Software Design Document (SDD) Template

Software design is a process by which the software requirements are translated into a representation of software components, interfaces, and the data necessary for the implementation phase. The SDD shows how the software system will be structured to satisfy the requirements. It is the primary reference for code development and, therefore, it must contain all the information required by a programmer to write code. The SDD is performed in two stages. The first is a preliminary design in which the overall system architecture and data architecture is defined. In the second stage, i.e. the detailed design stage, more detailed data structures are defined and algorithms are developed for the defined architecture. This template is an annotated outline for a software design document adapted from the IEEE Recommended Practice for Software Design Descriptions. The IEEE Recommended Practice for Software Design Descriptions have been reduced in order to simplify this assignment while still retaining the main components and providing a general idea of a project definition report. For your own information, please refer to IEEE Std 10161998 1 for the full IEEE Recommended Practice for Software Design Descriptions.

Group 21

Virtual Room Reservation Assistant

Software Design Document

Version 1.1

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

2021-12-23

Revision History

Date	Version	Description	Author
20/01/2022	1.1	1. Changes in Screen Images	范恩琪

TABLE OF CONTENTS

1. INTRODUCTION

The Software Design Document is a document to provide documentation which will be used to aid in software development by providing the details for how the software should be built. Within the Software Design Document are narrative and graphical documentation of the software design for the project including use case models, sequence diagrams, collaboration models, object behavior models, and other supporting requirement information

1.1 Purpose

This Software Design Document describes the architecture and system design of Virtual Room Reservation Assistant to allow for software development to proceed with an understanding on what to build and how it is expected to be. The Software Design Document provides necessary information and detail about the software and system to be built.

1.2 Scope

This Software Design Document is for a base level system which will work as a proof of concept for the use of a building system that provides a base level of functionality to show feasibility for large scale production use. This Software Design is focused on the base level systems and critical parts of the system. For this particular Software Design Document, the focus is placed on generation of documents and modifications of the documents. The system will be used in conjunction with other pre-existing systems and will consist largely of a document interaction facade that abstracts document interactions and handling of the document objects.

1.3 Overview

The Software Design Document is divided into 8 sections with various subsections. The sections of the Software Design Document are:

1 Introduction

2 System Overview

3 System Architecture

4 Data Design

5 Component Design

6 Human Interface Design

7 Requirements Matrix

1.4 Reference Material

- Software Design Document (SDD) Template

1.5 Definitions and Acronyms

Abbreviations	Meaning
IO	Input/output
SDD	Software Design Document
P2P	Peer-to-Peer
OS	Operating System
UI	User Interface

2. SYSTEM OVERVIEW

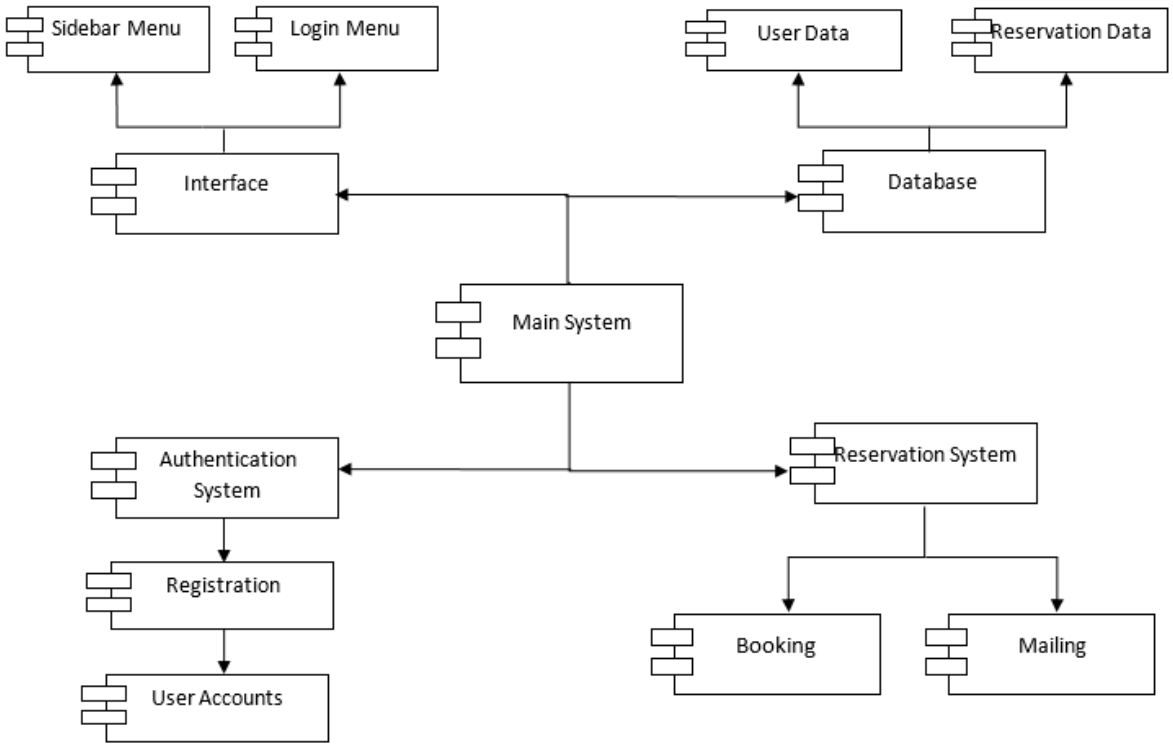
This system consists of functionalities that allow the user to book reservations, invite participants, edit reservations, cancel reservations, create accounts, edit accounts, delete accounts.

