



Faculty of Computing
UNIVERSITI TEKNOLOGI MALAYSIA

Programming Technique II
(SECJ1023)

SEMESTER x 2023/2024-2

Project and Assignment Software Documentation
UTM Hall Management System

By
Anjum Siddiqua Tanveer Siddiqui (A23CS0289)
Taqia Tahmida (A23CS0026)
Fatema Junaed (A23CS0016)

SECTION 06

Lecturer:
Ts. Dr. Johanna Binti Ahmad

Date: 30 June 2024

PART 1: INTRODUCTION

1.1 Synopsis Project:

Project Description:

The UTM Hall Booking System simplifies the process of reserving halls for events inside the UTM campus. The system offers users access to various information about available halls, including their locations and capacities aiding in informed decision-making. Users can check the real-time availability of their chosen halls through the booking system ensuring efficient planning of their events.

The project is implemented in C++ and incorporates various classes to represent different entities and their relationships enhancing the system's organization and efficiency.

Data Structure and Their Application:

- **Class Person:** Serves as the base class for User and Admin holding common attributes such as name and phone number.
- **Class User:** Inherits from Person and includes functionalities specific to users such as making, viewing, and canceling reservations.
- **Class Admin:** Inherits from Person and includes admin-specific functionalities like viewing all bookings and user details.
- **Class Hall:** Represents a hall booking with attributes like venue and methods to handle booking details and availability.
- **Class Login:** Manages login functionalities for both users and admins.

Inheritance:

- **Person to User and Admin:** User and Admin classes inherit from the Person class. This is an inheritance relationship where User and Admin are specific types of Person.

Composition:

- **Login Class has User and Admin Objects:** The Login class uses User and Admin objects to handle the login process. This is a composition relationship where Login contains and manages User and Admin objects.

Aggregation:

- **Hall Class Aggregates multiple Hall Objects:** Methods like fetchcompare and fetchAvailability in the Hall class work with arrays of Hall objects to manage multiple bookings. This is an aggregation relationship where the Hall class handles collections of its own objects.

Association:

- **Hall class methods interact with files:** The Hall class has methods that read from and write to files. This is an association relationship where Hall interacts with external files to store and retrieve booking information.
- **User and Admin Classes use Hall Class methods:** User and Admin classes call methods of the Hall class to perform operations like making reservations or viewing bookings. This is an association relationship where User and Admin interact with the Hall class to accomplish their tasks.

System Capabilities:

- **User Functions:** Users can log in, check availability, make, view, and cancel reservations.
- **Admin Functions:** Admins can log in, view all bookings, and manage user details.
- **Booking Storage and Retrieval:** The system can save booking details to a file (compare.txt) and retrieve them as needed.
- **User and Admin Authentication:** The system manages login details for both users and admins, ensuring secure access.

1.2 Objective of The Project

1. **Efficient Hall Booking Management:** Develop a system that allows users to create, manage, and store hall bookings with relevant details.
2. **Persistent Storage:** Implement file operations to save and retrieve booking data ensuring data recollection across sessions.
3. **User-Friendly Interface:** Provide methods for users to interact with the system easily including making and canceling reservations.
4. **Administrative Control:** Enable admins to effectively oversee and manage all bookings and user details.
5. **Data Integrity and Security:** Ensure accurate saving and retrieval of booking data, and secure login mechanisms for both users and admins.
6. **Flexibility:** Utilize appropriate data structures to handle a large number of bookings and user details efficiently.

PART 2: SYSTEM ANALYSIS AND DESIGN (USE CASE AND CLASS DIAGRAM)

2.1 System Requirements

There are two types of users in the UTM Hall Booking System. The first type is regular users which include students, UTM staff, and event organizers. These users can check the availability of halls, make reservations, cancel reservations, and retrieve their bookings. The second type of user are system administrators. These users have additional capabilities to view all bookings and user information and manage the overall system.

