Proposal for Hostel Management System

Purpose

The purpose of this proposal is to present a comprehensive hostel management system designed to streamline various aspects of hostel accommodation at our university. This system aims to simplify room selection, change requests, billing calculations, and cleaning service management for students and staff.

Audience

This proposal is intended for the university management department and the lecturer of the Structured Programming course (TFB1013-TEB1013). The system is designed to provide transparency, accountability, and flexibility in hostel management.

Project Overview

The hostel management system is a software solution developed to handle the complexities involved in managing student accommodations. It includes features for room selection, billing calculations, and cleaning service management, which are essential for both students and administrative staff.

Features

1. **Room Selection**: Allows students to choose their type of room (Shared/Single), village, floor, and specific room number.
2. **Room Change Requests**: Facilitates students to request a change of room with all necessary details.
3. **Billing Calculations**: Automatically calculates accommodation bills based on room type, air conditioning, and duration of stay (semester basis).
4. **Cleaning Service Management**: Enables students to opt-in or opt-out of cleaning services and manages the associated billing.

Complexity and Challenges

- **Room and Service Management**: Managing a large number of rooms, each with different attributes (type, village, block, floor, unit, room number), and ensuring accurate tracking of student allocations.
- **Billing Calculation**: Calculating bills accurately based on multiple factors (room type, air conditioning, cleaning services) and over different periods (semester duration).
- **Self-Service**: Providing a user-friendly interface that allows students to perform self-service checks for room availability, service requests, and billing status.

Implementation Details

1. **Room Details**: The system maintains detailed records for each room, including type, village, block, floor, unit, room number, air conditioning status, cleaning service status, and price.
2. **Data Persistence**: Room details and student allocations are saved to and loaded from a text file, ensuring data persistence across sessions.
3. **User Interaction**: The system provides a menu-driven interface for users to interact with, making it easy to navigate and use.
4. **Comprehensive Room Initialization**: The system initializes a comprehensive list of rooms across all villages, blocks, floors, units, and room numbers, ensuring all room details are available from the start.

Sample Code Snippet

```
void saveRoomDetailsToFile(const char *filename) {
    ofstream outFile(filename);
    if (outFile.is_open()) {
        outFile << roomCount << endl;
        for (int i = 0; i < roomCount; ++i) {
            outFile << rooms[i].type << " " << rooms[i].village << " "
                    << rooms[i].block << " " << rooms[i].floor << " " << rooms[i].unit
                    << " " << rooms[i].room << " " << rooms[i].airCond << " "
                    << rooms[i].cleaningService << " " << rooms[i].price << endl;
        }
        outFile.close();
        cout << "Room details saved to " << filename << endl;
    } else {
        cout << "Unable to open file for writing." << endl;
    }
}

void loadRoomDetailsFromFile(const char *filename) {
    ifstream inFile(filename);
    if (inFile.is_open()) {
        inFile >> roomCount;
        for (int i = 0; i < roomCount; ++i) {
            inFile >> rooms[i].type >> rooms[i].village >> rooms[i].block >>
                rooms[i].floor >> rooms[i].unit >> rooms[i].room >>
                rooms[i].airCond >> rooms[i].cleaningService >> rooms[i].price;
        }
        inFile.close();
        cout << "Room details loaded from " << filename << endl;
    } else {
        cout << "Unable to open file for reading." << endl;
    }
}
```

Benefits

- **Efficiency**: Reduces the manual workload for administrative staff by automating room assignments and billing processes.
- **Transparency**: Provides clear and accessible information to students regarding their accommodation details and costs.
- **Flexibility**: Allows students to manage their accommodation preferences and services with ease.

Conclusion

The hostel management system offers a robust solution for managing student accommodations, addressing the complexities and challenges associated with room assignments, billing calculations, and service management. This proposal aims to secure support and approval for the implementation of this system to enhance the efficiency and transparency of hostel management at our university.
