



PROJECT TITLE

MESS MANAGEMENT SYSTEM

A PROJECT REPORT

submitted by

K. SAI SRI HARSHA – 20BCB129

LAKSHYA VASWANI – 20BCB0132

MARLAPALLI RITIKA- 20BCB0134

SHUBHAM JAISWAL – 20BCB0124

SAGUTURU KAVYA – 20BCB0142

CSE1005 - Software Design and Development

Faculty: KUMAR K

Slot: C1+ TC1

MAY 2022

INDEX

1. Introduction

- 1.1 Abstract
- 1.2 Purpose.....
- 1.3 Literature Survey
- 1.4 Definitions, Acronyms & Abbreviations

2. Planning & Scheduling.....

- 2.1 SDLC model (20BCB0142).....
- 2.2 Work Breakdown Structure (20BCB0142)
- 2.3 Gantt Chart (20BCB0134).....
- 2.4 Pert Chart(20BCB0134).....
- 2.5 Use Case Model(20BCB0134).....
- 2.6 Data Flow Diagram(20BCB0124).....
- 2.7 Sequence Diagram(20BCB0129).....
- 2.8 Activity Diagram(20BCB0132).....
- 2.9 Class Diagram(20BCB0132).....
- 2.10 ER Model (20BCB0142).....
- 2.11 State Diagram(20BCB0129).....

3. Software Requirements Specification

- 3.1 Functional Requirements
- 3.2 Non-Functional Requirements
- 3.3 Assumptions & Constraints

INDIVIDUAL CONTRIBUTION :

20BCB0142 – ANALYSIS

20BCB0134 – DESIGN

20BCB0129 – CODE

20BCB0132 – IMPLEMENTATION

20BCB0142 – TESTING

Declaration

This report has been prepared on the basis of our own work. Wherever otherpublished and unpublished source materials have been used, these have beenacknowledged.

1. INTRODUCTION

1.1 ABSTRACT

To make the mess selection procedure transparent, easy and select the best mess according to your requirements. When it comes to mess selection there are various options to select from and anyone can get confused which one which offers him best food. There are various mess caterers in the campus and many types of mess like north veg, north non- veg, special, south Indian, and Paid mess. So, we are going to design and develop an android application to deal with all sorts of thing.

1.2 PROBLEM STATEMENT

For our problem statement we have decided to work upon something which would be relevant to all the students and could easily be implemented. One of the major issues which hostel students face is related to mess food and timings. It often happens that students skip their meal due to long waiting period at their hostel mess ,which may eventually lead to them missing

the next session or one may not like the food quality/ taste at a particular mess . Apart from this , there might be different preferences of cuisine by students from different states. For example south Indians may prefer south indian and so on.

We have tried to address this issue and came up with a software application which be a one stop solution for any problem around student mess in VIT. The app has 2 different interfaces for students and manager of the mess.

1.3 LITERATURE SURVEY

According to an automated food ordering system is proposed which will keep track of user orders smartly. Basically, they implemented a food ordering system for different type of restaurants in which user will make order or make custom food by one click only. By means of android application for Tablet PCs this system was implemented. The front end was developed using JAVA, Android and at the backend MySQL database was used. According to Customer using a Smartphone is considered as a basic assumption for the system. When the customer approach to the restaurant, the saved order can be confirmed by touching the Smartphone. The list of selected preordered items shall be shown on the kitchen screen, and when confirmed, order slip shall be printed for further order processing. The solution provides easy and convenient way to select pre-order transaction form customers. According to there was an attempt to design and implementation of digital dining in restaurants using android technology. This system was a basic dynamic database utility system which fetches all information from a centralized database. Efficiency and accuracy of restaurants as well as human errors were improved by this user-friendly application. Earlier drawbacks of automated food ordering systems were overcome by this system and it requires a onetime investment for gadgets. In an application of integration of hotel management systems by web services technology is presented. Ordering System Kitchen Order Ticket (KOT), Billing System, Customer Relationship Management system (CRM) are held together by the Digital Hotel Management. Add or expand of hotel software system in any size of hotel chains environment was possible with this solution. According to research work aims to design and

develop a wireless food ordering system in the restaurant. Technical operations of Wireless Ordering System (WOS) including systems architecture, function, limitations

2. PLANNING AND SCHEDULING :

2.1 SLDC MODEL :

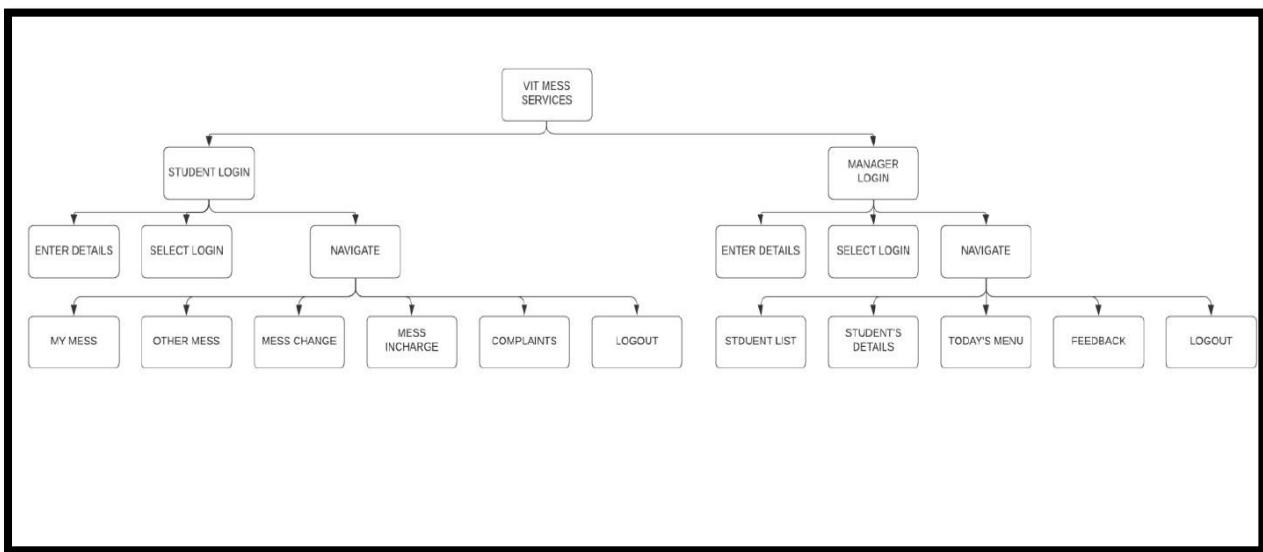
PROCESS MODEL : ITERATIVE WATERFALL

In our project, we are using the iterative waterfall model because it allows us the flexibility to rectify the mistakes at the same stage along with the scope of rollback which is very useful if you scroll to some other portion of the app and want to undo the changes.

- When errors are detected at some later phase, these feedback paths allow correcting errors committed by programmers during some phase. The feedback paths allow the phase to be reworked in which errors are committed and these changes are reflected in the later phases. But there is no feedback path to the stage – feasibility study, because once a project has been taken, does not give up the project easily.
- It is good to detect errors in the same phase in which they are committed. It reduces the effort and time required to correct the errors.
- Let's take an example where the manager can look for new registration and if there's an error it can be rectified in that same phase. The new registrations are added under his supervision and will be verified once.
- As a part of Testing purpose feedback portion is introduced in the app itself to keep check on the complaints/suggestion to improve.
- This is the most feasible model for the app and satisfies every property of iterative model from feasibility to testing and maintenance as it will be linked to server from VIT SDC which will be updated in accordance to the registration made by students from the VTOP site

2.2

WORK BREAKDOWN STRUCTURE:



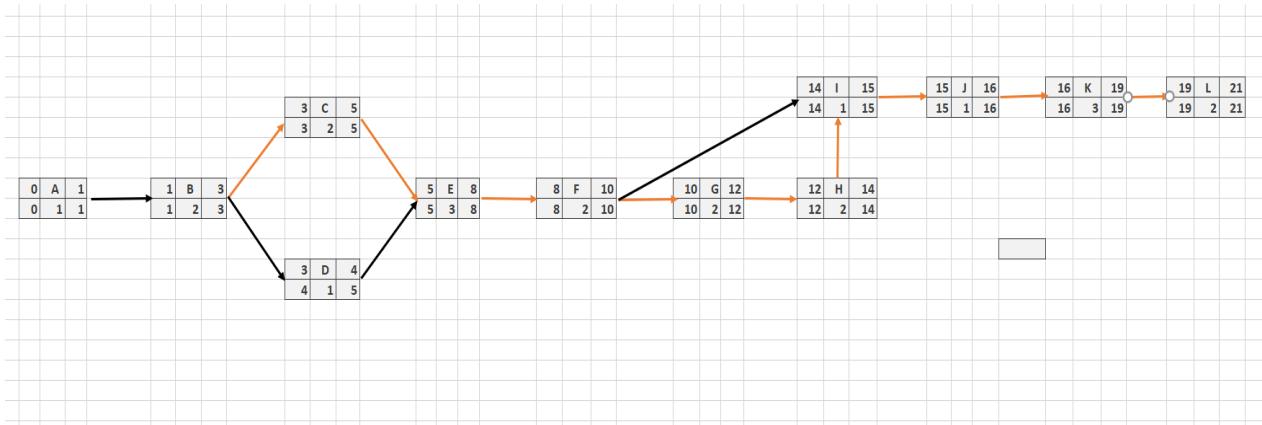
2.3

GANTT CHART :

ACTIVITIES		Dependencies	Duration(weeks)
Set up project acquisition team	A	NONE	1
Noting down the software requirements	B	A	2
Identifying the functional requirements	C	B	2
Identifying the non functional requirements	D	B	1
Developing a design plan and code implementation	E	CD	3
Developing software testing plan	F	E	2
Software customization phase 1	G	F	2
Testing the first release	H	G	2
Software customization phase 2	I	FH	1
Testing the second release	J	I	1
Updation of the software	K	J	3
Software finalisation and project sign off	L	K	2

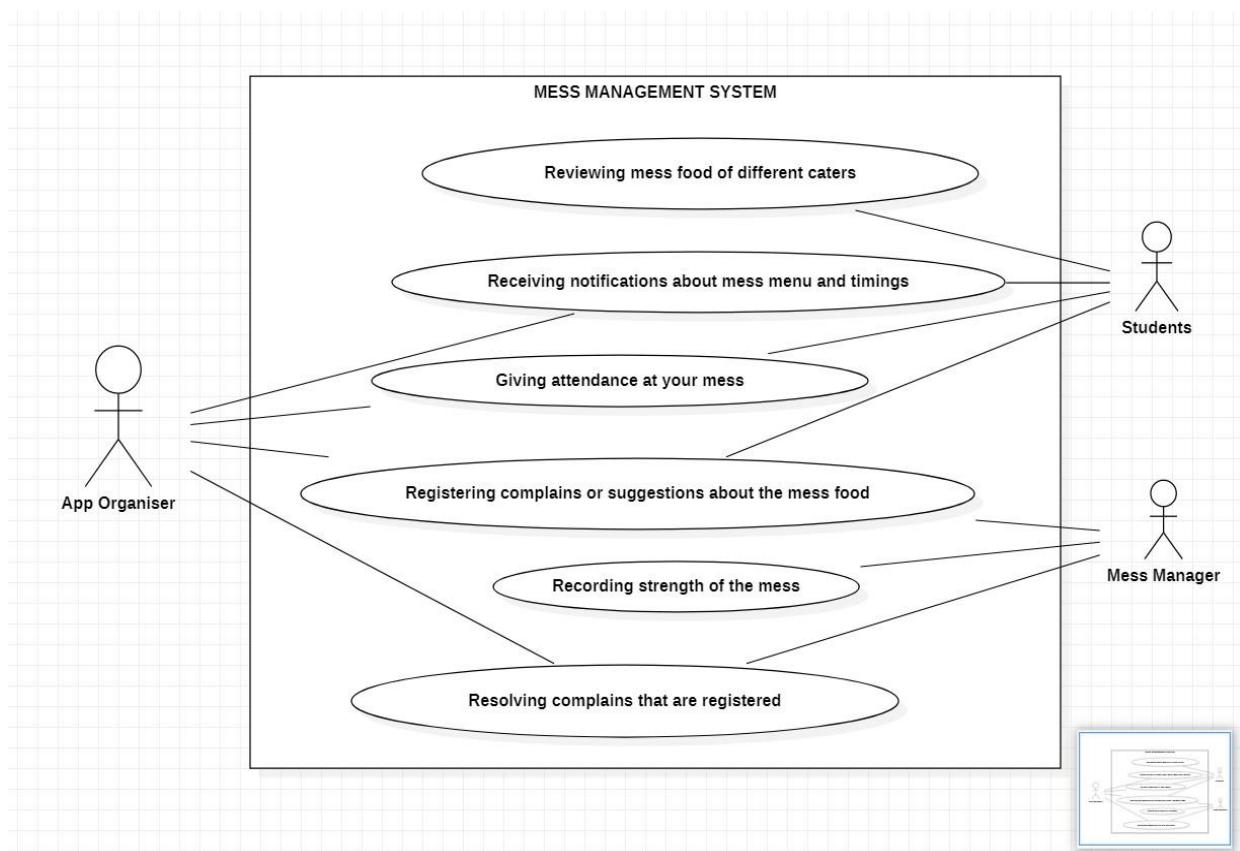
2.4 PERT CHART :

CRITICAL PATH AND MILESTONES :



