Virtual Room Reservation Assistant System

Software Architecture Document

Version 1.1

Revision History

Date	Version	Description	Author
20/01/2022	1.1	1. Changes in Deployment View 2. Changes in Implementation View 3. Changes in Process View	范恩琪 羅翡幸

Prepared by

Group 21

1. 李柏賢 B10815007
2. 羅翡幸 B10815009
3. 葉小婷 B10815011
4. 范恩琪 B10815013
5. 施佳妘 B10815014

National Taiwan University of Science and Technology (NTUST)

November 8, 2021

Table of Contents

Software Architecture Document

1.Introduction

Virtual Room Reservation Assistant is a platform that manages room reservation. The system has several functions which includes basic user authentication (covers: Login, Register, Logout) and room reservation (covers: Room Booking, Updating Reservation, Canceling Reservation, and Viewing Reservation Records).

This document elaborates the software architecture document for "Virtual Room Reservation Assistant". The software architecture is abstracted into many views and components which are explained in detail. This document follows the 4 + 1 view model as the reference model for this document

1.1. Purpose

This document provides a comprehensive architectural overview of the system, using a number of different architectural views to depict different aspects of the system. It is intended to capture and convey the significant architectural decisions which have been made on the system.

This document elaborates the architecture of the system in 5 different views from 4 + 1 view model, both static and dynamic behavior of the system is described in this document. 4 + 1 view model will make it possible to depict the software as accurately as possible, and it allows a wide range of stakeholders to find what they require in the architecture document.

1.2 Scope

The Software Architecture Document (SAD) applies to each static and dynamic aspect of the system. Since the 4 + 1 view model is used as the reference model, it will incorporate many views of the system, which make the document complete and consistent.

The scope of this document is to depict the architecture of the Virtual Room Reservation Assistant. Under the static behavior of the system, the document discusses the class diagrams, package diagrams, and other static architecture design. Dynamic aspects of the system are elaborated using use case realizations and system sequence diagrams.

1.3 Definitions, Acronyms and Abbreviations

Abbreviations	Meaning
UML	Unified Modeling Language

SAD	Software Architecture Document
QoS	Quality of Service
OS	Operating System
UI	User Interface
PC	Personal Computer

1.4 References

The 4 + 1 View Model of Architecture, Philippe B. Kruchten, November 1995, https://www.cs.ubc.ca/~gregor/teaching/papers/4+1view-architecture.pdf

1.5 Overview

Below are the topics that will be further discussed in later sections:

Section 2: Architectural Representation, describes the different views used in this document.

Section 3: Architectural Goals and Constraints, lists the constraints and objectives.

Section 4: Use Case View, illustrates the use cases.

Section 5: Logical View, describes the abstraction of the functional requirements.

Section 6: Process View, describes the general process flow.

Section 7: Deployment View, describes the connection between the physical components.

Section 8: Implementation View, describes the different layers and subsystems

Section 10: Size and Performance, discusses the memory size used and the general performance.

Section 11: Quality, discusses Quality of Service (QoS) attributes.

2. Architectural Representation

The views used to document the application are:

2.1 Logical view

Audience: Programmers, Designers.

Area: Functional requirement, system layers, describes the design’s object model.

2.2 Process view

Audience: Integrators, Programmers.

Area: Non-Functional requirements, describes the design's concurrency and synchronization.

2.3 Implementation view

Audience: Programmers.

Area: Software components, describes the layers and subsystems of the application.

2.4 Deployment view

Audience: Programmers.

Area: Topology, describes the mapping of the software into the hardware.

2.5 Use-case view

Audience: All the stakeholders of the system, including end users.

Area: Describes the use-cases of the system.

3. Architectural Goals and Constraints

This section describe the software requirements and objectives that have some significant impact on the architecture

3.1 Platform

The system will be hosted at a free web hosting space. As it is a web-based application, client OS can be any PC operating system. Clients need to have an internet connection and use a browser to access the system.

3.2 Security

The application will implement basic security measures such as authentication and authorization.

3.3 Persistence

Data persistence will be addressed using a relational database.

3.4 Reliability/Availability

The system will be tested before deployment to make sure that it is working.

3.5 Performance

The common processes such as login and making reservations shall be processed under 10 seconds.

3.6 Portability

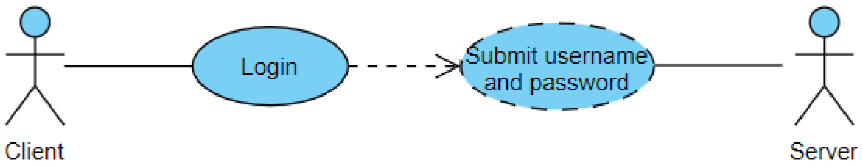
The system shall be available to access in any browser with a working internet connection.

4. Use-Case View

This section lists use cases or scenarios from the use-case model if they represent some significant, central functionality of the final system. There are some significant use cases in this virtual assistant room software such as login, register, logout, making reservation, edit reservation, cancel reservation, check room availability or room status and view user past and ongoing reservations.

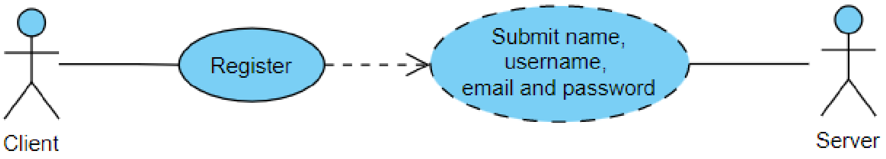
1. Login

   Clients enter the application and log in into their account by entering username and password into the system.



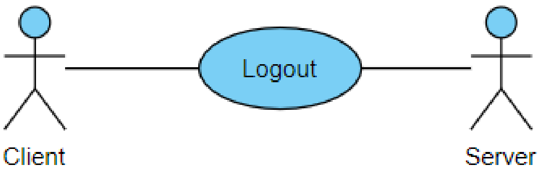
2. Register

   Clients enter the application and create a new account by entering username, email, and password , and then the user information will be saved by the system.



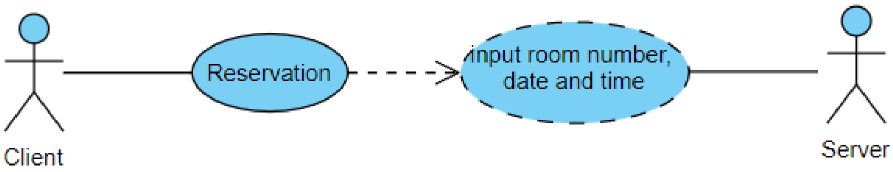
3. Logout

   Clients log out from their account and the system will log them out from their account.



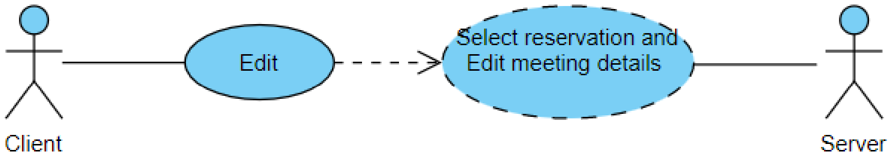
4. Reservation Booking

   Clients choose the room number, date and time that they desire to reserve. The server will accept client input and check whether the desired room is available during the specified time period. If it is, the reservation will be recorded to the database and the user will be redirected to the index page. If not, the user is notified with a failure notice.



