

AUTOMATED HOSTEL MESS BILLING SYSTEM

MINI PROJECT REPORT

Submitted by

HASEENA S (PJR23MCA2006)

to

The A P J Abdul Kalam Technological University

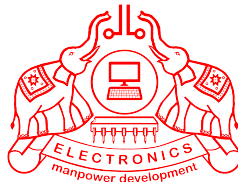


in partial fulfillment of the requirements for the award of the Degree

of

Master of Computer Applications

(MCA)



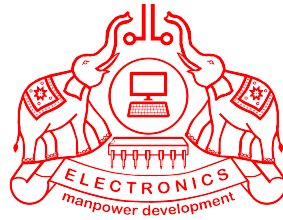
**DEPT. OF COMPUTER SCIENCE AND ENGINEERING
COLLEGE OF ENGINEERING POONJAR (CEP)**

POONJAR-686582

NOVEMBER 2024

College of Engineering Poonjar (CEP)

Poonjar (P.O), Kottayam-686582



CERTIFICATE

This is to certify that Mini Project entitled “**AUTOMATED HOSTEL MESS BILLING SYSTEM**” is a bonafide report of the Mini Project(20MCA245) presented during 3rd semester by **HASEENA S (PJR23MCA2006)** , in partial fulfillment of the requirements for the award of the degree of Master of Computer Applications.

Ms.Josymol Joseph

Project Guide

Ms.Anjana Sekhar

Project Co-ordinator

Dr.Annie Julie Joseph

Head of the Dept.

ACKNOWLEDGMENT

I hereby take the opportunity to thank those entire people who helped me to the completion of this project. First and Foremost, I express my gratitude to the Almighty God for His blessings, guidance, and the completion of this project. I extend my sincere appreciation to my guide Ms. Josymol Joseph, for her unwavering faith in my abilities and invaluable guidance throughout my project. I also wish to convey my heartfelt thanks to all our esteemed faculty members for their profound insights and encouragement. Furthermore, I deeply thankful to my friends for their steadfast support and cooperation. Their encouragement and moral backing have provided strength during challenging times. Lastly, I express my gratitude to all those who have contributed, directly or indirectly, to the successful completion of this project.

ABSTRACT

The Automated Hostel Mess Billing System is designed to address the inefficiencies of traditional hostel management, where mess bills are manually calculated, leading to time-consuming processes and frequent errors. This digital solution streamlines hostel administration by automating key tasks, including billing, leave tracking, and monthly expenditure management. Wardens can manage student admissions, configure billing settings, and automatically generate precise monthly bills that consider individual student attendance and approved leave, eliminating the need for manual mess fee adjustments. For students, the system offers a portal to view their detailed monthly fees, apply for leave, and update profiles, fostering transparency and ease of access. Additionally, the system enables wardens to record hostel expenditures, with descriptions and receipts, supporting organized financial management. By automating these essential functions, the Automated Hostel Mess Billing System enhances operational efficiency, reduces calculation errors, and strengthens communication between students and hostel administration.

Contents

List of Figures	iv
1 INTRODUCTION	1
1.1 Overview	1
1.2 Scope of project	2
1.3 Objectives	3
2 LITERATURE SURVEY	4
2.1 Background Study	5
2.1.1 Existing System	6
2.1.2 Proposed System	6
3 DESIGN	8
3.1 Problem Statement	8
3.2 Algorithm	9
3.3 Flowchart	10
3.4 System Requirements	10
3.4.1 Hardware Requirements	10
3.4.2 Software Requirements	10
3.5 Design Phase	11
3.6 Module Design	11
3.6.1 Warden Module	11
3.6.2 Student Module	12
3.7 Data Flow Diagram	13
3.7.1 Level 0	14
3.7.2 Level 1	15
3.7.3 Level 2 DFD for STUDENT	16
3.7.4 Level 2 DFD for WARDEN	16
3.8 ER Diagram	17
3.9 Database Design	19

4	IMPLEMENTATION	22
5	TESTING	23
5.1	Unit Testing	23
5.2	Integration Testing	24
5.3	System Testing	24
6	RESULTS AND ANALYSIS	26
7	CONCLUSION	27
8	FUTURE WORK	28
	Appendices	30
A	Sample Code	31
B	Screenshots	36

List of Figures

3.1	Flowchart	10
3.2	DFD Symbols	14
3.3	Level 0 DFD	14
3.4	Student Registration	15
3.5	Level 1 DFD	15
3.6	Level 2 DFD for Student	16
3.7	Level 2 DFD for Warden	17
3.8	ER Diagram Symbbols	18
3.9	ER Diagram	18
3.10	CustomUser Table	19
3.11	LeaveApplication Table	20
3.12	HostelBillSettings Table	20
3.13	HostelBillDetails Table	21
3.14	Expenditure Table	21
B.1	Login Page	36
B.2	Student Registration	37
B.3	Warden Dashboard	37
B.4	Active Students List	38
B.5	Mess fee List	38
B.6	Student Dashboard	38
B.7	Edit Profile	39
B.8	Apply for Leave	39
B.9	View Mess fee	39

Chapter 1

INTRODUCTION

1.1 Overview

The Automated Hostel Mess Billing System is an integrated software solution that streamlines hostel and mess administration by automating billing processes, leave management, student admissions, and financial tracking. In traditional hostel setups, mess bills are often calculated manually a task that is time-intensive and prone to errors, especially when adjustments for student leave need to be factored in. This project addresses these inefficiencies by enabling wardens to set up billing rules, manage hostel charges, and automatically generate monthly bills that reflect each student's attendance and approved leave, ensuring precise and transparent billing. The system also provides a dedicated portal for students, where they can view their monthly bills, apply for leave, and update their personal information. This improves communication and transparency, as students have continuous access to their financial details and leave status. Additionally, the system includes a module for monthly expenditure tracking, allowing wardens to record expenses, add descriptions, and upload receipts for better financial management. All expenditure data is securely stored, making financial record-keeping more organized and accessible.

By automating routine administrative tasks, the Automated Hostel Mess Billing System not only minimizes manual labor but also reduces calculation errors and centralizes hostel data, creating a more efficient and user-friendly environment for both students and hostel staff. This project ultimately enhances operational accuracy, saves time, and improves the overall hostel experience for everyone involved. It brings a digital transformation to hostel management by integrating billing and record-keeping into a single platform. Additionally, real-time access to information fosters a transparent environment, empowering both students and staff.

1.2 Scope of project

This System encompasses several key areas within hostel administration, aimed at enhancing efficiency, accuracy, and user experience for both hostel staff and students. The systems primary functionality revolves around automating mess billing by accurately calculating fees based on student attendance and approved leave. This removes the need for time-consuming, error-prone manual calculations and ensures that each students bill reflects only the actual services used.

In addition to billing, the project includes a comprehensive student management module that allows wardens to handle admissions, track active student records, and approve or reject new student applications. Through this module, wardens can keep up-to-date records of current residents, ensuring accurate billing and administrative management.

The systems leave management feature further supports billing automation by integrating approved leave records directly into the billing calculation, so mess fees can be automatically adjusted based on the leave duration. This provides a streamlined approach for both students and wardens, as students can apply for leave and monitor their status, while wardens can approve requests with ease.

A financial tracking module also allows wardens to log monthly hostel expenditures, complete with descriptions and digital uploads of receipts, creating a reliable record of hostel expenses. This functionality aids in budget planning, auditing, and expense monitoring for the hostel management.

For students, the system offers a self-service portal to view their billing details, apply for leave, and update their profiles. This ensures transparency and enhances user satisfaction, as students have real-time access to financial and administrative information related to their hostel stay.

Overall, the scope of this project covers essential aspects of hostel and mess management, aiming to automate and centralize various administrative tasks. This enables improved operational efficiency, greater accuracy in billing, enhanced financial transparency, and a better experience for both students and hostel staff.

1.3 Objectives

- The objective of our project is to automatically calculate and generate accurate mess bills based on student attendance and leave status.
- Efficiently manage student admissions and track active students, reducing manual data entry.
- Integrate leave approval and tracking to automatically adjust mess fees based on student leave.
- Provide students with real-time access to their bills and leave details for better transparency
- Record and manage hostel expenditures in an organized manner for better financial control.
- Automate repetitive administrative tasks to save time and reduce the workload for hostel staff.
- Improve overall efficiency by automating key processes and reducing manual errors in hostel management.