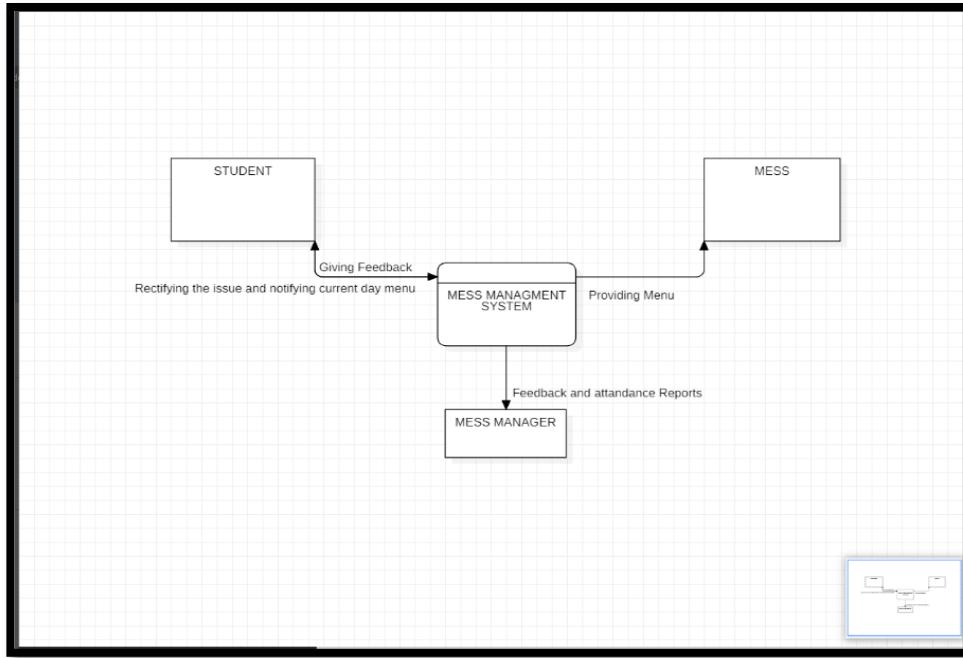
A → B → C → E → F → G → H → I → J → K → L

2.5 USE CASE MODEL :

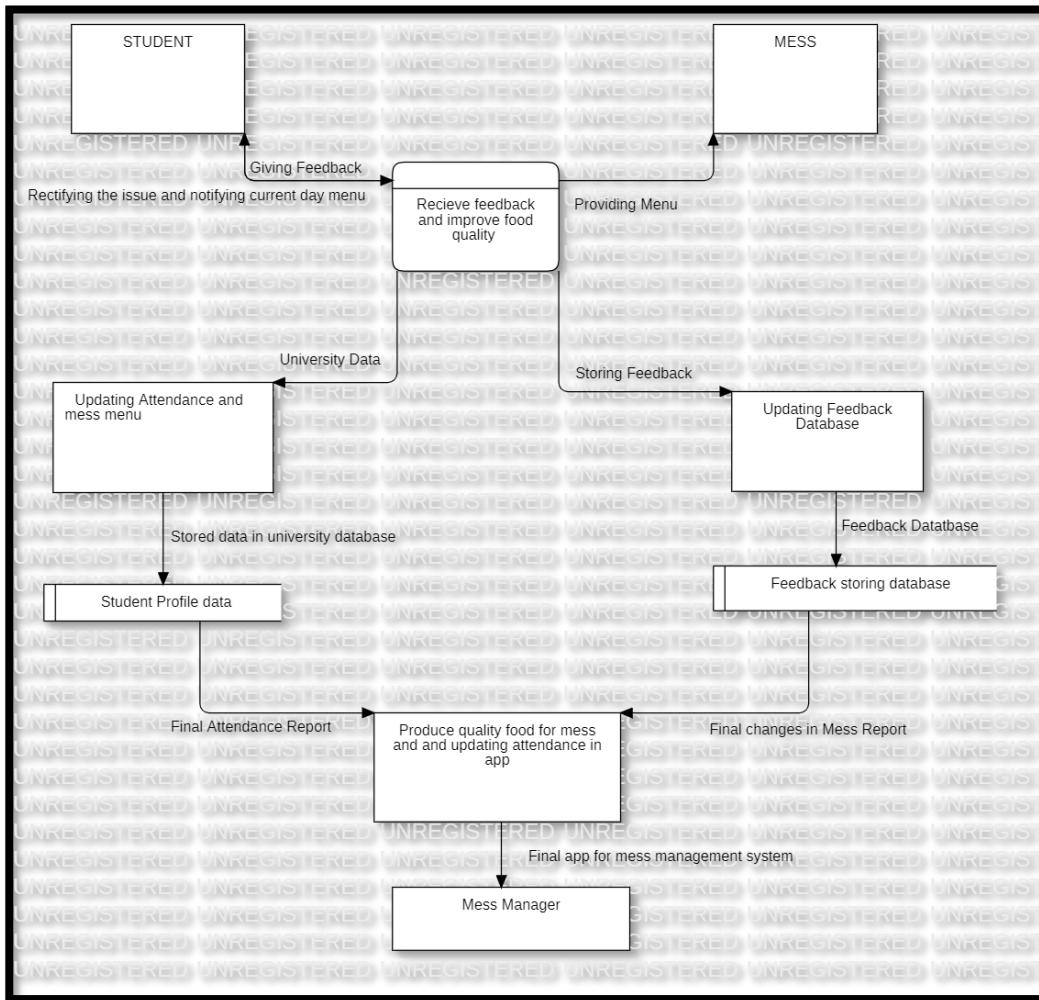


2.6 DATA FLOW DIAGRAM :

DFD LEVEL 0

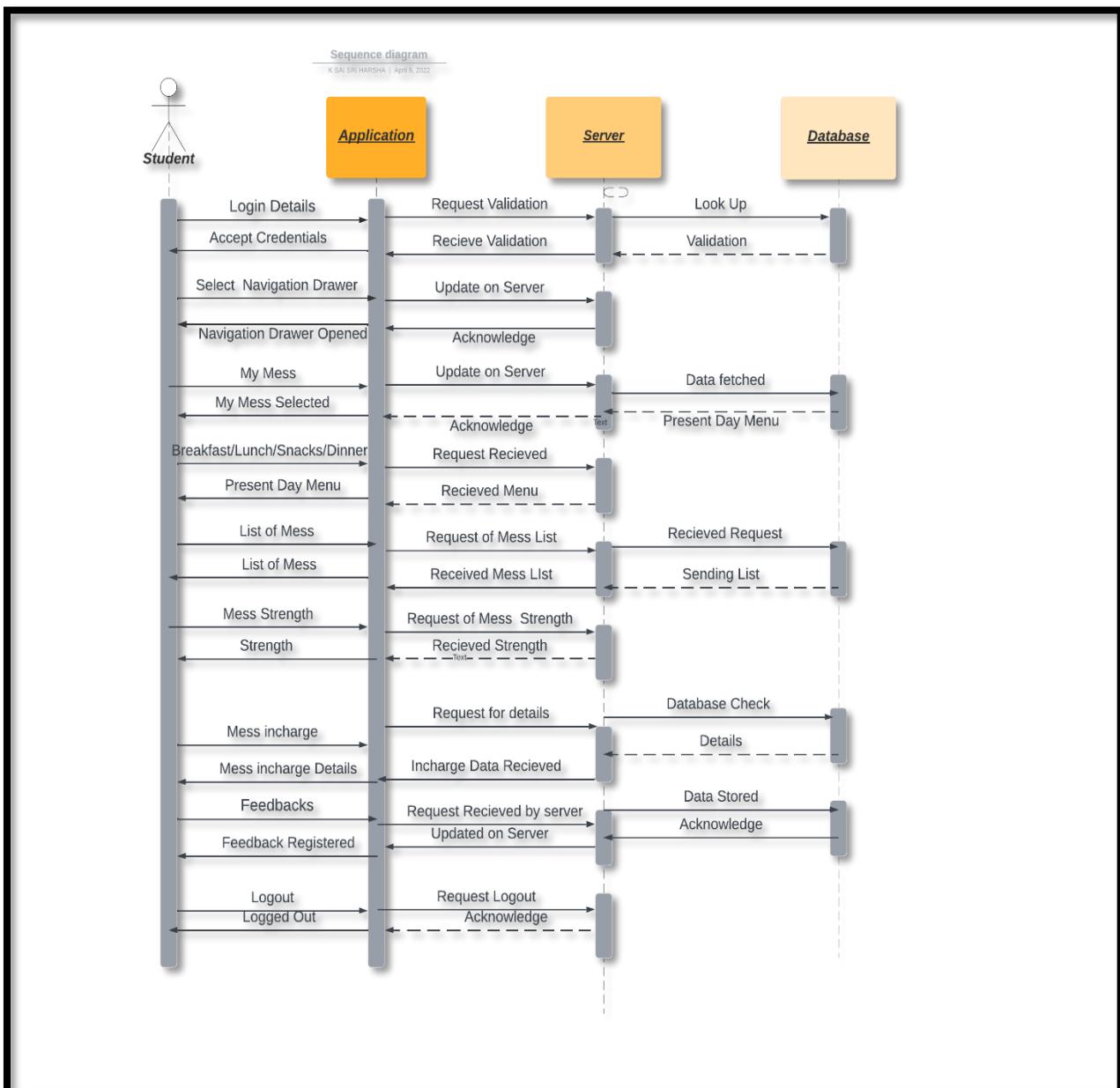


DFD LEVEL 1 :