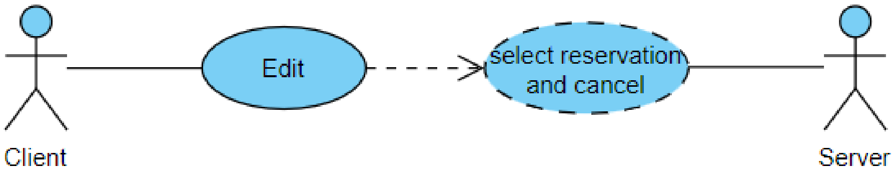
5. Edit

Clients choose which reservation to be edited, and the system will take them to a page to edit the reservation details. After clients change the reservation detail, the system will update the details of the reservation.



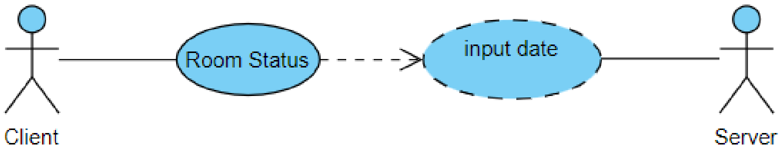
6. Cancel

Clients can cancel their selected reservation in the Edit page.The server will then process the client's request accordingly.



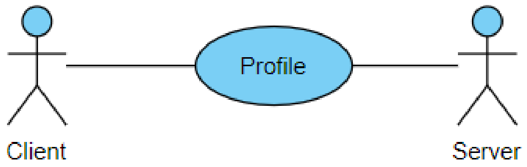
7. Room Status

Clients can see the schedule table of all rooms at the desired day by inputting the date to the server after clicking the "Room Status" button.



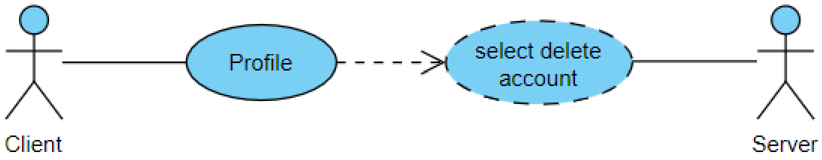
8. Profile

Clients can view their past reservations or delete their accounts.



9. Delete Account

   Clients can delete their account by going to the profile page and clicking the delete account button, the server will process the user's request and delete the user's account.



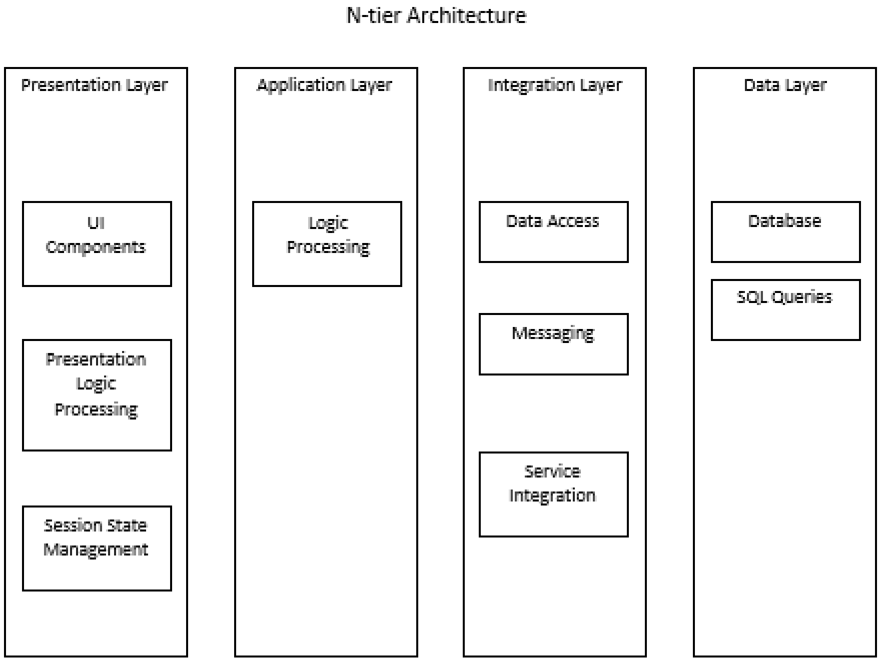
4.1 Use-Case Realization

Refer to section 5.2 to see how design elements provide the functionalities identified in the significant use cases.

5. Logical View

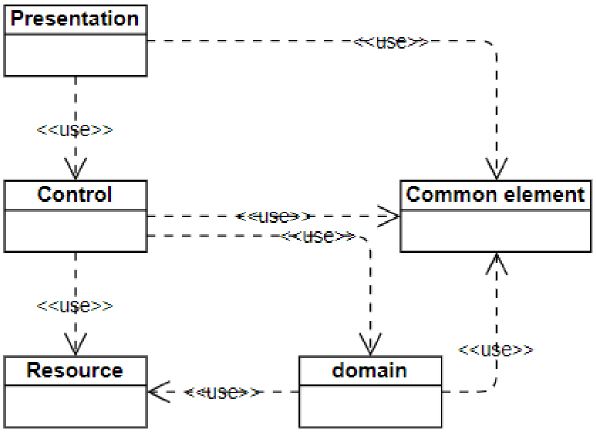
5.1 Overview

The virtual room reservation assistant software is divided into layers based on N tier architectures.



The layering model of the virtual room assistant is based on a responsibility layering strategy that associates each layer with a particular responsibility. This strategy is chosen because it isolates various system responsibilities from one another so that it improves both system development and maintenance.

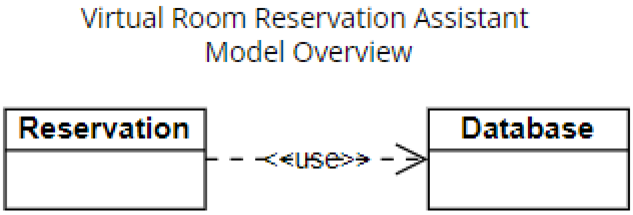
Architecture layer dependencies



Each layer has specific responsibilities:

- The presentation layer deals with the presentation logic and the pages rendering
- The control layer manages the access to the domain layer.
- The resource layer (integration layer) is responsible for access to the enterprise information system (databases or other sources of information).
- The domain layer (application layer) is related to the business logic and manages the accesses to the resource layer.
- The Common Elements layer gathers the common objects reused through all the layers.

The virtual room reservation assistant has 9 use cases which have been explained in part 4.

