Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

MESS MANAGEMENT SYSTEM

Team Members:

ANN MARRY JOHN – 20BCT0237
MYLIE MUDALIYAR – 20BCE2661
SWETA CHANDRASEKHAR – 20BCE2625

Report submitted

for the

Final Project Review of

Course Code: CSE3001 – SOFTWARE ENGINEERING
Slot: D2
Faculty Name: PROF. PREETHA EVANGELINE D

ACKNOWLEDGEMENTS

We are deeply grateful to our course faculty, PROF. Preetha Evangeline D, who provided us with the material and gave valuable inputs, and without whose guidance and help we would never have been able to complete the project.

We are also grateful to the management at VIT and the dean of SCOPE for providing us with this valuable opportunity to carry out our studies here at VIT.

Students Name:

Ann Marry John (20BCT0237
Mylie Mudaliyar (20BCE2661)
Sweta Chandrasekhar (20BCE2625)

INDEX

1. INTRODUCTION

1.1 ABSTRACT

To make the mess selection procedure transparent, accessible and select the best mess according to the user's requirements. When it comes to mess selection there are various options to choose from and anyone can get confused with the overwhelming options available, for example different types of mess like North Indian, South Indian, veg, non-veg, special, continental, etc. Also, many times students want to give feedback on the food/service provided to them. So, we are going to design and develop a user-friendly, easily and accessible android application that will deal with all the complexities a student goes through and make this mess process a smooth ride.

1.2 PROBLEM STATEMENT

● For our problems, we have decided to work on something which would be simple, effective, accessible 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.

● Many a time, students are unable to choose the correct mess type according to their liking due to a lack of knowledge about each mess type or miss their mess timings due to busy schedules, or are not aware of that day's menu and have to physically go check or ask their friends.

● Mess managers don't have access to required student details or unable to take regular feedback from the students to know about the food and service provided.

● Also, they find difficulties in manually logging students' attendance.

1.3 LITERATURE SURVEY

According to an automated food ordering system, it was proposed that it would keep track of user orders smartly. Basically, they implemented a food ordering system for different types of restaurants in which users will make orders or make custom food with one click only. By means of an android application for Tablet PCs, this system was implemented. The front end was developed using JAVA, and Android and at the backend MySQL database was used. According to the Customer using a Smartphone is considered a basic assumption for the system. When the customer approaches 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, the order slip shall be printed for further order processing.

The solution provides an easy and convenient way to select a pre-order transaction from 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 that fetches all information from a centralized database.

The 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 one-time investment for gadgets. An application of integration of hotel management systems by web services technology is presented. The Ordering System Kitchen Order Ticket (KOT), Billing System, and Customer Relationship Management system (CRM) are held together by Digital Hotel Management. Adding or expanding hotel software systems in any size of hotel chain environment was possible with this solution. Research work aims to design and develop a wireless food ordering system in restaurants. Technical operations of Wireless Ordering System (WOS) including systems architecture, function, and 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 for 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.

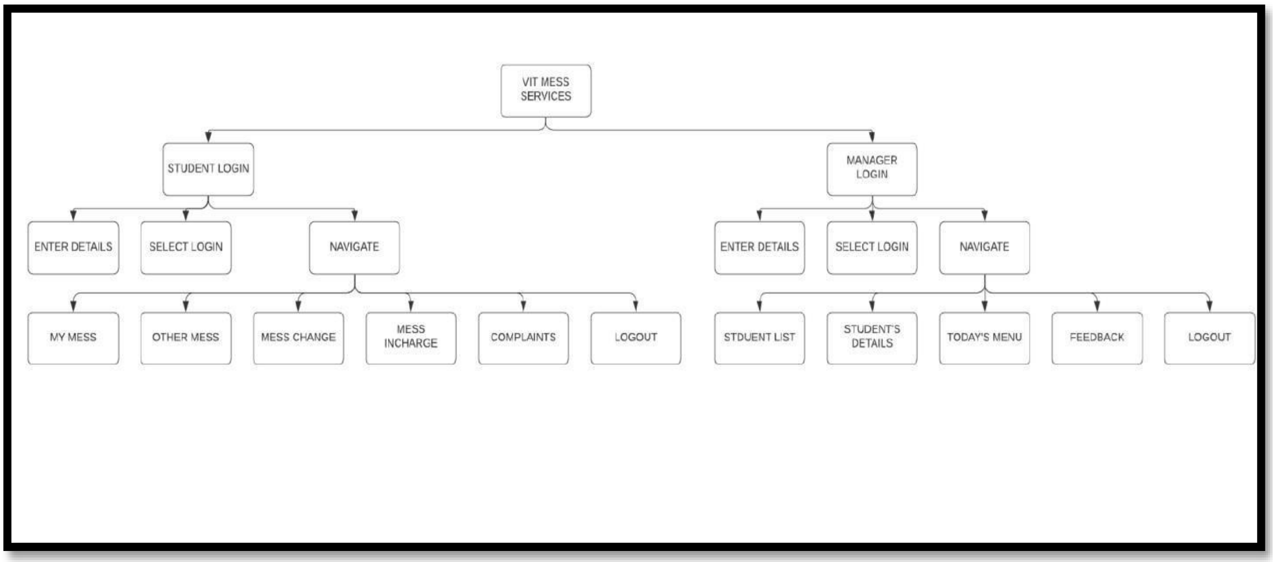
- It is good to detect errors in the same phase in which they are committed. It reduces the effort and time required to correct the errors.

- 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:



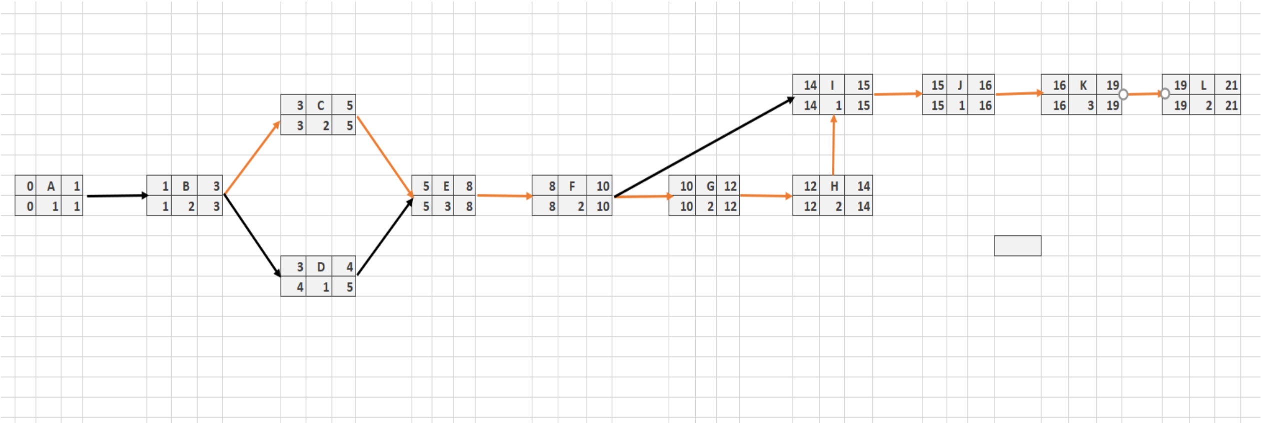
2.3 GANTT CHART:

ACTIVITIES		Dependencies	Duration(weeks)
Set up project acquisition team	A	NONE	1
Noting down the software requirements	B	A	2
Identifing the functional requirements	C	B	2
Identifing 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

ES	EF	LS	LF	1	3	5	8	10	12	14	15	16	17	18	19	20	21
0	1	0	1	■													
1	3	1	3		■												
3	5	3	5			■											
3	4	4	5			■											
5	8	5	8				■										
8	10	8	10					■									
10	12	10	12						■								
12	14	12	14							■							
14	15	14	15								■						
15	16	15	16									■					
16	19	16	19										■	■	■		
19	21	19	21													■	■

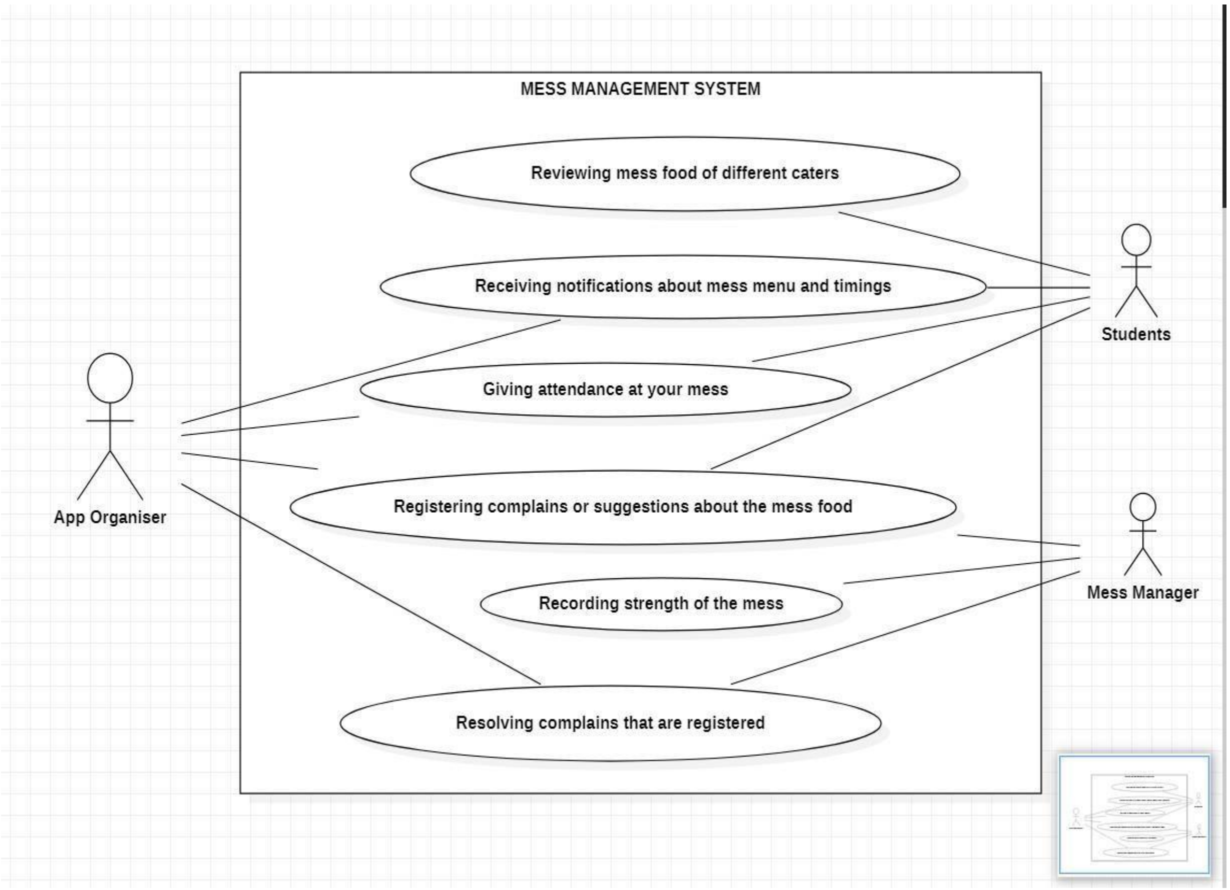
2.4 PERT CHART:

CRITICAL PATH AND MILESTONES :



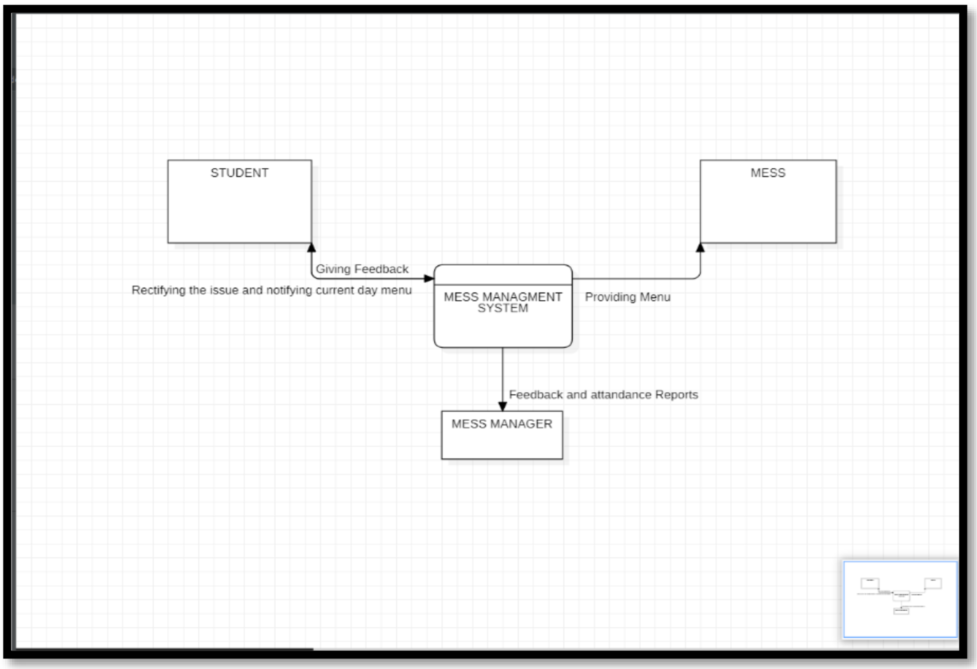
A→ B→C→E→F→G→H→I→J→K→L

2.5 USE CASE MODEL:

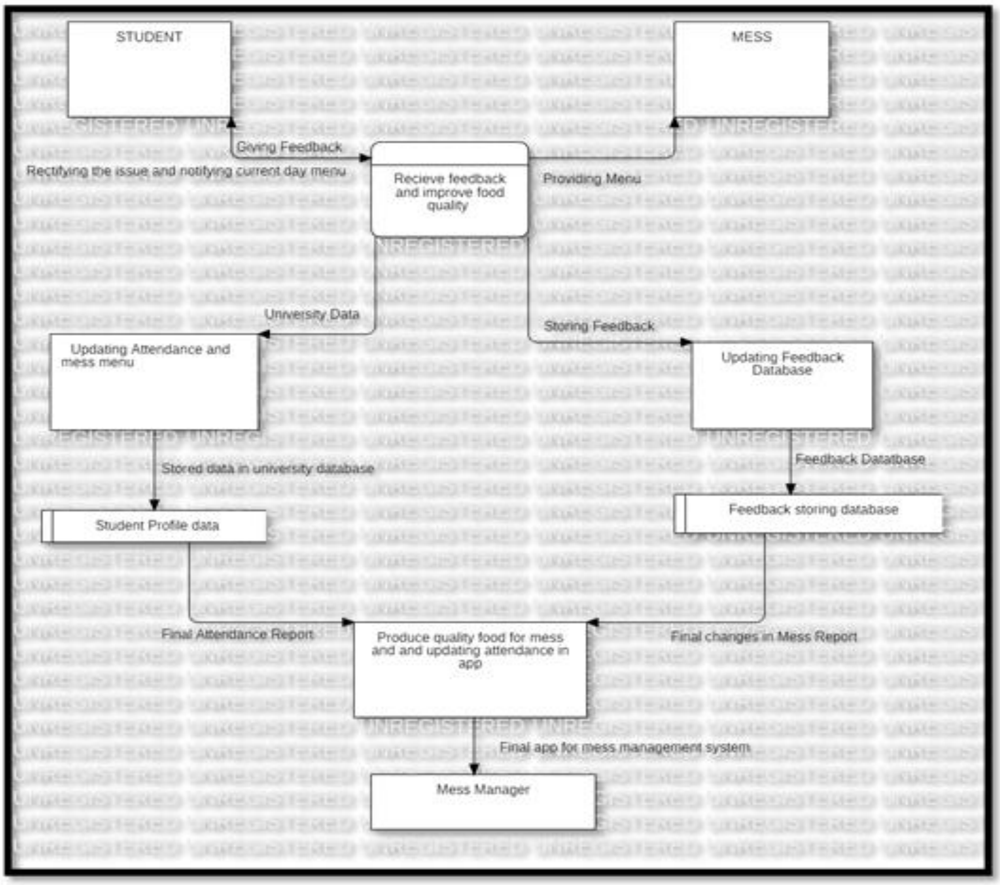

MESS MANAGEMENT SYSTEM
Reviewing mess food of different caters
Receiving notifications about mess menu and timings
Giving attendance at your mess
Registering complains or suggestions about the mess food
Recording strength of the mess
Resolving complains that are registered
App Organiser
Students
Mess Manager

2.6 DATA FLOW DIAGRAM:

DFD LEVEL 0:

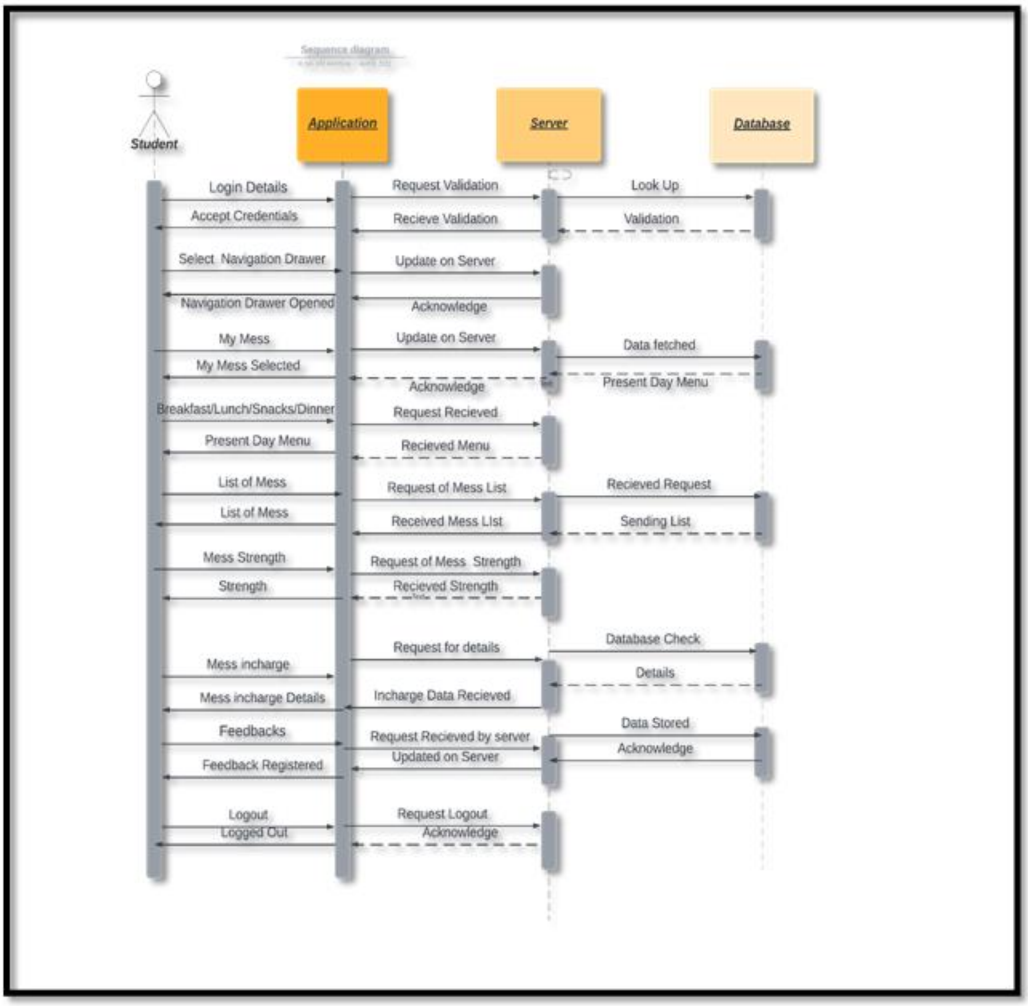

STUDENT
MESS
Giving Feedback
Rectifying the issue and notifying current day menu
MESS MANAGMENT SYSTEM
Providing Menu
Feedback and attandance Reports
MESS MANAGER

DFD LEVEL 1:

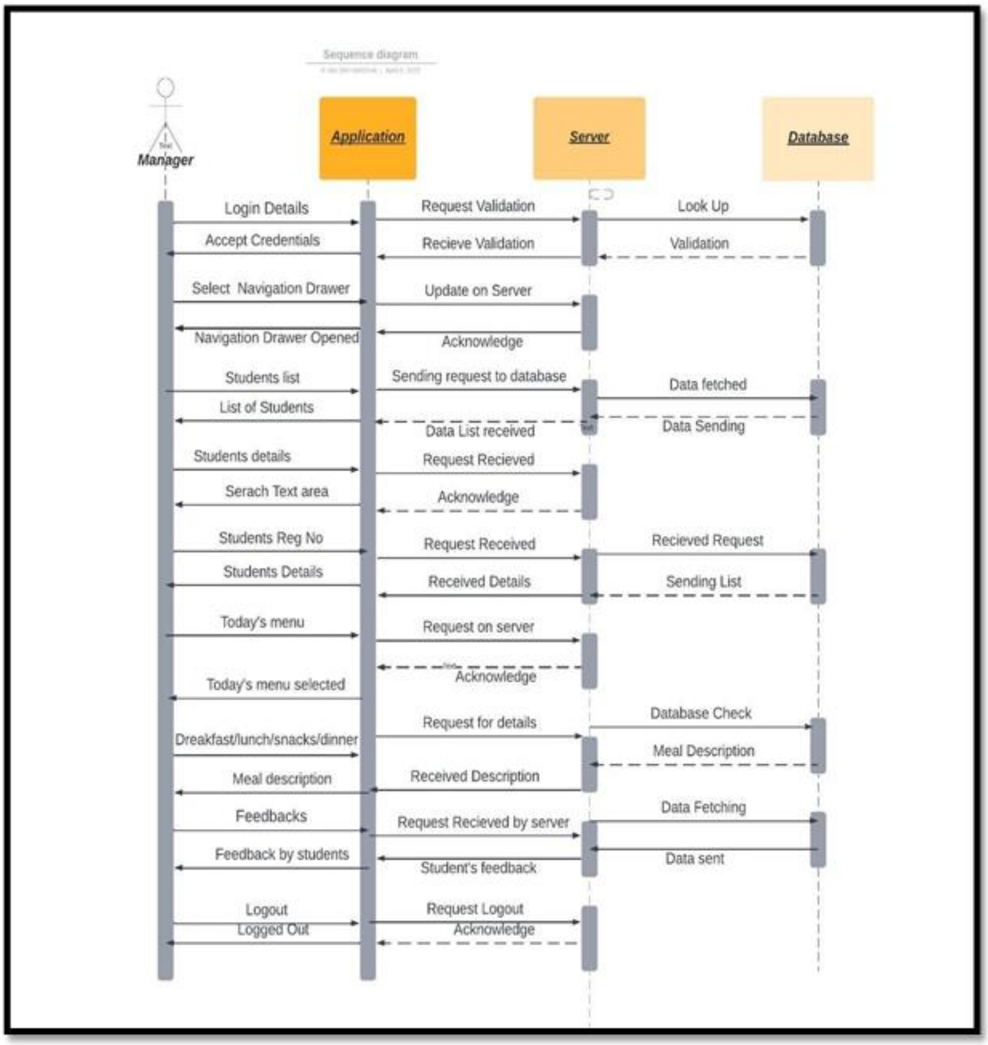

STUDENT
MESS
Giving Feedback
Rectifying the issue and notifying current day menu
Recieve feedback and improve food quality
Providing Menu
University Data
Storing Feedback
Updating Attendance and mess menu
Updating Feedback Database
Stored data in university database
Feedback Datatbase
Student Profile data
Feedback storing database
Final Attendance Report
Produce quality food for mess and and updating attendance in app
Final changes in Mess Report
Final app for mess management system
Mess Manager