Chapter 2

LITERATURE SURVEY

The literature survey explores existing research and systems related to hostel management, billing automation, and leave management, identifying key challenges and technological advancements in these areas. Traditional hostel management systems often rely on manual processes for student admissions, billing, leave tracking, and expenditure management, leading to inefficiencies, errors, and high administrative workload. Studies have shown that automating mess billing can significantly reduce errors and the time required for calculations, especially when adjusting for student leave. Research in the area of leave management indicates that integrating leave records with billing systems provides more accurate and transparent calculations, eliminating the need for manual adjustments. Additionally, the integration of financial tracking systems in hostel management has been widely recognized for improving budgeting, expense monitoring, and audit trails. Systems that allow students to access their billing details and update profiles have been shown to enhance communication and transparency. Several existing software solutions, such as student management and financial software, have incorporated these features, but challenges remain in integrating all aspects of hostel management into a cohesive, user-friendly system.

The literature highlights the need for an all-in-one solution that automates key administrative functions, reduces manual errors, and provides a seamless experience for both hostel staff and students, which is the core goal of the Automated Hostel Mess Billing System. Furthermore, recent advancements in digital record-keeping and cloud storage underscore the benefits of centralized, easily accessible data in improving operational efficiency. By consolidating billing, leave, and financial tracking under a single platform, the system offers a modern approach to hostel administration that aligns with these emerging best practices. This integrated model serves as a forward-looking solution to address ongoing challenges in hostel management.

2.1 Background Study

The Automated Hostel Mess Billing System is developed to address the challenges and inefficiencies in traditional hostel management systems. In conventional setups, hostel management often relies on manual processes for calculating mess bills, tracking student attendance, and managing leave records, leading to increased chances of errors and delays. Mess billing, in particular, requires frequent adjustments based on the number of days a student attends or stays absent, which can be complex and time-consuming when handled manually. This often results in discrepancies in fee calculation and increased administrative workload. Additionally, student admission management, leave requests, and expenditure tracking are often kept in separate systems or spreadsheets, making it difficult to centralize and streamline operations. As a result, hostel management can become disorganized, leading to inefficiencies, lack of transparency, and poor communication between hostel staff and students.

The growing adoption of digital tools has made it possible to automate many of these tasks. Previous studies and existing software solutions have shown that automating mess billing, leave management, and expenditure tracking can significantly reduce human error, save time, and improve operational efficiency. By integrating all functions into a single platform, it becomes easier to manage student records, apply leave, adjust mess bills based on absences, and track monthly expenditures. Moreover, providing students with access to their billing details and leave status promotes transparency and allows for a more user-friendly experience. This project builds on these advancements by offering a comprehensive solution that centralizes hostel management, reduces the administrative burden, and improves accuracy. The system's ability to automate key tasks and provide real-time data access to both hostel staff and students ensures an efficient and transparent process, enhancing the overall hostel experience for everyone involved.

Additionally, the system supports secure, long-term record-keeping, which is essential for auditing and financial accountability. By addressing both current challenges and future administrative needs, the project sets a robust foundation for scalable hostel management in the digital age.

2.1.1 Existing System

In existing hostel management systems, many processes, such as mess billing, student admissions, leave tracking, and financial management, are handled manually or through basic digital tools. Mess bills are calculated based on attendance, and adjustments are made manually for student leave. Student admissions and records are typically maintained in spreadsheets or basic databases, and leave applications are processed through paper forms or informal systems. Expenditures are tracked using manual entries or separate software tools, making it difficult to integrate all functions into a single, cohesive system.

Disadvantages

- Human Errors in Calculation
- Time-Consuming Process
- Difficulty in Handling Bulk Data
- Complex Leave-based Billing Adjustments
- Inefficient Student Management
- Difficulty in Tracking Monthly Expenditures
- Poor Transparency and Limited Access for Students
- High Risk of Data Loss or Misplacement

2.1.2 Proposed System

The Automated Hostel Mess Billing System is designed to streamline hostel administration by automating key functions such as student admissions, mess billing, leave management, and expenditure tracking. The system will automatically calculate mess fees based on student attendance and approved leave, minimizing errors and saving time. It also provides a self-service portal for students to view bills, apply for leave, and update profiles, promoting transparency and ease of access. By centralizing all data, the system ensures that hostel staff can manage tasks more efficiently and maintain accurate, up-to-date records.

Advantages

- **Error-Free Billing** : Automates mess bill calculations based on attendance and leave, reducing errors from manual calculations.
- **Time Efficiency** : Saves time by automating routine tasks such as leave tracking, bill generation, and expenditure logging.
- **Data Integration** : Centralizes student records, billing, leave, and expenditure data in one system for consistent and organized record-keeping.
- **Improved Transparency** : Allows students to access their billing details, leave status, and other personal records directly, fostering transparency.
- **Reduced Administrative Workload** : Automates repetitive administrative tasks, freeing hostel staff to focus on other important responsibilities.
- **Enhanced Data Security and Accuracy** : Securely stores data and minimizes the risk of data loss or duplication, ensuring more accurate records.
- **Financial Tracking** : Provides organized tracking of hostel expenses, supporting budget management and better financial oversight.

Chapter 3

DESIGN

The Automated Hostel Mess Billing System consists of three main components: an Admin Module for wardens, a Student Module for residents, and a Financial Management Module for expenditure tracking. The Admin Module is accessible only to authorized wardens through secure login credentials, allowing them to manage student admissions, configure mess billing, track leave requests, and record monthly expenditures. The Student Module enables students to log in, view their bills, update profiles, and apply for leave, with approved leave directly affecting their billing calculations. Additionally, students can provide feedback on services received, visible to the warden for quality assurance. Finally, the Financial Management Module allows wardens to record and monitor monthly hostel expenses, providing detailed expense tracking and financial transparency. The system ensures data security and streamlines hostel operations, making management efficient and accessible.

3.1 Problem Statement

The purpose of the project is to managing hostel operations manually poses significant challenges, as tasks like mess billing, leave tracking, and expense recording are prone to errors and consume a considerable amount of administrative time. Calculating mess bills manually, especially when adjusting for students' leave, often leads to inaccuracies and inconsistencies in billing. Furthermore, students lack easy access to their billing information and leave statuses, resulting in limited transparency and frequent communication issues with hostel administration. Additionally, the absence of an integrated system for managing admissions, billing, leave applications, and financial records makes it difficult to ensure accuracy and security across hostel data. This project aims to provide an automated solution that centralizes these processes, enhancing efficiency, reducing errors, and fostering transparency and organization in hostel management.

3.2 Algorithm

Steps

- 1 Start.
- 2 Access the Automated Hostel Mess Billing System website.
- 3 Login to the system.
 - 3.1 Login as Warden (Admin).
 - 3.2 Login as Student (Only if Warden has accepted the registration).
- 4 If Student:
 - 4.1 View current mess bill and billing history.
 - 4.2 Apply for leave (leave status will be updated in billing).
 - 4.3 Update profile details.
- 5 If Warden (Admin):
 - 5.1 Add new students to the database.
 - 5.2 Review and approve/reject student registrations.
 - 5.3 Once a student is accepted, they can log in to the system.
 - 5.4 Configure and update mess billing rates and settings.
 - 5.5 View and approve or reject student leave requests.
 - 5.6 Record hostel expenditures for monthly tracking.
 - 5.7 Update student payment status
- 6 End.

