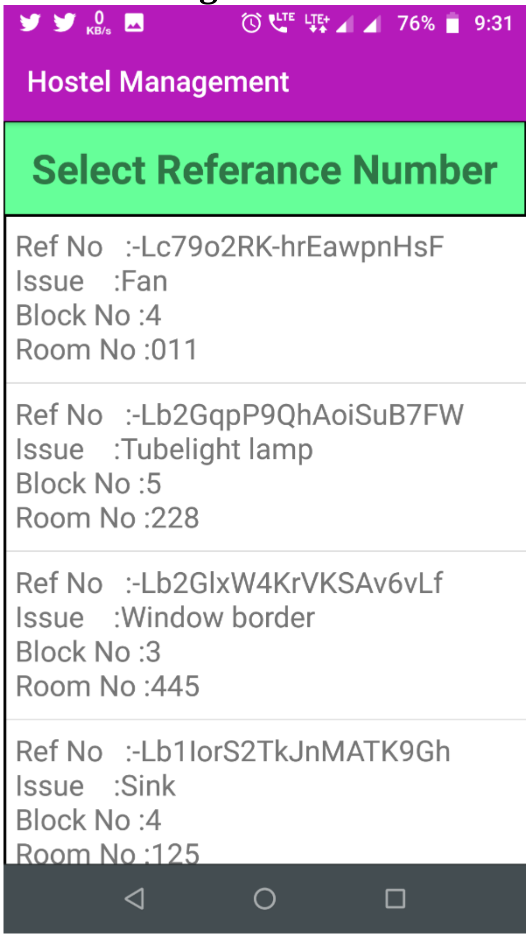
i. Add or delete Options



ii. Add register form



iii. List of registers added



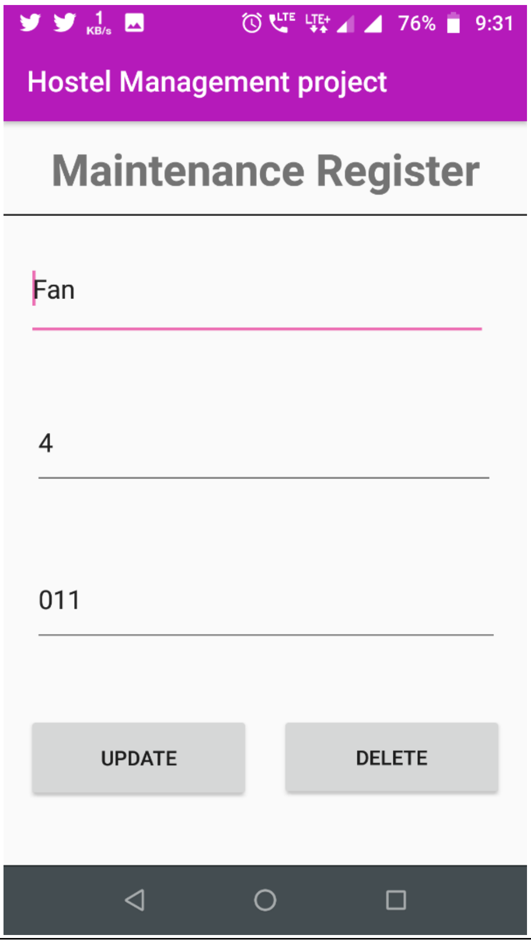
Ref No :-Lc79o2RK-hrEawpnHsF
Issue :Fan
Block No :4
Room No :011

Ref No :-Lb2GqpP9QhAoiSuB7FW
Issue :Tubelight lamp
Block No :5
Room No :228

Ref No :-Lb2GlxW4KrVKSAv6vLf
Issue :Window border
Block No :3
Room No :445

Ref No :-Lb1lorS2TkJnMATK9Gh
Issue :Sink
Block No :4
Room No :125

iv. Update form with a delete button

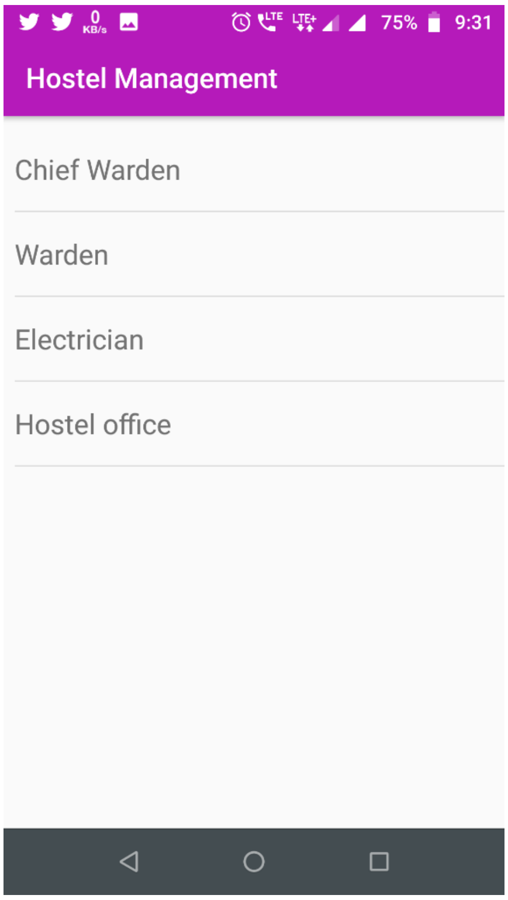


Fan

4

011

IX.Contact Info(on clicking on any view loads call intent)