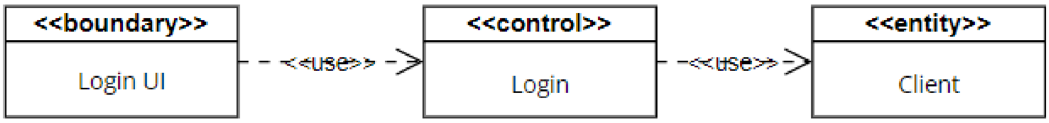


5.2 Architecturally Significant Design Packages

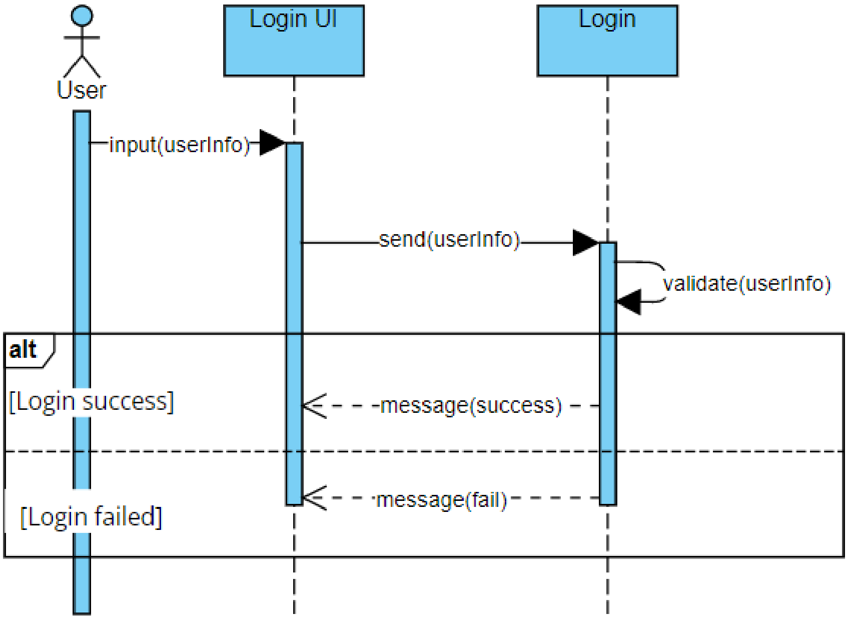
5.2.1 Login

This package is responsible for logging users into their account and accessing the system.

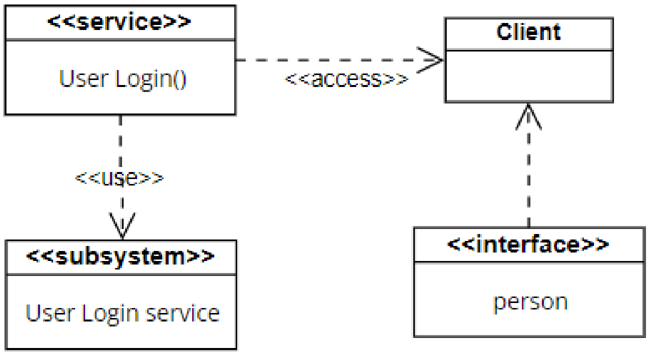
Participants:



Basic Flow:



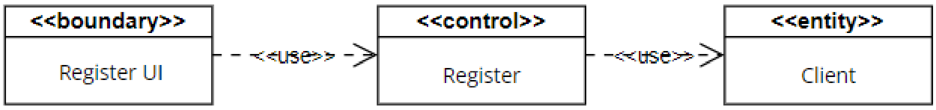
Design Model:



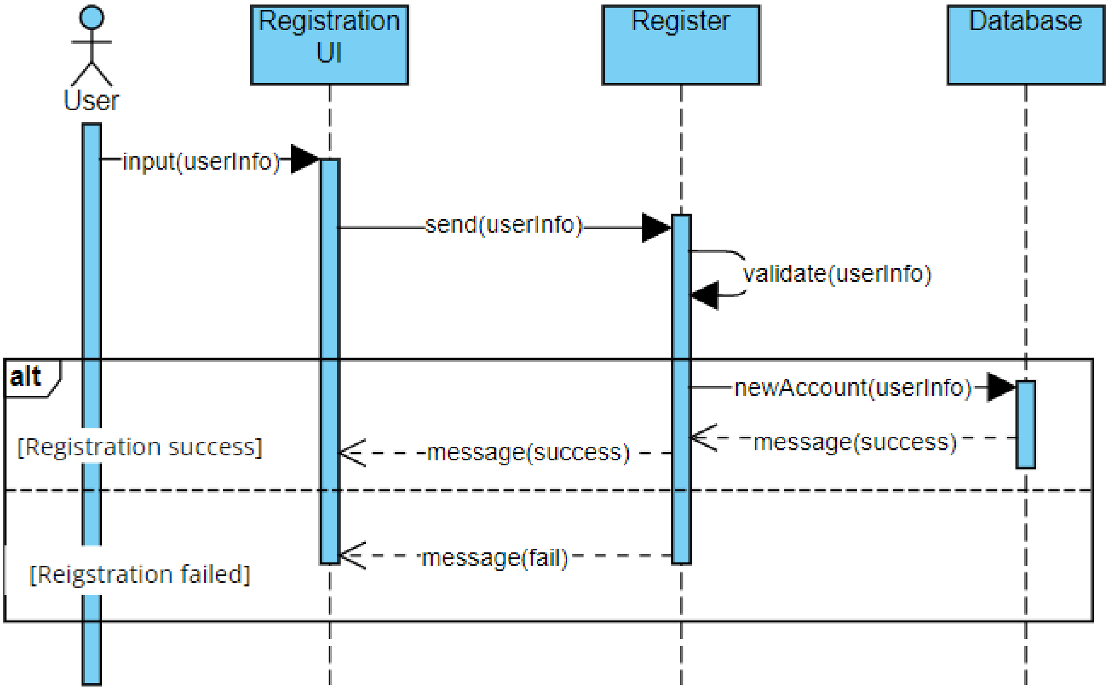
5.2.2 Register

This package is responsible for registering a new account for users and saving user's information into the server.

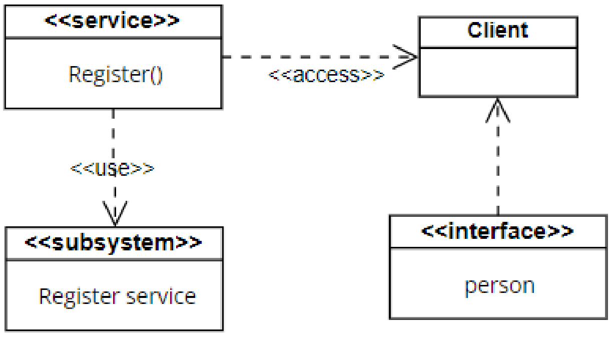
Participants:



Basic Flow:



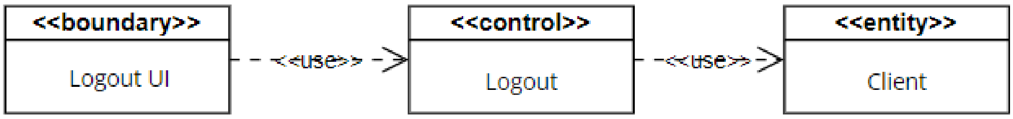
Design Model:



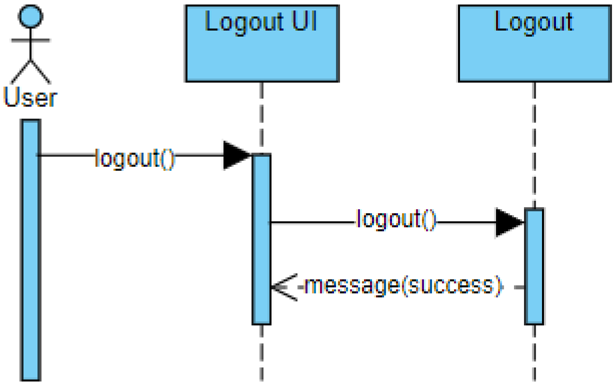
5.2.3 Logout

This package is responsible for logging users out from their account.