2.4 SEQUENCE DIAGRAM:

SEQUENCE DIAGRAM OF STUDENT

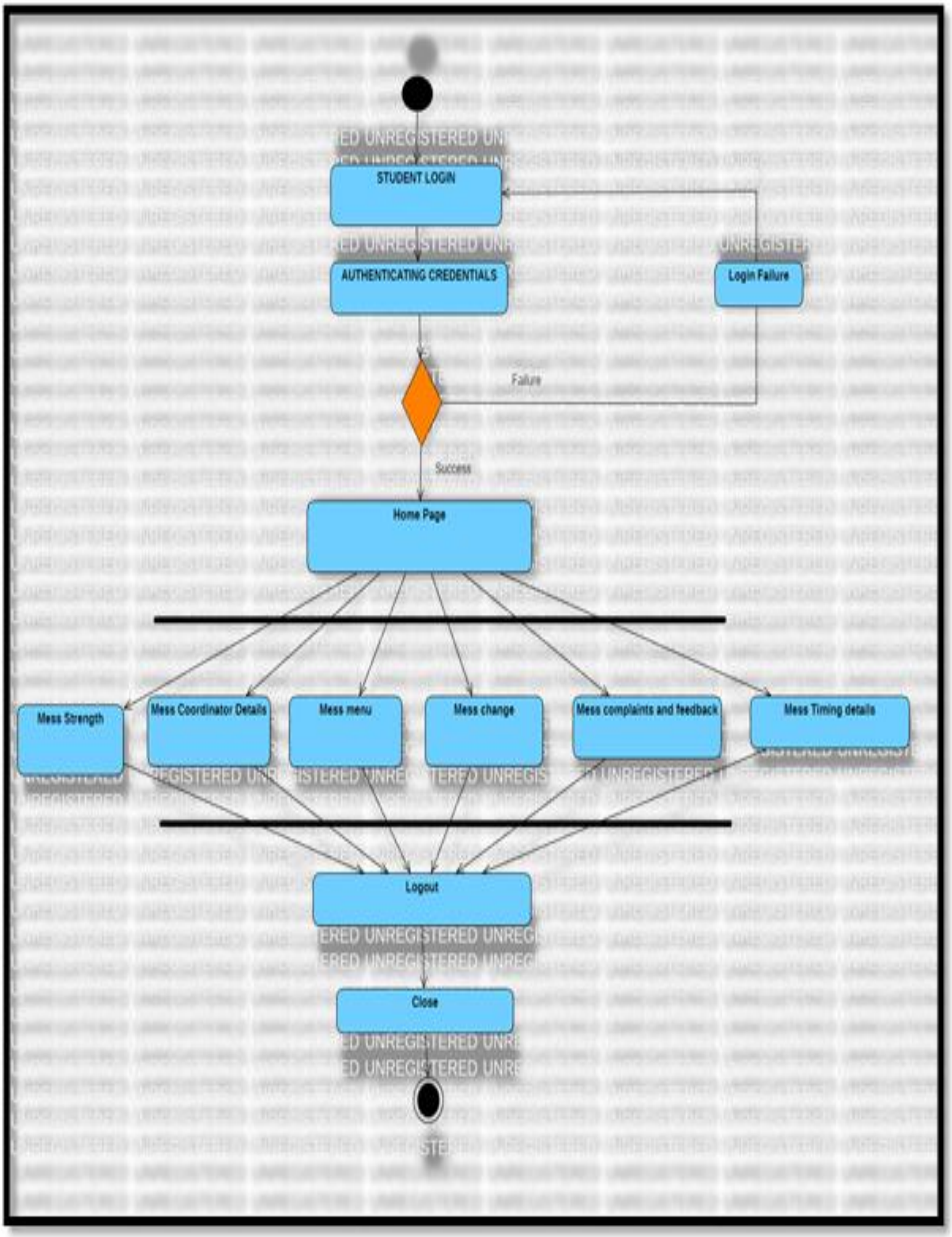


SEQUENCE DIAGRAM OF MANAGER

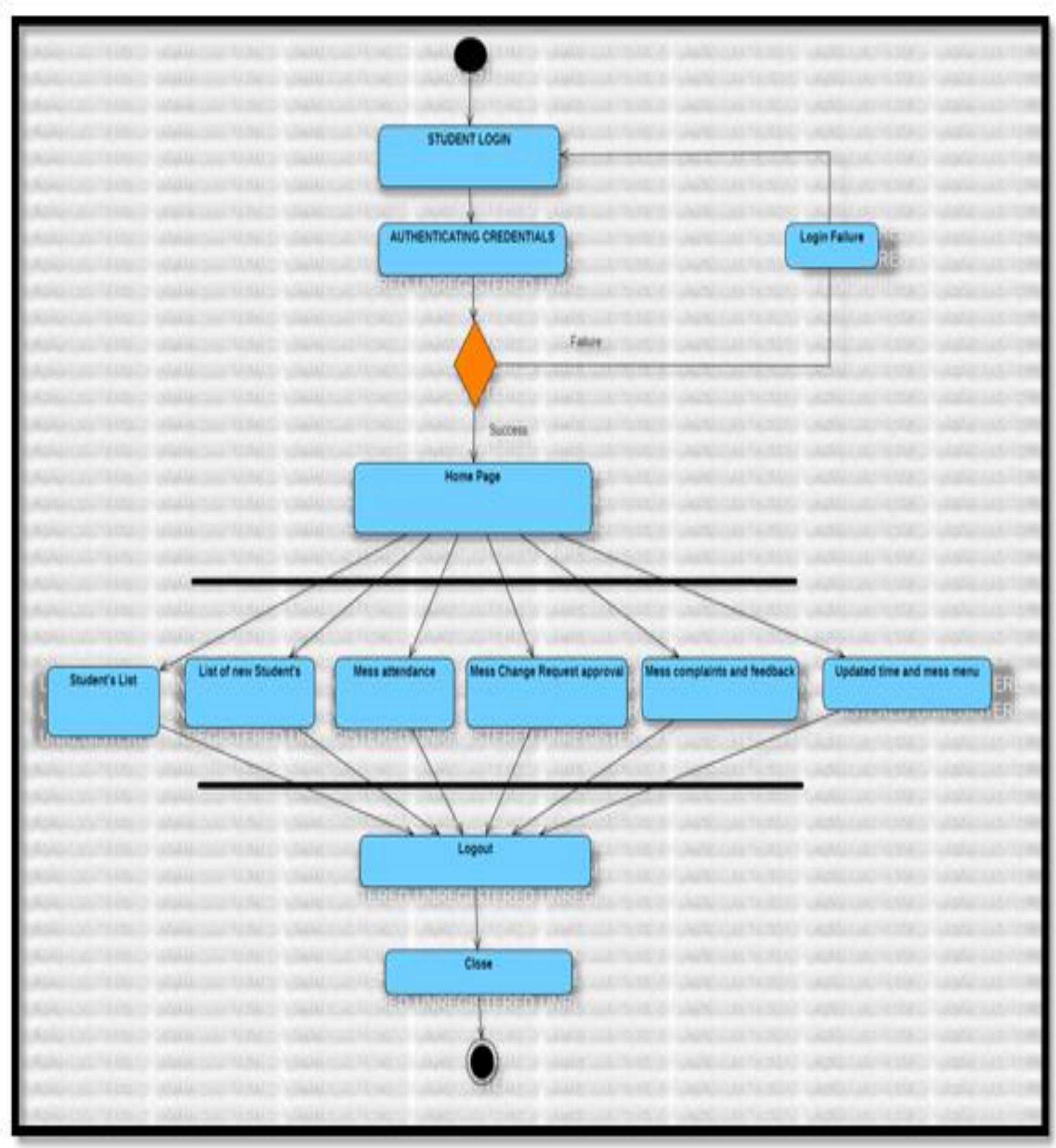


2.5 ACTIVITY DIAGRAM:

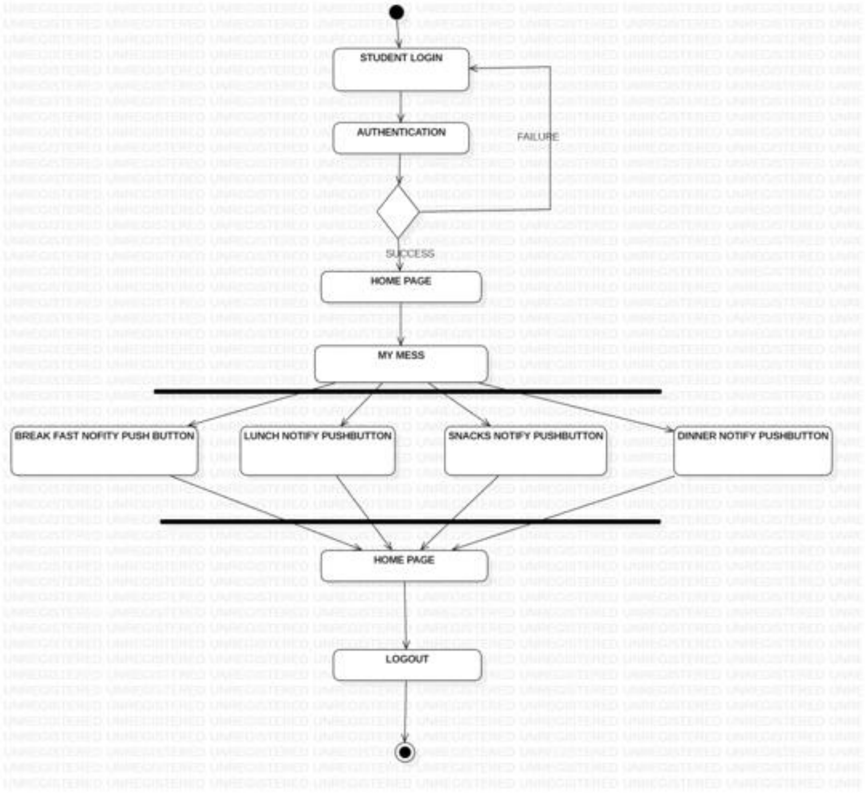
ACTIVITY DIAGRAM OF STUDENT



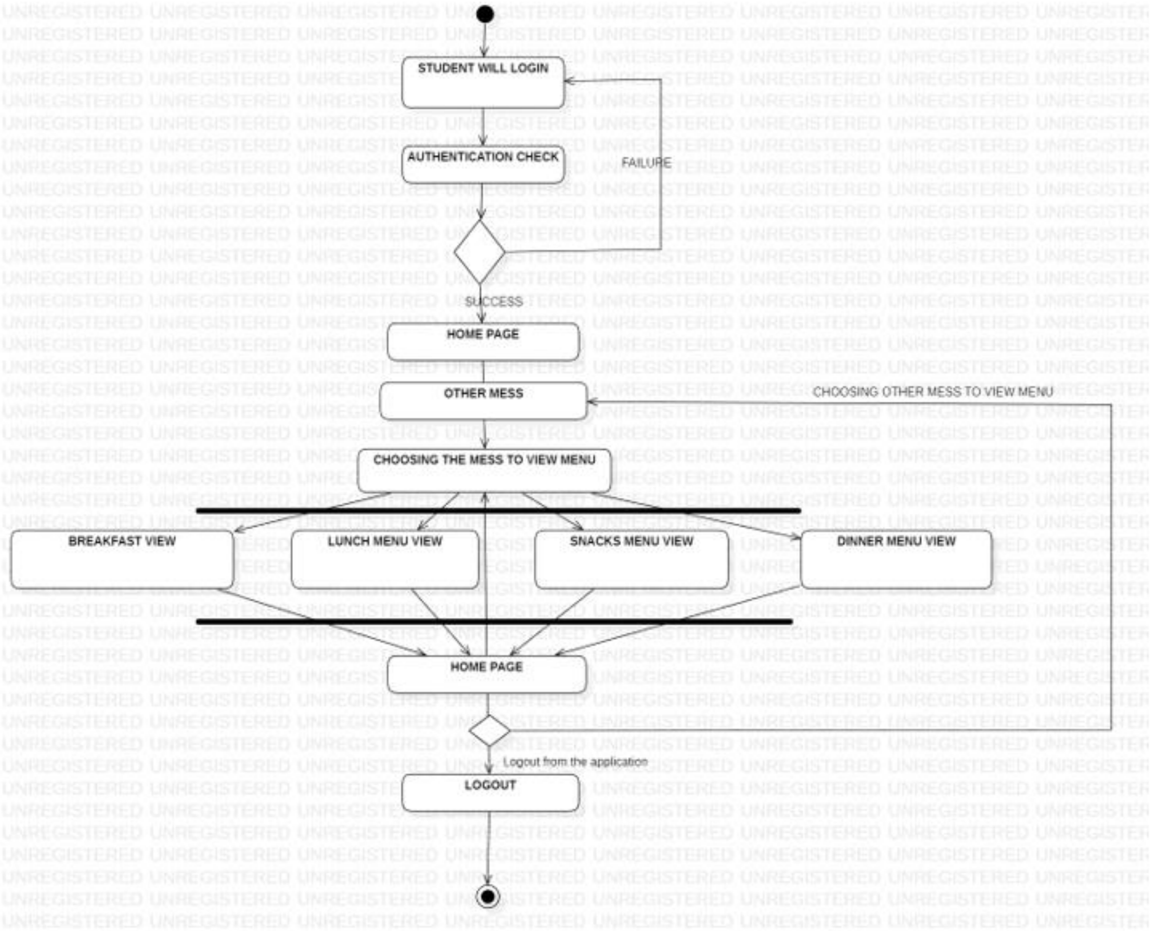
ACTIVITY DIAGRAM OF MANAGER



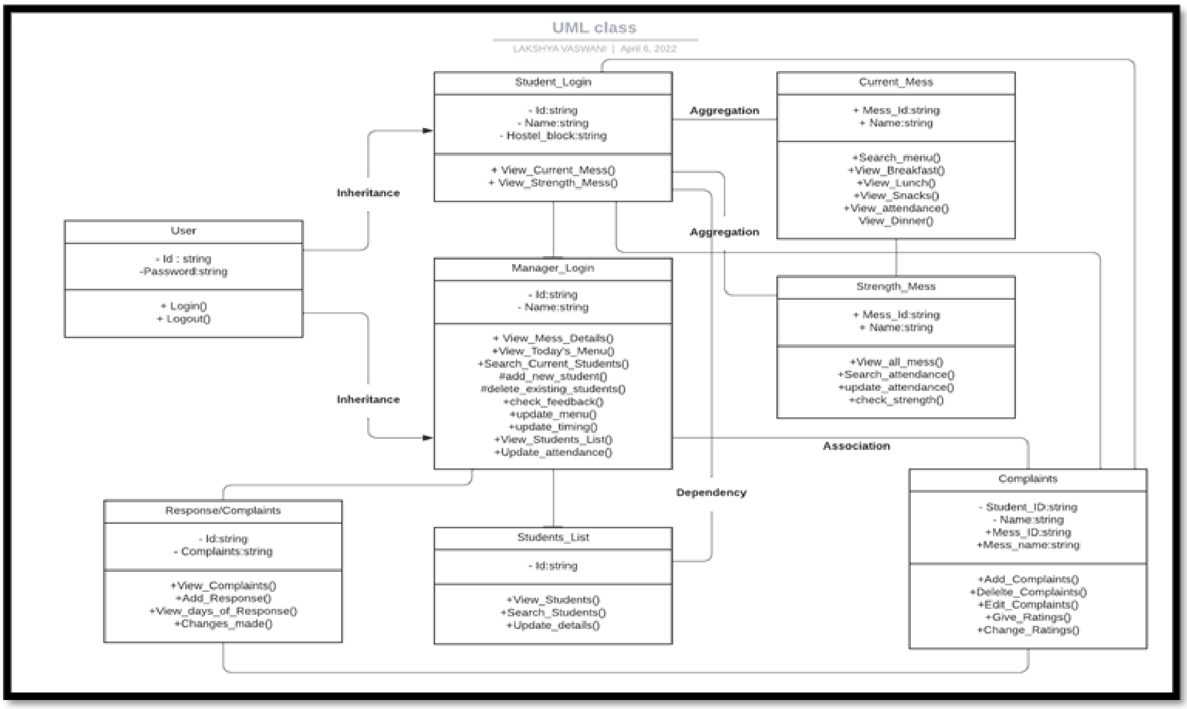
STUDENT MY MESS



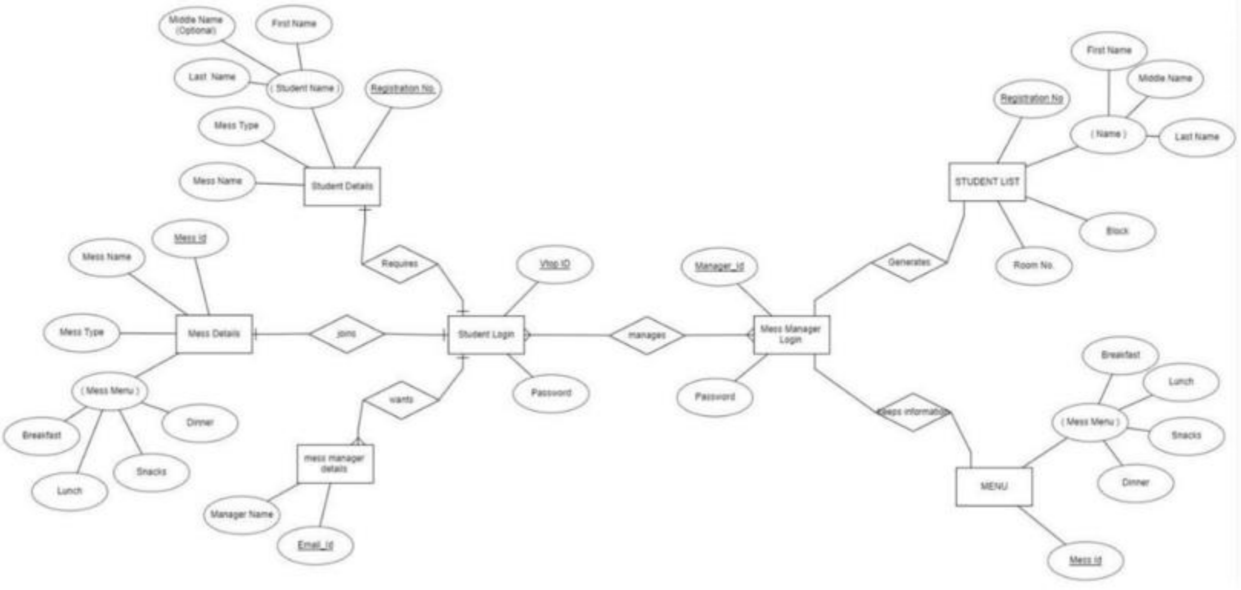
STUDENT OTHER MESS



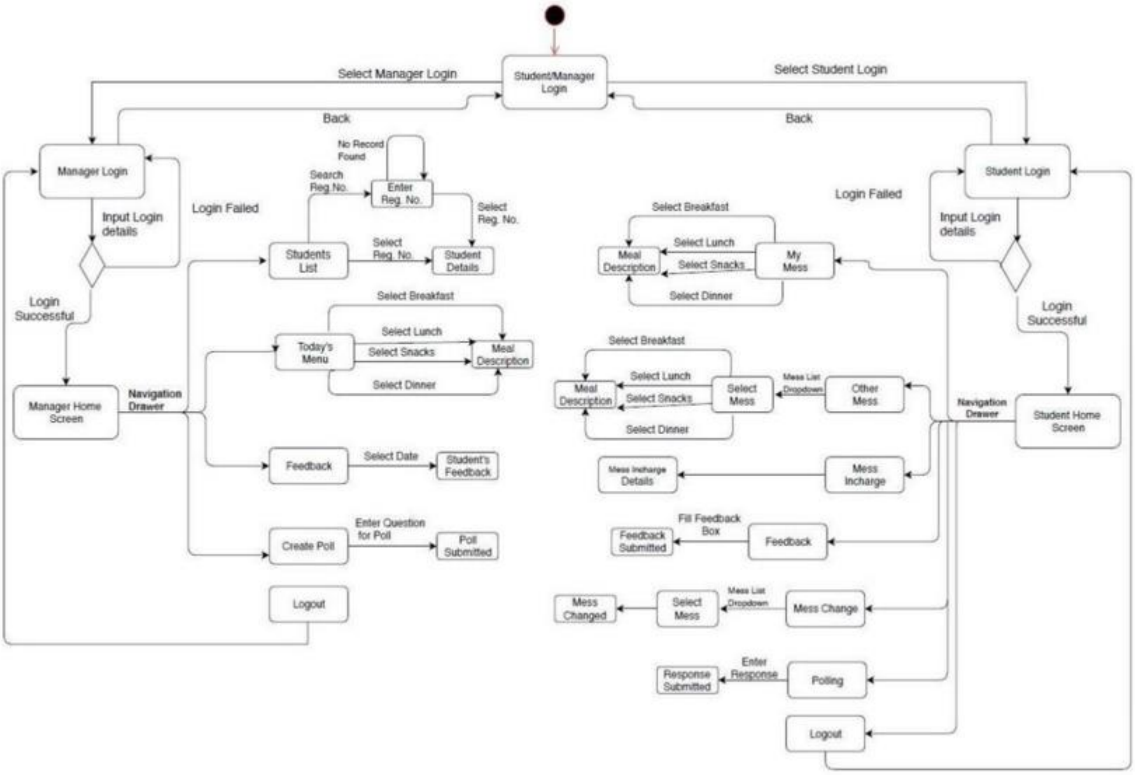
2.6 CLASS DIAGRAM:



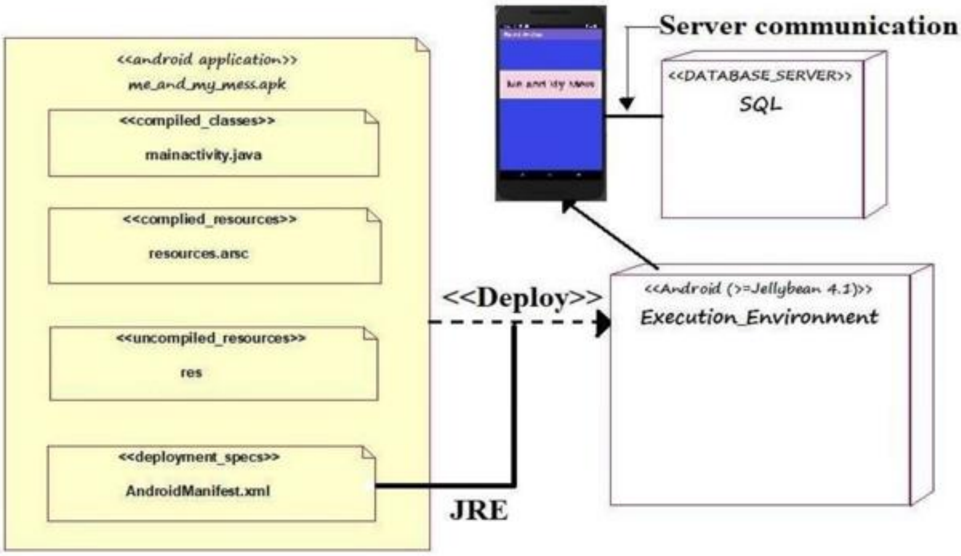
2.7 ER MODEL:



2.8 STATE DIAGRAM:



Deployment Architecture



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 the quality of the food, mess timing, and 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 them of their current day mess menu without checking manually in their mess.

4. Students will be provided with the manager's 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 a contact)

5. The manager will have a record of each and every student in his mess. He/she will even have a record of the students entering and leaving the mess and hence the strength of the mess.

3.2 NON-FUNCTIONAL REQUIREMENTS:

ORGANIZATIONAL 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 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 with the secure and smooth working of the app to make it glitch-free for the consumer which is students and managers over here. The time taken for the average person to open on click is mentioned below: In accordance with the GOMS Keystroke Level Model,