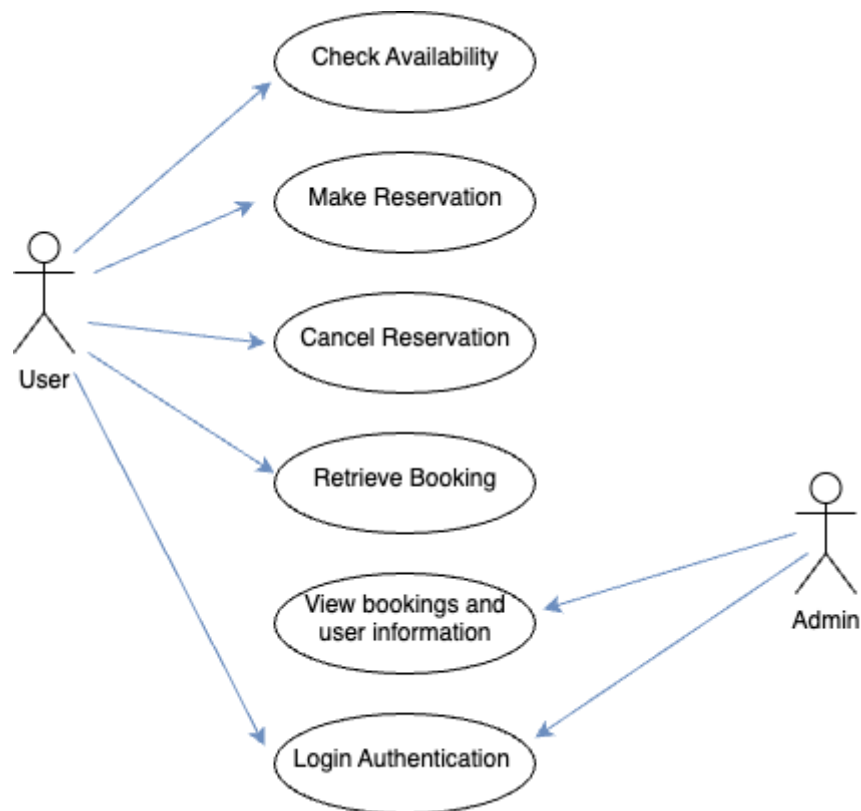


Figure 1: Use Case Diagram for UTM Hall Booking System

Detail Description for Each Use Cases

This system has 6 main use cases.

Use Case	Purpose
Check Availability	Users can search for available halls by picking their desired date and venue. The system checks and displays the availability status of the halls.
Make Reservation	Users can fill out a reservation form with details such as user name, date, and contact information. The system processes the request and confirms the booking.
Cancel Reservation	Users can enter their number and it will automatically cancel their reservation.
Retrieve Booking	Users can access their booking history to view details of upcoming reservations. This information can be used for planning and record-keeping purposes.
View Bookings & User Reservation	Admins can access a comprehensive view of all bookings and user details allowing them to manage and monitor the system effectively.
Login Authentication	Users and admins must authenticate themselves by providing login credentials. This ensures that only authorized personnel can access and manage the system.

