Online Mess Management System

A project submitted to

**UKA TARSADIA UNIVERSITY**

in partial fulfillment of the requirements for the degree of

# Bachelor of Science

in

# Information Technology

for

# 5 Years Integrated M.Sc.(IT)

By

**Patel Anvi A.**

202006100110081

**Desai Yash A.**

202006100110086

Guided by

**Dr. Sapan Naik**

**Assistant Professor**



**Babu Madhav Institute of Information Technology Uka Tarsadia University**

**Bardoli – 394350**

**January 2023**

**CERTIFICATE**



This is to certify that **Patel Anvi A. (202006100110081) and Desai Yash A. (202006100110086)** has submitted project entitled “**Mess Management System”** as the partial fulfillment for the award of the degree of Bachelor of Science in Information Technology for 5 Years Integrated M.Sc. (IT) in 2022 – 2023.

**Date: 27-01-2023**

**Place: Uka Tarsadia University**

# Dr. Sapan Naik Dr. Jitendra Nasriwala

Guide Programme Coordinator,

**External Examiner**

**Babu Madhav Institute of Information Technology, Uka Tarsadia University,**

**Bardoli – 394350**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Chapters** | **Particulars** | | | **Page no.** |
| **1** | **Introduction** | | | 1-2 |
| 1.1 | Problem Definition | | 1 |
| 1.2 | Project Objective | | 1 |
| 1.3 | Product Scope | | 2 |
|  | | | | |
| **2** | **Overall Description** | | | 4 |
| 2.1 | Product Perspective/Environment Description | | 4 |
| 2.1.1 | Hardware Interface/ Hardware Specification | 4 |
| 2.1.2 | Software Interface/ Software Specification | 4 |
|  | | | | |
| **3** | **System Specific Requirements** | | | 5-10 |
| 3.1 | Functional Requirement | | 5-9 |
| 3.2 | Non-functional Requirement | | 10 |
|  | | | | |
| **4** | **UML Diagrams** | | | 16-23 |
| 4.1 | Use case Diagram | | 16 |
| 4.2 | Activity Diagram | | 17-23 |
|  | | | | |
| **5** | **System Design** | | | 11-15 |
| 5.1 | Database Design | |  |
| 5.1.1 | Database Schema | 11 |
| 5.1.2 | Data Dictionary | 12-15 |
|  | | | | |
| **6** | **System Implementation** | | |  |
| 6.1 | Screenshots | |  |
|  | | | | |
| **7** | **Testing** | | |  |
| 7.1 | Test Cases | |  |
|  | | | | |
| **8** | **Future Enhancement** | | |  |
|  |  | | |  |
|  | **Conclusion** | | |  |
|  |  | | |  |
|  | **Bibliography** | | |  |

|  |  |  |
| --- | --- | --- |
| **Figure no.** | **Figure Description** | **Page no.** |
| **5.1.1** | **Use Case Diagram for System** | 16 |
| **5.2.1** | **Activity Diagram for Login** | 17 |
| **5.2.2** | **Activity Diagram for Registration** | 18 |
| **5.2.3** | **Activity Diagram for Meal Booking** | 19 |
| **5.2.4** | **Activity Diagram for Forgot Password** | 20 |
| **5.2.5** | **Activity Diagram for Change Password** | 21 |
| **5.2.6** | **Activity Diagram for Feedback** | 22 |

# List of Tables

|  |  |  |
| --- | --- | --- |
| **Table no.** | **Table Description** | **Page no.** |
|  | Data Dictionary Tables | 12-15 |
|  |  |  |
|  |  |  |
|  |  |  |

1. **INTRODUCTION**
   1. **Problem Definition:**

There are many institutions/Universities which provides facility of hostels for students who live far from their institutes. Hostel facilities includes Dining Hall, Wi-Fi, Playing Room, Reading Room, RO Water Supply etc for students living in hostel. Dining Hall facility particularly is known as Mess where students come together to eat their meals. Hostellers needs to physically go in the mess to check menu every-time. It may

also happen in case of hostellers that sometimes delicious menu is prepared and

students rush into mess all together due to which they need to wait in queue for hours

and hours which is very time consuming for students. When menu is not as good many students avoid going in mess and it may happen that food gets wasted in large quantity, because manually menu quantity is prepared based on attendance of the students in the hostel.