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

4. Implementation Code

Manager login

```
import android.content.Intent; importandroid.support.v7.app.AppCompatActiviy;
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,Main2Activity.class); startActivity(intent1);
}
public void openDialog() {
Ex2Dialog e = new Ex2Dialog(); e.show(getSupportFragmentManager(),"Dialog");
}
private Boolean validate() {
Boolean result = false;
```

Student login

```
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 Notification ON", Toast.LENGTH_SHORT).show(); }
else{
Toast.makeText(this,"BreakFast Notification OFF", Toast.LENGTH_SHORT).show(); }
break;
case R.id.switch2:
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

id) {
public void onItemSelected(AdapterView<?> parent, View view, int position, long
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.hsalf.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;
}
}
});
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);
```

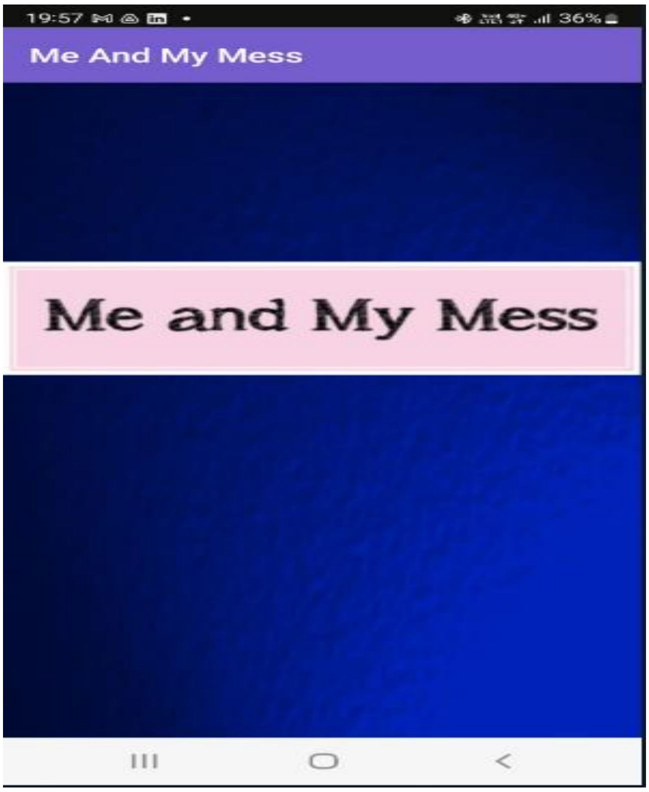
```
productList = new ArrayList<>();

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(
"ANN MARRY JOHN", "20BCT0237", "F-213", "VEG",

R.drawable.student123)); recyclerView.setLayoutManager(new LinearLayoutManager(this))
