Mess Management System

Project Report

1. Introduction

In shared living environments, effective management of shared resources, especially meals and expenses, is crucial for harmonious cohabitation. This report details the development of a user-friendly C++ console application, the Mess Management System, designed to streamline these essential operations. By automating tasks such as member management, expense tracking, meal logging, and bill calculation, this system empowers mess administrators with accurate records, transparent financials, simplifying complex calculations, reducing administrative burden, and promoting a more transparent and efficient management process.

2. Project Goals and Objectives

- **Goal:** Develop a user-friendly and efficient Mess Management System.
- **Objectives:**
  - Implement user authentication (login/registration).
  - Manage member information (add, view, remove, investment).
  - Record expenses (date, description, amount).
  - Record meal consumption per member.
  - Calculate individual member bills based on meal consumption and total expenses.
  - Calculate and display net expenses (bill - investment).

3. Features

1. **User Management:**
   - User registration and login system with data persistence.
   - Login required to access sensitive features.
2. **Member Management:**
   - Add new members with ID, name, and contact details.
   - View the list of members with their investment and meal count.
   - Remove a member from the system.
   - Add investment for a member.
3. **Expense Tracking:**
   - Record daily expenses with date and description.
   - View the list of all recorded expenses.
4. **Meal Management:**
   - Add meals consumed by members on specific dates.
   - View all meal records.
5. **Bill Calculation:**
   - Calculate the total expenses, meal rate, and individual bills.
   - Show net expenses for each member after considering their investments.

4. Technology Stack

- **Programming Language:** C++
- **File Handling:** Used for data storage and retrieval (e.g., users.txt, members.txt, expenses.txt, meals.txt).
- **Standard Libraries:**
  - <fstream>: For file operations.
  - <iomanip>: For formatting output.
  - <vector> and <string>: For efficient data management.
  - <windows.h>: For implementing delays and screen clearing.

5. Key Functionalities

5.1 User Authentication

- Allows users to register and log in.
- Ensures only authorized users can access critical features.

5.2 Member Operations

- Add a new member with their details.
- View the list of all members.
- Update investment amounts for specific members.
- Remove members when they leave the mess.

5.3 Expense Management

- Add daily mess expenses with a description and amount.
- View all recorded expenses.

5.4 Meal Tracking

- Add meals for specific members on given dates.
- View meal records.

5.5 Billing and Reports

- Calculate the meal rate by dividing total expenses by total meals.
- Display each member's total bill, investment, and net expense.
- Provide a comprehensive summary of the mess’s financials.

6. Methodology

The development of the Mess Management System followed these key phases:

6.1 Requirement Analysis

- Identified the core requirements for mess operations through stakeholder discussions.
- Defined the scope and limitations of the system.

6.2 System Design

- Designed the architecture and data flow of the application.
- Planned the modular structure using classes for better organization.

6.3 Implementation

- Implemented the system in C++ using object-oriented programming principles.
- Integrated file handling for data persistence.

6.4 Testing and Validation

- Tested the system under various scenarios to ensure functionality and accuracy.
- Validated results against manual calculations.

6.5 Deployment

- Compiled the code to create an executable file for easy distribution.

6.6 Budget, Time and Resource

Budget

This project is primarily a software development project with minimal direct costs. The budget is mainly related to time and resources:

- **Personnel (Developer Time):** This is a hypothetical cost for reporting purposes, as this project is done as a learning exercise and software project, so almost no cost is consumed.
- **Software and Tools:** The project uses free and open-source tools (e.g., C++ compiler, text editor). Therefore, there are no software licensing costs.

Time Management

As in the project plan, the project is estimated to take 3 days to complete. Regular progress tracking and communication will be essential to ensure timely completion.

Resource Management

- **Hardware:** A personal computer with a C++ compiler.
- **Software:**
  - C++ Compiler (e.g., g++, MinGW).
  - A text editor or IDE (e.g., Visual Studio Code, Code::Blocks).
- **Human Resources:** A software developer (or development team).

7. Project Architecture

7.1 Overview

The Mess Management System architecture is based on a layered structure:

- **Presentation Layer:** Handles user interaction through a console interface.

- **Application Layer:** Contains the core logic implemented in classes such as User
- Member, Expense, and Meal.
- Data Layer: Manages data storage and retrieval using text files.

7.2 Data Flow

1. **Input:** Users provide data through the console interface (e.g., member details, expenses, meals).
2. **Processing:** The application processes the input using the relevant class methods (e.g., addMember, addExpense).
3. **Storage:** Data is stored in text files for persistence and retrieval.
4. **Output:** Results such as meal rates and bills are displayed on the console.

7.3 Class Diagram

- **User**
  - login()
  - registerUser()
- **Member**
  - addMember()
  - viewMembers()
  - removeMember()
  - addInvestment()
- **Expense**
  - addExpense()
  - viewExpenses()
- **Meal**
  - addMeal()
  - viewMeals()
- **MessSystem**
  - calculateBill()
  - manageMemberExpense()

8. Testing and Validation

The system was tested under various scenarios:

1. Valid and invalid user login attempts.
2. Adding, viewing, and removing members.
3. Recording and viewing expenses.
4. Logging and viewing meals.
5. Accurate calculation of meal rates and net expenses.

9. Limitations

- Requires manual deletion of files if corrupted.
- No graphical user interface (GUI).
- Limited to essential text-file storage.

10. Future Enhancements

- Replace text files with a database for improved data integrity.
- Implement a graphical user interface for better user experience.
- Add support for exporting reports to PDF/Excel.
- Introduce additional security measures for user authentication.

11. Practical Application

The Mess Management System has a wide range of practical applications:

- **Student Hostels:** Helps manage daily expenses and meal tracking in student accommodations.
- **Shared Apartments:** These are useful for individuals sharing living spaces to manage shared costs and meals.
- **Small Organizations:** This can be adapted to track meals and expenses for canteens in small offices or organizations.

12. Benefits

The system offers several benefits:

- **Time-Saving:** Automates routine tasks like bill calculation and record maintenance.
- **Transparency:** Provides clear and accurate financial reports for all members.
- **Efficiency:** Reduces the risk of human errors in calculations and record-keeping.
- **User-Friendly:** Simplifies complex operations into a manageable and accessible interface.

13. Conclusion

The Mess Management System simplifies mess operations by automating tasks and providing accurate financial reports. It ensures transparency, reduces manual workload, and offers a reliable platform for managing mess activities effectively.

Submitted By:

Monjur Morshed: 220601 (ICE-14)

Apurbo Sharma: 220604 (ICE-14)