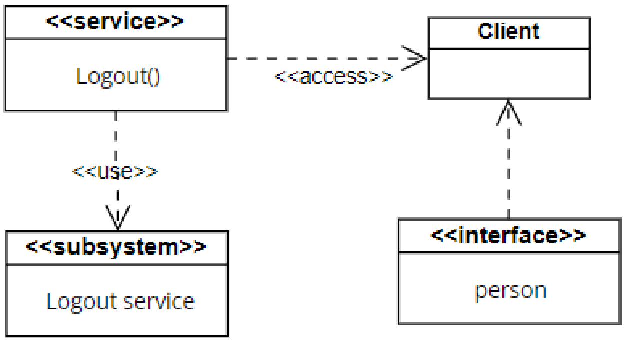
Participants:



Basic Flow:



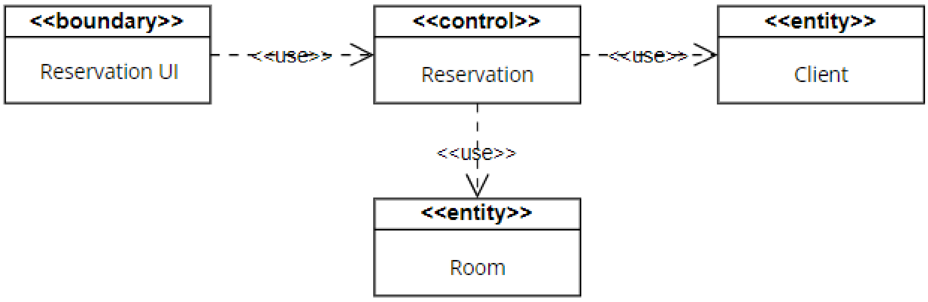
Design Model:



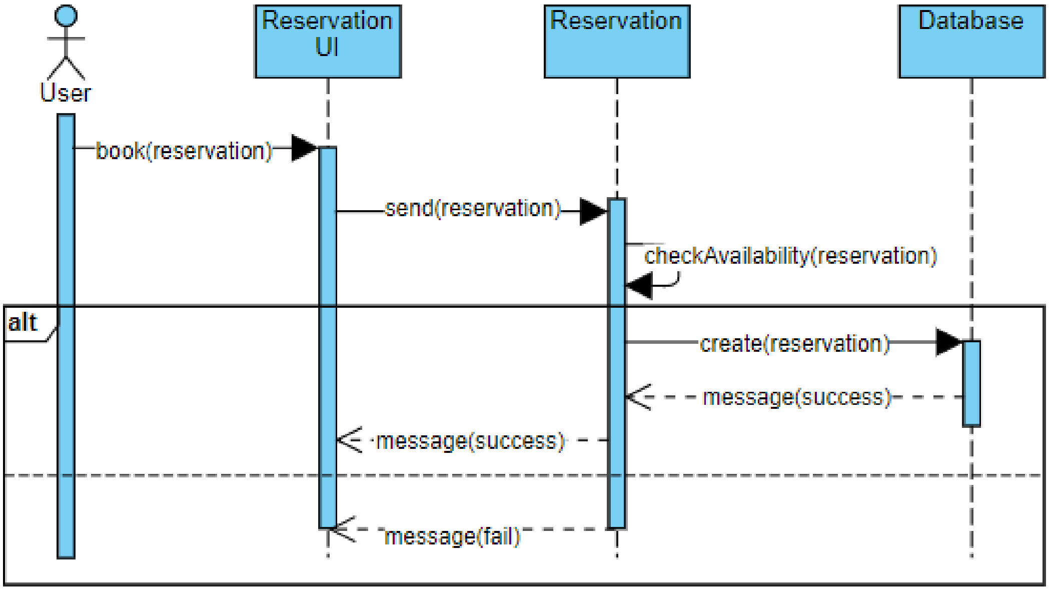
5.2.4 Reservation Booking

This package is responsible to reserve or book a room for the user and record it into the server.

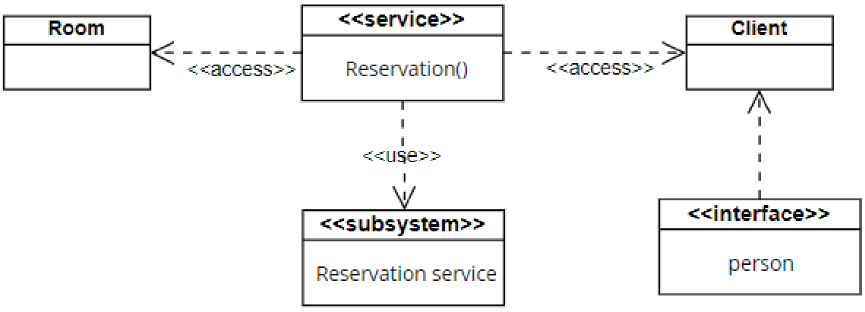
Participants:



Basic Flow:



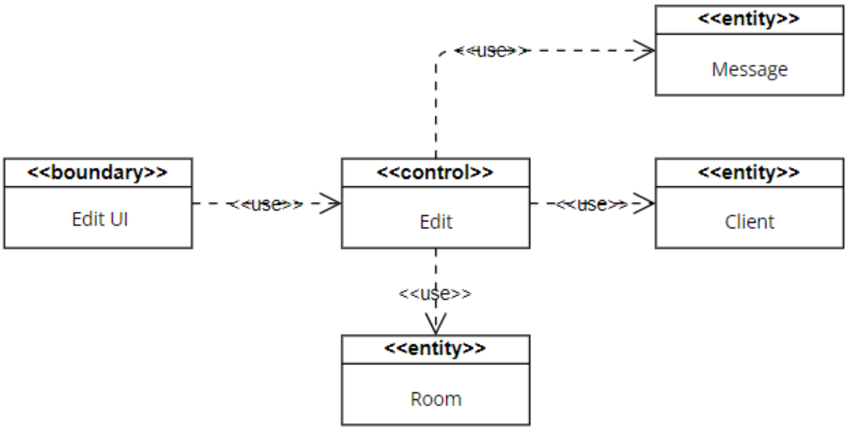
Design Model:



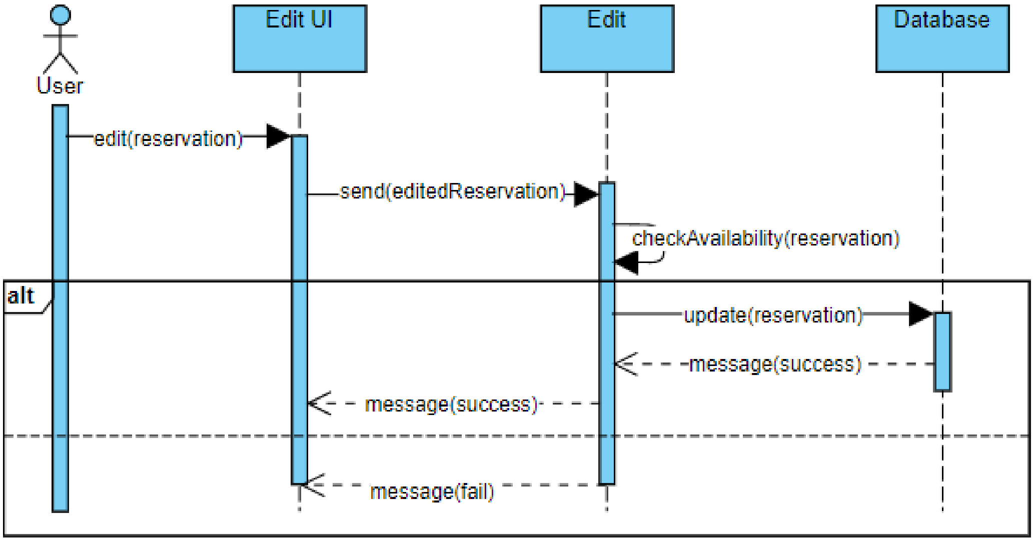
5.2.5 Edit

This package is responsible for editing meeting details and later the changes will be recorded in the server.