Managing records of the students who attended mess manually is very difficult for administration of the mess. Same facility of lunch is provided to day-scholars/teachers/ guests who come to visit university/hostel and pay per plate accordingly. Managing these outsiders, providing security to all the users, managing meals, and managing payments which has been made per plate to ensure efficient utilization of resource and monitoring inventory is timing consuming as well as possibly inaccurate at the administrative end.

* 1. **Project Objective:**

The purpose of developing software application is to atomise the hostel mess functionality and provide both the users and the admin a smart platform to interact

with each other. This platform will provide facility at administrative end to add itinerary of meals and at others end to view and book upcoming meals. Other users of the system except admin have facility to select the time slot which will help to avoid rush. With the help of the system meals will be prepared based on booked plates rather than attendance taken in the hostel which will help in minimal wastage of food.

It will provide administrators with real-time data on meal attendance, consumption trends, and inventory levels, helping them plan and budget effectively.

By automating payment procedures, the mess management system will ensure accurate

and timely financial transactions. Overall, the system aims to improve user satisfaction,

reduce administrative burdens and foster a seamless dining experience within the institution.

* 1. **Project Scope:**

The Mess Management System will look forward in managing meal plans handle online payments and managing records digitally. Key features of the system are user registration, booking meal, payment processing and feedback mechanism. There will be three users of the system namely Admin, Hostellers, Guests. This system is intended only for one campus having boys as well as girls’ hostel. Students shall book their meals on the daily basis. Guests shall book on every meal they need to grab.

* 1. **Initial Requirement Document:**

|  |  |
| --- | --- |
| **Title of the project** | Hostel Mess Management System |
| **Stakeholders involved in capturing requirements** | Mess Food Supplier, Hostel Warden, Hostel Manager |
| **Techniques used for requirement capturing** | Interviewing, Brainstorming |
| **Name of the persons along with designations** | -- |
| **Date** | 11th July, 2023 |
| **Version** | 1.0 |
| **Users of the system** | Admin, Food Supplier, Hostellers, Visitors |
| **Consolidated list of initial requirements:** | |
| 1. The system shall be able to generate login ID and password for the admin. 2. The admin shall be able to register Food Supplier, Hostellers for the system. 3. The admin shall be able to maintain all the details of the users of the system. 4. The admin shall provide username and password to other users of the system. 5. All the users shall login to the system with the help of provided username and password by the admin. 6. All the users shall be able to change password/forgot password once username and password provided by the admin. 7. Food Supplier shall add menu of meals to the system. 8. Food Supplier shall view total count of the booked meals as per the timing mentioned by the hostellers. 9. Food Supplier shall maintain all the details of the meals. 10. Hostellers shall be able to book their meal as per their time convenience. 11. Hostellers shall book their slot for visiting mess and having meal during breakfast/dinner. 12. Hostellers/Visitors shall be able to give feedback to the system. 13. Visitors shall book their meal only during lunch timing. | |

1. **Overall Description:**
   1. **Product Perspective/Environment Description**
      1. **Hardware Interface/ Hardware Specification**

|  |  |
| --- | --- |
| Utilities | Needs |
| RAM | 8 GB |
| Hard Disk | 25 GB |
| Operating System | All versions of Android IOS |
| Display | 1023 x 768 |

* + 1. **Software Interface/Software Specification**

|  |  |
| --- | --- |
| Type | Web-Based Application, Mobile-Based Application |
| Front-End | REACT JS, FLUTTER |
| Back-End | NODE JS, EXPRESS JS, MONGO DB |
| Operating System | All versions of Windows and Ubuntu |
| Tools | Postman , VS CODE, Android Studio |

1. **System Specific Requirement:**
   1. **Functional Requirements:**
      1. **Manage Login**
      2. **Manage Mess Handler**
      3. **Manage Hosteller**
      4. **Manage Guest**
      5. **Manage Menu**
      6. **Manage Meal Bookings**
      7. **Manage Order History**
      8. **Manage Payment**
      9. **Generate Bill**
      10. **Manage Report**

**Manage Login**

|  |  |  |
| --- | --- | --- |
| RN | Description | Comments |
| FR1 | System shall be able to authenticate Admin/Mess Handler/hostellers/guest using their email id as username and password. | Login\_Page |
| FR2 | System shall be able to generate new password if in case user forgets his old password. | ForgotPass\_Page ChangePass\_Page |
| FR3 | System shall allow users of the system to logout from the system. | All pages after  login |