Book reservation allows the client to make a reservation which also shows rooms and dates that are available. Invite participants allow the user to invite other participants to join the room that has been reserved. Edit reservation allows the user to edit the reservation which then checks the reservation's details such as rooms, date, time that are available and participants that join before deleting the previous one and assigning the new record to the database. Cancel reservation allows the user to cancel the reservation which later will delete the previous record of the details from the database.

Create account functionalities allow the user to create an account before making a reservation which consists of username, email, and password. Users also can edit their own account. Users can delete accounts which the room reservation that the user ever made will be deleted too and can't join the reservation that was invited by another user.

3. SYSTEM ARCHITECTURE

3.1 Architectural Design



The system is based on a client-server architecture pattern, which has two main components - a client and a server. The responsibilities of the system server are split between the interface, reservation and authentication subsystems.

The following are brief explanations of the subsystems.

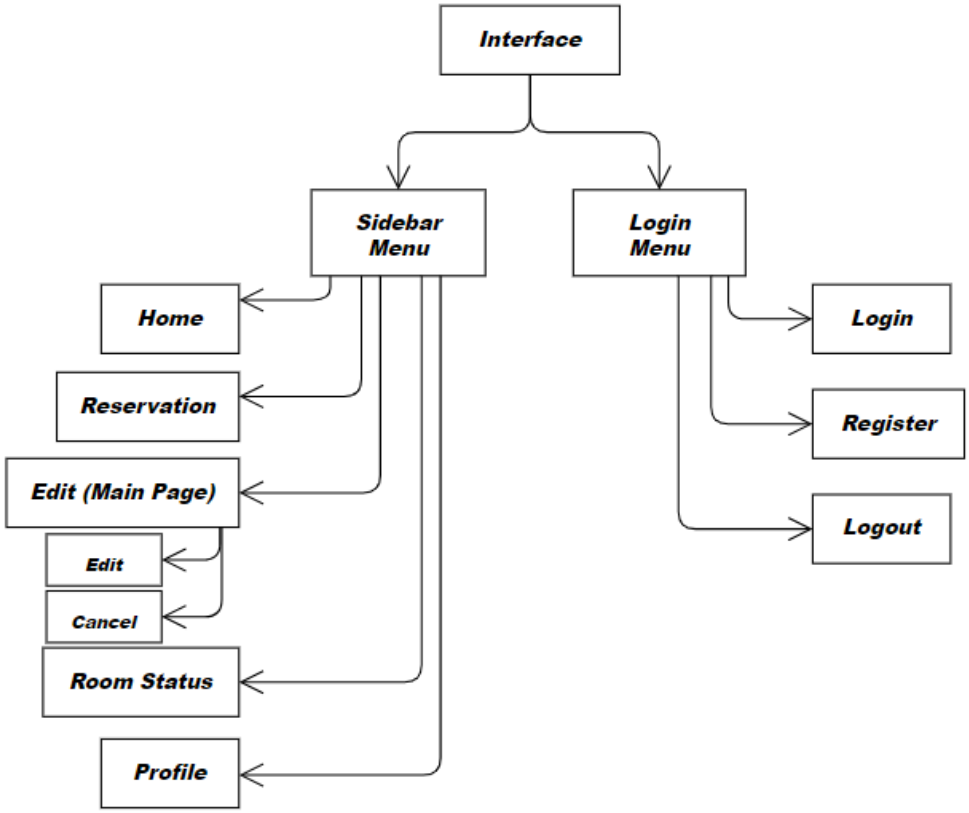
- The interface subsystem provides a user-friendly graphical user interface and menu bars containing triggers/buttons to initiate the functionalities of the system. These triggers call upon the specified use cases which mainly run on the reservation and the authentication subsystem .
- The reservation subsystem will then perform any booking related functionalities, including: reservation making, editing and cancellation. These actions would all trigger the mailing subsystem to notify the users' of all their actions in the booking subsystem.
- The authentication subsystem is responsible for registration, log I/O and account deletion.

All of the subsystems are allowed read and/or write access except for the interface as it can only access the database through the other two subsystems.