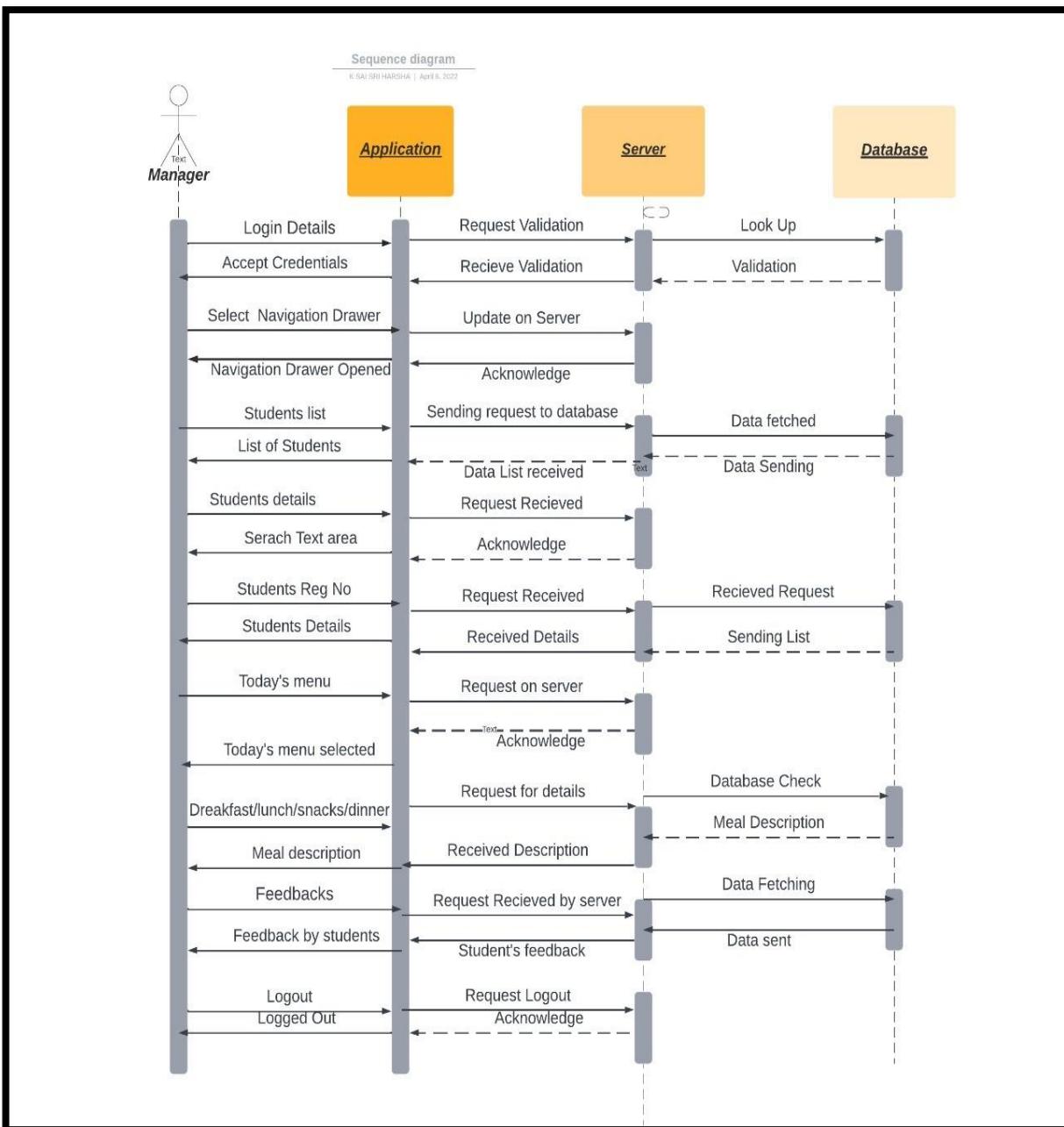


2.7 SEQUENCE DIAGRAM :

SEQUENCE DIAGRAM OF STUDENT

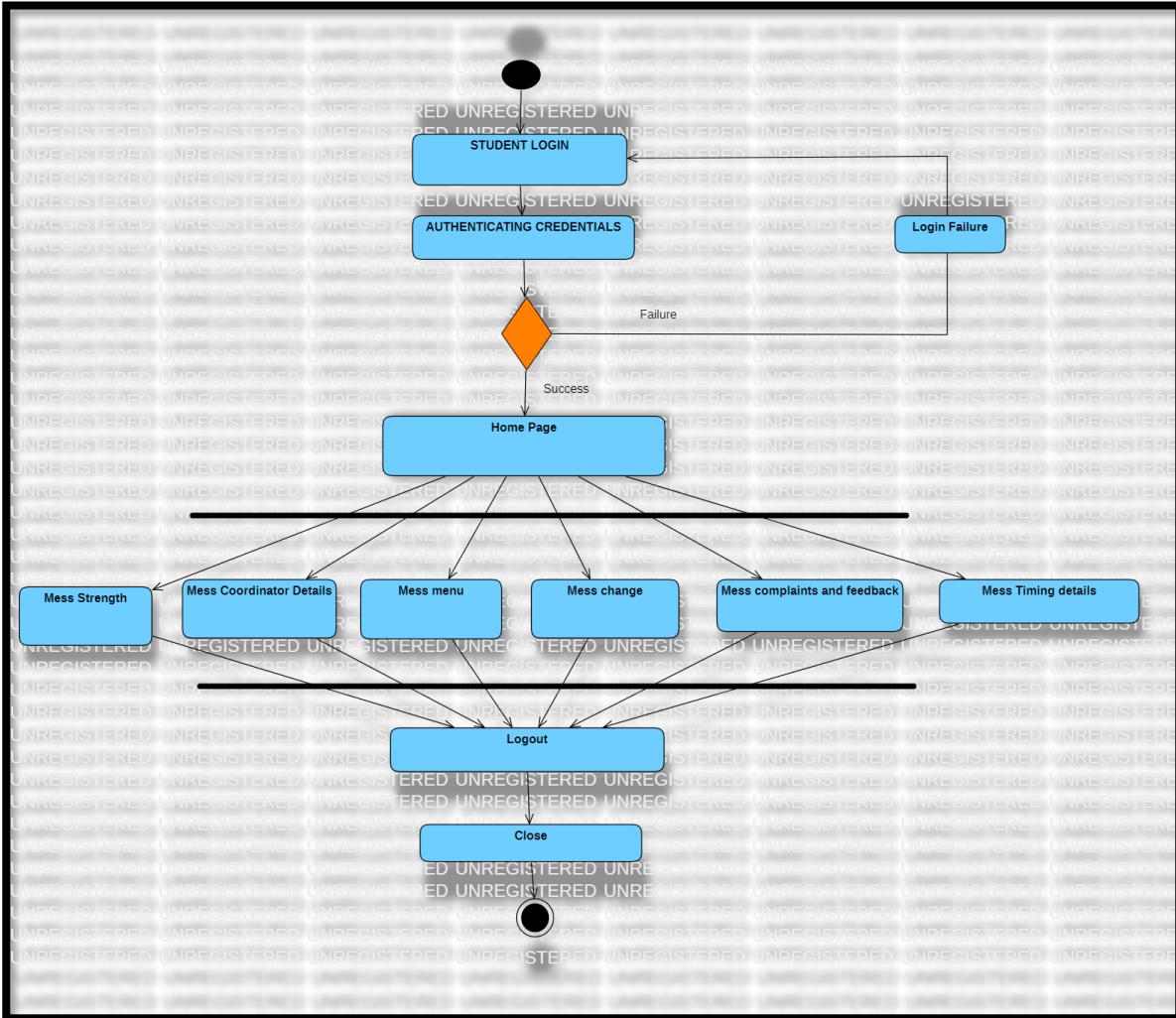


SEQUENCE DIAGRAM OF MANAGER

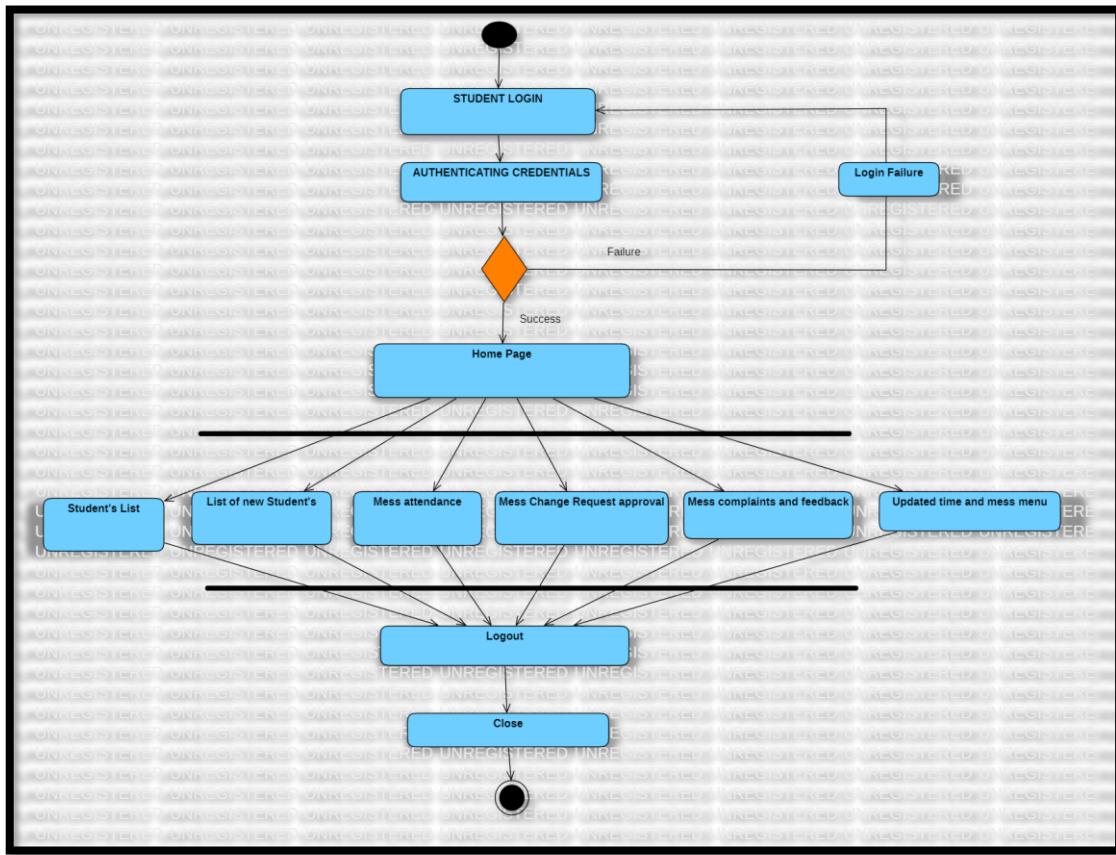


2.8 ACTIVITY DIAGRAM :

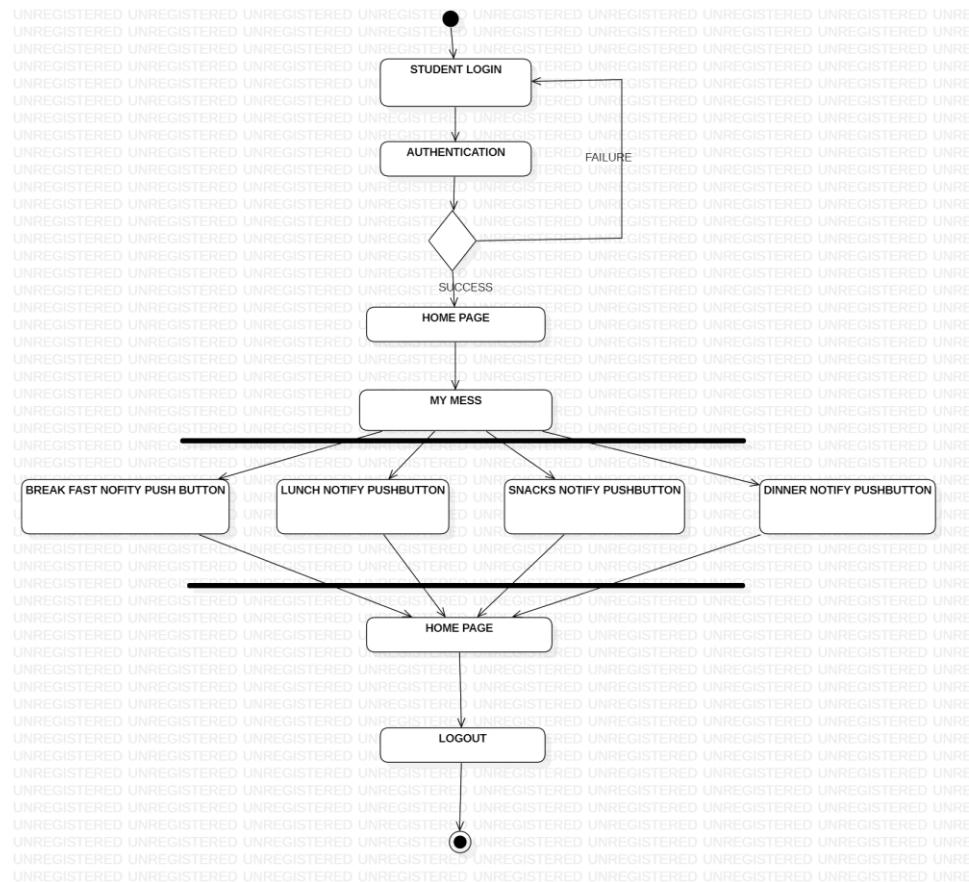
ACTIVITY DIAGRAM OF STUDENT



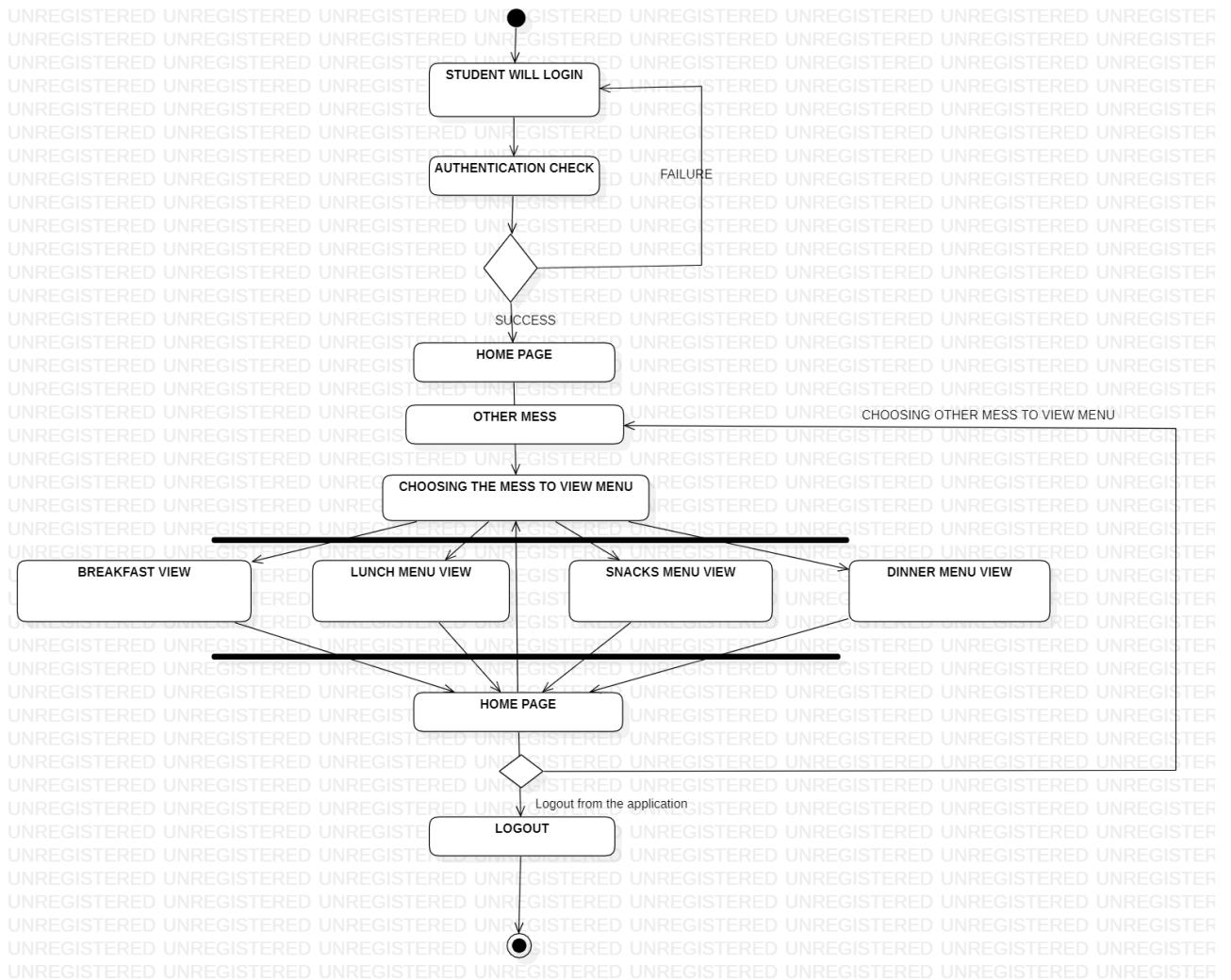
ACTIVITY DIAGRAM OF MANAGER



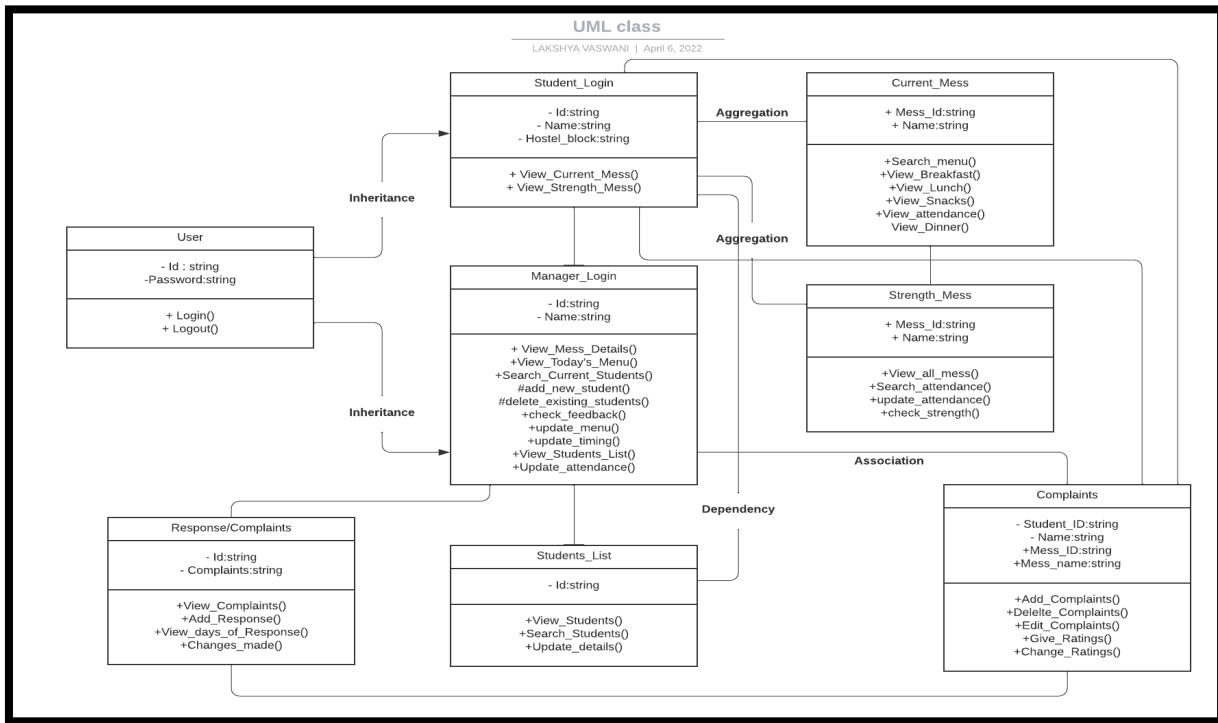
STUDENT MY MESS



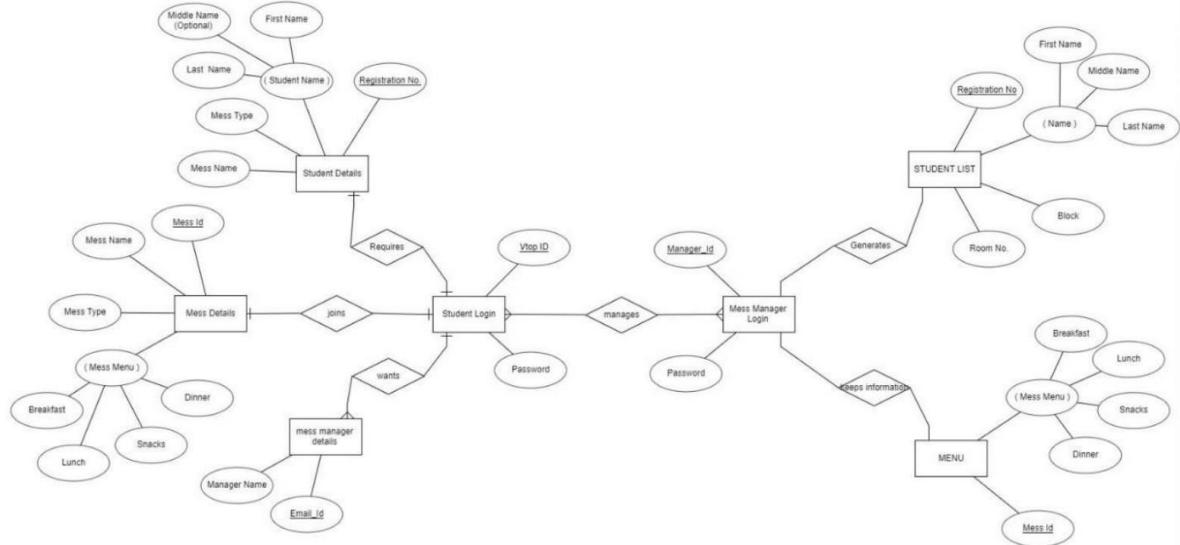
STUDENT OTHER MESS



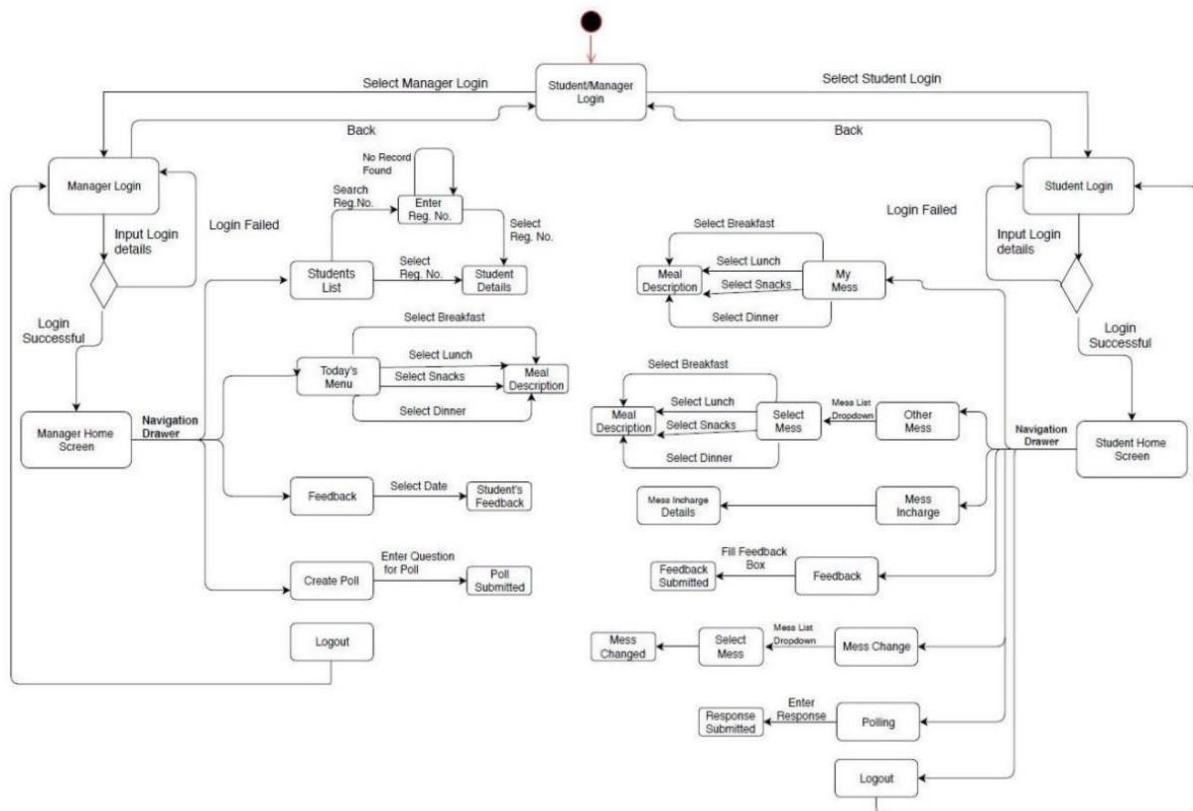
2.9 CLASS DIAGRAM :



2.10 ER MODEL :



2.11 STATE DIAGRAM :



3. SOFTWARE REQUIREMENT SPECIFICATIONS :

3.1 FUNCTIONAL REQUIREMENTS :

1. There are multiple mess caterers in VIT, this app provides transparent information about each mess, like quality of the food, mess timing, variety of dishes, cleanliness.
2. If a student does not like the current mess he is in, He can opt for the change of mess through our app.
3. This app sends the notification of the menu during the mess timings. So that can remind of their current day mess menu without checking manually in their mess.
4. Student will be provided with the manager details so that they can directly interact with them and, can convey their problems like cleanliness, food quality, etc. (Only Email id will be provided as contact)
5. The manager will have the record of each and every student in his mess. He/she will even have the record of the students entering and leaving the mess and hence the strength of the mess.

3.2 NON FUNCTIONAL REQUIREMENTS :

ORGANISATIONAL REQUIREMENTS:

The app is fully secure as it will be managed by the VIT Server and all the details will be end to end encrypted. The details such as Mess and Student's info will be taken altogether by VIT and hence there's no chance of insecurity or data loss.

SECURITY AND SIMPLER LOGIC:

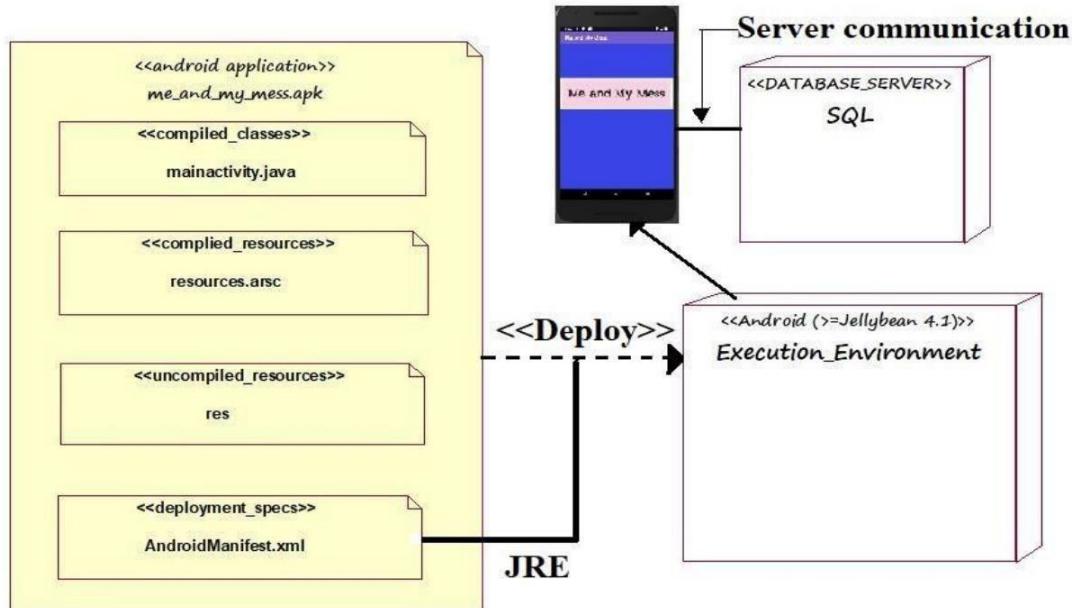
The backend is also designed in a way to serve in accordance to the secure and smooth working of the app to make it glitch-free for the consumer which is students and manager over here.

The time taken for the average person to open on click is mentioned below:

In accordance to the GOMS Keystroke Level Model,

- Opening of Main menu took <3 sec
- Login time varies from person to person and on the details of keystroke used in passwords and username
- The page layout will take all in all <=1 sec to open and undo
- The feedback page depends on the user feedback input time and submission.

Deployment Architecture



Implementation Code

Manager login

```
import android.content.Intent;  
import android.support.v7.app.AppCompatActivity;  
import android.os.Bundle;  
import android.view.View;
```

```
import
android.widget.Button;
import
android.widget.EditText;
import android.widget.Toast;
public class manager_login extends
AppCompatActivity { private Button;
private EditText
password,editText; @Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_manager_login);
getSupportActionBar().setTitle("Manager Login");
getSupportActionBar().setDisplayHomeAsUpEnabled(true);
button = (Button) findViewById(R.id.button4);
