A Project Report on

XPROXMESS - A Mess Management Application

Undergone at

National Institute of Technology Karnataka

Under the guidance of

Ms.Saumya Hegde

Submitted By

Divyansh Verma

(16CO110)

Pawan Rahangdale

(16CO237)

Indrajeet Ratnakar

(16CO116)

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

Bachelor of Technology
In
Computer Science & Engineering

INTRODUCTION

Purpose

The purpose of this document is to present a detailed description of the Mess Management System. It will explain the purpose and features of the system, the interfaces of the system, what the system will do, the constraints under which it must operate and what kind of system interactions take place.

Scope of Project

This software system will be a mess management system which consists of two separate parts, one to be used by the students who eat in the mess and one for the mess administrator. The student software allows students to login and choose the mess option for the whole semester, place orders for the night canteen online, book the mess for special occasions, view the daily mess menu and the night canteen (NC) menu, view food consumption statistics, inform the mess authorities when he/she won't be eating in the mess and submit feedback. The admin part of the software will allow the mess administrator to make changes to the daily mess menu, upload food consumption statistics and view net profit, view orders for the night mess and approve any special lunch orders as well along with reading feedback. The system will utilize a relational database for handling all the data such as menu items, prices, student IDs, consumption statistics, student feedback,etc.

Glossary

Term Definition

Database-Collection of all the information monitored by this system.

Mess Admin -The administrator in charge of the mess - updates the databases, uploads statistics, keeps track of orders, resets the system after every meal, etc.

Software Requirements Specification-A document that completely describes all of the functions of a proposed system and the constraints under which it must operate. For example, this document. Student A student avails the mess facilities.

Menu-A collection of all the food items which are available in the mess.

Order-A collection of food items from the menu that the student wants to buy.

Special Order-A list of meal preferences along with other data such as the date and time of the special lunch. A special order will typically be made for a club/department meeting or for special occasions such as festivals.

NC-Night Canteen (open from 7 pm to 2:00 am), sells items which are not on the daily mess menu in a pay-and-eat system.

Database Design:-

Database Design and Database Tables:-

- **admin_table:-**

Attribute Name	**Datatype**
id	Int(10) (Primary Key)
username	varchar(100)
password	varchar(100)
mess	varchar(100)

```
CREATE TABLE `admin_table` (
  `id` int(11) NOT NULL,
  `username` varchar(100) NOT NULL,
  `password` varchar(100) NOT NULL,
  `mess` varchar(256) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

- **Feedback table:-**

Attribute Name	**Datatype**
fid	int(11) (Primary Key)
roll_no	varchar(100)
mess	varchar(100)
complaint	varchar(100)
sub_date	date

```
CREATE TABLE `feedback` (
  `fid` int(11) NOT NULL,
  `roll_no` varchar(100) DEFAULT NULL,
  `mess` varchar(100) NOT NULL,
  `complaint` varchar(256) NOT NULL,
  `sub_date` date NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

- **FirstBlockMenu Table:-**

Attribute Name	**Datatype**
day	varchar(15)
breakfast	varchar(100)
lunch	varchar(100)
dinner	varchar(100)

```
CREATE TABLE `FirstBlockMenu` (
  `day` varchar(15) DEFAULT NULL,
  `breakfast` varchar(100) NOT NULL,
  `lunch` varchar(100) NOT NULL,
  `dinner` varchar(100) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=hp8;
```

- **FourthBlockMenu Table:-**

Attribute Name	**Datatype**
day	varchar(15)
breakfast	varchar(100)
lunch	varchar(100)
dinner	varchar(100)

- **MegaMessMenu Table :-**

Attribute Name	**Datatype**
day	varchar(15)
breakfast	varchar(100)
lunch	varchar(100)
dinner	varchar(100)

- **SecondBlockMenu table:-**

Attribute Name	**Datatype**
day	varchar(15)
breakfast	varchar(100)
lunch	varchar(100)
dinner	varchar(100)

- **Mess_allot Table:-**

Attribute Name	**Datatype**
mid	int(11) Primary Key
roll_no	varchar(100)
mess	varchar(100)
apply_date	date

```
CREATE TABLE `mess_allot` (
  `mid` int(11) NOT NULL,
  `roll_no` varchar(256) NOT NULL,
  `mess` varchar(100) NOT NULL,
  `apply_date` date NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=hp8;
```

- **Students Table:-**

Attribute Name	**Datatype**
roll_no	varchar(100)
email	varchar(100)
password	varchar(100)

- **Orders Table:-**

Attribute	**Datatype**
address	varchar(255)
phone	varchar(255)
food	varchar(255)
price	int(11)
quantity	int(11)
rollno	varchar(255)
tim	datetime

```
CREATE TABLE `orders` (
  `address` varchar(255) NOT NULL,
  `phone` varchar(255) NOT NULL,
  `food` varchar(255) NOT NULL,
  `price` int(11) NOT NULL,
  `quantity` int(11) NOT NULL,
  `rollno` varchar(255) NOT NULL,
  `tim` datetime NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=hp8;
```

- **Night_canteen_veg Table:-**

Attribute	**Datatype**
Food	varchar(50)
idd	int(11) Primary Key
price	int(4)
availability	int(2)
tag	varchar(50)

```
CREATE TABLE `night_canteen_nonveg` (
  `Food` varchar(50) NOT NULL,
  `price` int(4) NOT NULL,
  `availability` int(2) NOT NULL,
  `tag` varchar(50) DEFAULT NULL,
  `idd` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=hp8;
```

- **Night_canteen_nonveg Table:-**

Attribute	**Datatype**
Food	varchar(50)

idd	int(11) Primary Key
price	int(4)
availability	int(2)
tag	varchar(50)

- **Mess_count Table:-**

Attribute	**Datatype**
mess	varchar(256)
left_count	int(11)

```
CREATE TABLE `mess_count` (
  `mess` varchar(256) NOT NULL,
  `left_count` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=hp8;
```

Overall Description

System Environment

System Environment

The Mess Management System has two active actors and one relational database.The Students use a web portal to which they can login with their ID and

password and avail all the facilities available. The Mess Admin has access to a different interface on his/her end, with a login and password. A relational database is used to store all the data such as feedback, consumption statistics, list of registered students, menus with items and prices, etc.