**Manage Mess Handler**

|  |  |  |
| --- | --- | --- |
| RN | Description | Comments |
| FR1 | This module shall be managed by Admin.  Admin shall be able to register Mess Handler and edit his/her profile and shall be able to change password. | Mess Handler Registration\_Page |
| FR2 | The system shall be able to assign a unique id (mess handler id) to Mess Handler. | Mess Handler  Registration\_Page |
| FR3 | The admin shall be able to:  View/Search Mess Handler details. | Mess Handler Details Page |

**Manage Hostellers**

|  |  |  |
| --- | --- | --- |
| RN | Description | Comments |
| FR1 | This module shall be managed by Admin and Mess Handler.  Admin shall be able to register Hostellers and edit his/her profile and shall be able to change password. | Hosteller Registration Page |
| FR2 | The system shall be able to assign a unique id (hosteller id) to each Hostellers. | Hosteller Registration Page |
| FR3 | The Mess Handler shall be able to:  View/Search Hosteller’s details. | Hosteller Details Page |

**Manage Guest**

|  |  |  |
| --- | --- | --- |
| RN | Description | Comments |
| FR1 | This module shall be managed by Admin, Mess Handler, and Guests.  Guest shall be able to register himself and edit his/her profile. | Guest Registration Page |
| FR2 | The system shall be able to assign a unique id (guest id) to each guest. | Guest Registration Page |
| FR3 | The Admin/Mess Handler shall be able to:  View/Search guest’s details. | Guest Details Page |

**Manage Menu**

|  |  |  |
| --- | --- | --- |
| RN | Description | Comments |
| FR1 | Mess Handler shall be able to manage the menu details.  Mess Handler shall be able to perform following operations like (after login):  Add Menu Details  View/Search Menu Details  Active/De active Menu Details  Update Menu Details | Menu Details Page |
| FR2 | System shall be able to display menu to the Hostellers/Guests. | Home Page of Hosteller/Guest |

**Manage Meal Booking**

|  |  |  |
| --- | --- | --- |
| RN | Description | Comments |
| FR1 | System shall be able to open booking for meal before 3 hours of actual mess opening time. | Add Booking Page |
| FR2 | System shall be able to close booking for meal before 2 hours of actual mess opening time. | Add Booking Page |
| FR3 | After adding meals to the system, all the users of the system shall be able to view added meals and book their meals according. | Home page of the Hosteller /Guest |
| FR4 | Mess Handler shall be able to view total number of meals booked. | Booked Meals Details |
| FR5 | Mess Handler shall be able to view order history of Hostellers/Guests. | Order History of Meals Details |
| FR6 | Hostellers/Guests shall be able to book meal before 2 hours of mess opening time. | Add Booking Page |
| FR7 | Hostellers shall be able to book their meals as per their convenience according to the time slots provided for dinner. | Add Booking Page |

**Manage Order History**

|  |  |  |
| --- | --- | --- |
| RN | Description | Comments |
| FR1 | System shall be able to generate order history. | Order History of Meals Details |
| FR2 | Mess Handler shall be able to view order history. | Order History of Meals Details |
| FR3 | Hostellers/Guest shall be able to view their own booked meals. | Booked History Page |

**Manage Payment**

|  |  |  |
| --- | --- | --- |
| RN | Description | Comments |
| FR1 | After the generation of bill, the guests shall be able to make payment using one of the online payment gateways (compulsory in online order). | Payment Gateway Page |
| FR2 | The system shall be able to assign a unique id (payment id) to each guest’s order bill. | Payment Page |

**Generate Bill**

|  |  |  |
| --- | --- | --- |
| RN | Description | Comments |
| FR1 | After the confirmation of order placed by the guests the system shall be able to generate bill with all the necessary details of the Guests. | Guest Booking Page |
| FR2 | The system will assign a unique id (bill no) to each guest’s order bill. | Guest Booking Page |

**Manage Reports**

|  |  |  |
| --- | --- | --- |
| RN | Description | Comments |
| FR1 | The system should be able to generate reports like:  i. Day wise Booking Report  ii. All Bookings Report  iii. Hosteller/Guest Report | Admin Report Dashboard |