editText = (EditText) findViewById(R.id.editText2);
password = (EditText) findViewById(R.id.editText3);
button.setOnClickListener(new
View.OnClickListener() {
@Override
public void onClick(View v)
{ if(validate()) {
```

```
        openActivity1();
    ); finish();
}
else {
    openDialog();
    password.setText("");
);
editText.setText("");
}
}
});
}

public void openActivity1()
{
    Intent intent1 = new
Intent(this,MainActivity.class);
    startActivity(intent1);
}
public void openDialog()
{
    Ex2Dialog e = new Ex2Dialog();
    e.show(getSupportFragmentManager(),"Dialo
g");
}
private Boolean validate()
{
    Boolean result = false;
```

```
String name=editText.getText().toString().trim();
String pass=password.getText().toString();
if(name.isEmpty() && pass.isEmpty())
    Toast.makeText(this, "Please Enter All The
Details", Toast.LENGTH_SHORT).show();
}
else if((name.equals("Prmess")) && (pass.equals("hello")))
{
    Toast.makeText(this, "LOGIN SUCCESSFUL",
Toast.LENGTH_SHORT).show();
    result = true;
}
return result;
}
```

Student login

```
import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

public class studentLogin extends AppCompatActivity {
```

```
private Button;  
private EditText password,editText;  
String name,studentname,messname,pass1;  
DatabaseAccess a= new DatabaseAccess();  
@Override  
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_student_login);  
    getSupportActionBar().setTitle("Student Login");  
    getSupportActionBar().setDisplayHomeAsUpEnabled(true);  
    button = (Button) findViewById(R.id.button_login);  
    editText = (EditText) findViewById(R.id.edittext_username);  
    password = (EditText) findViewById(R.id.edittext_password);  
    button.setOnClickListener(new View.OnClickListener() {  
        @Override  
        public void onClick(View v) {  
            if(validate()) {  
                openActivity1();  
                finish();  
            }  
            else {  
                openDialog();  
                password.setText("");  
                editText.setText("");  
            }  
        }  
    }  
}
```

```
        }
    });
}

public void openActivity1()
{
    Intent intent1 = new Intent(this,student_Activity.class);

    startActivity(intent1);
}

public void openDialog()
{
    Ex2Dialog e = new Ex2Dialog();
    e.show(getSupportFragmentManager(),"Dialog");
}

public Boolean validate()
{
    name=editText.getText().toString().trim();
    Boolean result = false;
    studentname=a.student_name(name);
    pass1=a.password(name);
    messname=a.mess_name(name);
    Intent i = new Intent(studentLogin.this,Student_my_mess.class);
    i.putExtra("Value1",studentname);
    i.putExtra("Value2",messname);
    String pass=password.getText().toString();

    if(name.isEmpty() && pass.isEmpty()) {
```

```
        Toast.makeText(this, "Please Enter All The Details",
Toast.LENGTH_SHORT).show();

    }

    else if(pass.equals(pass1))

    {

        Toast.makeText(this, "LOGIN SUCCESSFUL", Toast.LENGTH_SHORT).show();
        result = true;
    }

    return result;
}

}
```

