

Software Architecture Document

for

Virtual Room Reservation Assistant

Version 2.0 approved

Prepared by:

B10504028 林哲豪

B10601120 許世佑

B10632011 陳彥瑜

B10632037 潘莫同

CS3025301 Software Engineering

2020/12/13

Table of Contents

Table of Contents	1
Revision History	2
1.Introduction	3
1.1 Purpose	3
1.2 Scope	3
1.3 Definitions, Acronyms and Abbreviations	3
1.4References	4
1.5 Overview	4
2.Architectural Representation	4
3.Architectural Goals and Constraints	5
3.1 Performance	5
3.2 Safety	6
3.3 Security	6
3.4 Privacy	6
3.5 Use of an Off-the-shelf Product	6
4.Use-Case View	7
4.1 Create Account	7
4.2 Login	7
4.3 Create Meeting	8
4.4 Cancel Meeting	9
4.5 Create Google Calendar Reminder	10
5.Logical View	10
5.1 Overview	11
5.1.1 Subsystems	11
5.1.1.1 User management subsystem	11
5.1.1.2 Meeting management subsystem	11
5.1.1.3 Meeting Room Management subsystem	11
5.1.2 Layering	11
5.2 Architecturally Significant Design Packages	13
5.2.1 Class Diagram	13
5.2.1.1 Model related class	13
5.2.2 Package Diagram	14
6. Process View	14
6.1 form create process	14
6.2 login process	15
6.4 forget password recovery process	16
6.5 edit meeting information process	16
7. Deployment View	17
8. Implementation View	18
8.1 Overview	18
8.2 Layers	18

9. Size and Performance	19
9.1 Volumes	19
9.2 Performance	19
10. Quality	19
10.1 Security	19
10.2 Reliability	20
10.3 Performance	20
10.4 Portability and reuse	20

Revision History

Date	Version	Description	Author
12/06/2020	1.0	first copy	B10632037
12/12/2020	2.0	reformat	B10504028

1.Introduction

This software will be a virtual room reservation system that provides users with reliable information so that they can track events, record messages, and manage meeting information. More specifically, this system is designed to allow a user to manage and communicate with members and publish meeting information to members by Email.

This document elaborates the software architecture document for the system “Virtual Room Reservation Assistant”.The system architecture is abstracted into many views and components which are explained in detail.The 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.Using 4+1 view model makes it possible to depict the software as accurately as possible.It allows a wide range of stakeholders to find what they require in the architecture document.

1.2 Scope

The software architecture document applies to each static and dynamic aspect of the system.Since 4+1 view model is used as the reference model. It incorporates many views of the system, thus makes the document complete and consistent.

Under the static behavior of the system, the document discusses the class diagrams, package diagrams and other static architecture designs. Dynamic aspects of the system are elaborated using use case realizations and system sequence diagrams.

1.3 Definitions, Acronyms and Abbreviations

Term	Definition
Physic Room	Meeting or discussing room in physics.
Virtual Room	Meeting or discussing rooms created by web conference software ,such as Zoom, Cisco Webex, Google Meet, which allow users to communicate and make discussion online.
Online calendar	An online application which can edit what, when and where an event is going to happen by user or other application.

Web browser	The application which can make users access to a web page or a web app.
Database	An organized collection of data which allows the user or other applications access electronically.
Web API	Application Programming Interface for a web service.
WSGI	Web Socket GateWay, an interface for web development framework to communicate with web server
template engine	A component of web development framework, which is used to change the context of certain placeholders to desired html code.

1.4References

SW Engineering Project List by Professor Kaliski

<https://www.slideshare.net/PasinduTennage/sample-software-architecture-document>

For more information, please refer to the manual referenced above.

1.5 Overview

The report will present a detailed analysis of the architecture of Virtual Room Reservation Assistant. Further section cover the architecture representation of the project including architectural representation, architectural goals and constraints and use case realizations. The later sections cover the detailed specific details of 4 main views (logical view, process view, deployment view and implementation view) of the system. Furthermore, the document describe the data view and performance, quality views.

2. Architectural Representation

This document presents the architecture as a series of views; use case view, process view, deployment view, and implementation view. These views are presented as the Unified Modeling Language (UML).