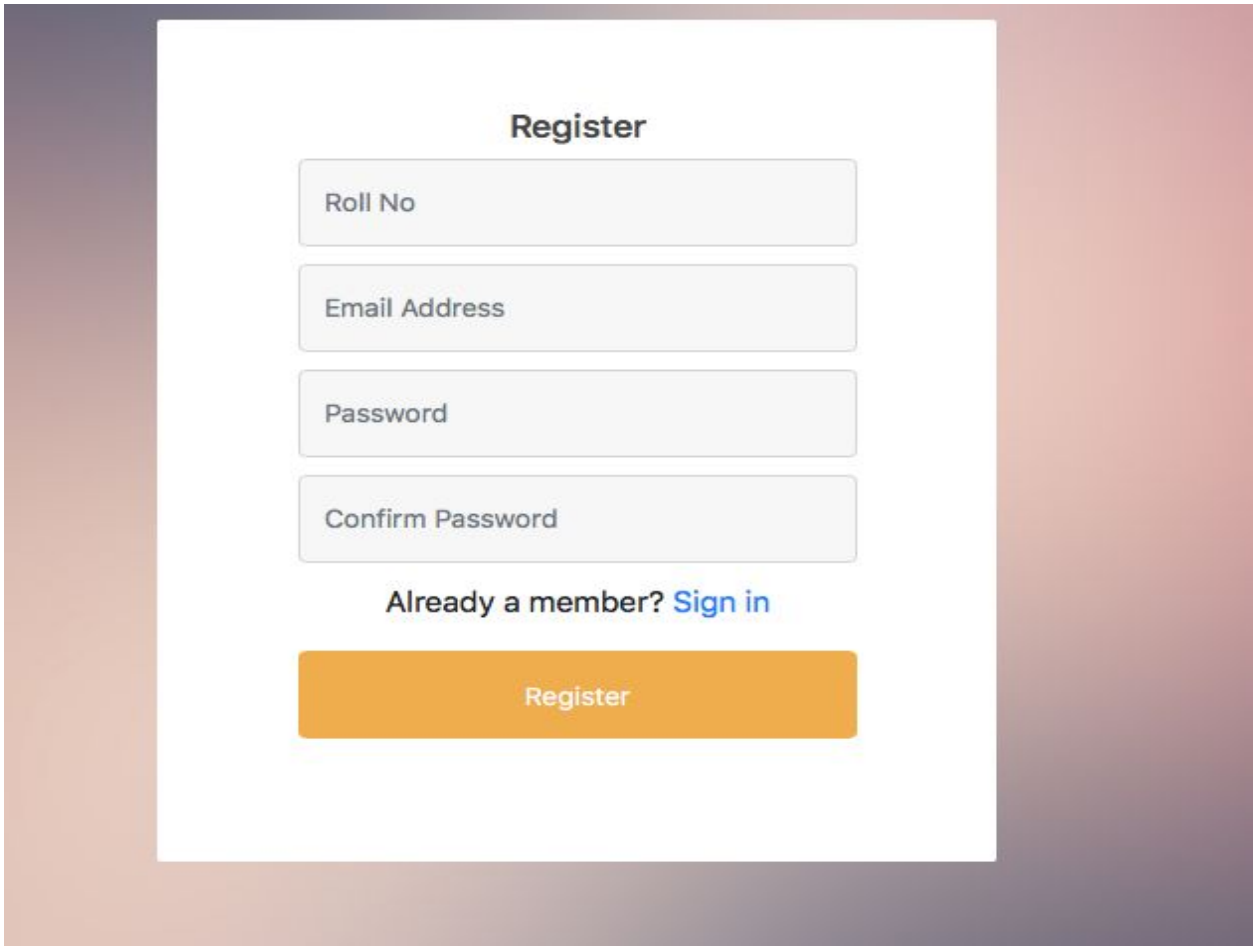
Functional Requirements Specification

This section outlines the use cases for each of the actors separately.
The Students have a different set of use cases (which are related to availing the mess facilities) than the Mess Admin (which are related to managing the various aspects of the mess such as special orders, NC orders, student mess option, menu updates, etc.).

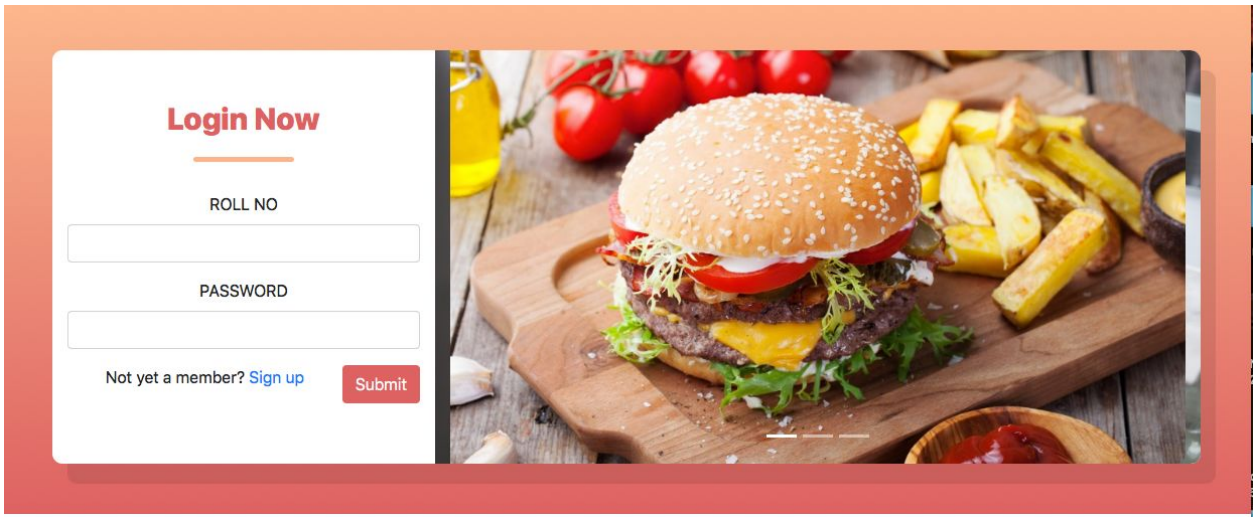
Student Use Cases

A brief description of all the Student Use Cases:

1)**Register -** The student needs to register for this application to use this software by filling out email Id , roll no and other credentials .