3.3 Flowchart

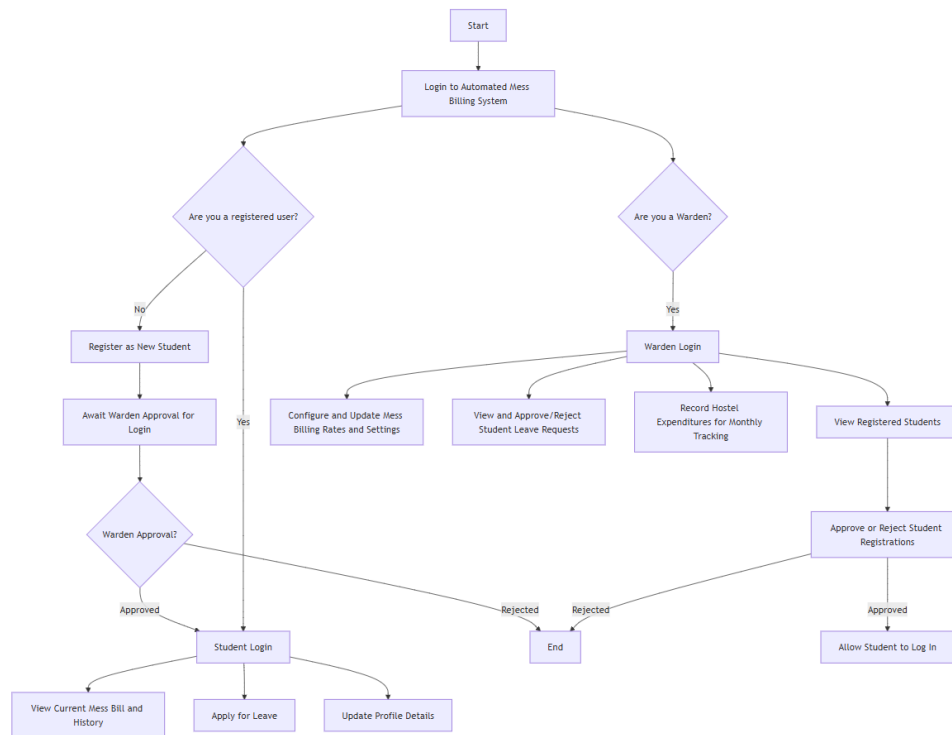


Figure 3.1: Flowchart

3.4 System Requirements

3.4.1 Hardware Requirements

Processor: Intel Core i3 or higher

RAM: 4 GB or more

Hard Disk: 250 GB HDD or SSD

Monitor : 15 inches or larger, supporting a resolution of at least 1280x720

3.4.2 Software Requirements

Front End-HTML, CSS, Javascript

Back End-Django

IDE-Visual Studio Code

Database-SQLite

3.5 Design Phase

The Automated Hostel Mess Billing System is built using a combination of modern technologies for efficiency and scalability. The system use SQLite to manage student records, mess bills, leave applications, and expenditures. The front-end is developed with HTML, CSS, and JavaScript for a responsive user interface, while the back-end is powered by Django, ensuring secure user authentication and smooth functionality.

3.6 Module Design

3.6.1 Warden Module

The Warden Module in the Automated Hostel Mess Billing System is designed to provide hostel administrators (wardens) with the tools necessary for managing hostel operations efficiently. The module allows the warden to perform various administrative tasks related to student management, billing, leave approvals, and expenditure tracking. **Functionalities:**

- **View Registered Students :** The warden can view a list of all registered students, including their personal and course details. This helps in keeping track of all students in the hostel.
- **Accept/Reject Registered Students :** The warden has the authority to approve or deny student registrations by verifying whether the student is a registered student of the college or not. Once accepted, students can log in to the system.
- **Manage Hostel Bill Settings :** The warden can update various billing parameters, such as rent, employee salaries, electricity charges, and other hostel-related expenses. This ensures that the billing system is aligned with the latest hostel policies and reflects any changes in operational costs.
- **Approve/Reject Student Leave :** The warden can approve or reject student leave applications, adjusting the mess bill accordingly. Leave approval affects the students billing, as they may be exempt from mess charges during their leave period.

- **Generate Mess Fee :** Based on attendance, leave applications, and configured settings, the warden generates the monthly mess fees for each student, adjusting for any absences or leave taken during the billing cycle.
- **Add Expenditure Details :** The warden can record monthly hostel expenditures (such as food, maintenance, etc.) into the system. This helps in tracking the overall financial health of the hostel and aids in budget planning.
- **Update Payment Status :** The warden can update the payment status of students' mess bills (e.g., paid, pending, overdue) to ensure accurate financial records and follow-up actions where necessary.
- **View Active Students :** The warden can view a list of currently residing students who are actively using the hostel services. This helps in monitoring the number of active students and their respective billing.
- **Delete Active Students :** The warden can remove students from the system who are no longer residing in the hostel, either because they have left or vacated the hostel. This ensures that only active students are billed for mess services.
- **View Vacated Students :** The warden can view a list of students who have vacated the hostel, helping in maintaining an updated record of past residents and closing out their accounts for final billing.

3.6.2 Student Module

The Student Module in the Automated Hostel Mess Billing System is designed to allow students to manage their hostel-related tasks, including viewing and paying mess fees, applying for leave, and updating their personal information. This module provides students with an interactive and transparent platform for easy access to their billing details, leave status, and profile management. **Functionalities:**

- **Register :** Students can create a new account on the system by providing necessary personal and hostel-related details. Registration must be approved by the warden before the student can log in and access the system.

- Login : After registration, students can log in to the system using their email id and password. This gives them access to their personalized dashboard, where they can manage their details and request services.
- Assign Leave : Students can apply for leave by submitting a leave request with the start and end dates. The warden will approve or reject the leave request, and the system will adjust the students mess bill accordingly.
- View Mess Fee : Students can view their current mess fee, which includes the charges for the month, and any adjustments based on their attendance or leave. This provides students with transparency about their financial obligations.
- Add Payment Details : Students can input and update their payment details, such as making a payment for the mess fee. This allows for easy tracking of financial transactions and ensures that payments are properly recorded.
- View Profile : Students can access their personal profile, which includes their contact details, course details, and mess billing history. This gives them a quick overview of their status and financial records.
- Update Profile : Students can update their profile information, such as changing their contact details or course details. This ensures that the system reflects the most current details about the student.

3.7 Data Flow Diagram

A data flow diagram (DFD) maps out the flow of information for any process or system. It uses defined symbols like rectangles, circles and arrows, plus short text labels, to show data inputs, outputs, storage points and the routes between each destination. Data flowcharts can range from simple, even hand-drawn process overviews, to in-depth, multi-level DFDs that dig progressively deeper into how the data is handled. They can be used to analyze an existing system or model a new one. Like all the best diagrams and charts, a DFD can often visually say things that would be hard to explain in words, and they work for both technical and nontechnical audiences, from developer to CEO. That's why DFDs remain so popular after all these years. While they work well for data flow software and

systems, they are less applicable nowadays to visualizing interactive, real-time or database-oriented software or systems.

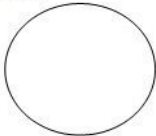



Symbols	Description
<u>Process</u> 	A process shows a transformation or manipulation of data flows within the system.
<u>External entity</u> 	External entities are outside the system, but they either supply input data into the system or use system output.
<u>Data flows</u> 	A data flow shows flow of information from source to destination. A data flow is represented by a line, with arrowhead showing the direction of flow.
<u>Data base</u> 	Database is a huge collection of data. It is used for storage purpose.

Figure 3.2: DFD Symbols

3.7.1 Level 0

The below figure 3.3 is the Level 0 DFD of the Automated Hostel Mess Billing System. This level represents the overall working of the project. Here, the users (students and wardens) request various activities, and the system responds accordingly by processing the requests, updating the data, and providing the necessary outputs, such as billing details, leave status, and financial records.

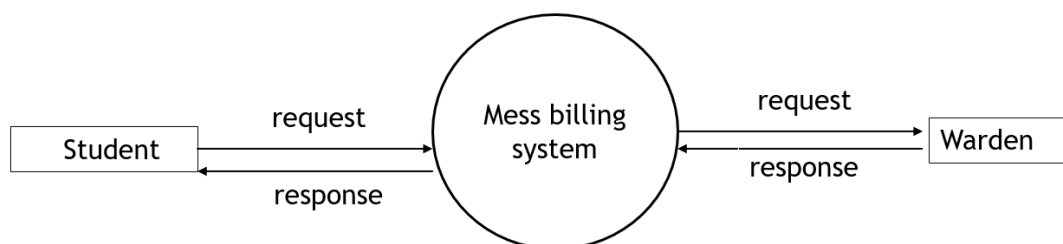


Figure 3.3: Level 0 DFD

3.7.2 Level 1

The below figure 3.4 is the Level 1 DFD for Student registration in the Automated hostel mess billing system. The warden has the authority to approve or deny student registrations by verifying whether the student is a registered student of the college or not. Once accepted, students can log in to the system.