3.2 Decomposition Description

Interface

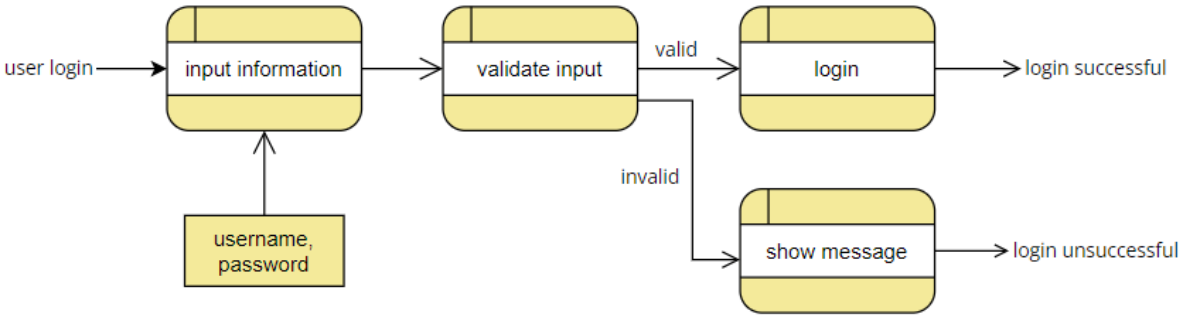
Structural decomposition



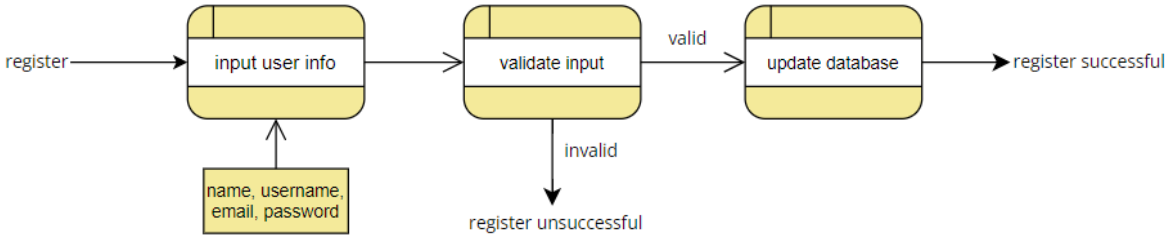
Data flow diagrams:

Login Menu

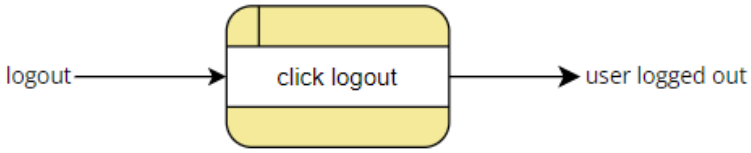
1. Login



2. Register

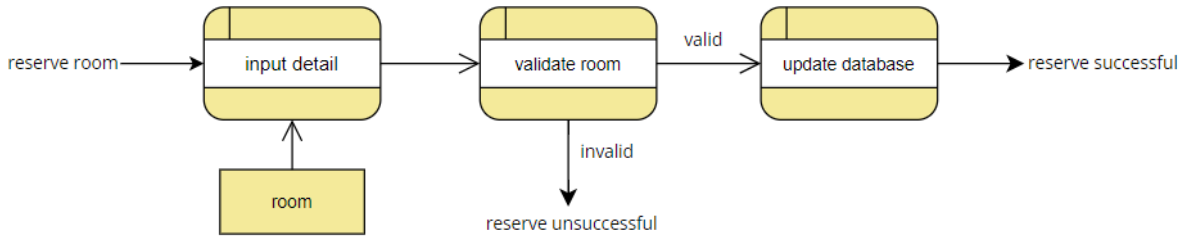


3. Logout

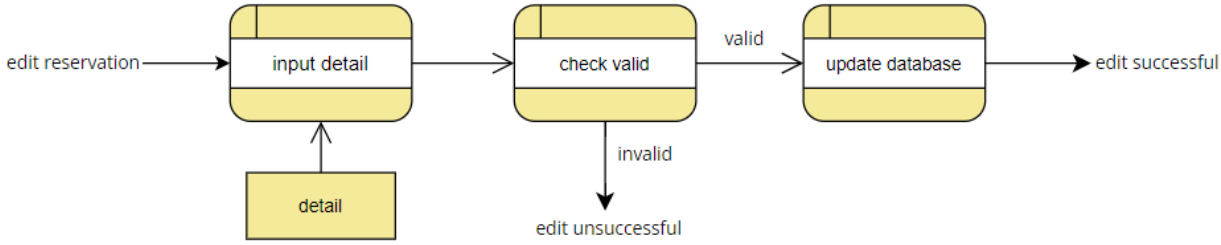


Sidebar Menu

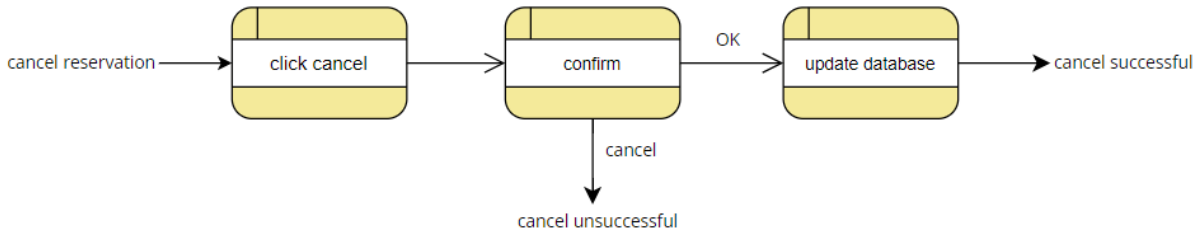
1. Home (No Data Flow)
2. Reservation



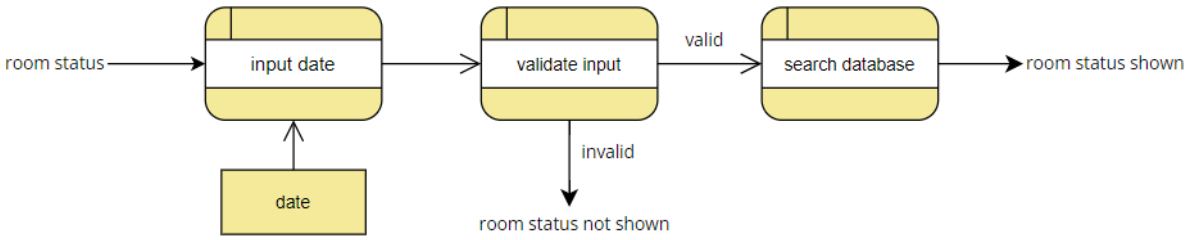
3. Edit (Main Page)
    a. Edit reservation