2)**Login –** The student logs into the system so that he can carry on with other options like choosing mess option or placing orders for the night canteen. The student has to enter his Roll no and password which will be then compared with the database entries to validate his login credentials.

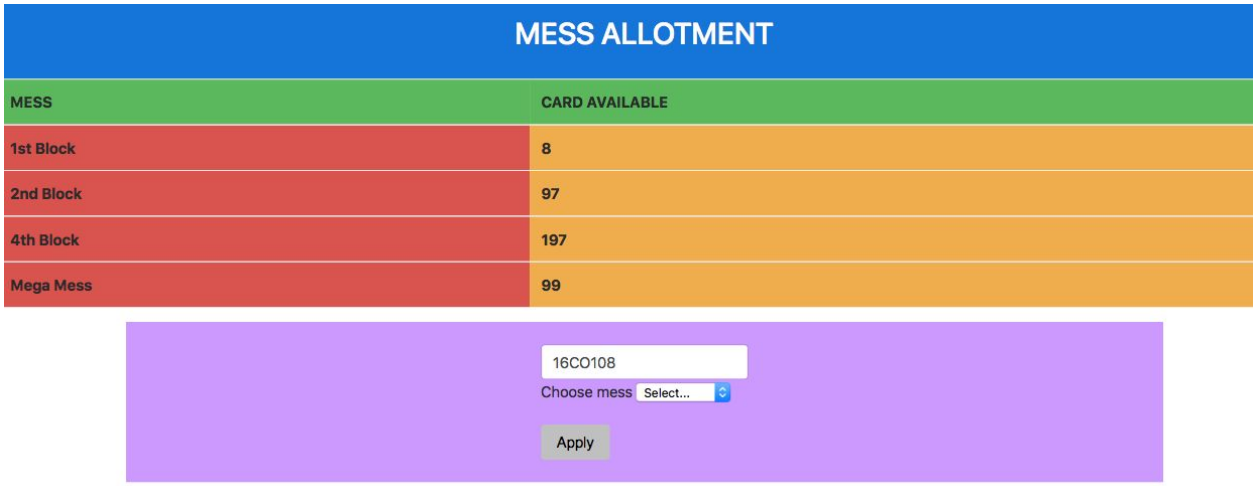


3)**HomePage/Log Out** - When student logs in ,Home page will be displayed. student logs out of the system so that nobody else can modify his mess option or book orders from his account.

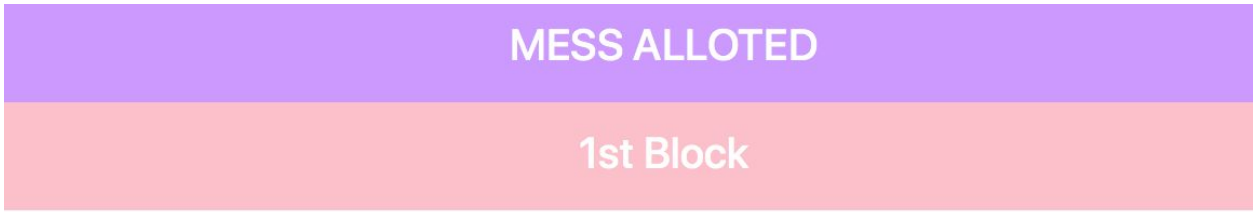
4)Choose Mess Option – Through this module, student chooses his mess option.If mess card for selected mess is available then that mess will be alloted to student and His choice is then sent to the database .If mess card is not available for chosen mess then student will have to select other mess.

When mess has been already allocated to student then this page will show a message that you have been allocated a particular mess.

If mess is not yet allocated to student then this module will show below page



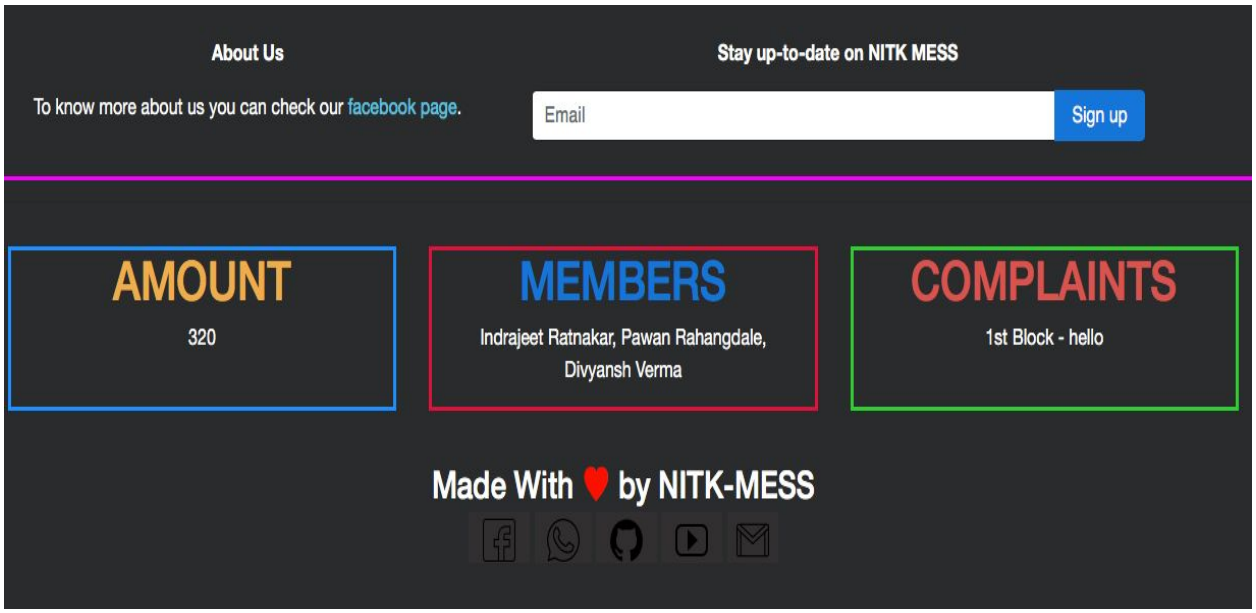
If mess is already allocated to student then this module will show below page



5)**View Menu –** The student gets to view the mess menu for the week by selecting a particular mess . He can view menu like Monday's lunch menu or the Night mess menu. The query from the student is sent to the database where the data is read and is sent back to the student to be displayed on the screen.