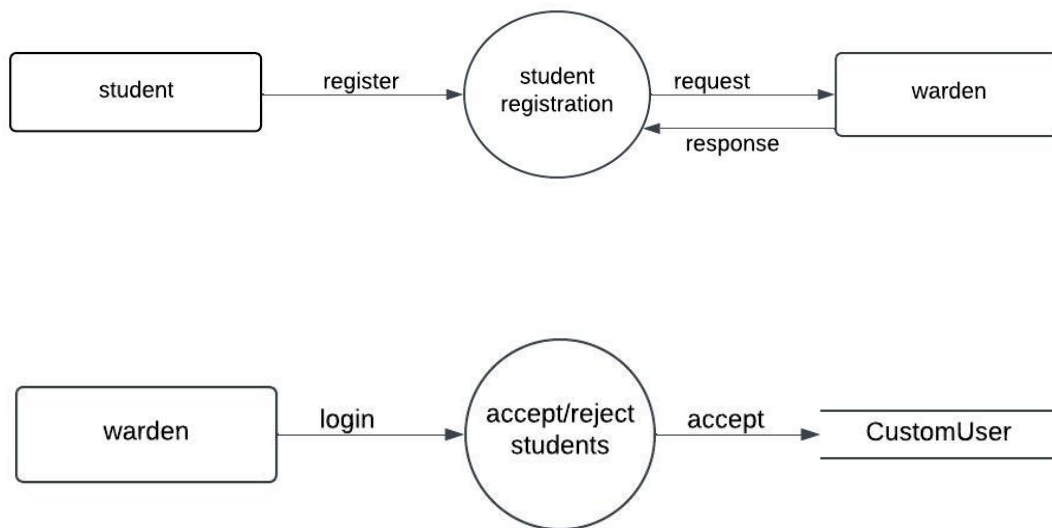


Figure 3.4: Student Registration

The below figure 3.5 is the Level 1 DFD for Automated Hostel Mess billing system.

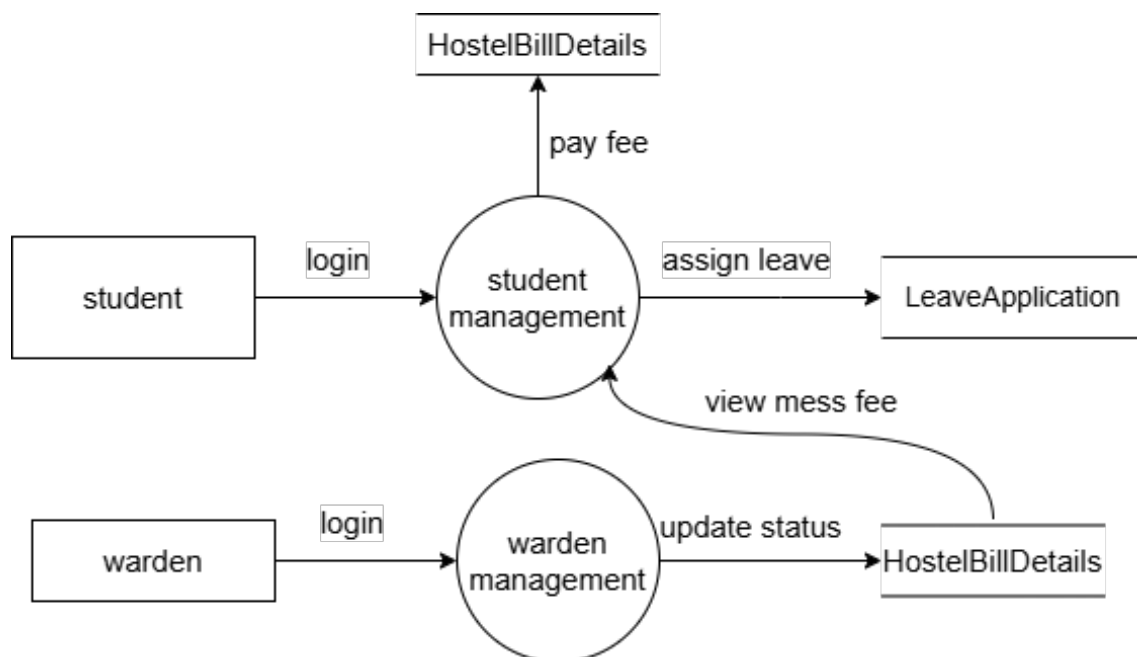


Figure 3.5: Level 1 DFD

3.7.3 Level 2 DFD for STUDENT

The figure 3.6 is the Level 2 DFD for the Student Module in the Automated Hostel Mess Billing System. When a student logs in, the system verifies their credentials, granting access only to approved students. Once logged in, students can submit leave applications for the wardens review, view their mess fee details, and update their profile information. Leave requests are stored in the 'LeaveApplication' table, mess fees in the 'HostelBillDetails' table, and profile updates in the 'CustomUser' table. This setup ensures that all student actions and data are efficiently organized and accessible within the system.

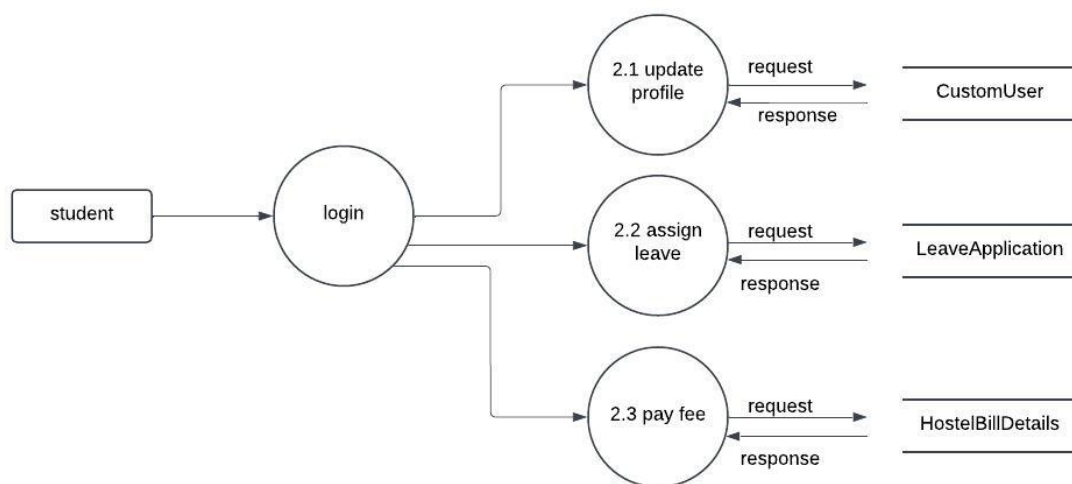


Figure 3.6: Level 2 DFD for Student

3.7.4 Level 2 DFD for WARDEN

The figure 3.7 is the Level 2 DFD for the Warden Module in the Automated Hostel Mess Billing System. This figure illustrates the wardens key functionalities within the system. When the warden logs in, the system verifies their credentials to ensure secure access. The warden can then view and approve or reject student registration requests; approved registrations are stored in the database table 'CustomUser'. Additionally, the warden can review and respond to student leave requests, with each decision being recorded in the 'LeaveApplication' table. The warden also manages hostel bill settings, such as mess fees, by updating relevant data in the 'HostelBillSettings' table. All actions by the warden, including registration approvals, leave decisions, and bill adjustments, are efficiently recorded in the systems designated tables, ensuring accurate tracking and management of

hostel services.