    b. Cancel reservation

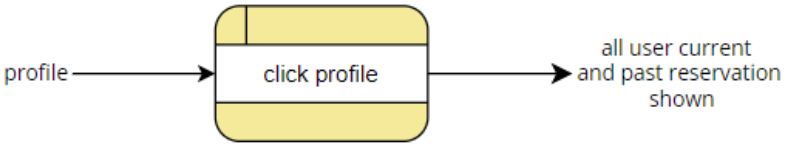


4. Room Status

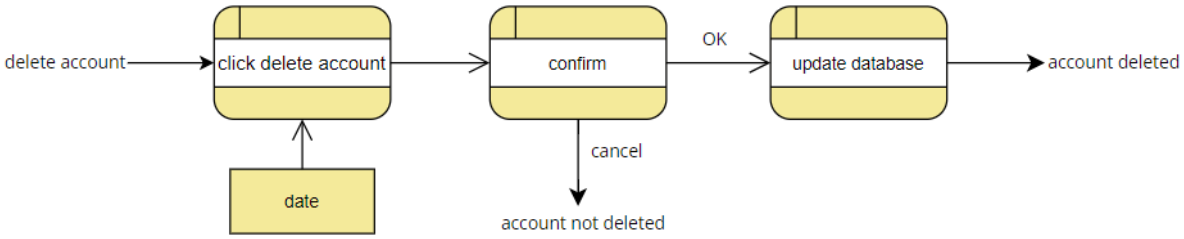


5. Profile

   a. View reservation history

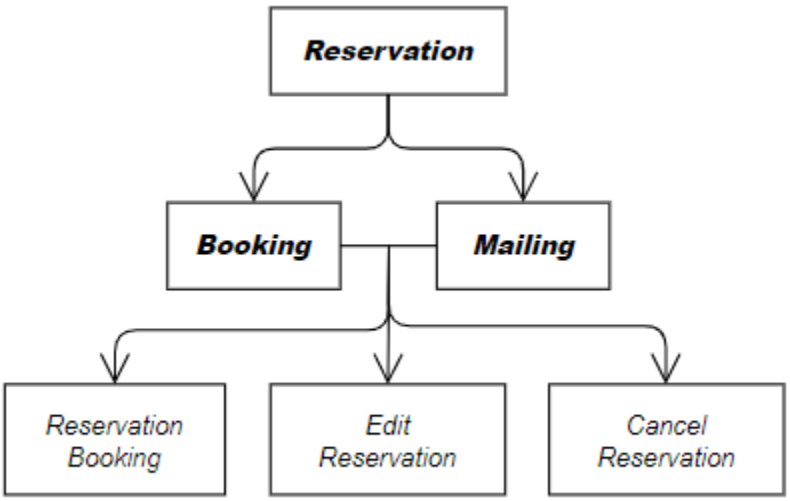


   b. Delete account



Reservation

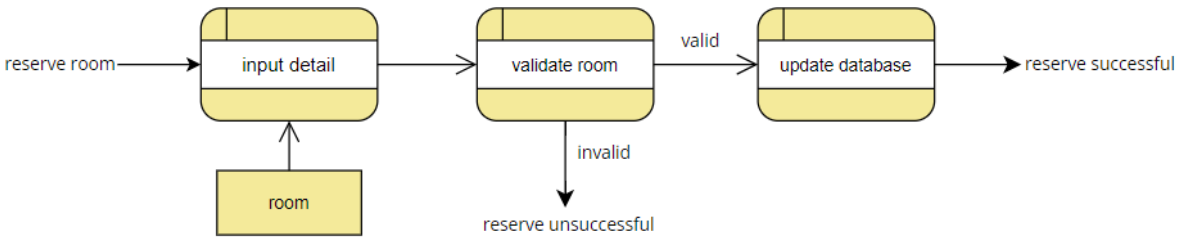
Structural decomposition



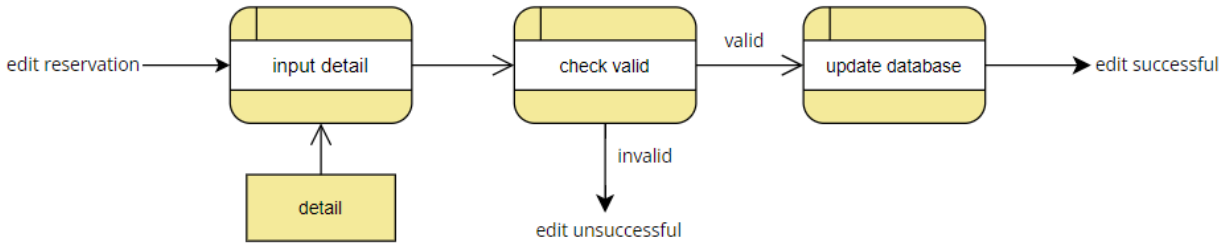
The booking and mailing subsystems run concurrently with both having the same data flow.