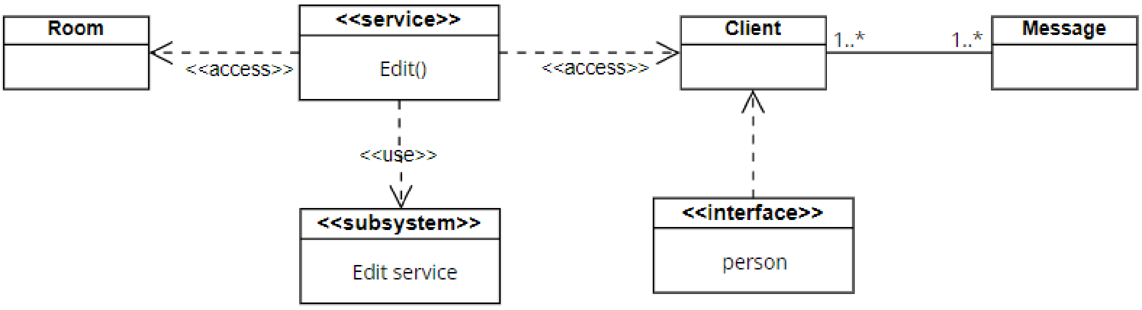
Participants:



Basic Flow:



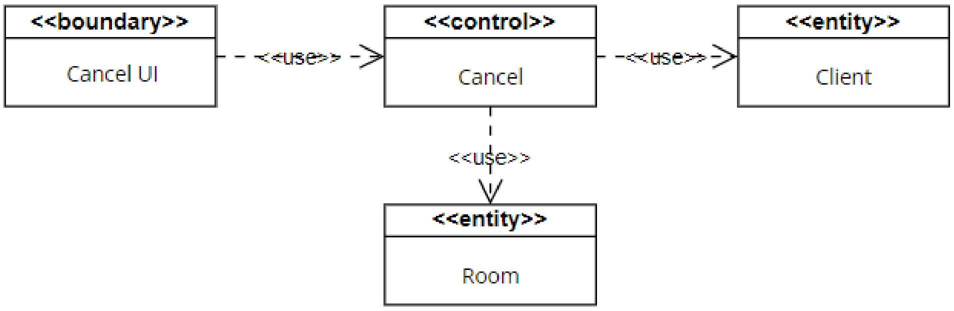
Design Model:



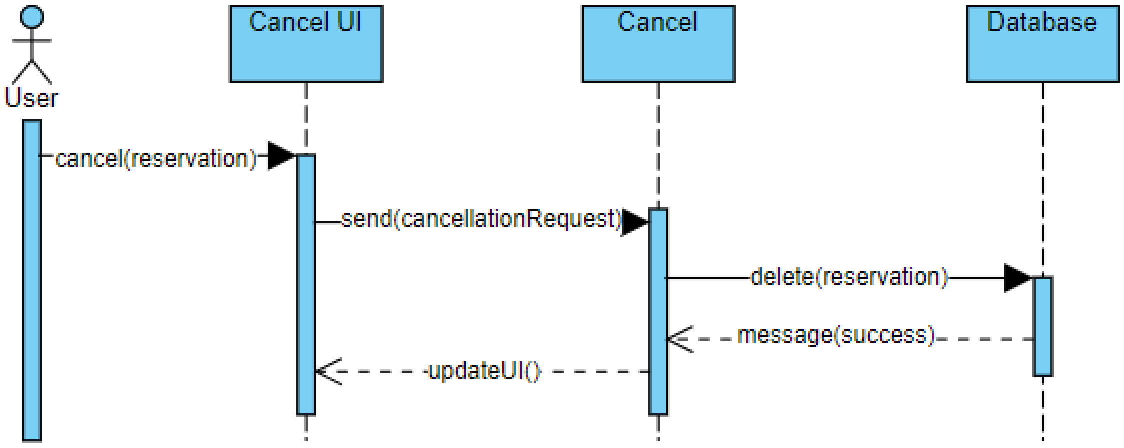
5.2.6 Cancel

This package is responsible for canceling the user's current reservation and it shall be recorded in the server.

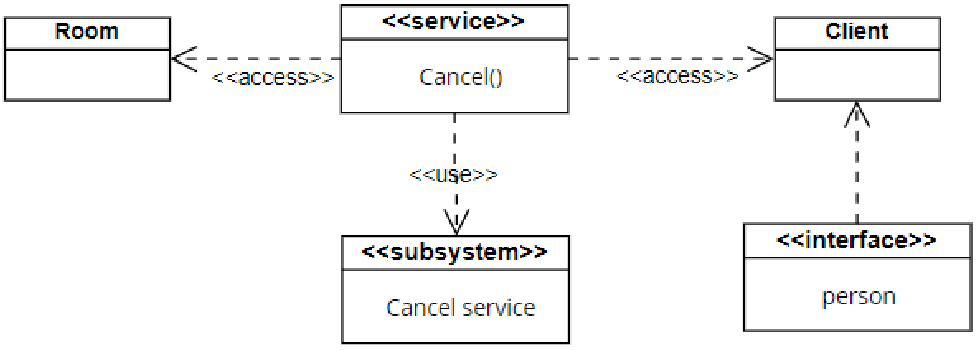
Participants:



Basic Flow:



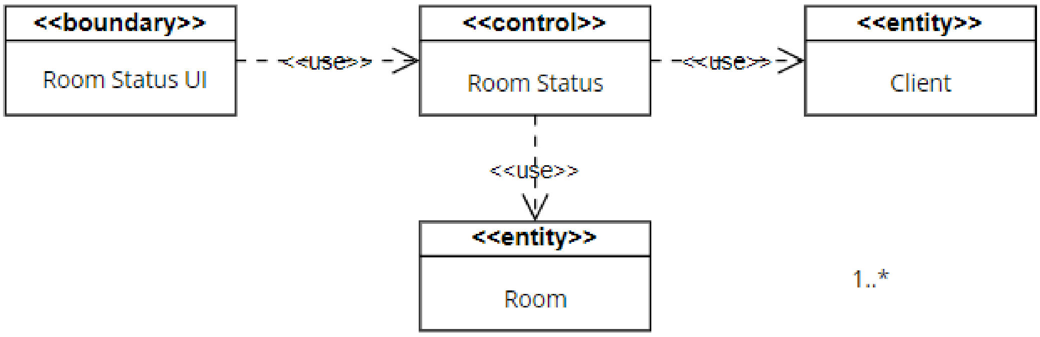
Design Model:



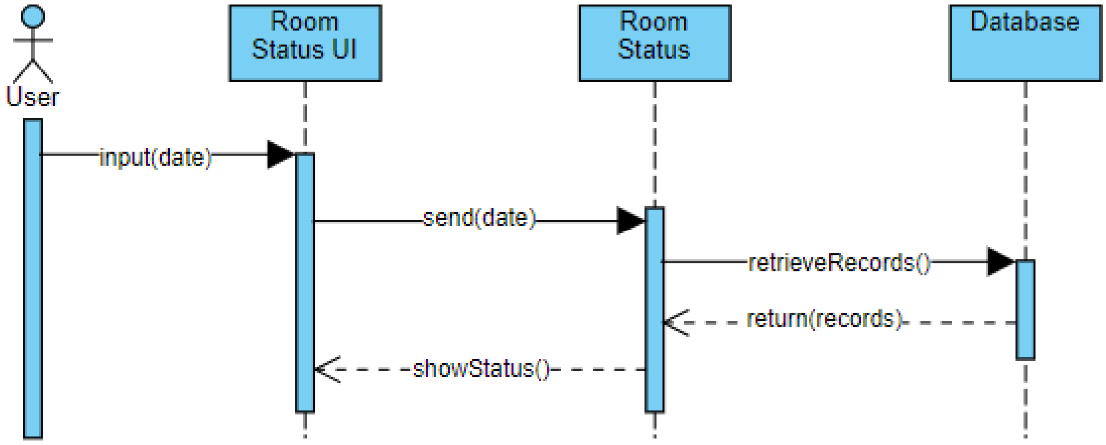
5.2.7 Room Status

This package is responsible for showing the schedule of all rooms at the desired date.

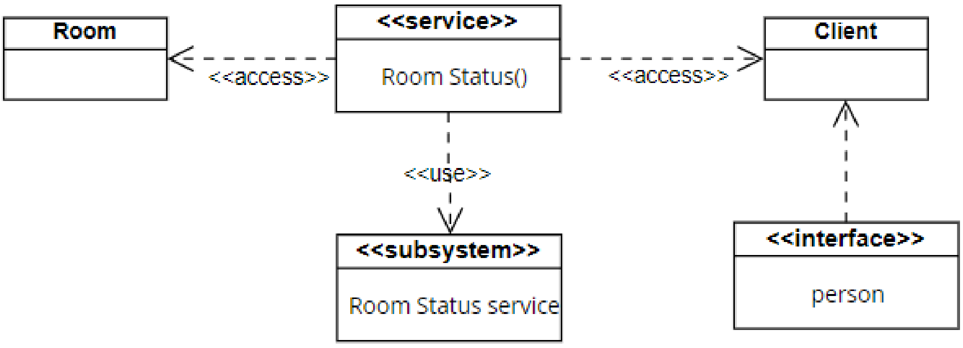
Participants:



Basic Flow:



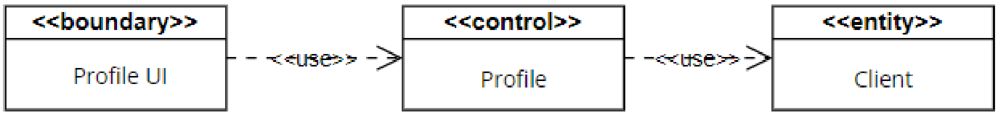
Design Model:



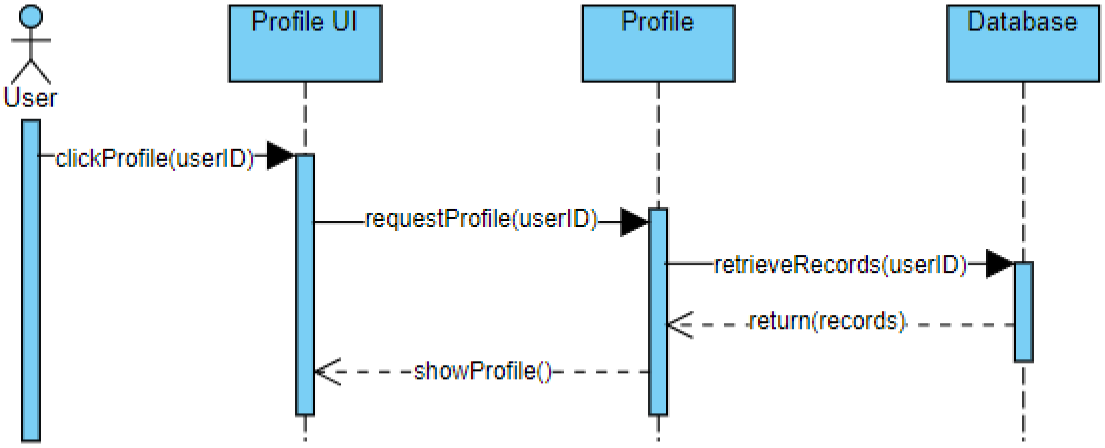
5.2.8 Profile

This package is responsible for showing the user past and current reservation.

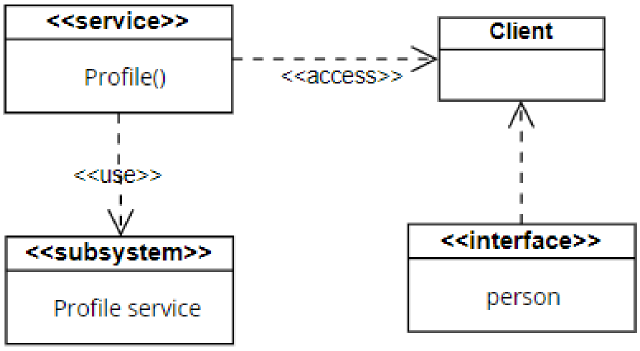
Participants:



Basic Flow:



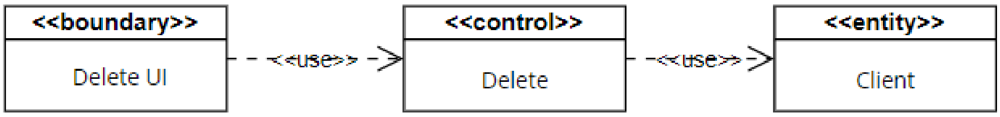
Design Model:



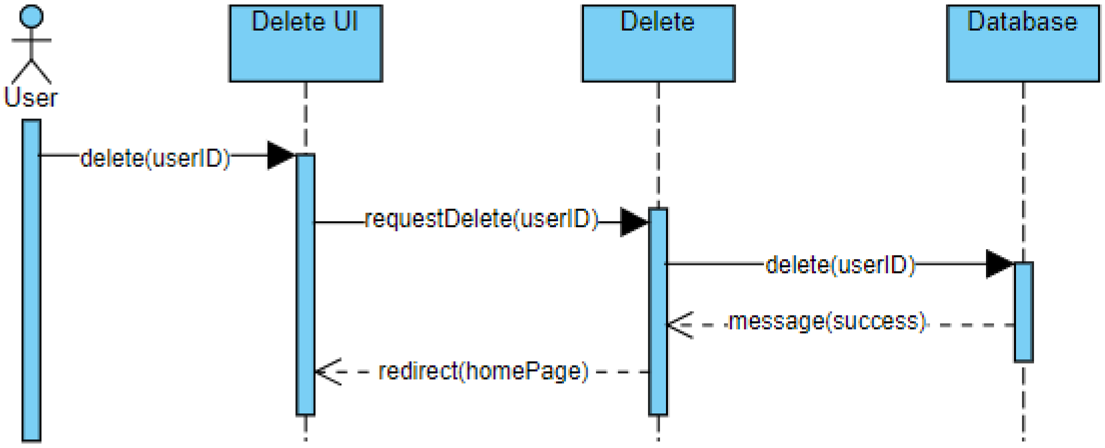
5.2.9 Delete Account

This package is responsible for deleting user accounts from the server.