Student_my_mess

```
import android.content.Intent;

import android.support.v7.app.AppCompatActivity;

import android.os.Bundle;

import android.view.View;

import android.widget.Button;

import android.widget.CompoundButton;

import android.widget.EditText;

import android.widget.Switch;

import android.widget.TextView;

import android.widget.Toast;

public class Student_my_mess extends AppCompatActivity implements

View.OnClickListener,CompoundButton.OnCheckedChangeListener {

    private Button button5,button6,button7,button8;

    private EditText Menu;

    private TextView t1,t2;
```

```
String l,m,mn,p;
DatabaseAccess k= new DatabaseAccess();
private Switch s1,s2,s3,s4;
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_student_my_mess);
    getSupportActionBar().setTitle("My Mess");
    getSupportActionBar().setDisplayHomeAsUpEnabled(true);
    s1 = (Switch) findViewById(R.id.switch1);
    s2 = (Switch) findViewById(R.id.switch2);
    s3 = (Switch) findViewById(R.id.switch3);
    s4 = (Switch) findViewById(R.id.switch4);

    t1=(TextView)findViewById(R.id.get_name);
    t2=(TextView)findViewById(R.id.mess_name1);
    Intent jk= getIntent();
    l=jk.getStringExtra("Value1");
    p=jk.getStringExtra("Value2");

    m=k.student_name("20BCB0129");
    mn=k.mess_name("20BCB0129");

    t1.setText(m);
    t2.setText(mn);

    button5 = (Button) findViewById(R.id.button5);
```

```
button6 = (Button) findViewById(R.id.button6);
button7 = (Button) findViewById(R.id.button7);
button8 = (Button) findViewById(R.id.button8);
Menu =(EditText) findViewById(R.id.fooditem);
button5.setOnClickListener(this);
button6.setOnClickListener(this);
button7.setOnClickListener(this);
button8.setOnClickListener(this);

s1.setOnCheckedChangeListener(this);
s2.setOnCheckedChangeListener(this);
s3.setOnCheckedChangeListener(this);
s4.setOnCheckedChangeListener(this);

}

DataAccessManager a= new DataAccessManager();
@Override
public void onClick(View v){
    switch(v.getId()){
        case R.id.button5:
            String breakfast=a.getbreakfast(mn);
            if(breakfast.equals("Not found")){
                Menu.setText("");
                Menu.setText("BREAKFAST");
            }
        else
    {

```

```
        Menu.setText(breakfast);
    }
    break;

case R.id.button6:
    String lunch = a.getlunch(mn);
    if(lunch.equals("Not found")) {
        Menu.setText("");
        Menu.setText("LUNCH");
    }
    else
    {
        Menu.setText(lunch);
    }
    break;

case R.id.button7:
    String snacks = a.getsnacks(mn);
    if(snacks.equals("Not found")) {
        Menu.setText("");
        Menu.setText("SNACKS");
    }
    else
    {
        Menu.setText(snacks);
    }
    break;

case R.id.button8:
    String dinner= a.getdinner(mn);
```

```
        if(dinner.equals("Not found")) {  
            Menu.setText("");  
            Menu.setText("DINNER");  
        }  
        else  
        {  
            Menu.setText(dinner);  
        }  
        break;  
    }  
}  
  
@Override  
public void onCheckedChanged(CompoundButton bv, boolean isChecked) {  
    switch(bv.getId())  
    {  
        case R.id.switch1:  
            if(isChecked){  
                Toast.makeText(this,"BreakFast  
Toast.LENGTH_SHORT).show();  
            }  
            else{  
                Toast.makeText(this,  
Toast.LENGTH_SHORT).show();  
            }  
            break;  
        case R.id.switch2:  
    }  
    Notification  
    ON",  
    "BreakFast  
    Notification  
    OFF",  
}
```

```
if(isChecked){  
    Toast.makeText(this, "Lunch Notification ON", Toast.LENGTH_SHORT).show()  
}  
else{  
    Toast.makeText(this, "Lunch Notification OFF", Toast.LENGTH_SHORT).show()  
  
}  
break;  
case R.id.switch3:  
if(isChecked){  
    Toast.makeText(this, "Snacks Notification ON", Toast.LENGTH_SHORT).show()  
}  
else{  
    Toast.makeText(this, "Snacks Notification OFF", Toast.LENGTH_SHORT).show()  
  
}  
break;  
case R.id.switch4:  
if(isChecked){  
    Toast.makeText(this, "Dinner Notification ON", Toast.LENGTH_SHORT).show()  
}  
else{  
    Toast.makeText(this, "Dinner Notification OFF", Toast.LENGTH_SHORT).show()  
  
}  
break;
```

```
    }
}
}
```

Student_other_mess

```
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.Spinner;
import android.widget.TextView;
import android.widget.Toast;

import java.util.ArrayList;
import java.util.List;

public class student_other_mess extends AppCompatActivity implements
View.OnClickListener{
    private Button button5,button6,button7,button8;
    private TextView Menu;
    String itemvalue;
    Spinner dropdownmenu;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
```

```
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_student_other_mess);
getSupportActionBar().setTitle("Other Mess");
getSupportActionBar().setDisplayHomeAsUpEnabled(true);
button5 = (Button) findViewById(R.id.button18);
button6 = (Button) findViewById(R.id.button19);
button7 = (Button) findViewById(R.id.button20);
button8 = (Button) findViewById(R.id.button22);
Menu =(TextView) findViewById(R.id.textView7);
button5.setOnClickListener(this);
button6.setOnClickListener(this);
button7.setOnClickListener(this);
button8.setOnClickListener(this);
```

```
dropdownmenu = (Spinner) findViewById(R.id.spinner4);
List<String> list=new ArrayList<>();
list.add("PR 1 - VEG MESS (B-ANNEX)");
list.add("PR 2 - SPECIAL MESS (B-ANNEX)");
list.add("SKC 1 - NON VEG MESS (OPP TO F-BLOCK)");
list.add("SKC 2 - SPECIAL MESS (OPP TO F-BLOCK)");
list.add("RSM 1 - VEG MESS (G-BLOCK)");
list.add("RSM 2 - SPECIAL MESS (G-BLOCK)");
list.add("CRC 1 - VEG MESS (H-BLOCK)");
list.add("CRC 2 - NON VEG MESS (H-BLOCK)");
list.add("CRC 3 - SPECIAL MESS (J-BLOCK)");
list.add("CRC 4 - FOOD PARKK (J-BLOCK)");
list.add("RRC 1 - VEG MESS (L-BLOCK)");
```

```
list.add("RRC 2 - NON VEG MESS (L-BLOCK)");
list.add("RRC 3 - SPECIAL MESS (L-BLOCK)");
list.add("RRC 4 - SPECIAL MESS (D- ANNEX)");
list.add("RRC 5 - FOOD MALL (L-BLOCK)");
list.add("MHPL 1 - VEG MESS (F-BLOCK)");
list.add("MHPL 2 - SPECIAL MESS (F-BLOCK)");
list.add("AAC 1 - NON VEG MESS (F-BLOCK)");
list.add("AAC 2 - SPECIAL MESS (F-BLOCK)");
list.add("ZEN 1 - VEG MESS (K-BLOCK)");
list.add("ZEN 2 - NON VEG (K-BLOCK)");
list.add("ZEN 3 - SPECIAL (K-BLOCK)");
list.add("ZEN 4 - FOODCY (K-BLOCK)");
list.add("PR 3 - VEG MESS (P-BLOCK)");
list.add("PR 4 - SPECIAL MESS (P-BLOCK)");
list.add("DR 1 - NON VEG (P-BLOCK)");
list.add("DR 2 - SPECIAL MESS (P-BLOCK)");
list.add("PRD 1- VEG MESS (Q-block)");
list.add("PRD 2- Buddies & Bites (Q-block);
```

```
ArrayAdapter<String> adapter = new ArrayAdapter<String>(this,
android.R.layout.simple_spinner_item, list);
```

```
adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
dropdownmenu.setAdapter(adapter);
```

```
dropdownmenu.setOnItemSelectedListener(new
AdapterView.OnItemSelectedListener() {
```

```
    @Override
```

```
public void onItemSelected(AdapterView<?> parent, View, int position, long id) {  
    itemvalue = parent.getItemAtPosition(position).toString();  
    Toast.makeText(student_other_mess.this,"Selected " + itemvalue,  
    Toast.LENGTH_SHORT).show();  
}  
  
@Override  
public void onNothingSelected(AdapterView<?> parent) {  
  
}  
});  
  
}  
DataAccessManager a =new DataAccessManager();  
@Override  
public void onClick(View v){  
    switch(v.getId()){  
        case R.id.button18:  
            String breakfast=a.getbreakfast(itemvalue);  
            if(breakfast.equals("Not found")) {  
                Menu.setText("");  
                Menu.setText("BREAKFAST");  
            }  
            else  
            {  
                Menu.setText(breakfast);  
            }  
    }  
}
```

```
        }
        break;
    case R.id.button19:
        String lunch = a.getlunch(itemvalue);
        if(lunch.equals("Not found")) {
            Menu.setText("");
            Menu.setText("LUNCH");
        }
        else
        {
            Menu.setText(lunch);
        }
        break;
    case R.id.button20:
        String snacks = a.getsnacks(itemvalue);
        if(snacks.equals("Not found")) {
            Menu.setText("");
            Menu.setText("SNACKS");
        }
        else
        {
            Menu.setText(snacks);
        }
        break;
    case R.id.button22:
        String dinner= a.getdinner(itemvalue);
        if(dinner.equals("Not found")) {
```

```
        Menu.setText("");
        Menu.setText("DINNER");
    }
    else
    {
        Menu.setText(dinner);
    }
    break;
}
}
}
```

Student_mess_change

```
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.Spinner;
import android.widget.TextView;
import android.widget.Toast;

import java.util.ArrayList;
import java.util.List;
```

```
public class Student_mess_change extends AppCompatActivity {  
    String itemvalue;  
    Spinner dropdownmenu;  
    TextView t1;  
    Button b1;  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_student_mess_change);  
        getSupportActionBar().setTitle("Mess Change");  
        getSupportActionBar().setDisplayHomeAsUpEnabled(true);  
        t1=(TextView)(findViewById(R.id.textView4));  
        b1=(Button)(findViewById(R.id.button9));  
        dropdownmenu = (Spinner) findViewById(R.id.spinner);  
        List<String> list=new ArrayList<>();  
        list.add("PR 1 - VEG MESS (B-ANNEX)");  
        list.add("PR 2 - SPECIAL MESS (B-ANNEX)");  
        list.add("SKC 1 - NON VEG MESS (OPP TO F-BLOCK)");  
        list.add("SKC 2 - SPECIAL MESS (OPP TO F-BLOCK)");  
        list.add("RSM 1 - VEG MESS (G-BLOCK)");  
        list.add("RSM 2 - SPECIAL MESS (G-BLOCK)");  
        list.add("CRC 1 - VEG MESS (H-BLOCK)");  
        list.add("CRC 2 - NON VEG MESS (H-BLOCK)");  
        list.add("CRC 3 - SPECIAL MESS (J-BLOCK)");  
        list.add("CRC 4 - FOOD PARKK (J-BLOCK)");  
        list.add("RRC 1 - VEG MESS (L-BLOCK)");  
        list.add("RRC 2 - NON VEG MESS (L-BLOCK)");
```

```
list.add("RRC 3 - SPECIAL MESS (L-BLOCK)");
list.add("RRC 4 - SPECIAL MESS (D- ANNEX)");
list.add("RRC 5 - FOOD MALL (L-BLOCK)");
list.add("MHPL 1 - VEG MESS (F-BLOCK)");
list.add("MHPL 2 - SPECIAL MESS (F-BLOCK)");
list.add("AAC 1 - NON VEG MESS (F-BLOCK)");
list.add("AAC 2 - SPECIAL MESS (F-BLOCK)");
list.add("ZEN 1 - VEG MESS (K-BLOCK)");
list.add("ZEN 2 - NON VEG (K-BLOCK)");
list.add("ZEN 3 - SPECIAL (K-BLOCK)");
list.add("ZEN 4 - FOODCY (K-BLOCK)");
list.add("PR 3 - VEG MESS (P-BLOCK)");
list.add("PR 4 - SPECIAL MESS (P-BLOCK)");
list.add("DR 1 - NON VEG (P-BLOCK)");
list.add("DR 2 - SPECIAL MESS (P-BLOCK)");
list.add("PRD 1- VEG MESS (Q-block)");
list.add("PRD 2- Buddies & Bites (Q-block)");
```

```
ArrayAdapter<String> adapter = new ArrayAdapter<String>(this,
    android.R.layout.simple_spinner_item, list);
```

```
adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
dropdownmenu.setAdapter(adapter);
```

```
dropdownmenu.setOnItemSelectedListener(new
AdapterView.OnItemSelectedListener() {
    @Override
```

```
public void onItemSelected(AdapterView<?> parent, View view, int position, long id) {  
    itemvalue = parent.getItemAtPosition(position).toString();  
    Toast.makeText(Student_mess_change.this,"Selected " + itemvalue,  
    Toast.LENGTH_SHORT).show();  
    t1.setText(itemvalue);  
}  
  
@Override  
public void onNothingSelected(AdapterView<?> parent) {  
  
}  
});  
  
b1.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
        DatabaseAccess g= new DatabaseAccess();  
        String p=g.mess_name("19BCB0015");  
        g.changemess(itemvalue,p);  
        Toast.makeText(Student_mess_change.this, "MESS SUCCESSFULLY CHANGED",  
        Toast.LENGTH_SHORT).show();  
    }  
});  
}  
}
```