Data flow diagrams:

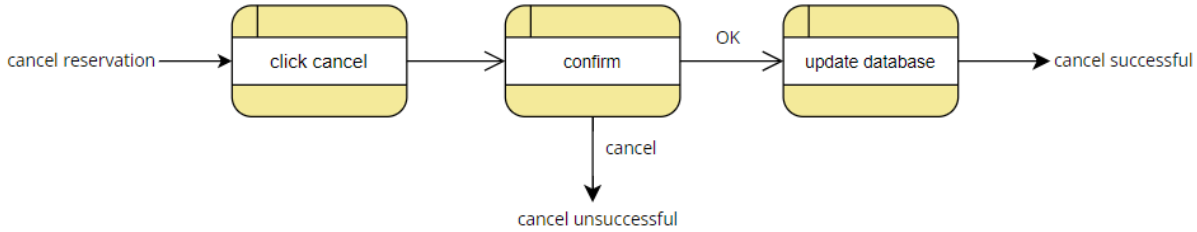
1. Reservation booking



2. Edit



3. Cancel



3.3 Design Rationale

By using a client-server architecture, any activities are conveniently centralized in the server. Multiple users are able to access the services from any device which allows an internet connection and the server can be easily managed and updated because of its centralized architecture.

However, this convenience comes with several trade-offs. For example, if the main server encounters a problem which halts its operation, other systems which depend on it would follow suit. This stops the entire server operations. The distribution of the service throughout various devices is also quite complicated as each device comes with different specifications.

One of the other architectures that was taken into consideration is the Peer-to-peer(P2P) architecture. Unlike client-server architecture, any individual component acts as both client and server. In this scenario(room reservation), P2P is not

preferable as it could cause traffic due to the possibility of unnecessary repetitive database querying.

4. DATA DESIGN

4.1 Data Description

The database of the system is stored into tables of objects. These objects are defined from two classes: User Class and Reservation Class. Their attributes are as below.

User Class attributes:

- id: Primary key
- username: User's username (unique)
- name: User's name
- email: User's email (unique)
- admin: Access to admin features

Reservation Class attributes:

- id: Primary key
- username: Booker's username
- room_id: Chosen room number
- booking_time: Time when booking is done
- booked_date: Chosen reservation date
- time_start: Reservation starting time
- time_end: Reservation ending time
- _party: Participants username and name represented as a string
- status: Reservation's current status represented by an integer (0: upcoming, 1: ongoing, 2: expired)

Both classes have IDs which act as their primary key in the database table. All users are defaulted to 'not an admin' and all reservations are defaulted to 'upcoming reservations'.

4.2 Data Dictionary

Alphabetically sorted list of all the attributes, functions and function parameters

R

Reservation: class

Reservation._party: string

Reservation.booked_date: date

Reservation.booking_time: datetime

Reservation.id: integer

Reservation.party(): member function

Reservation.room_id: integer

Reservation.status: integer

Reservation.time_start: time

Reservation.time_end: time

Reservation.username: string

U

User: class

User.admin: boolean

User.email: string

User.id: integer

User.is_authenticated(): member function

User.name: string

User.password: string

User.username: string

5. COMPONENT DESIGN

5.1 Overview of components

In the interface subsystem will handle the display information and it will be responsible to get user inputs and send them to the admin system. The admin subsystem will process all the inputs. Last, inputs from the user and process requests which are being processed connect to the database.

5.2 Summary of Algorithms

Routing algorithm

All actions are accessed through their own routes. Below is the general routing algorithm.

```
        action_route():
    display_action_form(form)
    if(form.is_submitted):
        if check_condition(form.data):
            database.commit_changes()
            redirecting_to(next_page)
        else:
            display_error(error)
    redirecting_to(action)
```

Checking for specified conditions

Each action needs to be checked before writing or modifying data to the database.

```
        check_condition(data_to_check):
            check_data = database.fetch(data_to_check)
            If check_data exists:
                return FALSE
            Return TRUE
```