		SecondBlockMenu	
DAY	BREAKFAST	LUNCH	DINNER
MON	Samosa	Masala Bhindi	Singhada
TUE	PudiSabji	Alu Chilli	Matar Paneer
WED	Poha	BEKAR SABJI	Rajma
THURS	Alu Paratha	Capsicum	Chicken 65
FRI	Dosa	Dam Alu	Manchurian
SAT	GoliBaji	Kadhahi Chicken	Karela Jon
SUN	Samosa	Masala Bhindi	Singhada

6) <u>View Amount and Feedbacks</u> – An amount will be displayed in the homepage which is the total price student will have to pay to the related mes by the end of the month and student can see feedbacks given by other students.



7)**Book NC order** – The student can view the night mess menu as different menus for veg and non-veg food items.These food items are categorized by using tag for example In veg food, a food item **Puri Sabji** comes under **Snacks** tag and In non-veg food, a food item **Chicken Biryani** comes under **Chicken** tag.Students can set quantity for a food item that they want to order from night canteen.Those food items will be added to cart and student will be redirected to a new page.

In this page student has to fill contact no and address where ordered food will be delivered. Total price of order will be shown in this page and student will have to confirm the order.

The student's order is sent to the database where it is processed by the system.

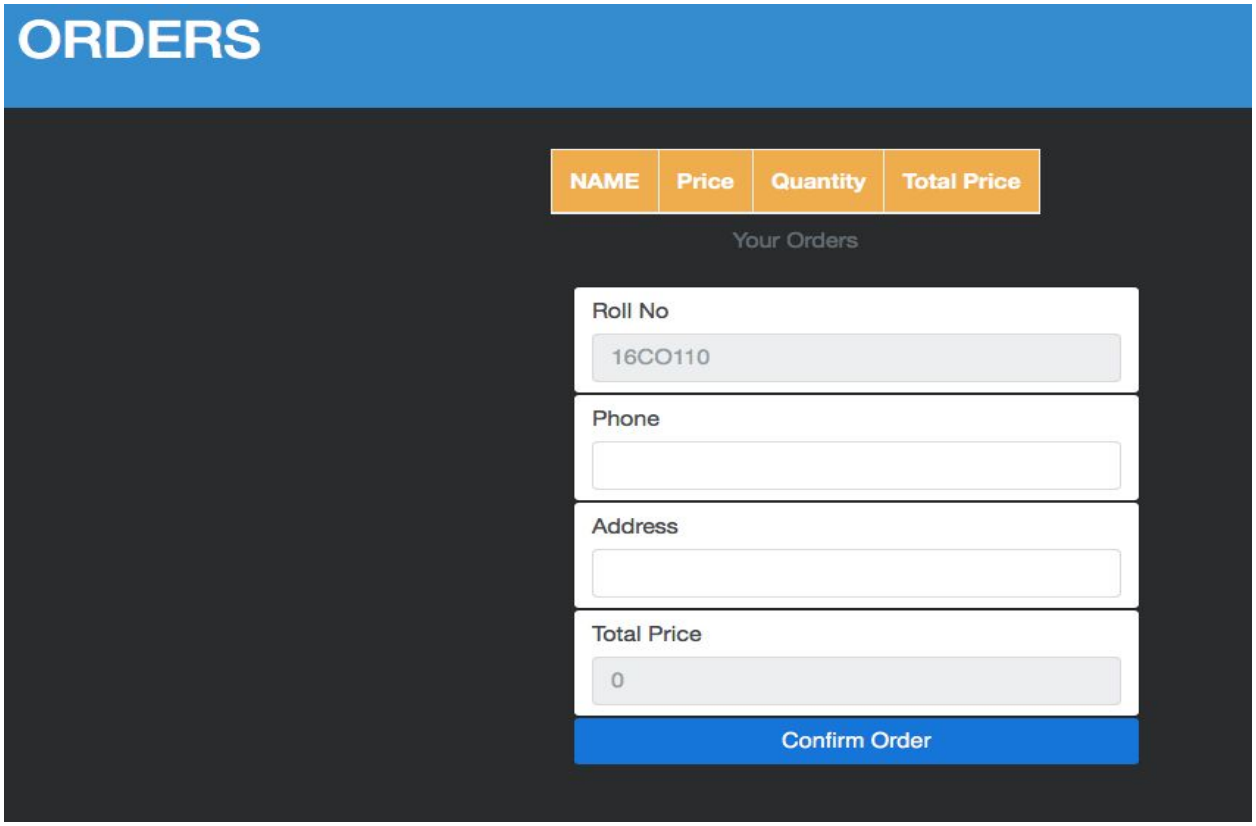
	Snacks		
FOOD ITEM	PRICE	AVAILABLE	QUANTITY
Puri Bhaji	40	1	0
Dosa	35	1	0

Order from Veg Night Canteen

	Chicken		
FOOD ITEM	PRICE	AVAILABLE	QUANTITY
Chicken Kurma	50	1	0
	Fish		
FOOD ITEM	PRICE	AVAILABLE	QUANTITY
Prawns	200	1	0
Fish Curry	50	1	0