**Manage Feedback**

|  |  |  |
| --- | --- | --- |
| RN | Description | Comments |
| FR1 | Hostellers/Guests shall also be able to give feedback for the mess food. | Give Feedback Page |
| FR2 | The system shall be able to assign a unique id (feedback id) to each feedback. | Give Feedback Page |
| FR3 | The feedback which is given by the Hostellers/Guests shall be able to be viewed by the Admin. | Admin Feedback Dashboard |

* 1. **Non-Functional Requirements:**

|  |  |  |
| --- | --- | --- |
| **RN** | **DESCRIPTION** | **COMMENTS** |
| NFR1 | 1. The system shall be password protected. 2. Only authenticated user shall be able to get access to the system. 3. Password of every user shall be stored in encrypted form. | Security |
| NFR2 | The system shall be easy to use as it shall provide easy navigation, interface and notifications. | Usability |
| NFR3 | The system shall be able to work on android as well as IOS operating system. | Compatibility |
| NFR4 | The system shall help users to reach t their goals i.e., to book meals in very few steps. | Efficiency |
| NF5 | The system shall take minimum response time and throughput for tasks such as:   1. While Hosteller/Guest is placing order. 2. While updating menus in the system. 3. While generating reports. | Performance |

1. **System Design:**
   1. **Database Design:**
      1. **Database Schema:**
2. tblUser (Userid[PK],U\_mail, U\_pass, U\_role, U\_status, contact)

FD -> U\_mail, U\_pass, U\_role, U\_status, contact

1. tblHosteller (Hid[PK],Hfname, Hlname, Hemail, Hcontact, Room\_No, Htype )

FD -> Hfname, Hlname, Hemail, Hcontact, Room\_No, Htype

1. tblGuest (Gid[PK], Gfname, Glname, Gcontact, Gemail)

FD -> Gfname, Glname, Gcontact, Gemail

1. tblHandler (MHid[PK], Mhname, mhlname, mhcontact, mhemail)

FD -> Mhname, mhlname, mhcontact, mhemail

1. tblMeals (Iid(PK), Iname, Icategory, Iprice)

FD -> Iname, Icategory, Iprice

1. tblGorder (Orid[PK],Gid[FK], order\_date, order\_day, quantity, price)

FD -> Gid[FK], order\_date, order\_day, quantity, price

1. tblHbook (Bookid[PK], Hid[FK], book\_date, book\_day, slot)

FD -> Hid[FK], book\_date, book\_day, slot

* + 1. **Data Dictionary:**

1. **tblUser**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| SR No. | Field Name | Datatype | Size | Constraints | Description |
| 1 | User\_id | int | - | Primary Key | Stores Id of particular  user |
| 2 | U\_mail | varchar | 100 | Not Null | Email of particular  user |
| 3 | U\_role | varchar | 50 | Not Null | Role of user |
| 4 | U\_status | int | - | Not Null | Status code  according to the role |
| 5 | U\_pass | varchar | 50 | Not Null | Password of the user |
| 6 | U\_contact | varchar | 10 | Not Null | Contact of the user |

1. **tblHosteller**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| SR No. | Field  Name | Datatype | Size | Constraints | Description |
| 1 | Hid | int | - | Primary Key | Stores Id of hosteller |
| 2 | Hfname | varchar | 20 | Not Null | First name of hosteller |
| 3 | Hlname | varchar | 10 | Not Null | last name of hosteller |
| 4 | Hemail | varchar | 255 | Not Null, Unique | Email of the hosteller |
| 5 | Hcontact | varchar | 50 | Not Null | Contact No of hosteller |
| 6 | Room\_no | varchar | 20 | Not Null | Room number of hosteller |
| 7 | Htype | varchar | 255 | Not Null | Name of Hostel |

1. **tblHandler**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| SR No. | Field Name | Datatype | Size | Constraints | Description |
| 1 | Mhid | int | - | Primary Key | Stores Id of Mess Handler |
| 2 | Mhname | varchar | 100 | Not Null | First Name of Mess Handler |
| 3 | Mhlname | varchar | 50 | Not Null | Last name of Mess Handler |
| 4 | Mhemail | varchar | 255 | Not Null | Email of Mess Handler |
| 5 | Mhcontact | varchar | 50 | Not Null | Contact No of Mess Handler |