2.2 System Design

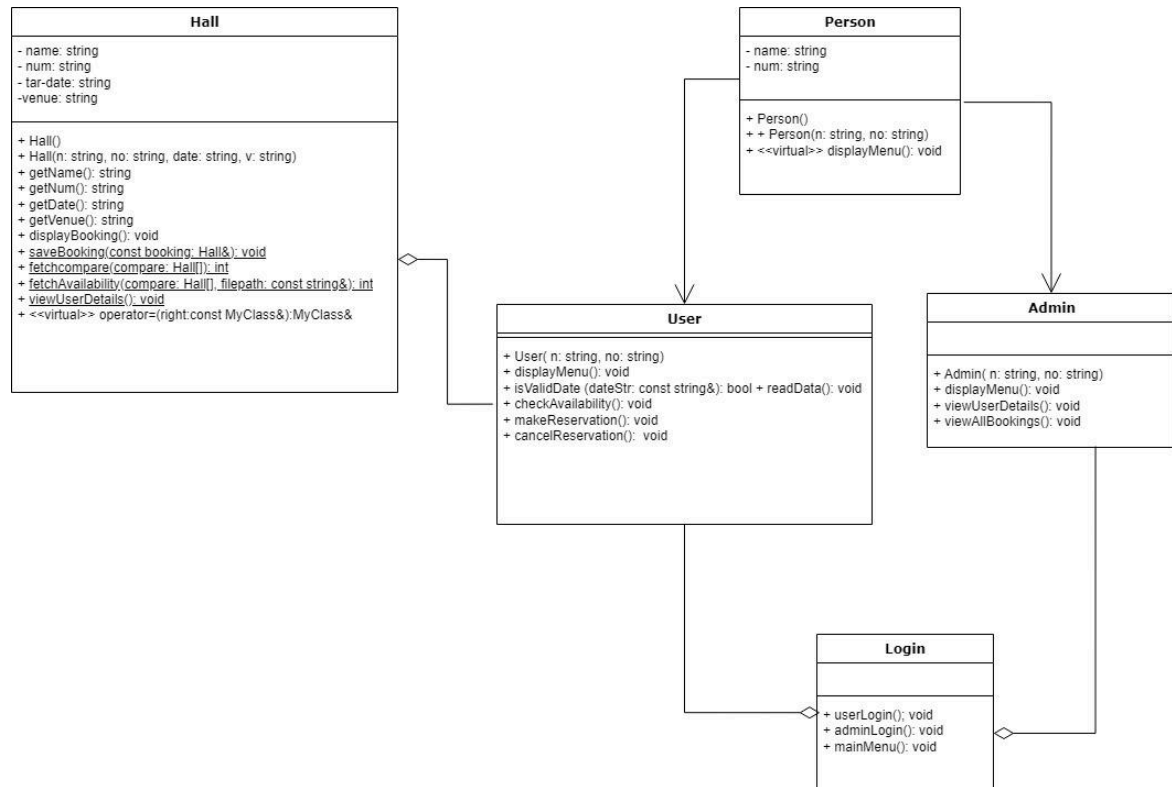


Figure 2: Class Diagram for UTM Hall Booking System

Description:

The System has 5 classes.

Class	Description
Person Class:	Base class for User and Admin Attributes: name, phone number Method: displayMenu()
User Class (Inherits from Person):	User-specific functionalities Methods: displayMenu(), isValidDate(), checkAvailability(), makeReservation(), cancelReservation(), retrieveBooking()
Admin Class (Inherits from Person):	Admin-specific functionalities Methods: displayMenu(), viewUserDetails(), viewAllBookings()
Hall Class:	Represents hall booking Attribute: venue Methods: getName(), getNum(), getDate(), getTime(), displayBooking(), saveBooking(), fetchCompare(), fetchAvailability(), viewUserDetails()
Login Class:	Manages authentication Methods: userLogin(), adminLogin(), mainMenu()

PART 3: SYSTEM PROTOTYPE

This section describes the system prototype. The following figures show an interface for the UTM Hall Booking System.

```

=====
Welcome to Hall Booking System
=====
1. User Login
2. Admin Login
3. Exit
Enter your role : 

```

Screen-1: Main Menu

Screen 1: The first screen displays the Login Menu of the UTM Hall Booking System. The person logged in must insert an integer value in the range 1-3 to choose which role they are entering the system as.

```

=====
User Login
=====
Enter your name: Fatema
Enter your phone number: 666666

```

Screen 2: User Login

Screen 2: If the person enters ,1 then the screen will display the User Login. Here the user will enter their name and their phone number.

```

=====
User Menu
=====
1. Check Availability
2. Make Reservation
3. Cancel Reservation
4. Retrieve Booking Info
5. Main Menu
Enter your choice : 1

```

Screen 3: User Menu

Screen 3: After the user enters their details and clicks enter it will take them to the User Menu where they must enter an integer value between 1-5.

```

=====
Check Availability
=====
Enter the venue: DSI
Enter the date (DD-MM-YYYY): 27-07-2024
Venue is available on 27-07-2024. You can proceed with booking.