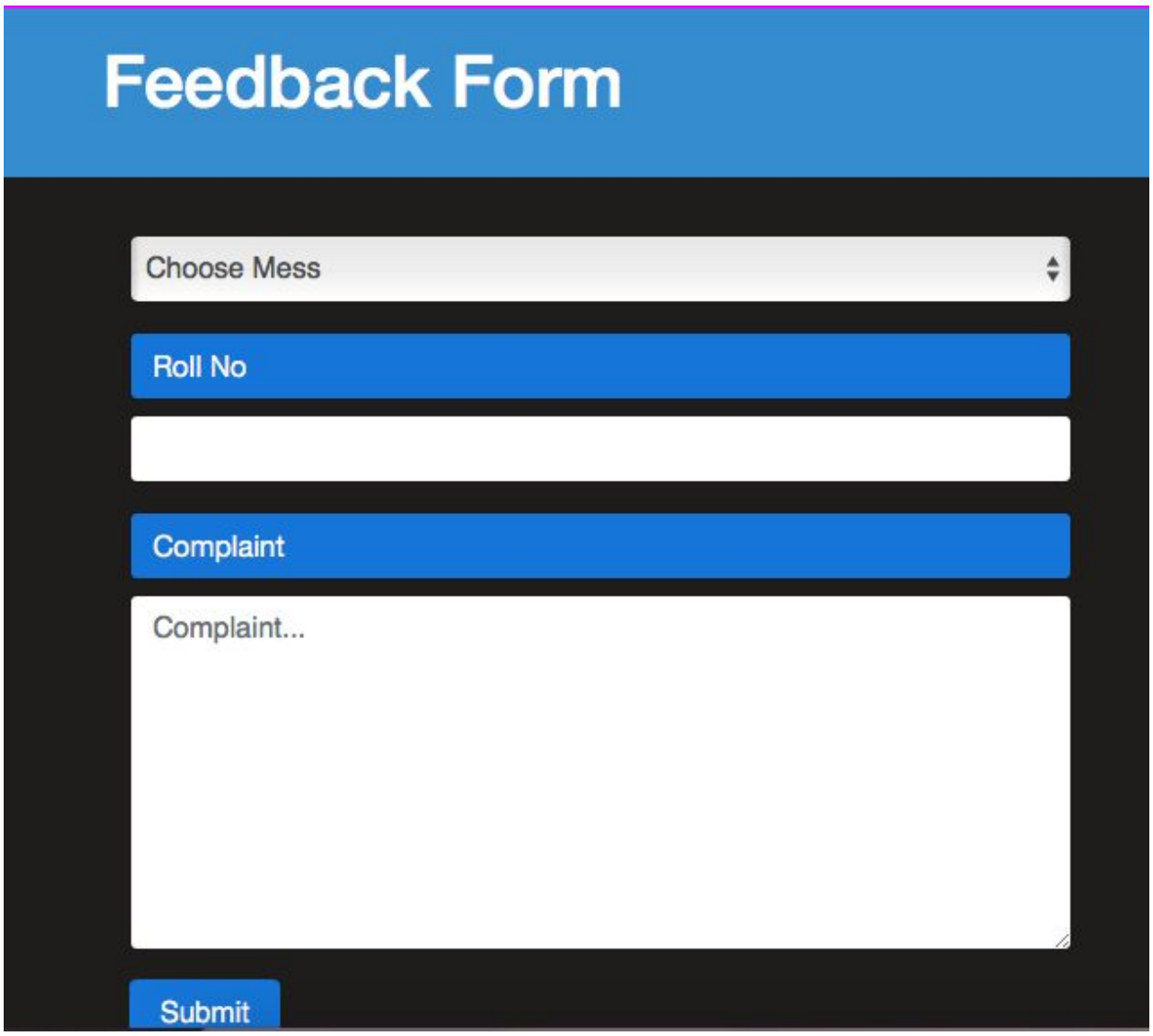
Order from Non Veg Night Canteen

ADD TO CART



8) <u>My Orders-</u> In this page, student can view his night canteen order details where for each ordered food item price , quantity and time of placed order will be shown and will get deleted when ordered food items are received by student.

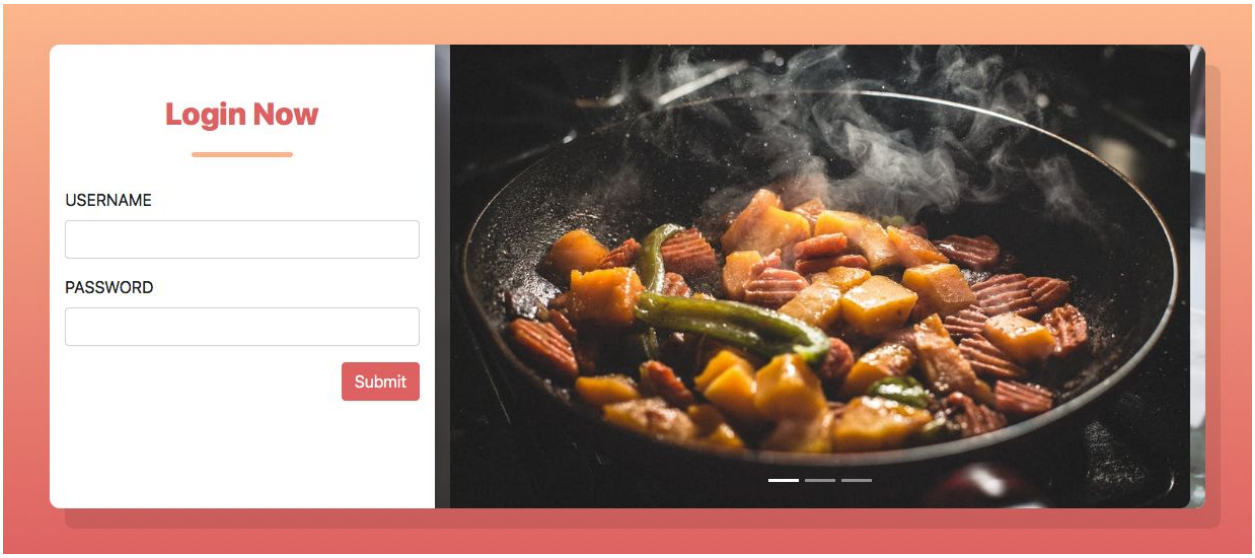
9) <u>Feedback-</u> Student can give feedback for a particular mess.This can be either anonymous or student can provide roll no also.feedback will be saved in the database and these feedbacks can be viewed in the home page


Feedback Form
Choose Mess
Roll No
Complaint
Complaint...
Submit

Mess Admin Use Cases

A brief description of the Mess Admin Use Cases:

1)**Login** – The administrator has to login to the system so that he can carry on with the other activities such as updating the mess menu,viewing and uploading statistics, viewing NC orders etc. The data entered will be matched with the fields in the database to validate the admin's credentials.