1. **tblGuest**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| SR No. | Field Name | Datatype | Size | Constraints | Description |
| 1 | Gid | int | - | Primary Key | Stores Id of Guest |
| 2 | Gfname | varchar | 100 | Not Null | First Name of Guest |
| 3 | Glname | varchar | 50 | Not Null | Last name of Guest |
| 4 | Gemail | varchar | 255 | Not Null | Email of Guest |
| 5 | Gcontact | varchar | 50 | Not Null | Contact No of Guest |

1. **tblMeals**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| SR No. | Field Name | Datatype | Size | Constraints | Description |
| 1 | Iid | int | - | Primary Key | Stores Id of  Item |
| 2 | Iname | varchar | 255 | Not Null | Name of an item |
| 3 | Icategory | varchar | 255 | Not Null | Category of an item |
| 4 | Iprice | Int | - | Not Null | Price of an item |

1. **tblGorder**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| SR No. | Field Name | Datatype | Size | Constraints | Description |
| 1 | Orid | int | - | Primary Key | Stores Id of guest order |
| 2 | Gid | int | - | Foreign Key | Id of guest |
| 3 | Order\_date | varchar | 50 | Not Null | Date of order |
| 4 | Order\_day | varchar | - | Not Null | Day of order |
| 5 | quantity | Int | - | Not Null | Total no of plate order |
| 6 | price | Int | - | Not Null | Total amount paid |

1. **tblHbook**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| SR No. | Field Name | Datatype | Size | Constraints | Description |
| 1 | Bookid | int | - | Primary Key | Stores Id of  Hosteller booking |
| 2 | Hid | int | - | Foreign Key | Id of hosteller |
| 3 | book\_date | varchar | 50 | Not Null | Date of booking |
| 4 | book\_day | varchar | - | Not Null | Day of booking |
| 5 | slot | Int | - | Not Null | Time slot booked |

1. **System Implementation:**
   1. **Use Case Diagram**
   2. **Activity Diagram**

**5.2.1 Activity Diagram for Login**

A diagram of a software system

Description automatically generated

**5.2.2 Activity Diagram for Registration**

**A diagram of a software company

Description automatically generated with medium confidence**

* + 1. **Activity Diagram for Meal Booking:**

**A diagram of a company

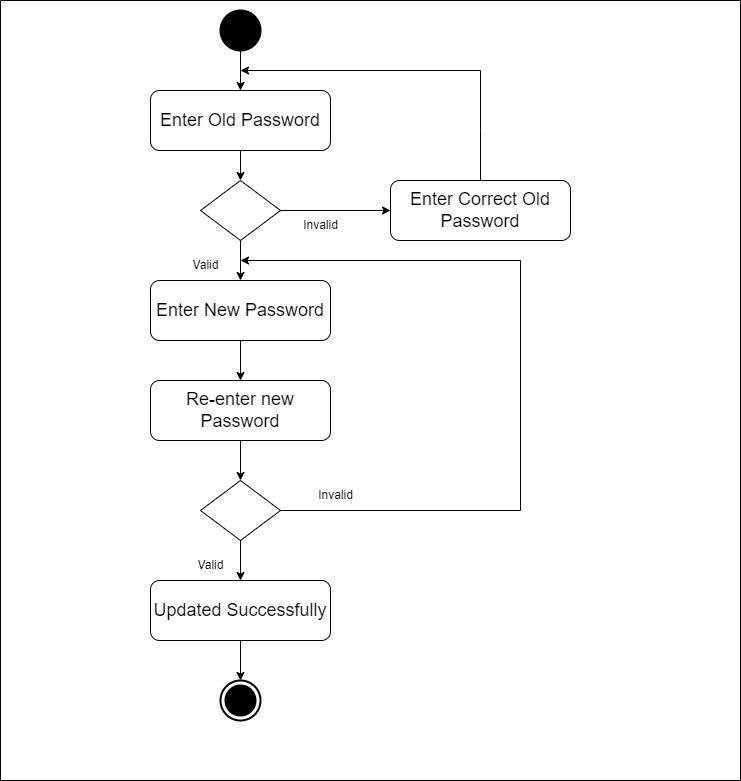
Description automatically generated**

* + 1. **Activity Diagram for Forgot Password:**

A diagram of a software flowchart

Description automatically generated

* + 1. **Activity Diagram for Change Password:**



* + 1. **Activity Diagram for Feedback:**

A diagram of a process

Description automatically generated