productList.add(
new Product(
"MYLIE MUDALIYAR", "20BCE2661", "B-232", "SPECIAL",
R.drawable.student123)); recyclerView.setLayoutManager(new LinearLayoutManager(this))
productList.add(
new Product(
"SWETA CHANDRASEKHAR", "20BCE2625", "H-338", "VEG",

R.drawable.student123)); recyclerView.setLayoutManager(new LinearLayoutManager(this))
productList.add(
new Product(
"ANN MARRY JOHN", "20BCT0237", "L-351", "PAID",

R.drawable.student123)); recyclerView.setLayoutManager(new LinearLayoutManager(this))
productList.add(
new Product(

"MYLIE MUDALIYAR", "20BCE2661",  "D-521", "SPECIAL",
R.drawable.student123)); recyclerView.setLayoutManager(new LinearLayoutManager(this))
productList.add(
new Product(
"SWETA CHANDRASEKHAR", "20BCE2625", "F-452", "NON VEG",

R.drawable.student123)); recyclerView.setLayoutManager(new LinearLayoutManager(this))
productList.add(
new Product(
"ANN MARRY JOHN", "20BCT0237", "K-231", "PAID",

R.drawable.student123)); recyclerView.setLayoutManager(new LinearLayoutManager(this))
productList.add(
new Product(

"MYLIE MUDALIYAR", "20BCE2661", "P-213", "SPECIAL",
R.drawable.student123));
recyclerView.setLayoutManager(new LinearLayoutManager(this)); productList.add(
new Product(
"SWETA CHANDRASEKHAR", "20BCE2625", "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

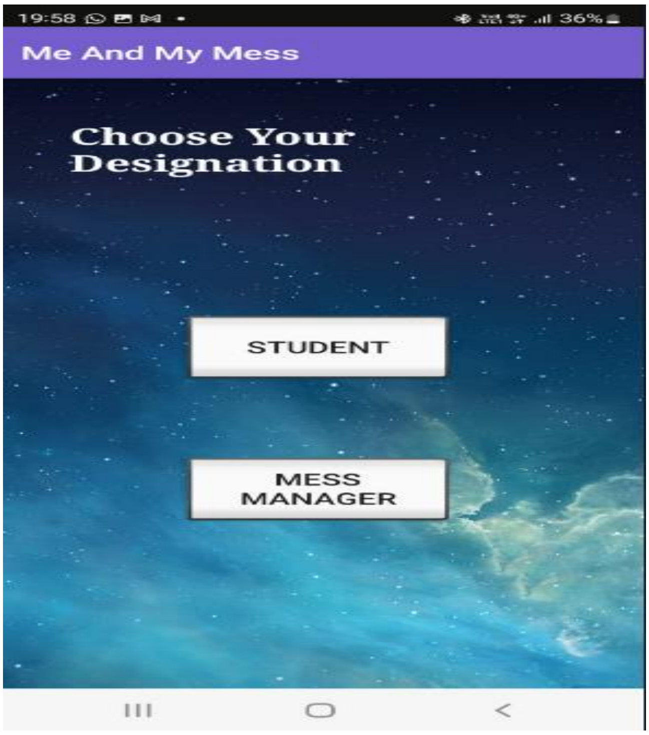
```
import android.support.annotation.NonNull;