6. HUMAN INTERFACE DESIGN

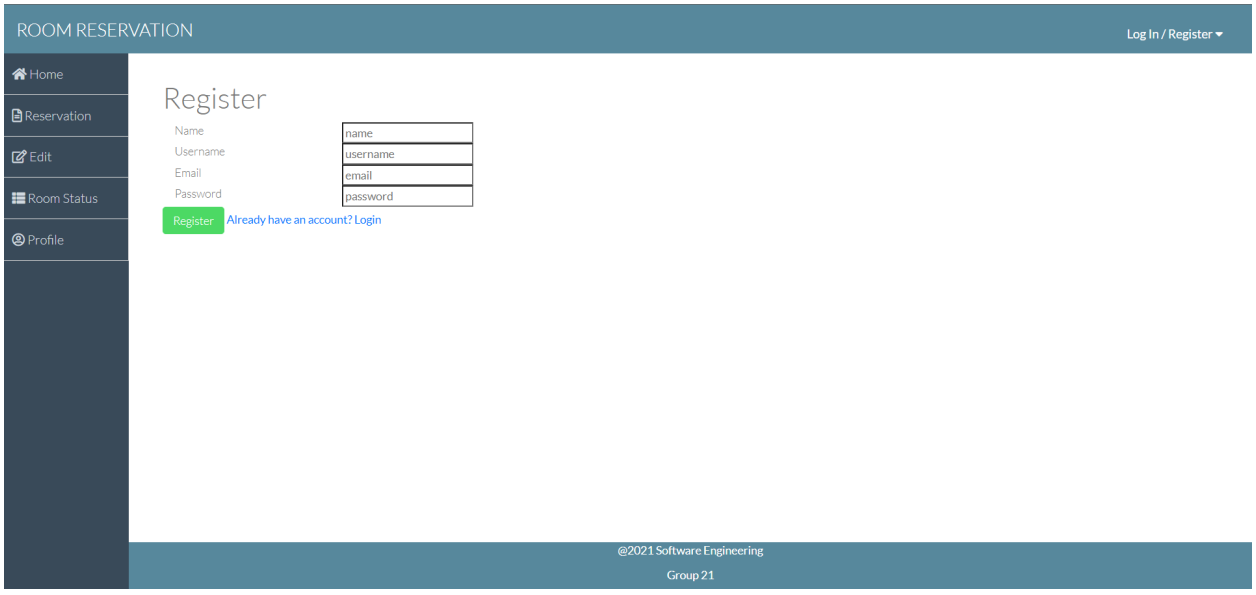
6.1 Overview of User Interface

Used GUI components are menus, submenus, buttons, text boxes, check boxes, links, and tables. The only means of access to the entire database, by all the users, is through this UI.