Logical view

Audience: Designers, Programmers, Testing staff

Area: Functional Requirements: describes the design's object model. Also describes the most important use-case realizations.

Related Artifacts: Design model

Process view

Audience: Programmers, Integrators

Area: Non-functional requirements: describes the design's concurrency and synchronization aspects. Elaborates the run time behavior of the system.

Implementation view

Audience: Programmers.

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

Related Artifacts: Implementation model, components

Use Case view

Audience: all the stakeholders of the system, including the end-users.

Area: describes the set of scenarios and/or use cases that represent some significant, central functionality of the system.

Related Artifacts : Use-Case Model, Use-Case documents

Physical view

Audience: Database administrators, System engineers, Deployment managers

Area: Persistence: describes the architecturally significant persistent elements in the data model. Describes the mapping of the software onto the hardware and shows the system's distributed aspects

3. Architectural Goals and Constraints

3.1 Performance

Our system is available through web browser. The system may have multiple online users using the system simultaneously. All action about connecting to users' online calendar account should be done without too much lag. The notification of successfully booking a room should be sent to the attendee and host in 30 minutes after the users successfully book a room. A room reservation must be upgraded to the database immediately without obviously lag.

3.2 Safety

Our system will ask the user to link their online calendar account. The system can't access any information that users don't want to provide and also can't require any unnecessary authority. Our system may ask the user to share their calendar, so the other user can know if a user will be free for meeting.

3.3 Security

Our system will not store the information that user provide include their email address, phone number, calendar etc. for improper usage. The virtual room is created by external web conference software, therefore our system won't use any method to monitor users' meeting. However, we can't guarantee that user's online meeting won't be monitor through the web conference software.

3.4 Privacy

This service would be installed on a local machine for testing. The database should be installed on a cloud environment for testing. User's password must be encrypted before being stored into a database to ensure the security of the user's personal data. The webpage should be compatible with common web browsers like Google Chrome, Apple Safari, Microsoft Edge.

3.5 Use of an Off-the-shelf Product

The only external system of this service is online calendar, the app should connect to the user's online calendar service, such as Google calendar, outlook etc., with web API provided by these cloud services. Web API usually comes as REST style, which a standard http request is made and a json response is returned, this service should help users make the request and handle it with the return data.

4.Use-Case View

4.1 Create Account

Use Case Name	Create Account
Priority	Essential
Trigger	The User selects to create a new account to the database.
Precondition	The User has accessed the login page.
Basic Path	<ol style="list-style-type: none">1. Click “Signup” button.2. Fill out the User name.3. Fill out the User’s email.4. Fill out the account number of the User.5. Fill out the password of the User.6. Fill out the password again.7. Click “Create” button.
Alternative Paths	None.
Postcondition	The User’s account has been added to the database.
Exception Paths	The User may abandon the operation at any time.

4.2 Login

Use Case Name	Login
Priority	Essential
Trigger	The User selects to login his / her account.
Precondition	The User has accessed the login page.
Basic Path	<ol style="list-style-type: none"> 1. Fill out the account number of the User. 2. Fill out the password of the User. 3. Click “Login” button.
Alternative Paths	None.
Postcondition	The User has logged in his / her account.
Exception Paths	The User may abandon the operation at any time.

4.3 Create Meeting

Use Case Name	Create Meeting
Priority	Essential
Trigger	The User selects to create the meeting.
Precondition	The User has accessed the Home page.

Basic Path	<ol style="list-style-type: none"> 1. Click right most button on the top right corner. 2. Fill out title of the meeting. 3. Fill out start time of the meeting. 4. Fill out end time of the meeting. 5. Fill out participants of the meeting. 6. Choose an available room or create a virtual room. 7. Click “Create” button.
Alternative Paths	None.
Postcondition	The meeting information has been added to the database.
Exception Paths	The User may abandon the operation at any time.

4.4 Cancel Meeting

Use Case Name	Cancel Meeting
Priority	Essential
Trigger	The User selects to cancel the meeting.
Precondition	The User has accessed the meeting information page.
Basic Path	<ol style="list-style-type: none"> 1. Select unwanted meeting. 2. Click “Cancel” button. 3. Fill out the title of the unwanted meeting. 4. Fill out the start time of the unwanted meeting. 5. Fill out the end time of the unwanted meeting. 6. Click “Send” button.
Alternative Paths	None.