import android.support.v7.app.AppCompatActivity;

import android.os.Bundle;
import android.widget.ArrayAdapter;
import android.widget.ListView;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;

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

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) {

}

});

}
```

DEMONSTRATION OF PROJECT

- **Welcome Page**

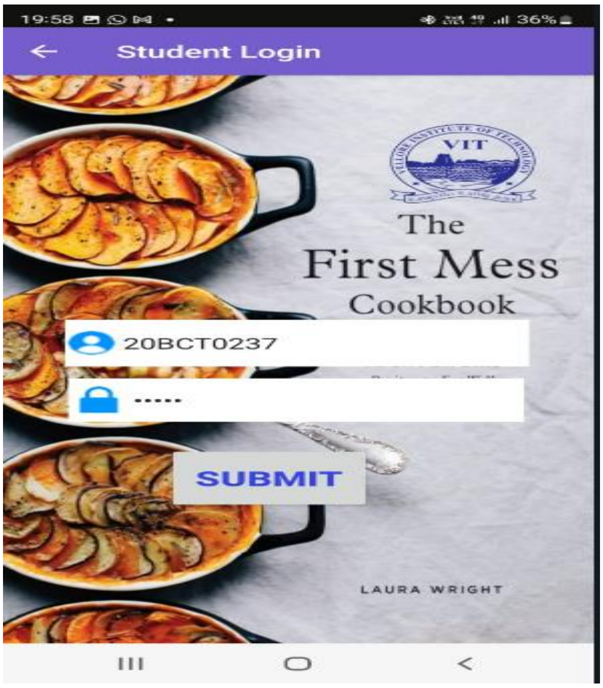


- **Home Page**



Student View

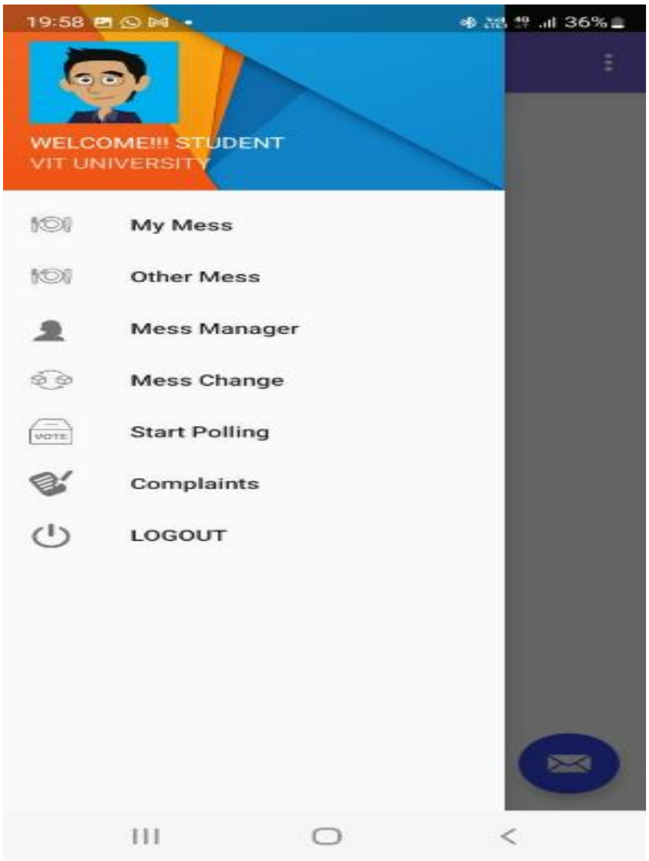
- **Student Log In Page**



Student Login

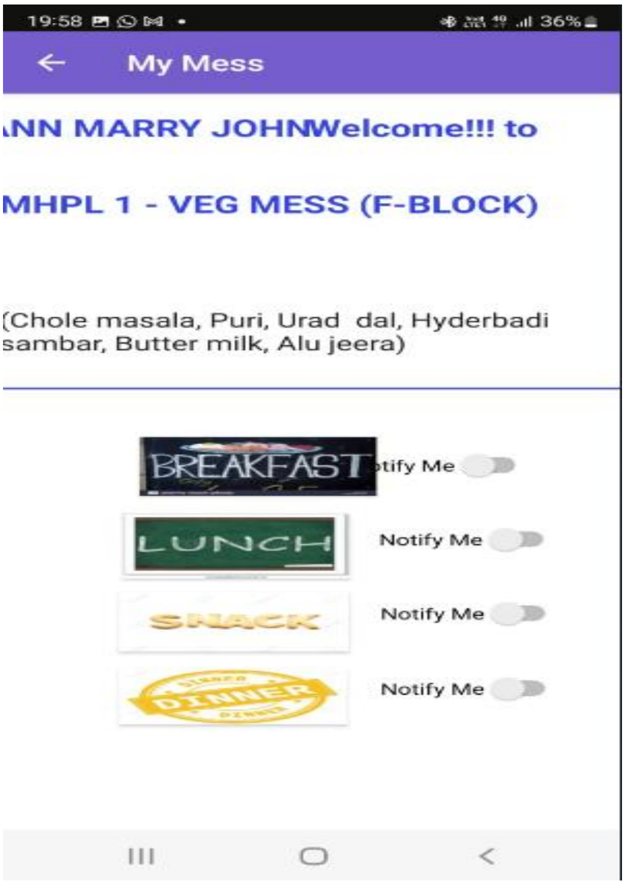
20BCT0237

- **Student Home Page**



WELCOME!!! STUDENT
VIT UNIVERSITY

- **Student Mess Page**



19:58 ... 36%

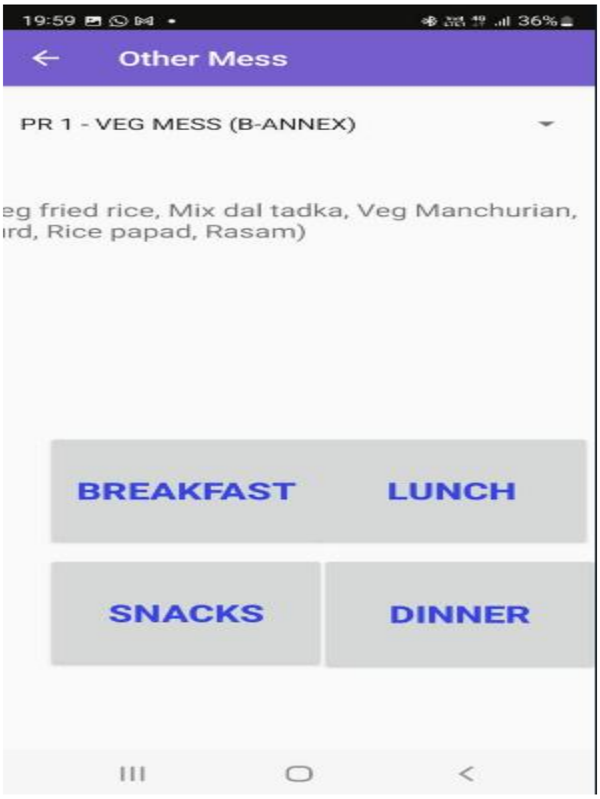
My Mess

NN MARRY JOHNWelcome!!! to

MHPL 1 - VEG MESS (F-BLOCK)

(Chole masala, Puri, Urad dal, Hyderbadi sambar, Butter milk, Alu jeera)

- **Student Other Mess Page**



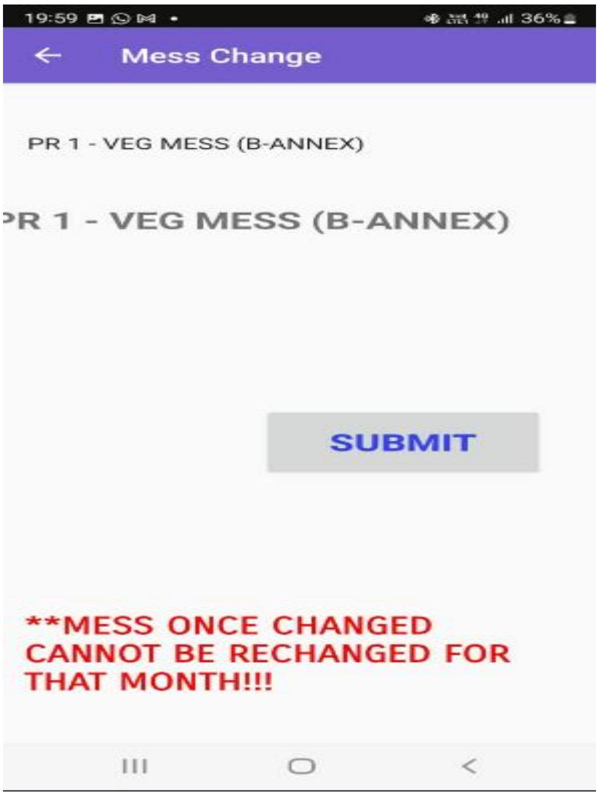
19:59 ... 36%

Other Mess

PR 1 - VEG MESS (B-ANNEX)

eg fried rice, Mix dal tadka, Veg Manchurian, ırd, Rice papad, Rasam)

- **Student Mess Change**


19:59

36%

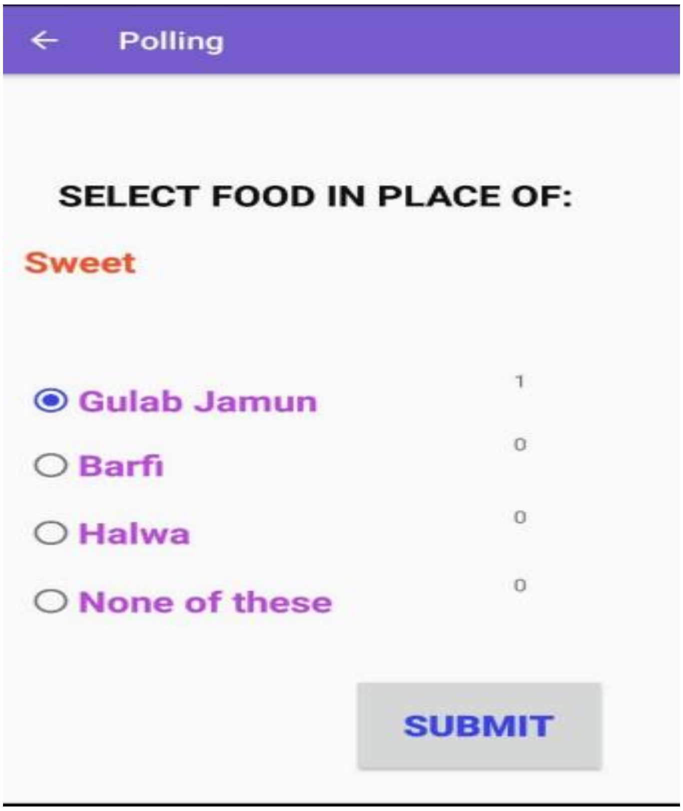
Mess Change

PR 1 - VEG MESS (B-ANNEX)

PR 1 - VEG MESS (B-ANNEX)

**MESS ONCE CHANGED CANNOT BE RECHANGED FOR THAT MONTH!!!

- **Polling Page**

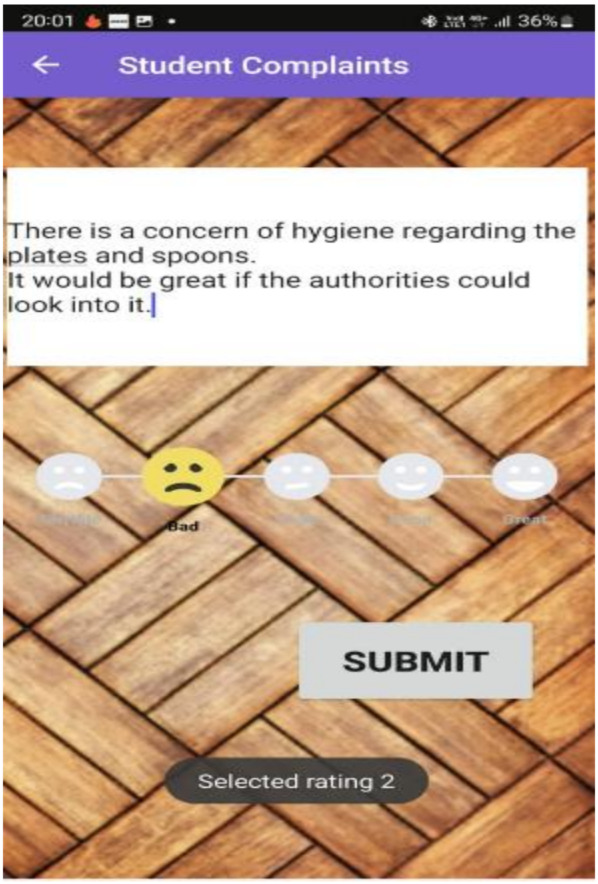

Polling

SELECT FOOD IN PLACE OF:

Sweet

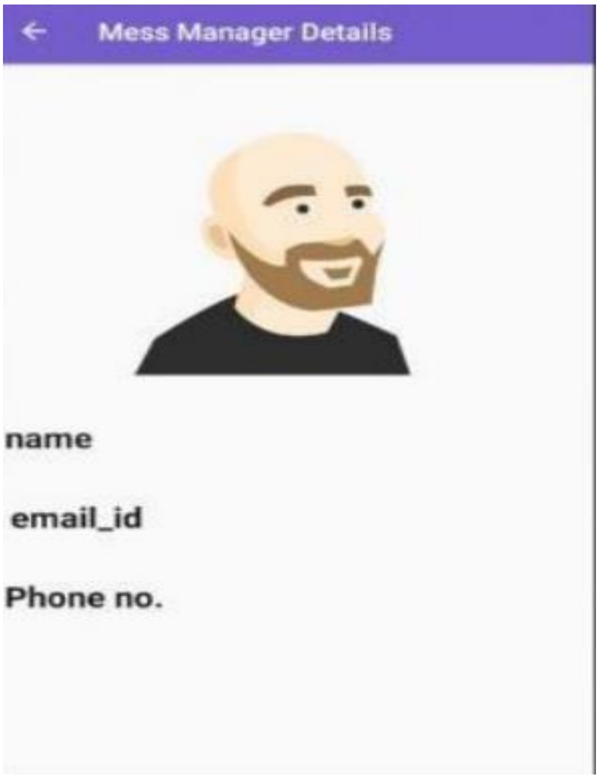
● Gulab Jamun 1
○ Barfi 0
○ Halwa 0
○ None of these 0

- **Student Complaints Page**



There is a concern of hygiene regarding the plates and spoons.
It would be great if the authorities could look into it.

- **Mess Manager Page**



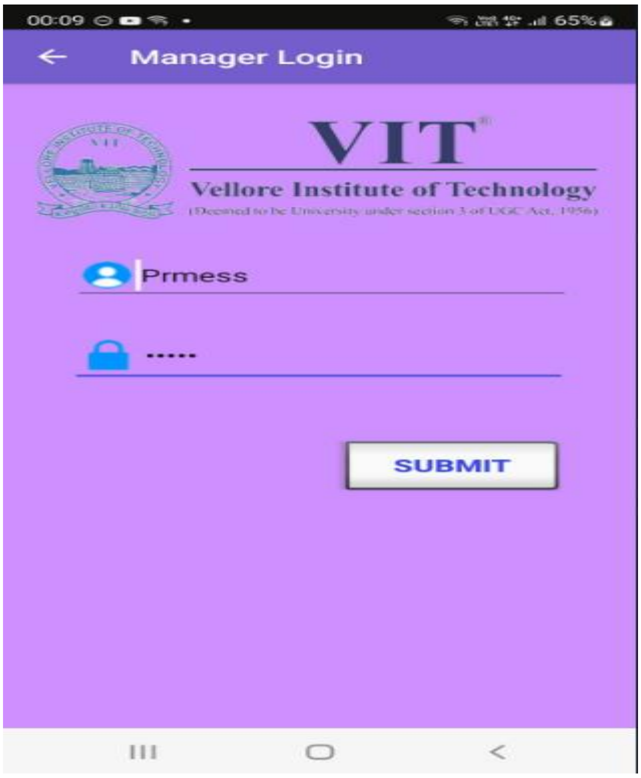
name

email_id

Phone no.

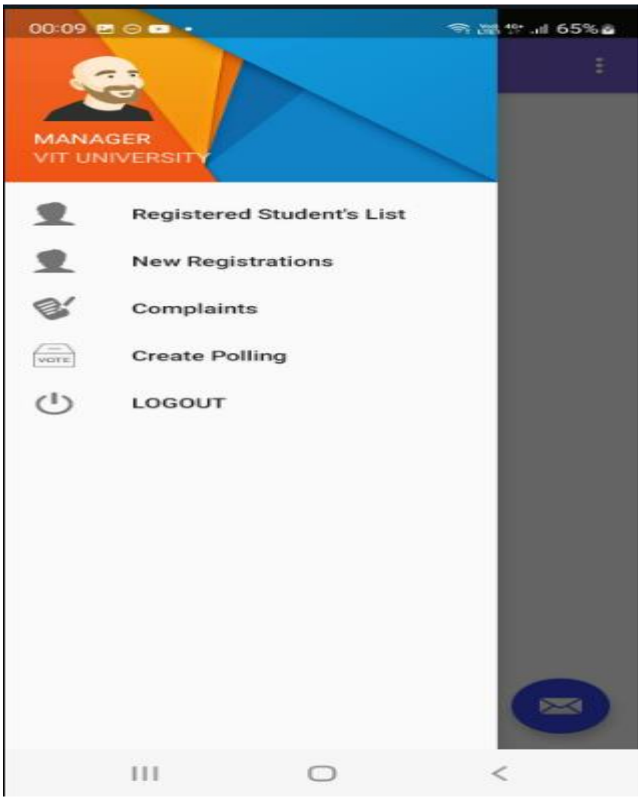
Manager View

- **Mess Manager Login Page**

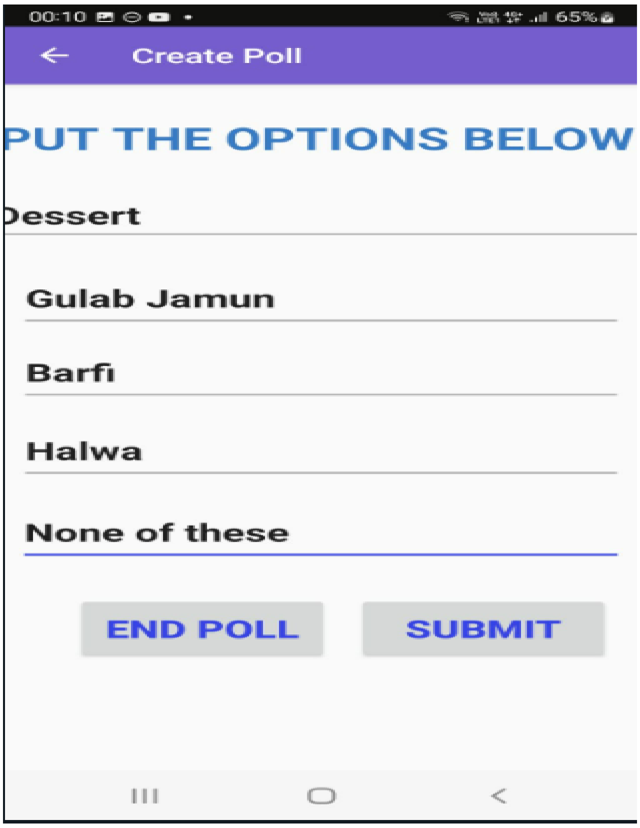


00:09
65%
Manager Login
VIT
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)
Prmess
•••••
SUBMIT

- **Manager Home Page**



- **Create Poll Page**



PUT THE OPTIONS BELOW

Dessert

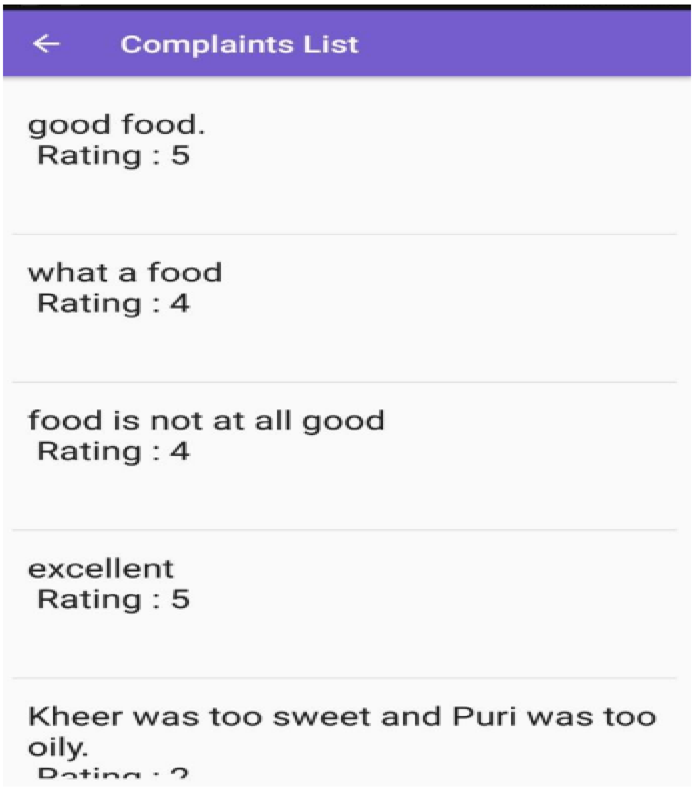
Gulab Jamun

Barfi

Halwa

None of these

- **Complaints Page**



good food.
Rating : 5

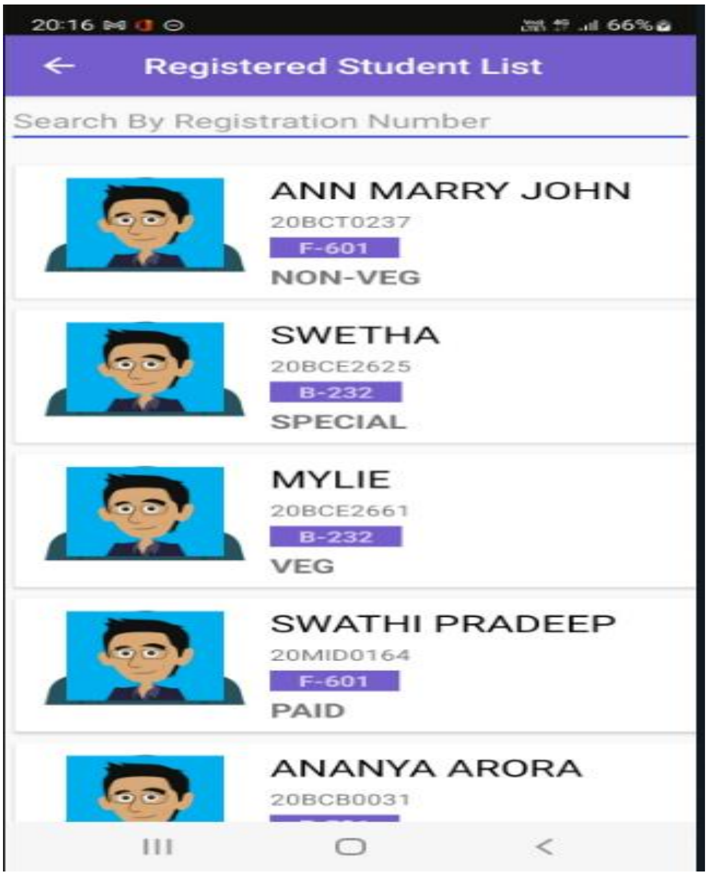
what a food
Rating : 4

food is not at all good
Rating : 4

excellent
Rating : 5

Kheer was too sweet and Puri was too oily.
Rating : 2

- **Registered Students List**



ANN MARRY JOHN
20BCT0237
F-601
NON-VEG

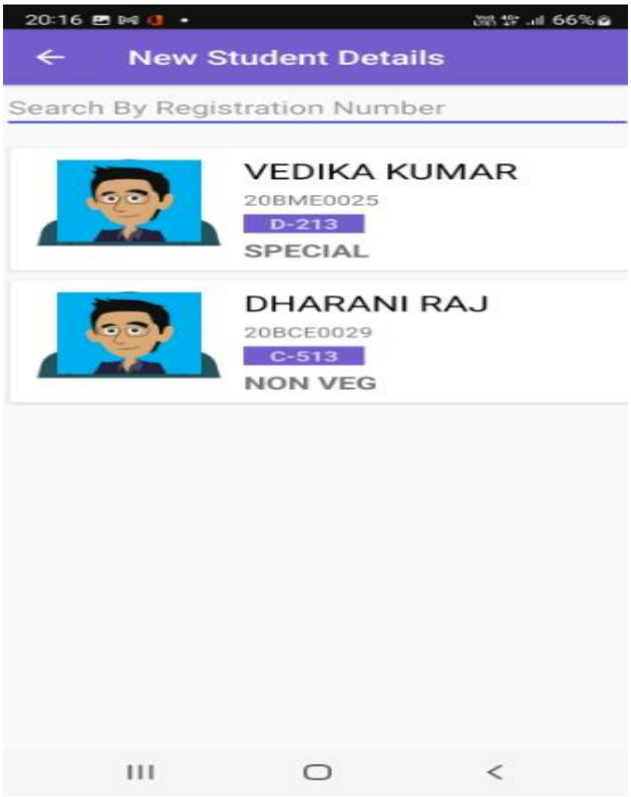
SWETHA
20BCE2625
B-232
SPECIAL

MYLIE
20BCE2661
B-232
VEG

SWATHI PRADEEP
20MID0164
F-601
PAID

ANANYA ARORA
20BCB0031

- **New Registration Page**



VEDIKA KUMAR
20BME0025
D-213
SPECIAL

DHARANI RAJ
20BCE0029
C-513
NON VEG

CONCLUSIONS:

To conclude, we have developed a fully functional mess management app for our college mess system. We have successfully developed an android app that has two functional features: Student and manager.

We have tried to include all the necessary functionalities for the app that can make the current existing mess management more efficient. We have also tried to resolve all the troubles that we are facing concerning the mess and this application This app is a solution that resolves all of the trouble that a scholar faces regarding mess like having troubles with mess meals, recommendations to be provided, mess changes, complaints speaking to the supervisor. This app can be a preventative answer for any troubles concerning the mess. We have attempted to create a user-pleasant interface so as to be easily handy and act as a mediator among mess supervisors and students. Our interface design is quite simple and understandable making it reliable and efficient to use for all students, faculties,mess managers and all the other concerning authorities. Future scope of our project that could be included are voice enabled mess assistant, a more effective polling strategy for choosing mess menus, easier transition of mess changes,etc.

REFERENCES:

[1] Abhishek Singh1, Adithya R2, Vaishnav Kanade3, Prof. Salma Pathan4 “ONLINE FOOD ORDERING SYSTEM “International Research Journal of Engineering and Technology (IRJET)

[2] Prathamesh Jagannath Mane” Study of Online Food Delivery App like Zomato & Swiggy and their effect on Casual Dining.” International Journal of Scientific Research and Engineering Development.

[3] Shweta Shashikant Tanpure* “Automated Food Ordering System with Real-Time Customer Feedback”, International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, Issue 2, February 2013, Department of CSE, J.S.P.M,Pune

[4] Varsha Chavan, Priya Jadhav, Snehal Korade, Priyanka Teli, ”Implementing Customizable Online Food Ordering System Using Web Based Application”, International Journal of Innovative Science, Engineering Technology(IJISET) 2015

[5] Kirti Bhandge, Tejas Shinde, Dheeraj Ingale, Neeraj Solanki, Reshma Totare,”A Proposed System for Touchpad Based Food Ordering System Using Android Application”, International Journal of Advanced Research in Computer Science Technology (IJARCST 2015).

[6] Ashutosh Bhargave, Niranjan Jadhav, Apurva Joshi, Prachi Oke, S. R Lahane,“Digital Ordering System for Restaurant Using Android”, International Journal of Scientific and Research Publications 2013.

[7] Khairunnisa K., Ayob J., Mohd. Helmy A. Wahab, M. Erdi Ayob, M. Izwan Ayob, M. Afif Ayob,”The Application of Wireless Food Ordering System” MASAUM Journal of Computing