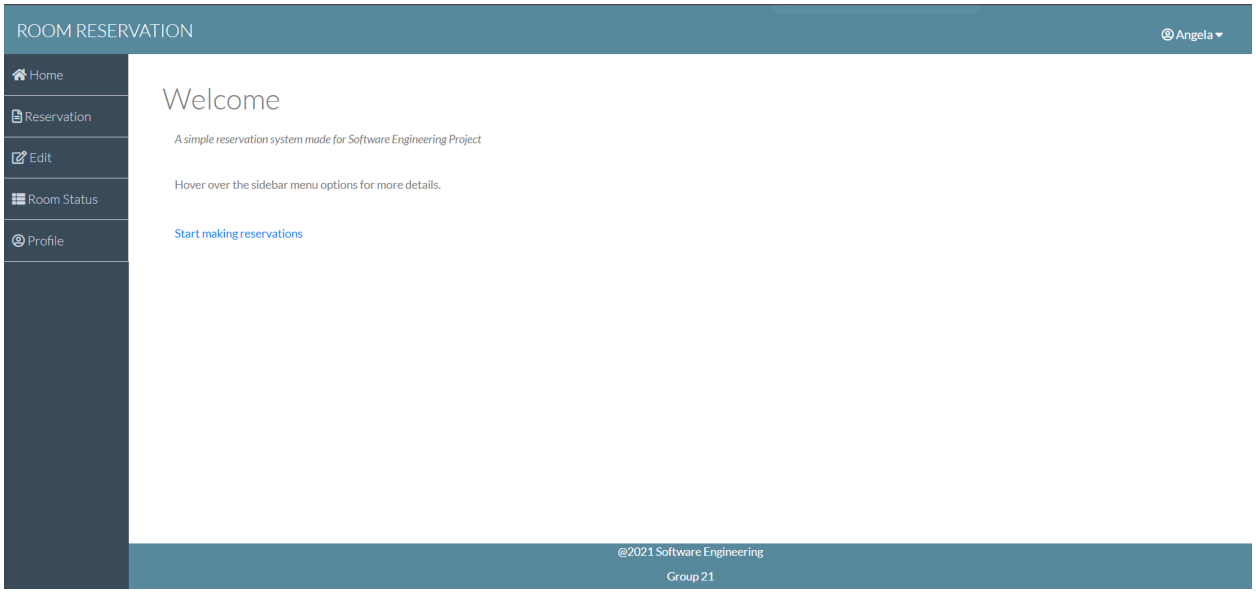
6.2 Screen Images

1. Login Page

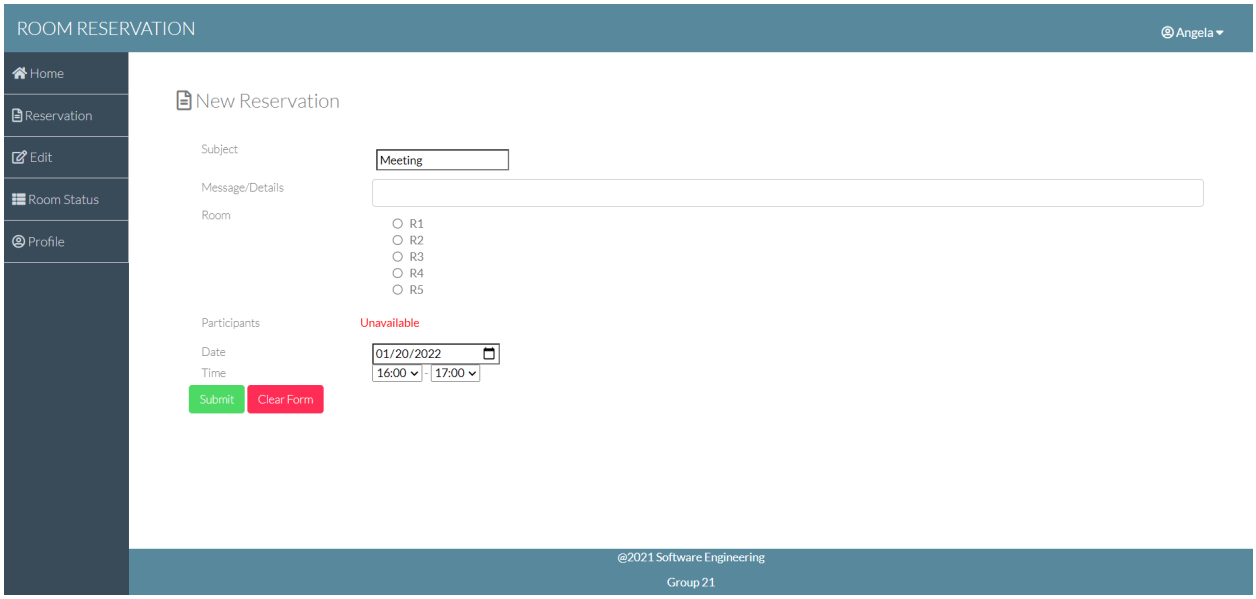
2. Registration Page



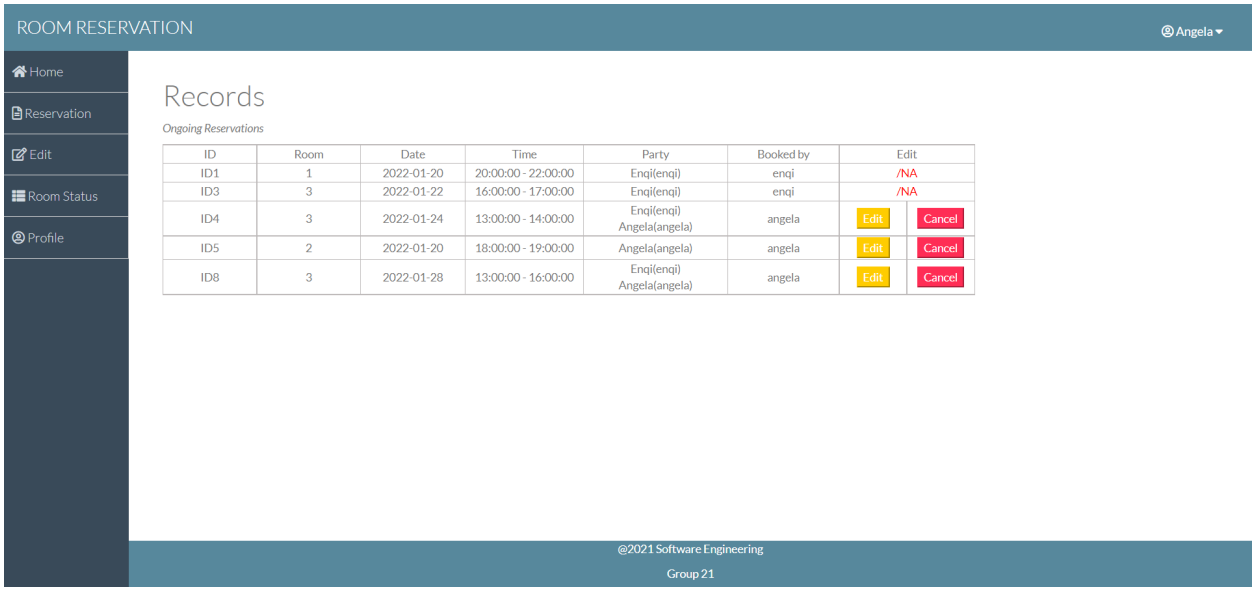
3. Home Page



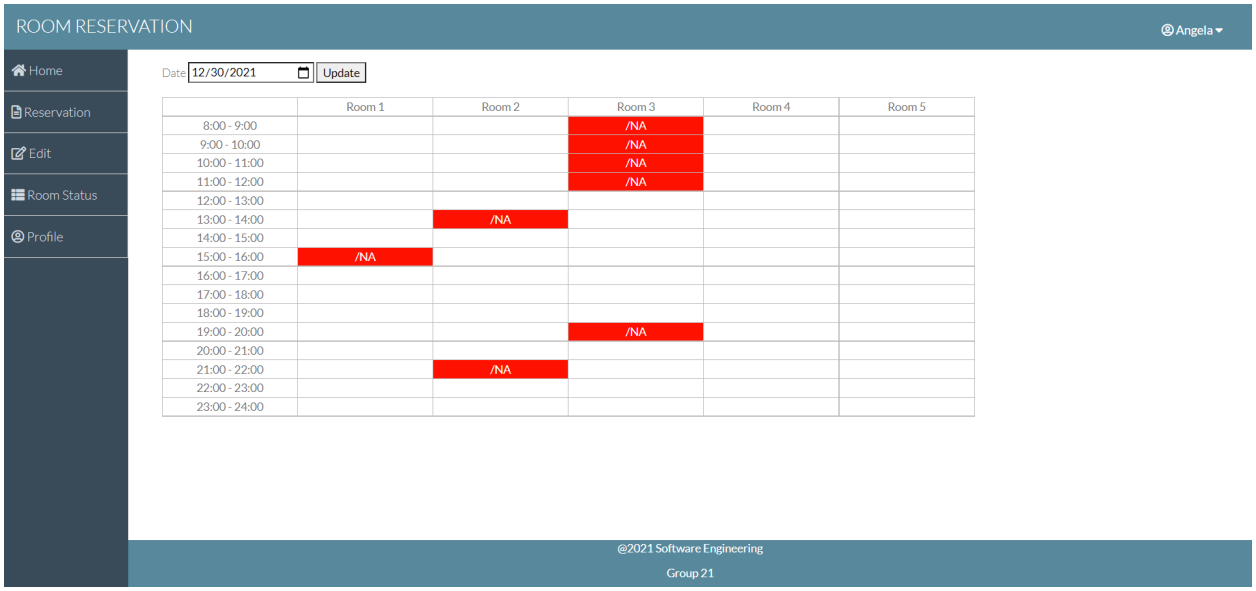
4. Reservation/Booking Page



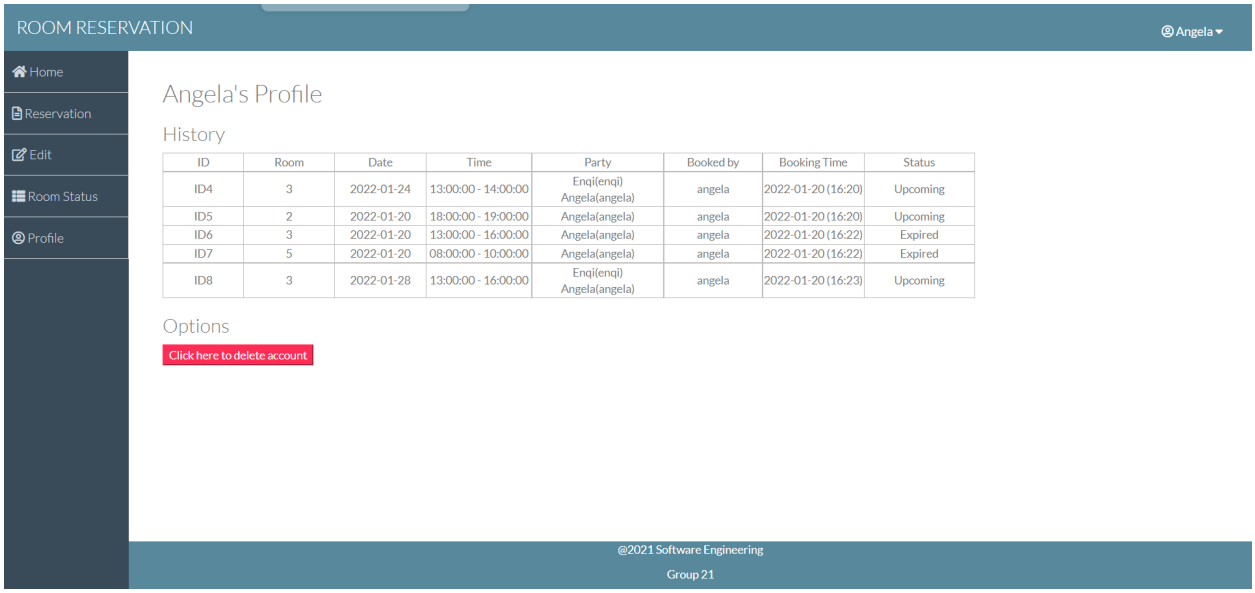
5. Edit Page



6. Room Status Page



7. Profile Page



6.3 Screen Objects and Actions

1. Login Page: Consists a field for users to input username and password, and a button to log users into the system.
2. Registration Page: Consists a field for users to input name, username, email and password, and a button to register a user account into the system.
3. Home Page: Consist a side menu for all the pages and a greeting after users log into the system.
4. Reservation/Booking Page: Consists a field for users to select which room they desired, participants to invite, message or meeting detail, date and time. And it also has a button to submit your reservation and a button to clear all the fields.
5. Edit Page: Consist of all users ongoing reservation, and users can edit their meeting details or cancel their reservation.
6. Room Status: Consists a field for users to input the date that they desire and a button to update the table to show all the ongoing reservation of each room for selected date.
7. Profile Page: Consists the user reservation past and ongoing reservation. It also has a button for users to delete their account.

7. REQUIREMENTS MATRIX

SRS Section	Requirement	Description	SDD Section	
			Data Flow Diagram	User Interface Design
3.1.1	Home Page	Design “main” page	-	6.2.3
3.1.2	Reservation Form Page	Design page for room reservation	-	6.2.4
3.1.3	Edit Page	Design page for user to edit the reservation room	-	6.2.5
3.1.4	Room Status Page	Design page that show the selected rooms’ status corresponding to the selected date.	-	6.2.6
3.1.5	Profile Page	Design page to show the list of user’s reservation	-	6.2.7
3.2.1	Login Page	Design page for user to login	-	6.2.1
3.2.2	Register Page	Design page for user to register account	-	6.2.2