2)**Home Page/Log out-** When admin logs in ,Home page will be displayed. Admin logs out of the system so that nobody else can modify his mess option or book orders from his account.

2) **Update Mess Cards available -** The admin can update available mess cards for each mess.

3)**View Allotment-** Administrator can also view list of students who have been alloted a mess ,name of alloted mess along with their mess allocation date.

ALLOTMENT

ROLL NO	MESS	APPLY DATE
16CO110	1st Block	2018-11-12
16CO116	2nd Block	2018-11-12

4)**Update Menu –** The administrator can update the mess menu for any meal for any day of the week of a particular mess.The changes after the updation are saved in the database so that the students can see the updated menu.

SecondBlockMenu

DAY	BREAKFAST	LUNCH	DINNER
MON	Samosa	Masala Bhindi	Singhada
TUE	PudiSabji	Alu Chilli	Matar Paneer
WED	Poha	BEKAR SABJI	Rajma
THURS	Alu Paratha	Capsicum	Chicken 65
FRI	Dosa	Dam Alu	Manchurian
SAT	GoliBaji	Kadhahi Chicken	Karela Jon
SUN	Samosa	Masala Bhindi	Singhada

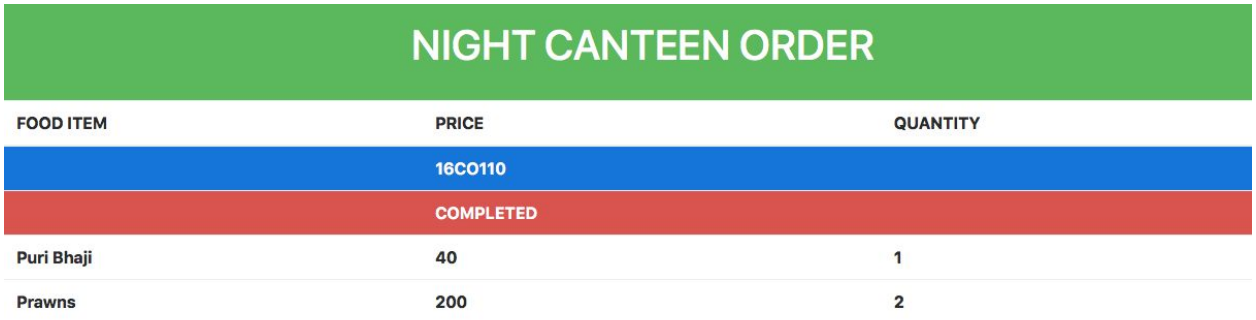
Update

MENU .....

Main page

5)**View NC Orders –** The admin can check the number of orders received for the night canteen mess and also pass along that to the kitchen staff so that they can prepare the meals according to those specifications. The admin can then update the

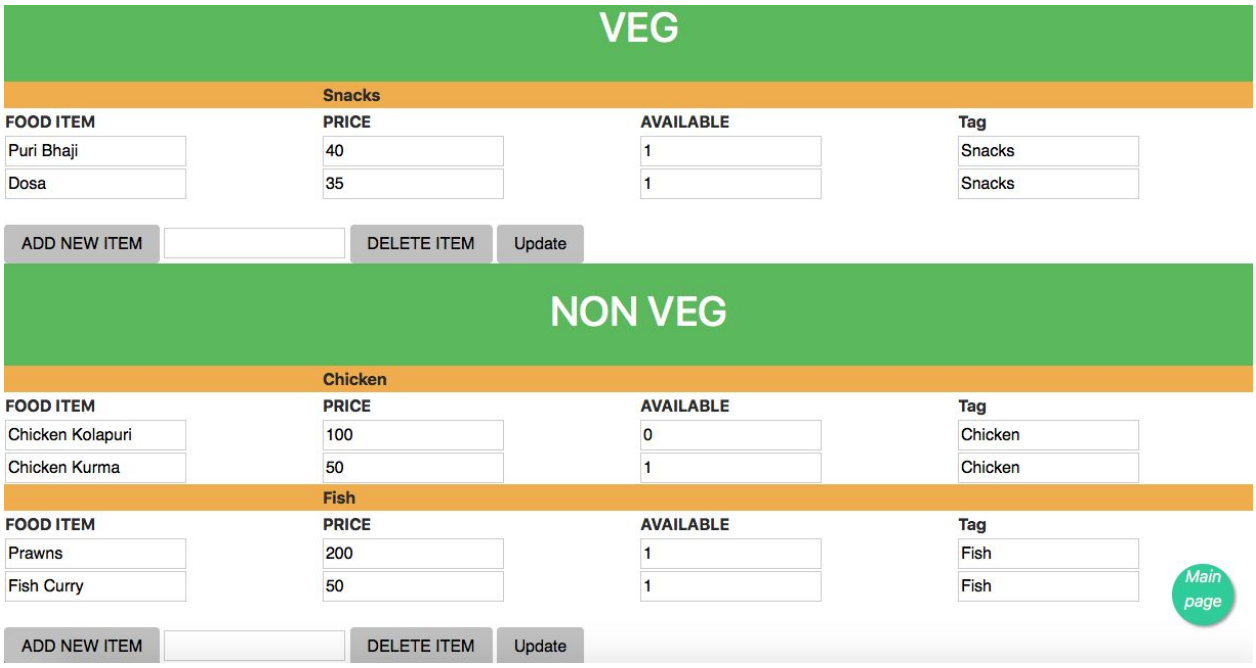
order status in the database to completed so that the student can come and pick it up after which its status will be delivered.



NIGHT CANTEEN ORDER

FOOD ITEM	PRICE	QUANTITY
	16CO110	
	COMPLETED	
Puri Bhaji	40	1
Prawns	200	2

6)**<u>Update Night Canteen Menu -</u>** The admin can add new food items based on food item's category like Chicken Kurma as Non-veg item in chicken category, delete some food items , change the availability whether a particular food item is available or not.