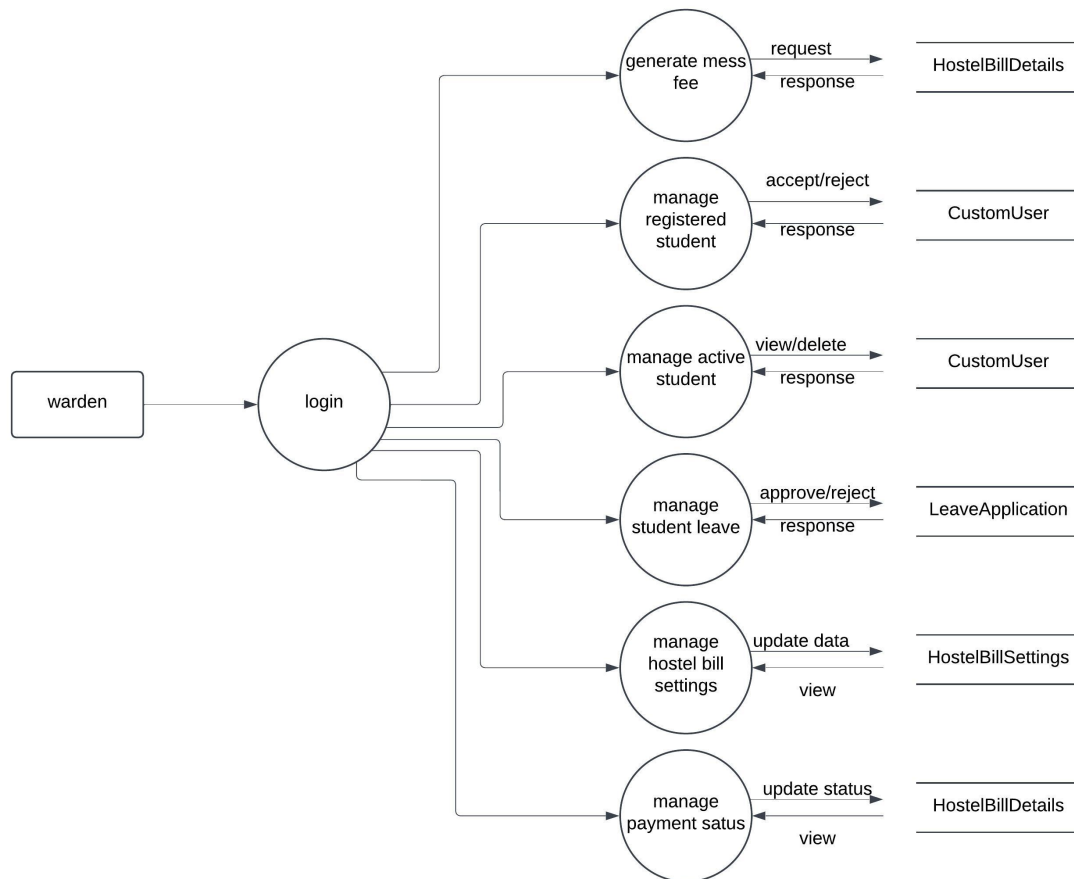


Figure 3.7: Level 2 DFD for Warden

3.8 ER Diagram

An Entity-Relationship (ER) Diagram is a graphical tool used in database design to illustrate and organize the relationships between entities within a system. Entities represent objects or concepts, such as "Student," "Course," or "Order," and are visualized as rectangles. Attributes, which describe properties of each entity, are connected to entities with ovals, defining unique features like "Student ID" or "Name." Relationships between entities are depicted with diamonds, showing how different entities, like students and courses, interact within the system. Cardinality notations, such as one-to-one or one-to-many, are often included on connecting lines to specify the nature of these relationships. ER diagrams help developers create a clear and structured layout of a database by providing a blueprint that maps out all entities, their attributes, and relationships, allowing for efficient organization, retrieval, and management of data.

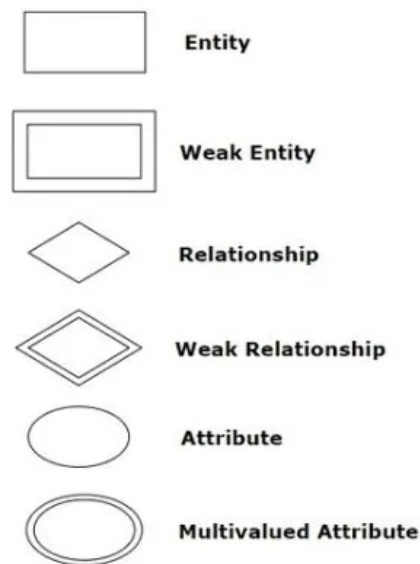


Figure 3.8: ER Diagram Symbols

The below figure 3.9 is the ER Diagram for Automated Hostel Mess billing system .

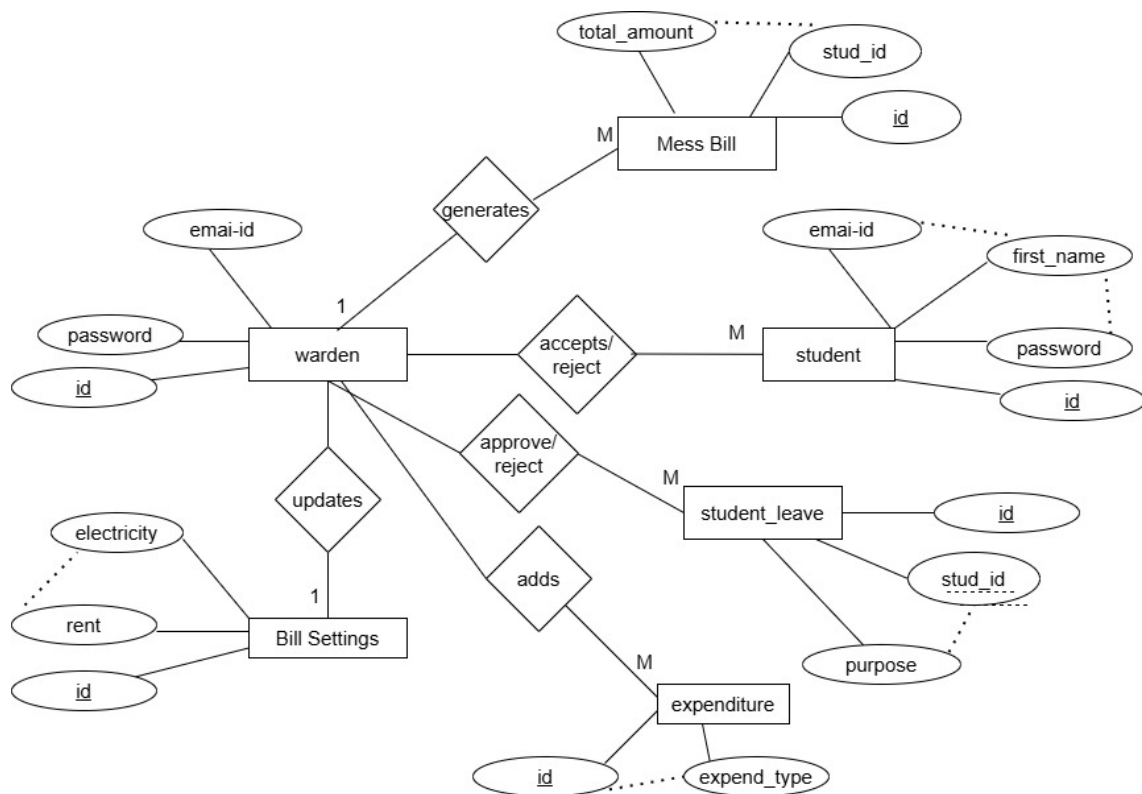


Figure 3.9: ER Diagram

3.9 Database Design

Database design is the process of structuring a database to store, manage, and retrieve data efficiently. It involves defining tables, fields, relationships, and constraints to ensure data is organized logically and can support application requirements. A well-designed database minimizes redundancy, maintains data integrity, and optimizes query performance. The design process includes creating an Entity-Relationship (ER) diagram to visualize entities, their attributes, and relationships, helping developers understand data flow and plan for scalable and maintainable storage solutions.

The below figure 3.10 is the CustomUser table contains detailed information about students in the system. It includes attributes such as the student's unique ID, name, contact information, registration status, and login credentials.

Column Name	Data Type	Constraints
id	INTEGER	PRIMARY KEY, Auto-increment
password	VARCHAR(128)	NOT NULL
username	VARCHAR(150)	UNIQUE, NOT NULL
first_name	VARCHAR(150)	
last_name	VARCHAR(150)	
email	VARCHAR(254)	UNIQUE, NOT NULL
dob	DATE	
gender	CHAR(1)	CHECK (gender IN ('M', 'F'))
address	TEXT	
course	VARCHAR(4)	CHECK (course IN ('CSE', 'EC', 'EEE', 'CHE', 'EL', 'DEEE', 'MCA'))
reg_no	VARCHAR(15)	UNIQUE, NOT NULL
sem	INTEGER	CHECK (sem >= 0)
start_year	INTEGER	
end_year	INTEGER	
ph_no	VARCHAR(15)	
guard_name	VARCHAR(100)	
guard_phn	VARCHAR(15)	
is_active	BOOLEAN	DEFAULT FALSE
past_stud	BOOLEAN	DEFAULT FALSE

Figure 3.10: CustomUser Table

The below figure 3.11 is the LeaveApplication table, which contains details about student leave requests. It includes attributes such as the student's unique ID, leave start and end dates, purpose of leave, a description of the leave, the leave status (approved or rejected), and the date the leave was applied.