Student complaints

```
import android.support.v7.app.AppCompatActivity;  
import android.os.Bundle;  
import android.text.TextUtils;  
import android.view.View;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.RatingBar;  
import android.widget.Toast;  
  
import com.google.firebase.FirebaseApp;  
import com.google.firebase.database.DatabaseReference;  
import com.google.firebase.database.FirebaseDatabase;  
import com.hself.smilerating.SmileRating;  
  
public class Student_complaints extends AppCompatActivity {  
    SmileRating sr;  
    EditText et;  
    Button bu;  
    DatabaseReference databaseComplaints;  
    int i;  
  
    @Override  
    public void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        FirebaseApp.initializeApp(this);  
        setContentView(R.layout.activity_student_complaints);  
    }  
}
```

```
databaseComplaints = FirebaseDatabase.getInstance().getReference("complaints");
et = (EditText) findViewById(R.id.editText4);
getSupportActionBar().setTitle("Student Complaints");
getSupportActionBar().setDisplayHomeAsUpEnabled(true);
sr = (SmileRating) findViewById(R.id.smile_rating);
bu = (Button) findViewById(R.id.button12);
sr.setOnSmileySelectionListener(new SmileRating.OnSmileySelectionListener() {

    @Override
    public void onSmileySelected(int smiley, boolean reselected) {
        switch (smiley) {
            case SmileRating.BAD:
                Toast.makeText(Student_complaints.this,
                        "BAD",
                        Toast.LENGTH_SHORT).show();
                break;
            case SmileRating.GOOD:
                Toast.makeText(Student_complaints.this,
                        "GOOD",
                        Toast.LENGTH_SHORT).show();
                break;
            case SmileRating.GREAT:
                Toast.makeText(Student_complaints.this,
                        "GREAT",
                        Toast.LENGTH_SHORT).show();
                break;
            case SmileRating.OKAY:
                Toast.makeText(Student_complaints.this,
                        "OKAY",
                        Toast.LENGTH_SHORT).show();
                break;
            case SmileRating.TERRIBLE:
                Toast.makeText(Student_complaints.this,
                        "TERRIBLE",
                        Toast.LENGTH_SHORT).show();
                break;
        }
    }
})
```

```
        Toast.makeText(Student_complaints.this,
                      "TERRIBLE",
                      Toast.LENGTH_SHORT).show();
    break;
}
}
});
sr.setOnRatingSelectedListener(new SmileRating.OnRatingSelectedListener() {
    @Override
    public void onRatingSelected(int level, boolean reselected) {
        i = level;
        Toast.makeText(Student_complaints.this, "Selected rating " + level,
                      Toast.LENGTH_SHORT).show();
    }
});
bu.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        addDetails();
    }
});
}
}

public void addDetails()
{
    String complaint = et.getText().toString().trim();
    String id= databaseComplaints.push().getKey();
```

```
if(!TextUtils.isEmpty(complaint))
{
    Complaints comp = new Complaints(id,complaint,i);
    databaseComplaints.child(id).setValue(comp);
    Toast.makeText(this,"Feedback Added",Toast.LENGTH_LONG).show();
}
else
{
    Complaints comp = new Complaints(id,complaint,i);
    databaseComplaints.child(id).setValue(comp);
    Toast.makeText(this,"Feedback Added",Toast.LENGTH_LONG).show();
}
}
```

Manager_student_list

```
import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.support.v7.widget.LinearLayoutManager;
import android.support.v7.widget.RecyclerView;
import android.support.v7.widget.SearchView;
import android.text.Editable;
import android.text.TextWatcher;
import android.view.View;
import android.widget.ArrayAdapter;
import android.widget.EditText;
import android.widget.ListView;
```

```
import java.util.ArrayList;
import java.util.Arrays;
import java.util.List;

public class Manager_student_list extends AppCompatActivity {
    RecyclerView;
    ProductAdapter adapter;
    List<Product> productList;
    int kd;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_manager_student_list);
        getSupportActionBar().setTitle("Registered Student List");
        getSupportActionBar().setDisplayHomeAsUpEnabled(true);

        productList = new ArrayList<>();

        recyclerView= (RecyclerView) findViewById(R.id.recyclerView);
        recyclerView.setHasFixedSize(true);

        EditText = findViewById(R.id.edittext);
        editText.addTextChangedListener(new TextWatcher() {
            @Override
            public void beforeTextChanged(CharSequence s, int start, int count, int after) {
```

```
}

@Override
public void onTextChanged(CharSequence s, int start, int before, int count) {

}

@Override
public void afterTextChanged(Editable s) {

    filter(s.toString());
}

recyclerView.setLayoutManager(new LinearLayoutManager(this));
productList.add(
    new Product(
        1,
        "Lakshya Vaswani",
        "20BCB0132",
        "F-213",
        "VEG",
        R.drawable.student123));
recyclerView.setLayoutManager(new LinearLayoutManager(this));
productList.add(
    new Product(
        1,
```

```
        "Sri harsha",
        "20BCB0129",
        "B-232",
        "SPECIAL",
        R.drawable.student123));

recyclerView.setLayoutManager(new LinearLayoutManager(this));
productList.add(
    new Product(
        1,
        "Subham Jaiswal",
        "20BCB0124",
        "H-338",
        "VEG",
        R.drawable.student123));

recyclerView.setLayoutManager(new LinearLayoutManager(this));
productList.add(
    new Product(
        1,
        "Saguturu Kavya",
        "20BCB0142",
        "L-351",
        "PAID",
        R.drawable.student123));

recyclerView.setLayoutManager(new LinearLayoutManager(this));
productList.add(
    new Product(
        1,
```

```
        "M. Ritika",
        "20BCB0134",
        "D-521",
        "SPECIAL",
        R.drawable.student123));

recyclerView.setLayoutManager(new LinearLayoutManager(this));
productList.add(
    new Product(
        1,
        "HARSH GULATI",
        "20BCT0210",
        "F-452",
        "NON VEG",
        R.drawable.student123));

recyclerView.setLayoutManager(new LinearLayoutManager(this));
productList.add(
    new Product(
        1,
        "ANISH KHANNA",
        "20BCI0108",
        "K-231",
        "PAID",
        R.drawable.student123));

recyclerView.setLayoutManager(new LinearLayoutManager(this));
productList.add(
    new Product(
        1,
```

```
        "LAKSHIT KOTHARI",
        "20BDS0077",
        "P-213",
        "SPECIAL",
        R.drawable.student123));

recyclerView.setLayoutManager(new LinearLayoutManager(this));
productList.add(
    new Product(
        1,
        "ISHAN TIWARI",
        "20BDS0071",
        "P-513",
        "NON VEG",
        R.drawable.student123));

adapter = new ProductAdapter(this, productList);
recyclerView.setAdapter(adapter);

}

private void filter(String text)
{
    ArrayList<Product> filteredList = new ArrayList<>();

    for(Product item : productList)
    {
        if(item.getRegno().toLowerCase().contains(text.toLowerCase()))
        {
```

```
        filteredList.add(item);
    }
}
adapter.filterlist(filteredList);
}
}
```

Manager_complaints

```
import android.support.annotation.NonNull;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.widget.ArrayAdapter;
import android.widget.ListView;

import com.google.firebaseio.database.DataSnapshot;
import com.google.firebaseio.database.DatabaseError;
import com.google.firebaseio.database.DatabaseReference;
import com.google.firebaseio.database.FirebaseDatabase;
import com.google.firebaseio.database.ValueEventListener;

import java.util.ArrayList;
import java.util.List;

public class Manager_complaints extends AppCompatActivity {

    ListView;
    FirebaseDatabase database;
```

```
DatabaseReference ref;
ArrayList<String> list;
ArrayAdapter <String> adapter;
Complaints;
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_manager_complaints);

    complaints= new Complaints();
    getSupportActionBar().setTitle("Complaints List");
    getSupportActionBar().setDisplayHomeAsUpEnabled(true);
    listView = (ListView) findViewById(R.id.list);
    database = FirebaseDatabase.getInstance();
    ref= database.getReference("complaints");
    list = new ArrayList<>();
    adapter = new ArrayAdapter<String>(this,R.layout.list_layout,R.id.textView5,list);
    ref.addValueEventListener(new ValueEventListener() {
        @Override
        public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
            for(DataSnapshot ds: dataSnapshot.getChildren())
            {
                complaints=ds.getValue(Complaints.class);
                list.add(complaints.getFeedback().toString()+" \n " + "Rating : "+
complaints.getRating());
            }
            listView.setAdapter(adapter);
        }
    });
}
```

```

    }

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

}

});
}

```

Java Source files for remaining classes -

<https://drive.google.com/drive/folders/1HpGY9WAWhnWEkxRhhJ7GrlQctUlhwXG?usp=sharing>

Test Case Documentation

Unit Testing Table					
S.no	Test case Description	input	Output	Remarks	Status
1.	Welcome Page Animations Check	Nil	Animations Displayed	Attractive and Fluid Animations displayed on welcome page encouraging user to explore the website further	Pass
2.	Welcome Page Buttons Check	Nil	Buttons Displayed and Functional	Responsive buttons help user to Navigate through the website successfully	Pass

3.	Displayed Content Responsive Check	Nil	Content is Legible	The content displayed on the website is Attractive across screen dimensions	Pass
4.	Login Page for new user Check	Email, Password	Error Message	New and unregistered users are not allowed to proceed further before signing up. The user is given a link to go to the Sign Up Page	Pass
5.	User Registration Check on Login	Email, Password	Dashboard displayed or error message displayed	If conditions including that the email is not already registered, and password does not match with the database, the dashboard is displayed else an error message is displayed	Pass
6.	User registration when Signing Up	Name, Email, Password, Confirm Password	Error message displayed	When a user registers with an email account that does not pass the regular expression conditions, an invalid error message is displayed	Pass

7.	User registration check on Signing Up	Name, email, Password, confirm password	User registered successfully	Conditions to be satisfied. Email must not be already registered, password and confirm password should match, password must be at least 6 characters long.	Pass
7.	Redirection to Sign In Page Check	Submit Button Click	Sign In page displayed	User is successfully redirected to the sign in page when they register and sign up successfully	Pass
8.	Dashboard Display Test on Logging in	Nil	Dashboard Displayed	Welcome Message along with attractive images and option to record speech displayed on logging in. Options to record speech, View profile, view recorded speeches or logout visible on Navbar	Pass
9.	Speech Title Addition Check	Speech Title	Speech Title Displayed	The added speech title can be seen being added on the UI successfully	Pass
10.	Recording Speech Check for Simple English words containing commonly used words	Click on 'Start Recording your Speech Button'	Microphone turned on for Audio Input	The Microphone is successfully able to capture and display the entered speech in text form in the case of an input containing commonly used English words	Pass
11.	Recording Speech Check on a long term basis with Unique and non- English words	Click on 'Start Recording your Speech Button'	Microphone turned on for Audio Input	The audio input has inconsistent patterns in successfully capturing the user speech, especially in the case of non-English words or unique accents	Fail

12.	Display message check when user opens Saves Speeches without Recording any	Nil	The message "You do not have any Speeches! Save using Record tab" is displayed	An appropriate error message is displayed when user tries to access saved speeches without recording any	Pass
13.	Saves speeches visibility check	Click on Records tab	Saved Speeches displayed	The saved speeches are visible in attractive boxes	Pass
14.	Convert to Sign Button Check	Click on Convert to Sign Button	Sign Language Output Displayed	The output of all the letters in Sign Languages is visible clearly with letters right below with each sign for clarity. This page would help the user for direct communication with deaf people	Pass
15.	Navbar Buttons Check	Clicking on any button	Appropriate page displayed	The buttons are functioning successfully and are redirecting the user to the desired pages	Pass
16.	Edit this Speech Option Check	Click on Edit this Speech Button	Intended Page displayed	A creative and user friendly interface is displayed that prompts the user to edit the speech as per their convenience, update the speech and save it	Pass
17.	Delete this Speech Button Check	Click on Delete this Speech Button	Speech deleted	The speech is successfully deleted from the profile when the delete this speech button is clicked	Pass
18.	Profile Page display check	Clicking Profile option	Page displayed	The profile page is successfully displayed when the option is clicked and various of its features are visible	Pass

19.	Total saved speeches and total visits display check	Nil	Successful Display	The total speeches and total visits are successfully displayed on every visit	Pass
20.	Name Updation Check	Name	Name Updated	The name can be updated by the user successfully when desired	Pass
21.	Profile Pic updation check	Profile pic	Profile pic updated	The profile pic on the dashboard is successfully updated on the display	Pass

Project Demonstration

Student's Dashboard –

The dashboard is divided into several sections:

- Student Login:** Shows a profile picture, "WELCOME!!! STUDENT VIT UNIVERSITY", and a "SUBMIT" button.
- My Mess:** Displays "Sri Harsha" and "CRC 1 - VEG MESS (H-BLOCK)" with a menu of items: Palak corn, Toor daal (without ginger garlic paste), Kachha bhindi, Rasam, Apple, Boondi (adood).
- Notify Me:** Options for Breakfast, Lunch, Snack, and Dinner with toggle switches.
- Other Mess:** Shows "PR 1 - VEG MESS (B-ANNEX)" with a list of items: Veg fried rice, Mix dal tadka, Veg Manchurian, Tandoori chicken, Tandoori paneer, Rice papad, Rasam.
- Student Complaints:** A message: "Quality was perfect today just the butter milk was less salty". Below it is a rating scale from Poor to Great, with "Good" highlighted.
- Polling:** A section titled "SELECT FOOD IN PLACE OF:" with a heading "Sweet". It lists options: Gulab Jamun (1), Barfi (0), Halwa (0), and None of these (0). A "SUBMIT" button is at the bottom.