VEG

Snacks

FOOD ITEM	PRICE	AVAILABLE	Tag
Puri Bhaji	40	1	Snacks
Dosa	35	1	Snacks

ADD NEW ITEM | DELETE ITEM | Update

NON VEG

Chicken

FOOD ITEM	PRICE	AVAILABLE	Tag
Chicken Kolapuri	100	0	Chicken
Chicken Kurma	50	1	Chicken

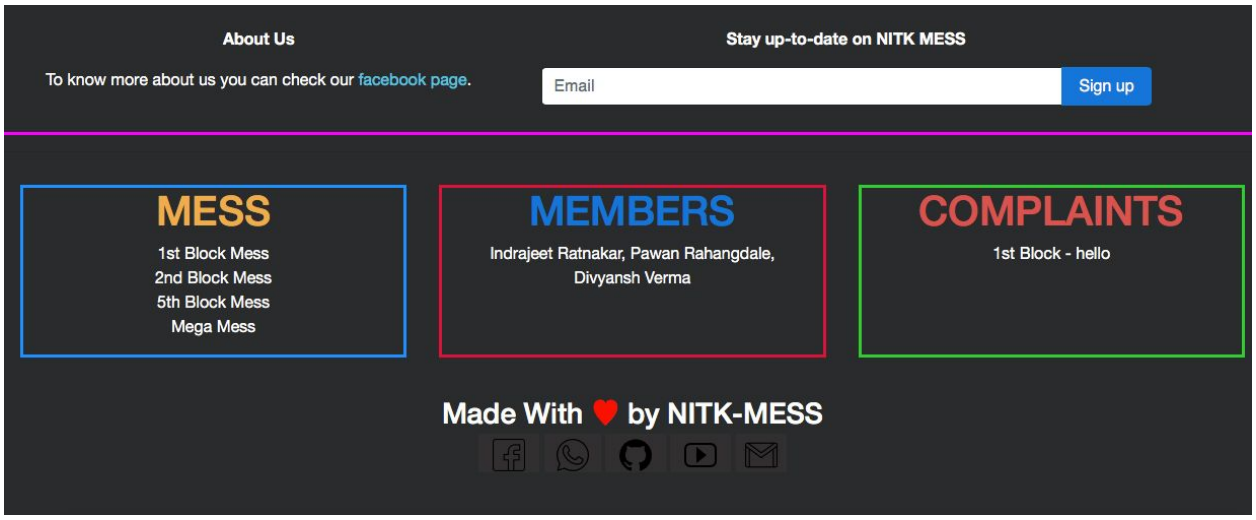
Fish

FOOD ITEM	PRICE	AVAILABLE	Tag
Prawns	200	1	Fish
Fish Curry	50	1	Fish

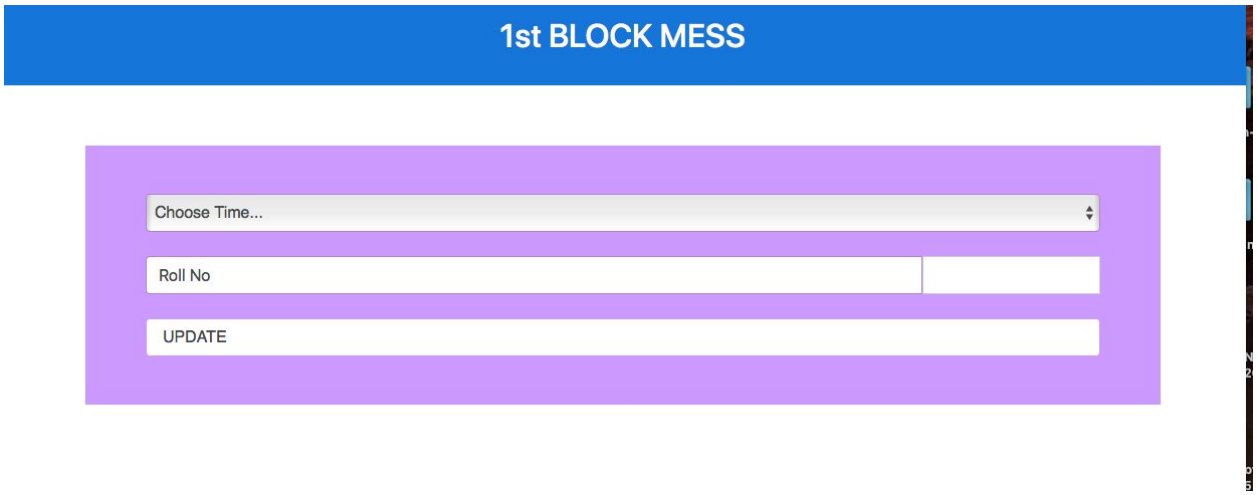
Main page

ADD NEW ITEM | DELETE ITEM | Update

7)**<u>View Feedback –</u>** The mess admin can view the feedback submitted by the students. The database will be queried for feedback and complaints and the admin can check all these forms and take the required action on them.



8) **<u>Update amount of students</u>**- Admin will update amount of student whenever student visits mess which has to be paid by student by the end of month.



<u>9)**Reset System –**</u> The mess admin can trigger a hard reset of the system where the list of people who have eaten their meal is erased. This will typically occur automatically after every meal and so this hard reset will seldom be used.

<u>User Characteristics</u>

The Students and Mess Admin are expected to have a basic working

knowledge of a computer in order to use this software. The User

Interface will be quite intuitive, so any advanced knowledge will not be

necessary.

Non-Functional Requirements

The physical machine to be used in the mess needs to have internet access in order to connect to the database. This software will not assume that a code scanner hardware is available on the machine, and so the ID input will be done via keyboard. Students need internet access on their devices as well, since all the data will be stored on the database which the software will need to connect to.

Requirements Specification

External Interface Requirements

User Interfaces : The Interface will be in the form of a webapp. It is designed to be functional and minimal in its styling. All options will be displayed in a menu based format. HTML and CSS will be used to setup the page layout and add minimal styling to make the interface user friendly.