Column Name	Data Type	Constraints
id	SERIAL	PRIMARY KEY
user_id	INTEGER	FOREIGN KEY (references CustomUser), NOT NULL, ON DELETE CASCADE
purpose	VARCHAR(255)	NOT NULL
description	TEXT	NOT NULL
status	VARCHAR(10)	CHECK (status IN ('pending', 'approved', 'rejected')), DEFAULT 'pending', NOT NULL
applied_date	TIMESTAMP WITH TIME ZONE	NOT NULL, DEFAULT CURRENT_TIMESTAMP
start_date	DATE	NOT NULL
end_date	DATE	NOT NULL

Figure 3.11: LeaveApplication Table

The below figure 3.12 is the HostelBillSettings table, which contains details about the billing settings for the hostel. It includes attributes such as mess rates, rent, employee salaries, electricity charges, and other related billing parameters. The data in this table is crucial for generating accurate hostel bills for students based on the applicable charges.

Column Name	Data Type	Constraints
id	INTEGER	PRIMARY KEY, Auto-increment
perDay_fixedCharge	DECIMAL(10, 2)	NOT NULL
rent	DECIMAL(10, 2)	NOT NULL
electricity	DECIMAL(10, 2)	NOT NULL
broadband	DECIMAL(10, 2)	NOT NULL
maintenance	DECIMAL(10, 2)	NOT NULL
emp_salary	DECIMAL(10, 2)	NOT NULL

Figure 3.12: HostelBillSettings Table

The below figure 3.13 is the HostelBillDetails table, which contains detailed information about the individual hostel bills generated for students. It includes

attributes such as the student's unique ID, bill amount, billing period, and the status of the bill (paid or pending).

Column Name	Data Type	Constraints
id	INTEGER	PRIMARY KEY, Auto-increment
student_id	INTEGER	FOREIGN KEY (references CustomUser), NOT NULL, ON DELETE CASCADE
total_amount	DECIMAL(10, 2)	NOT NULL
bill_details	TEXT	NOT NULL
generated_date	TIMESTAMP WITH TIME ZONE	NOT NULL, DEFAULT CURRENT_TIMESTAMP
billing_month	DATE	NOT NULL
electricity	INTEGER	NOT NULL
broadband	INTEGER	NOT NULL
maintenance	INTEGER	NOT NULL
total	INTEGER	NOT NULL
paid_amount	DECIMAL(10, 2)	DEFAULT 0.00, NOT NULL
transactionID	TEXT	NOT NULL
status	VARCHAR(20)	DEFAULT 'unpaid', NOT NULL
due	DECIMAL(10, 2)	DEFAULT 0.00, NOT NULL

Figure 3.13: HostelBillDetails Table

The below figure 3.14 is the Expenditure table, which stores detailed records of the hostel's expenses. It includes attributes such as the expenditure date, type of expenditure (e.g., maintenance, utilities, etc.), description of the expenditure, the amount spent, and the associated bill image (if applicable).

Column Name	Data Type	Constraints
id	INTEGER	PRIMARY KEY, Auto-increment
expenditure_date	DATE	NOT NULL
expenditure_type	VARCHAR(20)	NOT NULL
description	TEXT	NOT NULL
amount	DECIMAL(10, 2)	NOT NULL
bill_image	IMAGE	NULLABLE
created_at	TIMESTAMP WITH TIME ZONE	NOT NULL, DEFAULT CURRENT_TIMESTAMP

Figure 3.14: Expenditure Table

Chapter 4

IMPLEMENTATION

The Automated Hostel Mess Billing System leverages Django's model-view-template (MVT) architecture to provide a comprehensive solution for hostel management. The system's core functionality revolves around well-structured models, such as Student, Leave, Bill, and Expenditure, which handle key aspects of hostel operations. The Student model tracks personal details and associations with mess billing and leave records, while the Leave model manages student leave requests, including approval status and dates. The Bill model automatically generates and updates monthly bills based on the students attendance and approved leave, ensuring that mess charges are accurately calculated. Additionally, the Expenditure model allows wardens to record and track hostel-related expenses for better financial management. These models are interconnected through Django's Object-Relational Mapping (ORM), ensuring efficient database operations and seamless data management.

The views in Django handle business logic, such as registering students, applying leave, and generating bills. For instance, when a student applies for leave, the system adjusts their monthly mess bill based on the approved leave days. The Warden module provides functionality for managing student registrations, leave approvals, and expenditures. Through Django's templating engine, the system generates user-friendly interfaces that are role-specific: students have a personalized dashboard to view bills, apply for leave, and manage their profiles, while wardens and admins have dashboards for monitoring and managing hostel operations. The seamless integration of these modules ensures that the system is not only efficient but also transparent, reducing manual errors and improving overall hostel management efficiency.

Chapter 5

TESTING

Testing in an Automated Hostel Mess Billing System is essential to verify that each module Warden and Student functions accurately according to requirements. Given the system's responsibilities, such as billing management, student leave tracking, and profile handling, a robust testing strategy is applied across Unit Testing, Integration Testing, Functional Testing, and User Acceptance Testing (UAT). Using Django's built-in testing framework, each module undergoes thorough testing to ensure reliable calculations, secure data handling, and smooth user experiences, contributing to an efficient and user-friendly billing process.

5.1 Unit Testing

Unit Testing focuses on validating the smallest units of functionality such as functions, methods, or models by testing them independently. In the Automated Hostel Mess Billing System, unit tests ensure that logic within each module functions as expected without reliance on other components. For example:

- **Models:** Unit tests validate models like `HostelBillDetail`, `LeaveApplication`, and `Expenditure`, checking that field constraints (e.g., unique registration numbers, correct status updates) are applied accurately. Tests also confirm that methods within these models, like calculating total bills or tracking payment statuses, produce the correct outputs.
- **Forms:** Form validation is essential, especially for student registrations, leave applications, and billing submissions. Unit tests verify that forms enforce required fields, ensure valid dates, and prevent duplicate records, providing accurate feedback to users.
- **Views and Utilities:** Unit tests for view functions check that each view renders the appropriate templates and displays messages or redirects users based

on conditions such as payment completion or leave approval. This testing ensures a consistent user experience and accurate system functionality.

5.2 Integration Testing

Integration Testing ensures that various components in the Automated Hostel Mess Billing System function together seamlessly, allowing smooth data flow and communication between models, views, and forms. Examples include:

- **Student Registration and Approval Process:** Integration tests verify interactions between the registration form, the wardens approval view, and student login access. When a student registers, integration testing ensures that their profile remains pending until the warden approves, after which they gain full access to system features.
- **Billing Calculation and Payment Status:** Integration tests for `HostelBillSettings`, `HostelBillDetail`, and payment views confirm that any updates to settings (such as rate changes) are accurately reflected in each students monthly bill. Tests ensure that partial payments, pending balances, and status updates (e.g., paid, unpaid) synchronize correctly throughout the billing and payment modules.
- **Leave Application and Bill Adjustments:** Integration tests verify that when students apply for leave and the leave is approved, the billing system adjusts the final amount accordingly. These tests ensure accurate communication between the leave model, approval views, and billing system, providing students with precise billing details.

5.3 System Testing

System Testing for the Automated Hostel Mess Billing System evaluates the application as a whole, ensuring that all components and functionalities work together seamlessly in a realistic environment. This testing phase validates complete workflows, focusing on interactions between modules, data flow, security, and usability for a cohesive experience across user roles.

- **End-to-End Workflow Validation:** System tests simulate a complete billing cycle from warden configuration of mess billing rates and expenditure track-

ing to student billing view and payment. These tests verify that settings entered by the warden are accurately reflected in students' bills and that payments update billing statuses and due amounts correctly.

- **Role-Based Access Control:** Tests confirm that each user role (warden and student) has access only to their designated functions. The warden should access administrative features like registration approvals, bill settings, and expenditure tracking, while students should view billing and apply for leave. Unauthorized access attempts are blocked, ensuring security.
- **Data Consistency Across Modules:** System tests verify data integrity across the modules, ensuring that leave applications impact billing accurately, expenditure records update financial reports correctly, and warden-approved student registrations enable system login. This ensures that workflows like leave applications flow smoothly without data inconsistencies.
- **User Interface and Usability:** Testing ensures that the UI provides a user-friendly experience with clear navigation for both warden and student users. The warden should easily navigate between settings, approval queues, and expenditure reports, while students should intuitively find billing and leave applications.
- **Performance and Load Testing:** System tests also include performance tests to assess how the application handles concurrent users, especially during peak times like billing cycles. This ensures smooth performance when multiple students check their bills or apply for leave simultaneously.
- **Security Testing:** Testing verifies security protocols, such as encrypted login credentials, data privacy, and secure access to sensitive financial and user data. Tests ensure that only authorized users access specific data and that stored sensitive data remains protected from unauthorized access.