Manager's Dashboard -

The Manager's Dashboard displays five screenshots of an Android application interface:

- Manager Login:** A purple-themed screen with a VIT logo. It shows fields for "Prmess" and a password, along with a "SUBMIT" button.
- Manager Dashboard:** A screen titled "MANAGER VIT UNIVERSITY" featuring a user profile icon. It includes links for "Registered Student's List", "New Registrations", "Complaints", "Create Polling", and "LOGOUT".
- Registered Student List:** A list of registered students with their details:
 - Lakshya Vaswani (20BCB0132, F-213, VEG)
 - Sri harsha (20BCB0129, B-232, SPECIAL)
 - Subham Jaiswal (20BCB0124, H-338, VEG)
 - Saguturu Kavya (20BCB0142, L-351, PAID)
 - M. Ritika (20BCB0134)
- New Student Details:** A screen for adding new student details. It shows two examples:
 - Garv Mittra (20BCE2122, P-213, SPECIAL)
 - VIVIAN JOSEPH (20BCE2309, P-513, NON VEG)
- Complaints List:** A list of complaints with ratings:
 - good food.
Rating : 5
 - what a food
Rating : 4
 - food is not at all good
Rating : 4
 - excellent
Rating : 5
 - Kheer was too sweet and Puri was too oily.
Rating : ?
- Create Poll:** A screen for creating a poll with options:

PUT THE OPTIONS BELOW

 - Sweet
 - Gulab Jamun
 - Barfi
 - Halwa
 - Rasgulla

END POLL **SUBMIT**

Student's Layout –

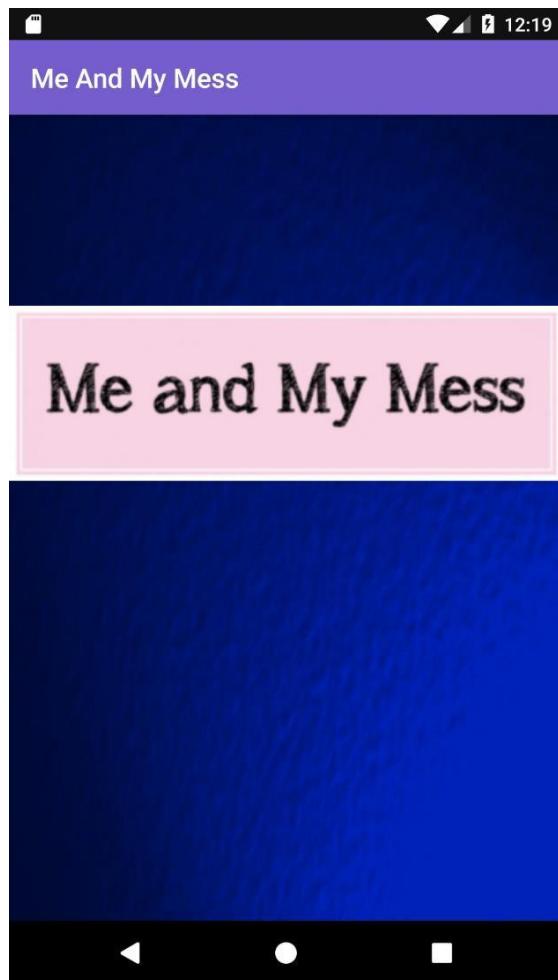


Figure 1 : Start Up Page of the app. This screen is just showing the logo of the app before the app launches.

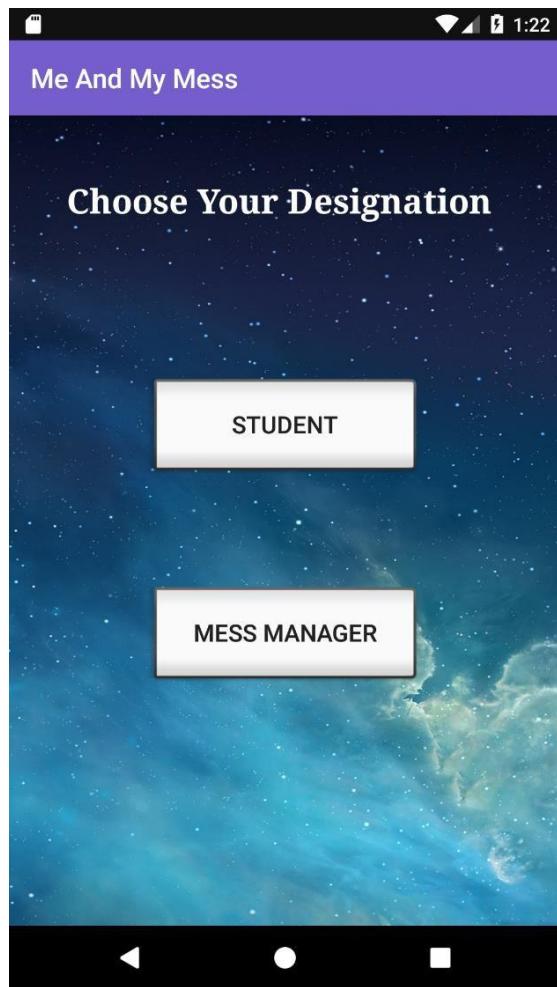


Figure 2 : Selection: Once the user starts the app and gets welcomed by the previous start screen, he/she will have the option to choose his/her designation weather he/she is a student of the university or It's the mess manager as the app aims to provide app usability to both of them

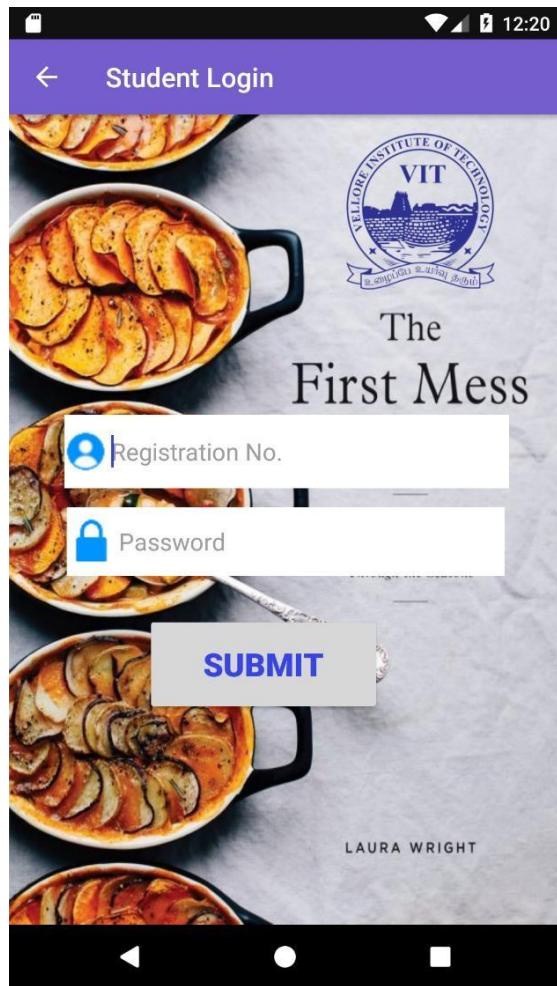


Figure 3 : Student Login: This page enables the students of VIT to log in to their mess profile. For this they've their respective registration number as their Username and password of their choice.

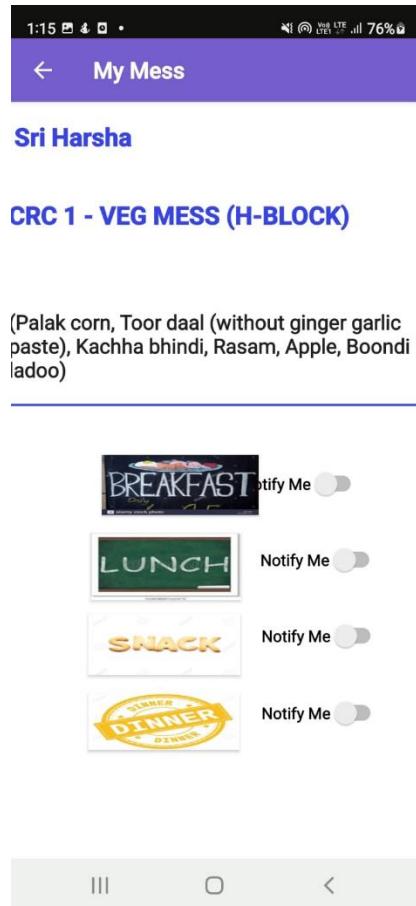


Figure 4 : Student Welcome screen: This page shows the user that's logged in and display the current mess he/she has enrolled. This page also provides the user the option to get notified about the different meals of the day

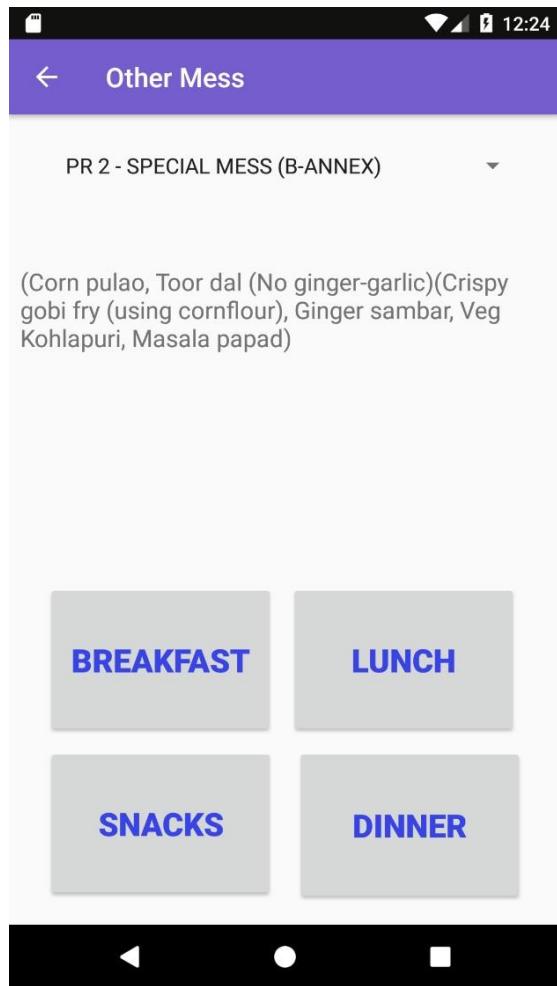


Figure 5 : Other mess Menu: This functionality will be accessible to the students in their respective login which can be accessed by scrolling the left portion option menu. Here the user can see the served meals menu of other mess for that particular day.



Figure 6 : Mess Manager Details: This option will be provided to the students in their login so that they can contact the manager through their email and contact info provided. For new enrolled students, they can see who's their mess manager is and can contact if necessary.

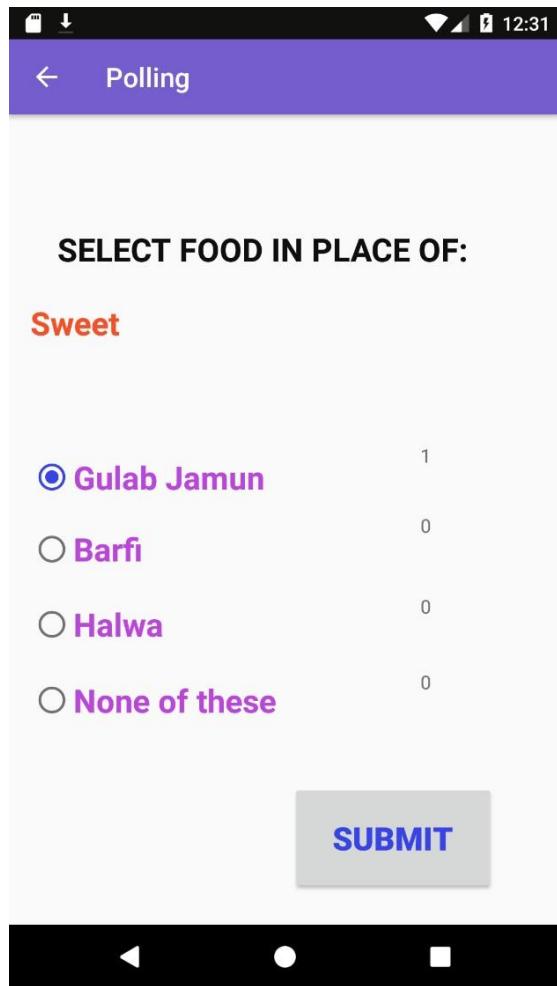


Figure 7 : Polling Option: This is another option that's provided for the students enrolled in certain mess that they can poll for certain food items served to them. Provides the user the opportunity for choosing food in place of the one already in the menu. The results will be stored and checked upon by the mess manager in his/her login.