Hardware Interfaces : A webserver will be required so that the students and the mess admin can connect to it to exchange information. The server have a database to store all the data entries. The Server will have to have a high speed 1 Gigabit ethernet connection to the college's local network.But here in our project for simplicity purposes we are going to run it on out local server and then we would host it on github.

Software Interfaces : It will also have a MySQL relational database. The main backend processing will be done using node js including connecting to and accessing the database and processing requests.

Communications Interfaces : The main communication protocol will be Hyper Text Transfer Protocol (HTTP). This will be used to transfer information back and forth from the client to the server. HTTP GET and POST will be used to send the information.

Detailed Non-Functional Requirements

Logical Structure of the Data

Student database will consist of Name,Year , Institute Roll no,the amount that would be refunded,Their statistics of eating in the mess.

Mess Database will consist of the Mess memu,Statistics of Student visiting the mess on a particular day,etc.

Similarly The Night Canteen Database will consist of the Nc menu,Prices,Orders placed,Status of the order,etc.

Security Measures

1.PREVENTION OF SQL INJECTIONS

SQL injection, also known as SQLI, is a common attack vector that uses malicious SQL code for backend database manipulation to access information that was not intended to be displayed. This information may include any number of items, including sensitive company data, user lists or private customer details.
The impact SQL injection can have on a business is far reaching. A successful attack may result in the unauthorized viewing of user lists, the deletion of entire tables and, in certain cases, the attacker gaining administrative rights to a database, all of which are highly detrimental to a business.

Prevention of SQL Injection in our Application:-

In our Web Application 'Xproxmess' we have used "mysqli_real_escape_string " to prevent sql injection .Escapes special characters in a string for use in an SQL statement, taking into account the current charset of the connection.This is helpful in bypassing the sql injections since the input string doesn't interfere with the sql query in action.

2. PREVENTION OF HTML CODE IN INPUT USING HTMLENTITIES

Whenever you allow your users to submit text to our website either for login purpose or for registration , you need to be careful that you don't leave any security holes open for malicious users to exploit. If you are ever going to allow user submitted text to be visible by the public you should consider using the *htmlentities* function to prevent them from running html code and scripts that may be harmful to your website .

In our XPROXMESS Application we have used Htmlentities so that our input doesn't interfere with the Html code .It encodes the special characters where are the part our html markup syntax so that out input is considered only as the string and not as as any html.

3 . SESSION MANAGEMENT IN OUR XPROXMESS APPLICATION

Whenever a user logins into our XPROXMESS Application we initialise a session variable for that user considering the user email as the primary key as the email ids are unique for all the users .So whenever a user tries to access the web pages that requires authentication ,first the session variable is checked whether it is initialised is not ,if it is initialised then the user is directed to the requested web page otherwise not. Broken authentication and session management encompass several security issues, all of them having to do with maintaining the identity of a user. If authentication credentials and

session identifiers are not protected at all times an attacker can hijack an active session and assume the identity of a user.

4. INSECURE DIRECT OBJECT REFERENCES

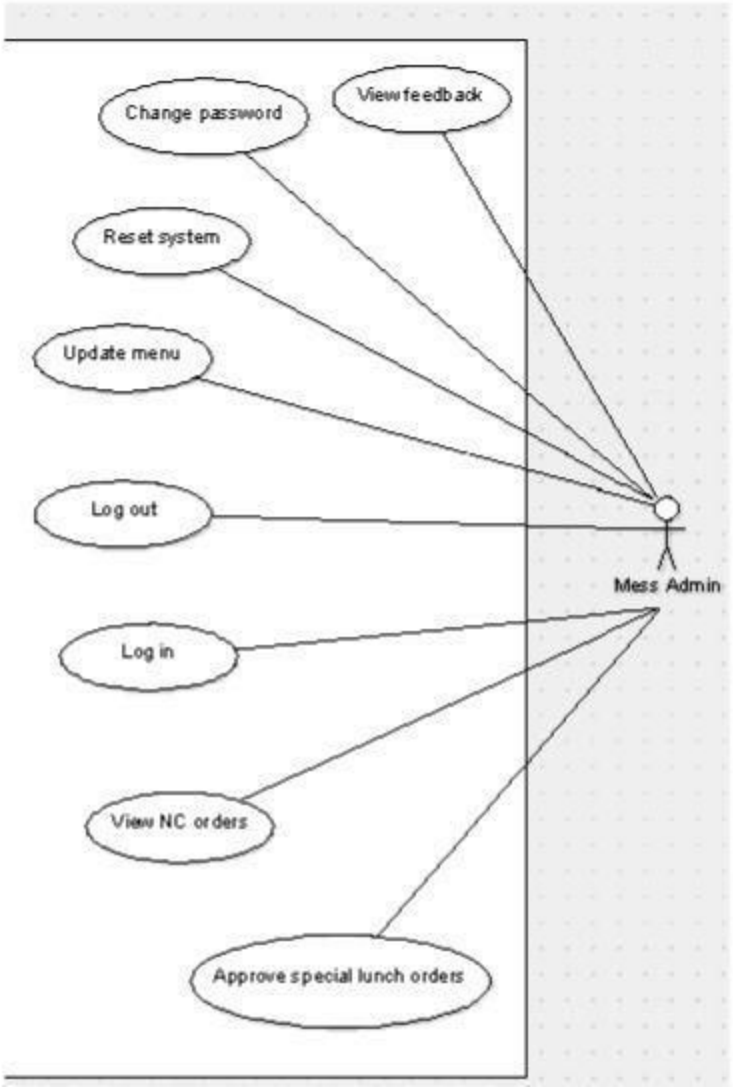
Nothing critical data such as the database password or keys is explicitly mentioned in the url while make get requests so it prevents our Application from getting hacked .Internal Implementations of all the database queries also prevents from insecure data direct object references.

5. PASSWORD ENCRYPTION

We have encrypted all the passwords stored in our database using md5 hashing algorithm which prevents the account being getting hacked since the passwords stored in the database can't be apprehended properly by humans as it can't be decrypted without a special key which is not known.

Use case Diagrams:-

Mess Admin Use case Diagram



Student Use case Diagram