Chapter 6

RESULTS AND ANALYSIS

The Automated Hostel Mess Billing System has been designed with efficiency and performance in mind, ensuring that the system can handle large volumes of data while maintaining quick response times. The time complexity of various operations has been optimized to minimize delays and ensure smooth user experience, especially during bill generation and leave status checks. Database queries are optimized using Django's query optimization techniques, and indexes have been applied to frequently accessed fields such as student IDs and billing records. This ensures that even with a large number of students and records, the system can function without performance bottlenecks.

Comprehensive testing has been conducted throughout the development process to ensure the reliability and accuracy of the system. Unit tests have been written for individual components such as models, forms, and views, verifying that the logic within each module functions correctly in isolation. Integration testing has confirmed that the interaction between different components, such as student registrations, leave management, and billing, works seamlessly. Functional testing has simulated real-world user scenarios, ensuring that both students and hostel staff can perform their tasks effectively, from registering students to generating monthly bills. Finally, the system has undergone User Acceptance Testing (UAT) to ensure that it meets the expectations of both students and administrative users. This thorough testing approach ensures that the system operates reliably and accurately, reducing errors and improving overall user satisfaction.

Chapter 7

CONCLUSION

In conclusion, the Automated Hostel Mess Billing System not only simplifies the complex tasks of managing hostel services but also provides a high degree of accuracy and transparency in hostel operations. By automating key administrative tasks such as bill calculations, leave management, and expenditure tracking, the system drastically reduces human error, minimizes manual workload, and enhances the overall efficiency of the hostel management process. The intuitive user interfaces for both students and staff ensure that each user can easily navigate and access the information they need, promoting a seamless experience.

The real-time data access and automated billing features foster greater transparency, allowing students to track their financial details and leave status effortlessly, while the warden and admin can make informed decisions based on up-to-date information. Furthermore, the system's adaptability ensures that it can scale as the hostel population grows, and as new requirements emerge, making it a sustainable solution for long-term use.

As the system continues to evolve, future updates could include additional features like automated notifications, more advanced reporting tools, and integration with other campus management systems to further improve operational efficiency. With continuous development, the Automated Hostel Mess Billing System stands as a future-ready solution that offers both immediate benefits and the capacity for future enhancements, ensuring long-term satisfaction and optimal management of hostel services.

Chapter 8

FUTURE WORK

The future work of the Automated Hostel Mess Billing System focuses on further enhancing its functionality, improving user experience, and incorporating additional features to make it more efficient and user-friendly. As the system evolves, several improvements can be made to meet the growing demands of students and administrative staff while addressing emerging challenges.

- **Integration with Payment Gateways:** Incorporating an online payment system for students to pay their mess bills directly through the platform, making it more convenient and reducing manual work.
- **Mobile Application Development:** Developing a mobile app for students and wardens to access the system on the go, increasing accessibility and improving user experience.
- **Real-Time Notifications:** Implementing a notification system to inform students about bill status, leave approvals, and other updates in real-time.
- **Leave Management Enhancements:** Adding features to allow students to track their leave history and request for multiple leave types (e.g., short-term, long-term).
- **Data Analytics and Reporting:** Introducing data analytics to generate reports on hostel expenditures, student payments, and usage patterns to help make informed decisions.
- **Improved User Interface:** Enhancing the system's interface to make it more intuitive and user-friendly for both students and administrators.
- **Security Enhancements:** Implementing advanced security measures to safeguard sensitive data and protect against cyber threats.

Bibliography

- [1] Prof. Deepali Narkhede, Rutuja Bamgude, Mayuri Sonawane, Mandar Shevade. Hostel Management System (HMS) .*International Journal for Research in Applied Science and Engineering Technology*.ISSN:2321-9653
<https://www.ijraset.com/best-journal/hostel-management-system>

- [2] Bhad Nikita Babasaheb,Bankar Prajakta Santosh,Kolhe Sonali Arjun,Gaikwad Gayatri Sukhdev,Bhad Sainath.SMART HOSTEL AUTOMATION. *International Journal of Advance Research and Innovative Ideas in Education*.ISSN:2395-4396

- [3] Kola Ayanlowo¹,O. Shoewu ,Segun O.Olatinwo ,Olusegun O.Omitola , Damilola D.Babalola .”Development of an Automated Hostel Facility Management System ” *Southeast University Journal of Science and Engineering*.
ISSN: 2331-5172

Appendices

Appendix A

Sample Code

"""

Django settings for mess-mate project.

Generated by 'django-admin startproject' using Django 5.1.2.

For more information on this file, see

<https://docs.djangoproject.com/en/5.1/topics/settings/>

For the full list of settings and their values, see

<https://docs.djangoproject.com/en/5.1/ref/settings/>

"""

```
from pathlib import Path
```

```
from django.contrib.messages import constants as messages
```

```
import os
```

```
# Build paths inside the project like this: BASE_DIR / 'subdir'.
```

```
BASE_DIR = Path(__file__).resolve().parent.parent
```

```
# Quick-start development settings - unsuitable for production
```

```
# See https://docs.djangoproject.com/en/5.1/howto/deployment/checklist/
```

```
# SECURITY WARNING: keep the secret key used in production secret!
```

```
SECRET_KEY = 'django-insecure-$41bp!cud2nkvh-sqh(z68vb3^&rq7f
```

```
sgxon2l3684z5%zkiot '
```

```
# SECURITY WARNING: don't run with debug turned on in production!
```

```
DEBUG = True
```

```
ALLOWED_HOSTS = []
```

```
# Application definition
```

```
INSTALLED_APPS = [
```

```
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'mess_app',
    'student',
```

```
]
```

```
MIDDLEWARE = [
```

```
    'django.middleware.security.SecurityMiddleware',
    'django.contrib.sessions.middleware.SessionMiddleware',
    'django.middleware.common.CommonMiddleware',
    'django.middleware.csrf.CsrfViewMiddleware',
    'django.contrib.auth.middleware.AuthenticationMiddleware',
    'django.contrib.messages.middleware.MessageMiddleware',
    'django.middleware.clickjacking.XFrameOptionsMiddleware',
    'django.contrib.messages.middleware.MessageMiddleware',
```

```
]
```

```
ROOT_URLCONF = 'mess_mate.urls'
```

```

TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': ['templates'],
        'APP_DIRS': True,
        'OPTIONS': {
            'context_processors': [
                'django.template.context_processors.debug',
                'django.template.context_processors.request',
                'django.contrib.auth.context_processors.auth',
                'django.contrib.messages.context_processors.messages',
                'mess_app.context_processors.admin_status',
                # Add your context processor here
            ],
        },
    },
]

```

```

WSGIAPPLICATION = 'mess_mate.wsgi.application'

```

Database

<https://docs.djangoproject.com/en/5.1/ref/settings/#databases>

```

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.sqlite3',
        'NAME': BASE_DIR / 'db.sqlite3',
    }
}