Postcondition	The meeting information has been removed from the database and it will be sent the notification to participants.
Exception Paths	The User may abandon the operation at any time.

4.5 Create Google Calendar Reminder

Use Case Name	Create Google Calendar Reminder
Priority	Essential
Trigger	The User selects to create Google Calendar Reminder.
Precondition	The User has created the meeting.
Basic Path	1. Fill out the reminder time. 2. Click “Confirm” button.
Alternative Paths	None.
Postcondition	The Google Calendar Reminder has been created and it will remind User the meeting at the corresponding time.
Exception Paths	The User may abandon the operation at any time.

5.Logical View

A description of the logical view of the architecture. Describes the most important classes, their organization in service packages and subsystems, and the organization of these subsystems into layers. Also describes the most important use-case realizations, for example, the dynamic aspects of the architecture. Class diagrams may be included to illustrate the relationships between architecturally significant classes, subsystems, packages and layers.

5.1 Overview

5.1.1 Subsystems

System (Virtual Room Reservation Assistant) can be divided into three main subsystems.

1. User Management System
2. Meeting Management System
3. Meeting Room Management System

5.1.1.1 User management subsystem

This system provides facilities that cover all the login management functionalities. Main use case that comes under this subsystem includes

1. user login
2. new user creation
3. forget password

Each user that interacts with the system is an actor of these use cases. Depending on the accessing levels, some actions will not be permitted.

5.1.1.2 Meeting management subsystem

This subsystem involves the main operations on the Meeting object. Main use case that comes under this subsystem includes

1. add new meeting
2. modify/delete meeting
3. create google calendar reminder

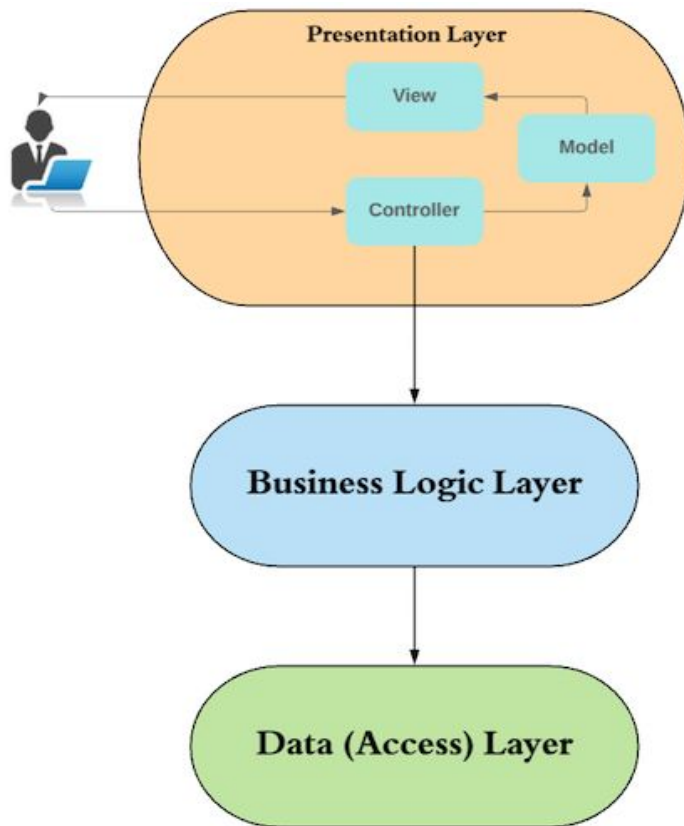
5.1.1.3 Meeting Room Management subsystem

There are two kinds of meeting rooms in this service: virtual meeting room and physical meeting room. The following operation depends on which kind of meeting room they are

1. adding meeting room
2. deleting meeting room
3. allocate meeting room to a meeting

5.1.2 Layering

Virtual Room Reservation Assistant system is divided into three main layers based on MVC framework. The layering model of the Virtual Room Reservation Assistant application is based on a responsibility layering strategy that associates each layer with a particular responsibility. This strategy has been chosen because it isolates various system responsibilities from one another, so that it improves both system development, reusability and maintenance.



Each layer has specific responsibilities.