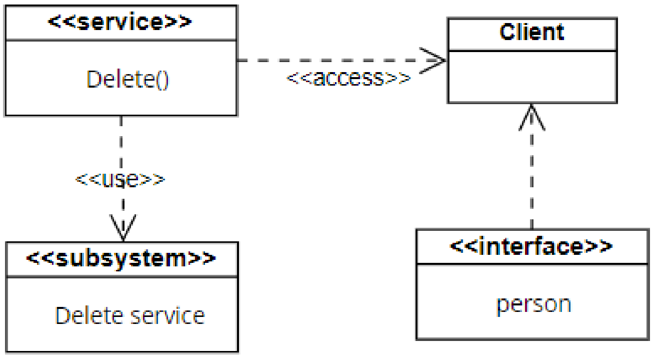
Participants:



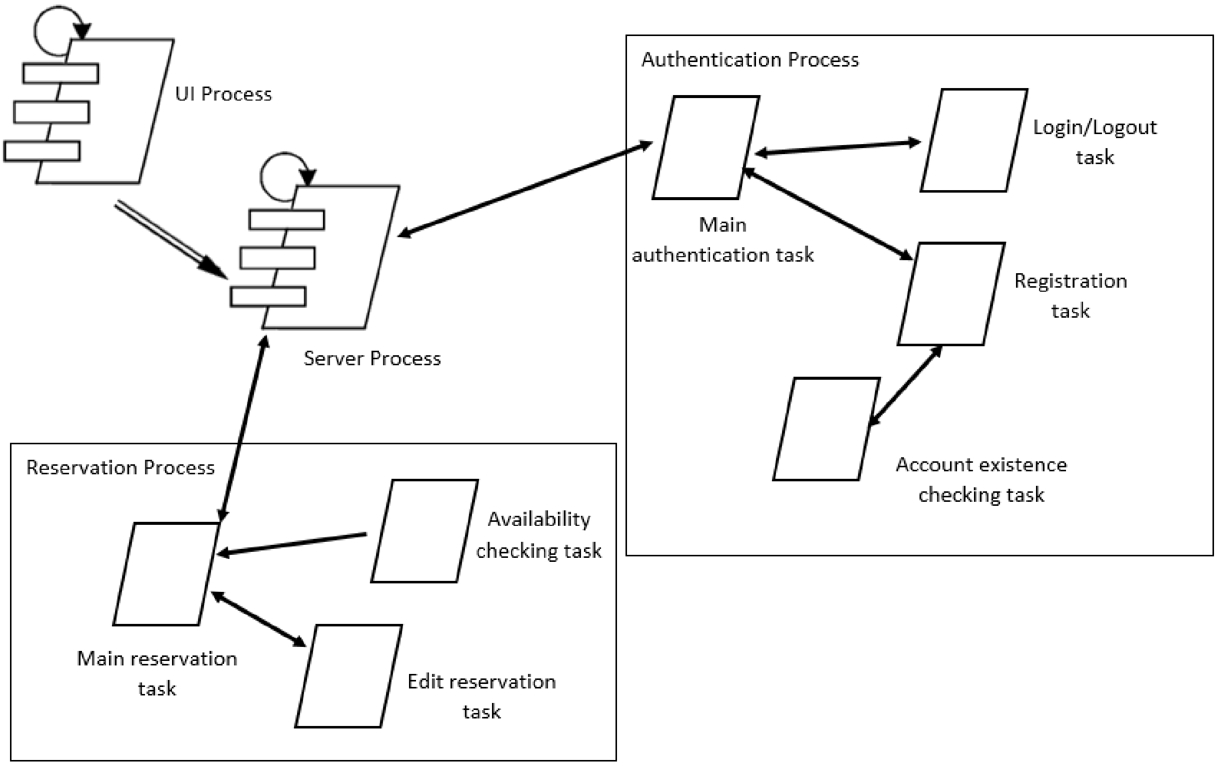
Basic Flow:



Design Model:

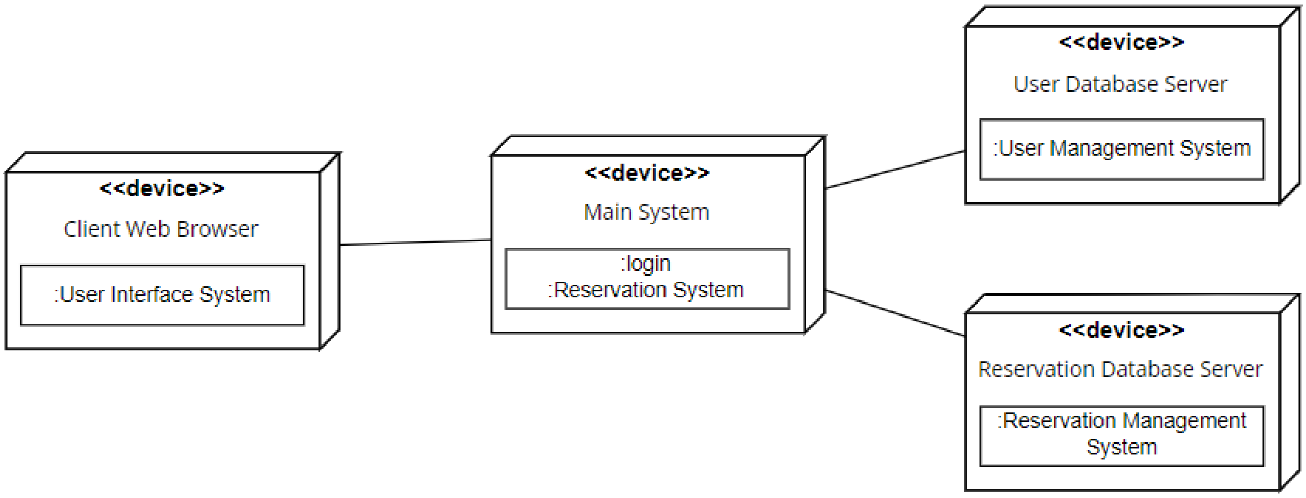


6. Process View



The above shows the general flow of the process in the system. All terminals are handled by the terminal process named *Server Process*. The main method of communication among the processes used is message passing. Message passing allows the processes to exchange messages which are sending and receiving messages. A process can send messages to the *Server Process* which the message can be received by another process. *Server Process* gives the services to clients by responding to the client messages and returning it back again to the client.

7. Deployment View



The system is divided into 4 devices which includes the Client server, Main server, User DB Server, and Reservation DB Server. It is only connected to one network, as the system is run via localhost.

7.1. Client Web Browser

Users access our service by using a browser (can be any PC Operating System), and need to have an internet connection.

7.2. Main System

Main system will include all the interfaces that will connect users to the system.