```

Password validation

<https://docs.djangoproject.com/en/5.1/ref/settings/>


```
#auth-password-validators
```

```
AUTH_PASSWORD_VALIDATORS = [  
    {  
        'NAME': 'django.contrib.auth.password_validation.  
UserAttributeSimilarityValidator',  
    },  
    {  
        'NAME': 'django.contrib.auth.password_validation.  
MinimumLengthValidator',  
    },  
    {  
        'NAME': 'django.contrib.auth.password_validation.  
CommonPasswordValidator',  
    },  
    {  
        'NAME': 'django.contrib.auth.password_validation.  
NumericPasswordValidator',  
    },  
]
```

```
# Internationalization
```

```
# https://docs.djangoproject.com/en/5.1/topics/i18n/
```

```
LANGUAGE_CODE = 'en-us'
```

```
TIME_ZONE = 'UTC'
```

```
USE_I18N = True
```

```
USE_TZ = True
```

```
# Static files (CSS, JavaScript, Images)
```

```

# https://docs.djangoproject.com/en/5.1/howto/static-files/

STATIC_URL = 'static/'

STATICFILES_DIRS = [
    os.path.join(BASE_DIR, "static"),
    # Folder where your static files are located
]

# Default primary key field type
# https://docs.djangoproject.com/en/5.1/ref/settings/
# default-auto-field

DEFAULT_AUTO_FIELD = 'django.db.models.BigAutoField'

# settings.py

# AUTHENTICATION_BACKENDS = ['mess_app.backends.EmailBackend',
#                             'django.contrib.auth.backends.ModelBackend'
#                             ]

# AUTH_USER_MODEL = 'mess_app.CustomUser'
AUTH_USER_MODEL = 'mess_app.CustomUser'

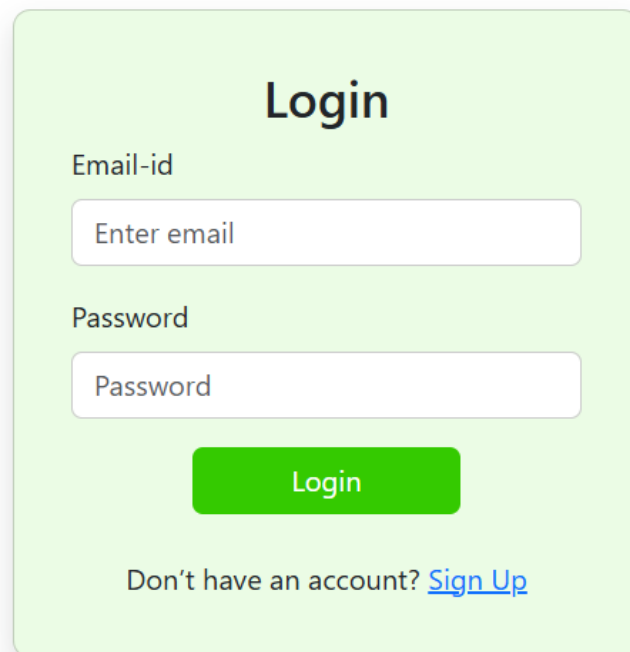
MESSAGE_TAGS = {
    messages.ERROR: 'danger',
}

MEDIA_URL = '/media/'
MEDIA_ROOT = os.path.join(BASE_DIR, 'media')
# The folder where media files will be stored

```

Appendix B

Screenshots



A screenshot of a login form. The form is titled "Login" in bold black text. Below the title, there are two input fields: "Email-id" with a placeholder "Enter email" and "Password" with a placeholder "Password". Both fields are white with a light gray border. Below the password field is a green button with the text "Login" in white. At the bottom of the form, there is a link that says "Don't have an account? [Sign Up](#)". The entire form is set against a light green background with rounded corners and a subtle shadow.

Figure B.1: Login Page

SIGNUP

First Name	Last Name
<input type="text" value="Hari"/>	<input type="text" value="K"/>
Date of Birth	Gender
<input type="text" value="13-11-2000"/>	<input type="text" value="Male"/>
Address	
<input style="width: 100%;" type="text" value="TVM"/>	
Course	
<input type="text" value="MCA"/>	
Register Number	Semester
<input type="text" value="PJR23MCA23"/>	<input type="text" value="3"/>
Class Starting Year	Class Ending Year
<input type="text" value="2023"/>	<input type="text" value="2025"/>
Email	Mobile Number

Figure B.2: Student Registration

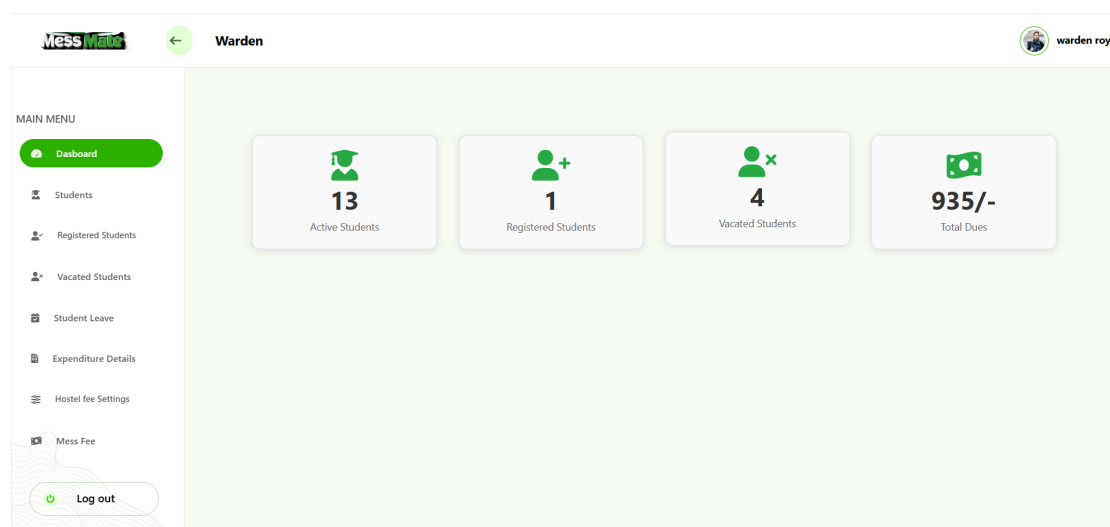


Figure B.3: Warden Dashboard

MESS MENU		warden roy				
MAIN MENU		ACTIVE STUDENTS				
Dashboard	Students	Registered Students	Vacated Students	Student Leave	Expenditure Details	Hostel fee Settings
Mess Fee	Log out					
REGISTER NO	NAME	COURSE	PHONE NUMBER	EMAIL ID	ADDED DATE	
PJR23CHE2334	ANAGHA M	Diploma in Computer Hardware Engineering	6282314531	anagha123@gmail.com	2024-09-03 16:	
PJR23CSE234	DONAL S	B.Tech(Computer Science Engineering)	9745335899	donal@gmail.com	2024-09-02 16:	
PJR23EEE545	DIYAN S	Diploma in Electrical & Electronics Engineering	0123456789	diyan@gmail.com	2024-09-10 05:	
PJR23EEE553	JINSMON ANTONY	Diploma in Electrical & Electronics Engineering	0969696969	jins@gmail.com	2024-09-18 07:	
PJR23BTECH45	SAI K	B.Tech(Electronics & Communication Engineering)	0628231753	sai@gmail.com	2024-09-11 15:1	
PJR23BCE2009	ANOOJA S	B.Tech(Electronics & Communication Engineering)	0123456789	anooja@gmail.com	2024-09-11 05:1	

Figure B.4: Active Students List

MESS MENU		warden roy					
MAIN MENU		MESS FEE					
Dashboard	Students	Registered Students	Vacated Students	Student Leave	Expenditure Details	Hostel fee Settings	Mess Fee
Log out							
REG NO	NAME	COURSE	MONTH	GRAND TOTAL	PAID AMOUNT	DUE	
PJR23MCA234	DRISHYA S	S4 MCA	October 2024	8415.00	0.00	0.00	
PJR23CSE234	DONAL S	S5 CSE	October 2024	8415.00	0.00	0.00	
PJR23EEL8234	ADHI S	S3 EL	October 2024	8415.00	0.00	0.00	
PJR23CHE2334	ANAGHA M	S4 CHE	October 2024	8415.00	0.00	0.00	
PJR23ECE234	ARYA M	S2 EC	October 2024	8415.00	0.00	0.00	
PJR23MCA200	ATHIRA S	S4 MCA	October 2024	7935.00	7000.00	935.00	

Figure B.5: Mess fee List

MESS MENU		Student				Athira s	
MAIN MENU		<div> <div>Rs 7935.00/- Total Amount</div> <div>✓ 7000.00/- Paid Amount</div> <div>! 935.00/- Due</div> <div>i Partially_paid Status</div> </div>					
Dashboard	Leave	Fees	Profile	Log out			

Figure B.6: Student Dashboard

Edit My Profile

First Name: Athira

Guardian Name: venu

Last Name: s

Address: TVM

Email ID: athira@gmail.com

Guardian Phone Number: 6282317512

Phone Number: 0123456789

Semester: 4

Buttons: Cancel, Submit

Figure B.7: Edit Profile

Apply for Leave

Leave Purpose: fever

Start Date: 01-11-2024

End Date: 12-11-2024

Leave Description: fever

Buttons: Cancel, Submit Leave

Figure B.8: Apply for Leave

MESS FEE

MONTH	TOTAL	ELECTRICITY	WIFI	MAINTENANCE	STATUS	DUE
October 2024	7735	50	30	120	partially_paid	935.00

Buttons: Log out

Footer: Show 10 Entries, 1 2 3 4 5

Figure B.9: View Mess fee