```

```
=====
                        Check Availability
=====
Enter the venue: DSI
Enter the date (DD-MM-YYYY): 12-07-1998
Invalid booking date.
```

Screen 4: Check Availability

Screen 4: If the user enters 1 in the User Menu, it will take them to Check the Availability of the venues on the specific day they want to book it. If the user inputs a date that is too much in the past or future it will showcase the message “Invalid booking date.”

```
=====
                        Make Reservation
=====
Enter your name: Fatema
Enter your phone number: 666666
Enter the date (DD-MM-YYYY): 27-07-2024
Enter the venue: DSI
Booking successfully made!
```

Screen 5: Make Reservation

Screen 5: If the user enters 2 in the User Menu, it will take them to Make Reservation where they can enter their name, phone number, hall booking date, and venue to reserve the venue for their chosen date.

```
=====
                        Cancel Reservation
=====
Enter your phone number: 666666
Reservation cancelled!
```

Screen 6: Cancel Reservation

Screen 6: If the user enters 3 in the User Menu, it will take them to Cancel Reservation. The user can enter their phone number to cancel the reservation.

```
=====
                        Retrieve Booking Information
=====
Enter your phone number: 666666
Booking Details:
Name: Fatema
Phone Number: 666666
Date: 27-07-2024
Venue: DSI
```



```

=====
Retrieve Booking Information
=====
Enter your phone number: 666666
No booking found for the provided information.

```

Screen 7: Retrieve Booking Information

Screen 7: If the user enters 4 in the User Menu, then it will take them to Retrieve Booking Information, where it will ask them to enter their phone number. After entering their phone number they can view the booking details of their reservation of the venue. If the reservation was canceled, then the message “No booking found for provided information” will be displayed.

```

=====
User Menu
=====
1. Check Availability
2. Make Reservation
3. Cancel Reservation
4. Retrieve Booking Info
5. Main Menu
Enter your choice : 5

```

Screen 8: User Menu

Screen 8: If the user enters 5 in the User Menu, it will take them to the Main Menu.

```

=====
Welcome to Hall Booking System
=====
1. User Login
2. Admin Login
3. Exit
Enter your role : 2

```

Screen 9: Main Menu

Screen 9: If the person enters 2 in the main menu it will take them to the Admin Login Page.

```
=====
                        Admin Login
=====
Enter your name: Admin1

Enter your phone number: 601127

Invalid Admin Details
```

```
=====
                        Admin Login
=====
Enter your name: Admin1

Enter your phone number: 1234
```

Screen 10: Admin Login

Screen 10: Since the system already has a specific name and phone number for the admin, they must enter that for the admin information. If the admin enters incorrect details for their name and phone number then it will display the message “Invalid Admin Details.”

```
=====
                        Admin Menu
=====
1. View User Details
2. View All Bookings
3. Main Menu
Enter your choice : █
```

Screen 11: Admin Menu

Screen 11: After the admin enters their name and phone number, it will display the Admin Menu for them where they must choose from an integer between 1-3.

```
=====
                        Admin Menu
=====
1. View User Details
2. View All Bookings
3. Main Menu
Enter your choice : 1
Name: Fatema, Phone Number: 666666
```

Screen 12: User Details

Screen 2: If the admin chooses 1, it will display the user details who have made a reservation, on the Admin Menu.

```

=====
                          All Bookings
=====

Booking 1:
Name: Fatema
Phone Number: 666666
Date: 27-07-2024
Venue: DSI

=====
                          All Bookings
=====
No bookings found.

```

Screen 13: All Bookings

Screen 2: If the admin enters 2, it will display the All Bookings that are made by the users. If no booking is made, then it will display "No bookings found."

```

=====
                          Admin Menu
=====
1. View User Details
2. View All Bookings
3. Main Menu
Enter your choice : 3
=====
                          Welcome to Hall Booking System
=====
1. User Login
2. Admin Login
3. Exit
Enter your role : 

```

Screen 14 and 15: Admin Menu and Main Menu

Screen 2: If the admin enters 3, it will take them back to the main menu.

PART 4 : APPENDIX

Hard copy of the source code attached with report in elearning.

---- END OF DOCUMENTATION ----