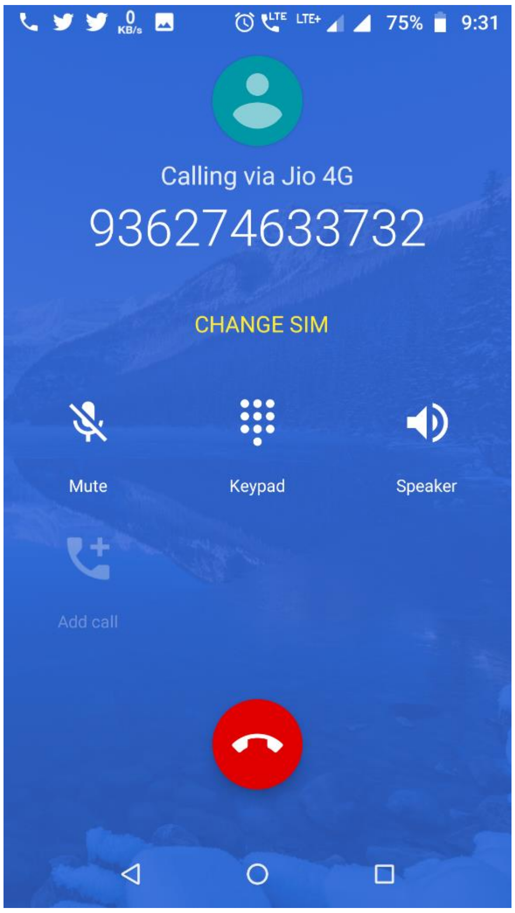

Hostel Management

Chief Warden

Warden

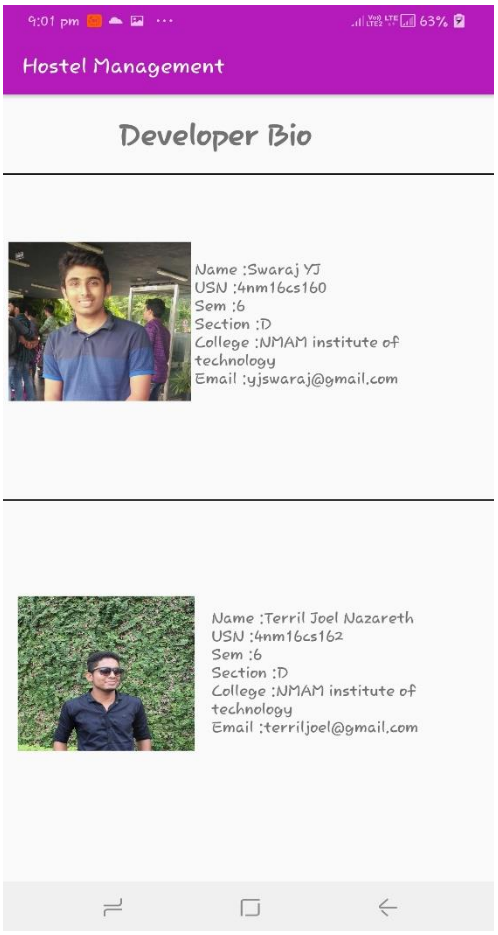
Electrician

Hostel office


Calling via Jio 4G

936274633732

X. Developer Bio


Hostel Management

Developer Bio

Name :Swaraj YJ
USN :4nm16cs160
Sem :6
Section :D
College :NMAM institute of technology
Email :yjswaraj@gmail.com

Name :Terril Joel Nazareth
USN :4nm16cs162
Sem :6
Section :D
College :NMAM institute of technology
Email :terriljoel@gmail.com

CHAPTER 8

CONCLUSION AND FUTURE WORK

8.1 Results/conclusion:

- Hostel Management app is an android application used to manage the activities of the hostel.
- Movement register is used to keep track of student when they go out from the hostel.
- Maintenance register used to record any damage done to the utilities of the hostels.
- Circulars are used to send important notices to the students leaving in the hostel.
- Mess time gives the update menu of the dishes served in the hostel
- It also has contact information of the in-charge of the hostel.

8.2 Future Works

This mobile can be further developed by adding more features like GPS enabled student locator, Notifications etc. For now back end i.e., firebase is managed manually using firebase console which can improved by developing web page for the administrator.

8.3 References

- https://howtofirebase.com/what-is-firebase-fcb8614ba442
- https://stackoverflow.com/
- https://www.geeksforgeeks.org/