7.3. User Database Server

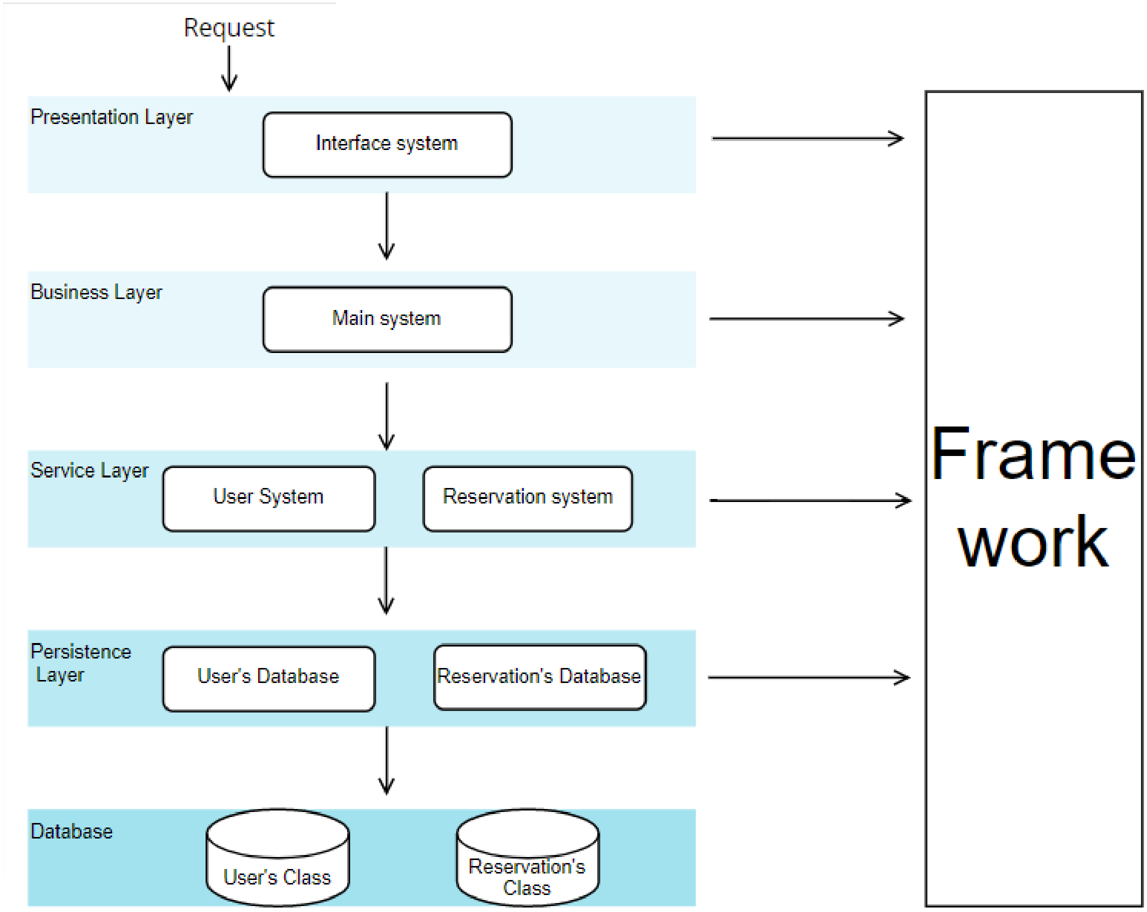
The database will store user information such as username, password, email, and name.

7.4. Reservation Database Server

The database will store reservation information including participant’s name, participant’s email, date, etc.

8. Implementation View

8.1 Overview



8.2 Layers

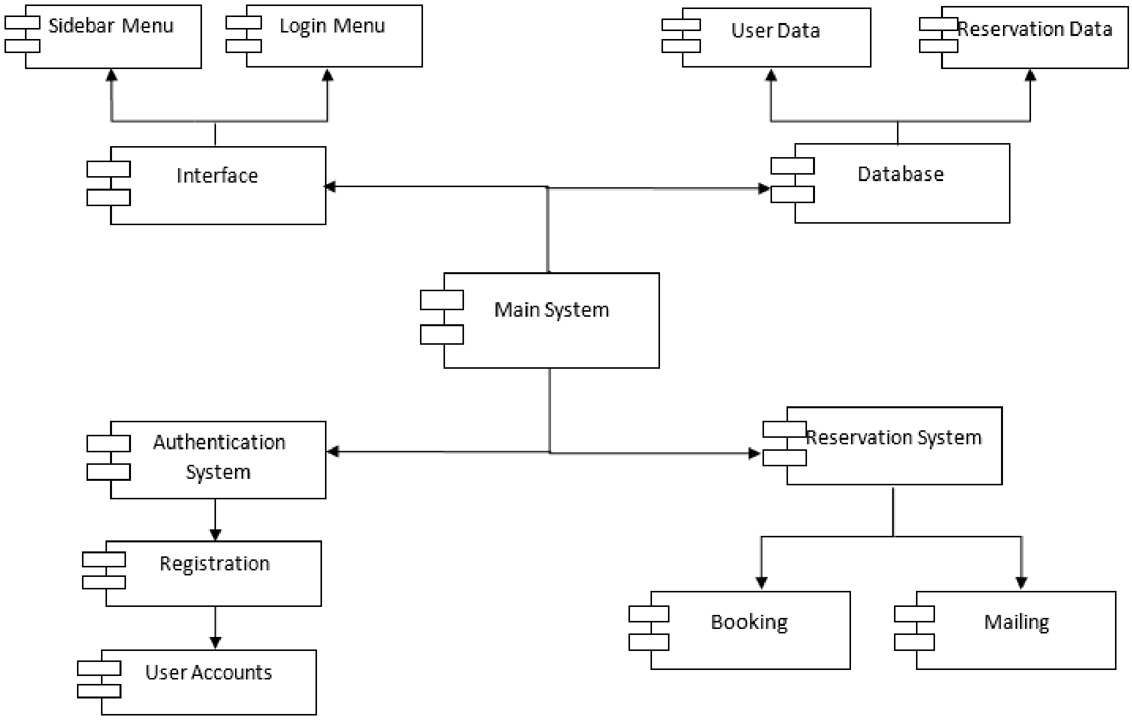
Client's request first goes into the **presentation layer** which consists of an interface system.

The **business layer** consists of the Main system.

Service layer consists of User database and Reservation database and only the business layer can access the service layer.

Persistence layer consists of User database and Reservation database.

Database Layer consists of User's class and Reservation's class.



9. Data View (Optional)

10. Size and Performance

As it is a web based application, the software will be locally run from the user's device. Software size shall not exceed 10MBs and there should be no lag when accessing new functions.

11. Quality

Below are the quality of services provided in the system.

Extensibility

The system is extensible in such a way that features could be added with proper understanding of the framework used by the system.

Reliability

- Ensures that there are no losses of personal data
- Ensures that the room is available for booking, if there is any emergency maintenance, then inform the user that the booking is canceled due to maintenance.
- Ensures that when 2 users at the same time are booking the same room for the same time slot, there should be a time duration for the first user to do the booking before the second user does the booking.

Portability

The system is compatible with a number of browsers, including Google Chrome, Firefox, Microsoft Edge, Opera, etc.

Security

- The system will not monitor or collect any un-prompted data from the user.
- The system requires users to register an account before being given access to the reservation system.
- Passwords will be encrypted before recording it into the database..

Simplicity

The user interface of the system shall be designed for ease-of-use and appropriate for a computer-literate user community with minimal additional training on the system.

Privacy

Accounts are used for authentication and authorization