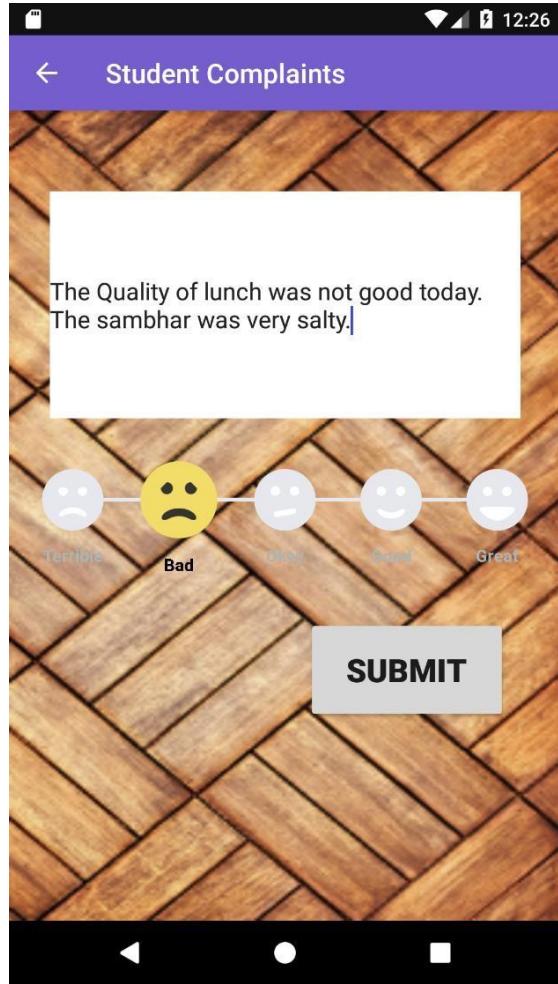


Figure 8 : Student's Complaints: In this page student can give feedback in word format about his/her opinion about the meals served and can suggest on a daily basis.

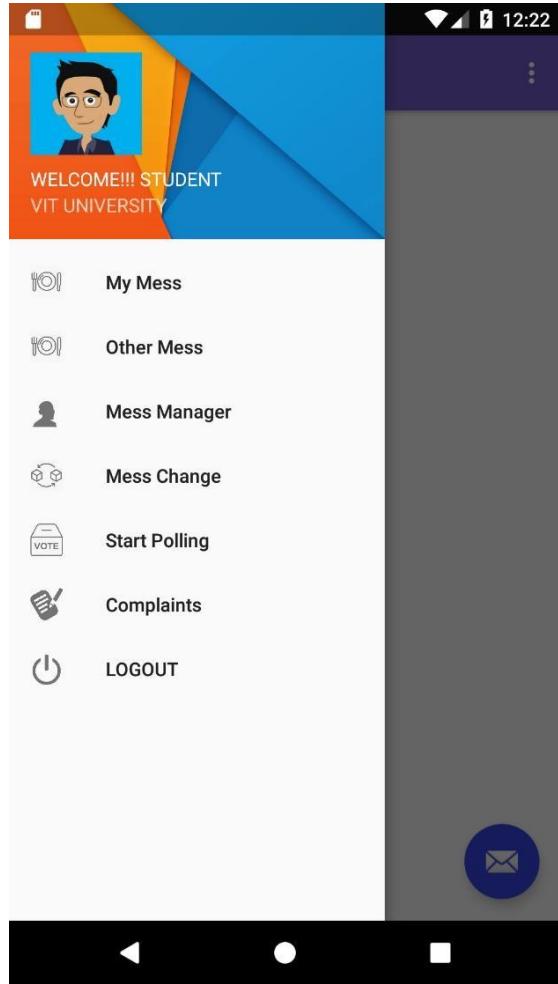


Figure 9 : Student Navigation Drawer: This is the snap of the options the student has access to after login into his/her account. All the sub menus mentioned here has been discussed above and the user can log out if wishes to.

Manager's Layout –



Figure 10 : Manager Login: This page is for the manager to login into his mess account. The manager has a unique username and a password of choice which give him the access to this if authentication is successful

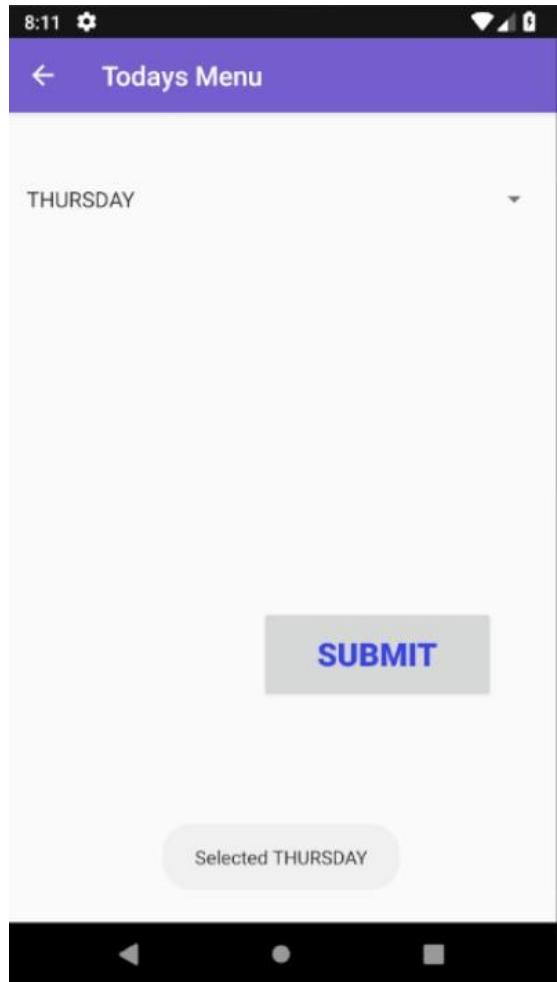


Figure 12 : Today's Menu: This comes under Manager section where he can select the day and submit the food menu for the respective day so that the meal served on the day is known to him as he's the one who gets the feedback and polling result from the information. So, it'll be him who'll make this choice.

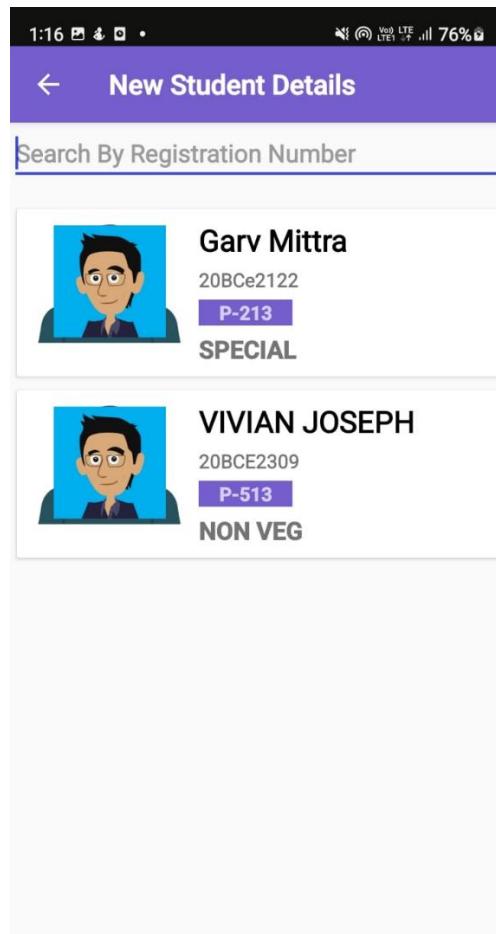


Figure 13 : New Enrolment: This comes under the manager section where manager can see the newly enrolled student list in case, he wants to keep check that the student is in that particular mess or not. The manager can also search by respective registration number makes him easier to find if two students having the same name.

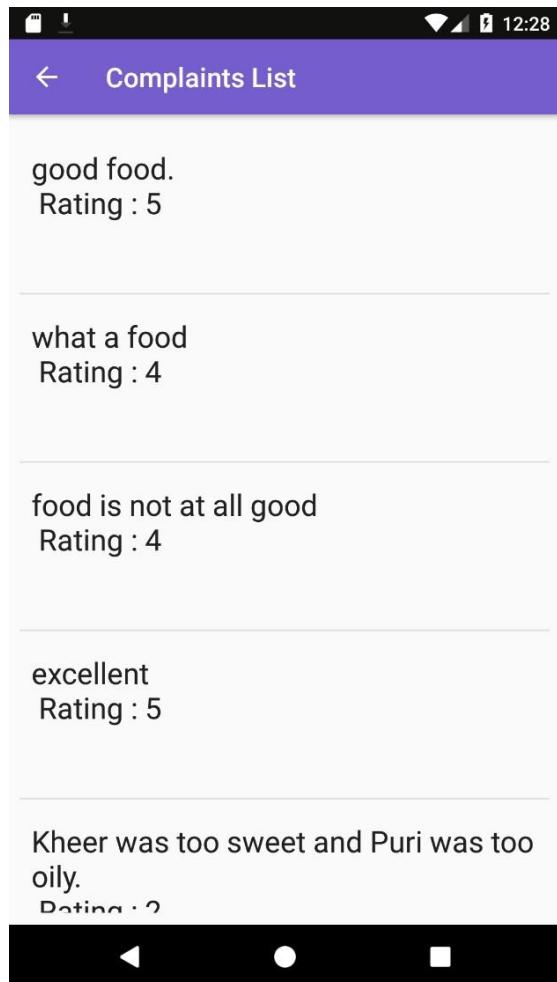


Figure 14 : Complaints List: The complaints registered by the students in their respective login is displayed in this section to the manager along with the ratings. Any feedback either good or bad will be shown so that the manager can take necessary action from his/her side if viable to do so.

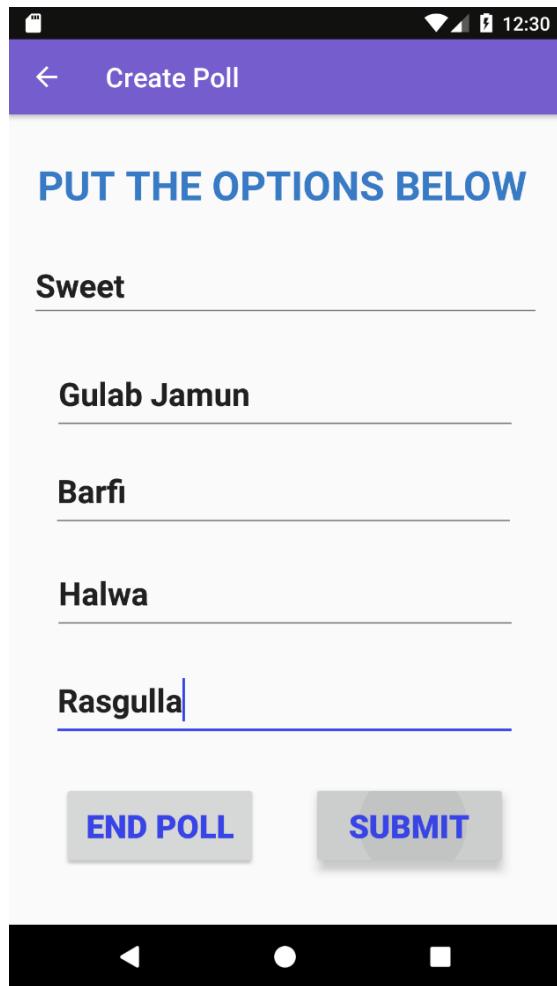


Figure 15 : Polling: The polling results will be shown here. The food to be changed and recommended replacements will be shown to the manager over here. It's up to him/her weather to make some amendments in the existing food items as suggested or to go on with the same. The End poll is done by the manager which will stop the poll from the student and the final result can be displayed. This poll can again be started and students will be informed about this or it'll be weekly process.

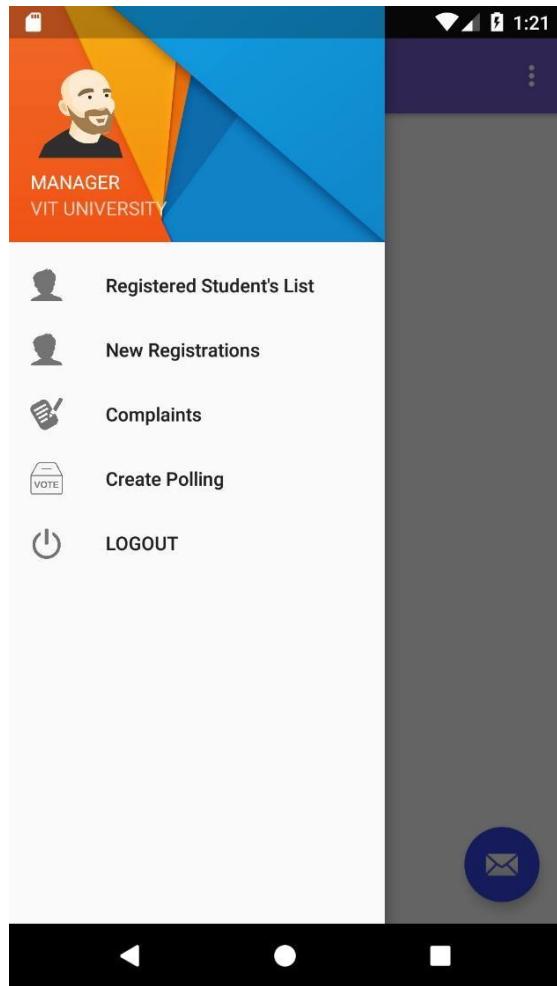


Figure 16 : Manager Navigation Drawer: This is how the UI for manager section for look like when he/she logged into their respective mess accounts. All the sub-menus here are discussed above. The message inbox in the blue box will be for any queries or suggestion regarding the app to the developers.

CONCLUSION:

We have looked into all the problems that we face regarding mess and this application Mess Management System resolves all the problem that a student faces with the mess like having issues with mess food, suggestions to be provided, complaints about the mess food and talking to the manager. This app will be a one stop solution for any issues regarding the mess. We have tried to create a user-friendly interface which will be easily accessible and act as a mediator between mess manager and students. For the future perspective of our app, we have tried to incorporate food delivery system to student rooms, for those who are ill or cannot make it to mess.