The presentation layer is responsible for handling all user interface and browser communication logic

The business logic layer is responsible for executing specific business rules associated with the request

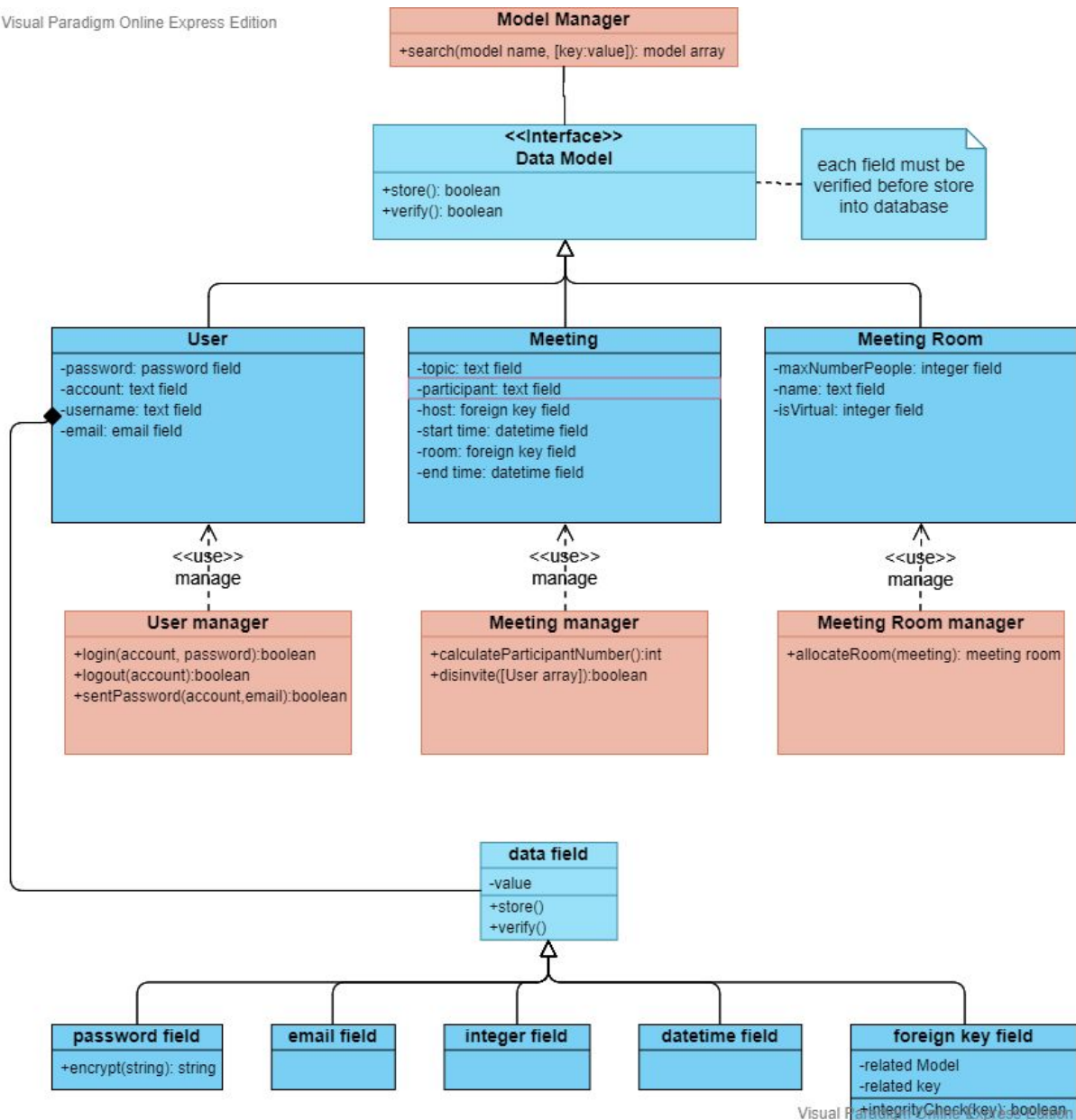
The data (access) layer is responsible for providing access to the persistence layer. This layer is only ever invoked by Service objects. Objects in the data access layer do not know about each other.

5.2 Architecturally Significant Design Packages

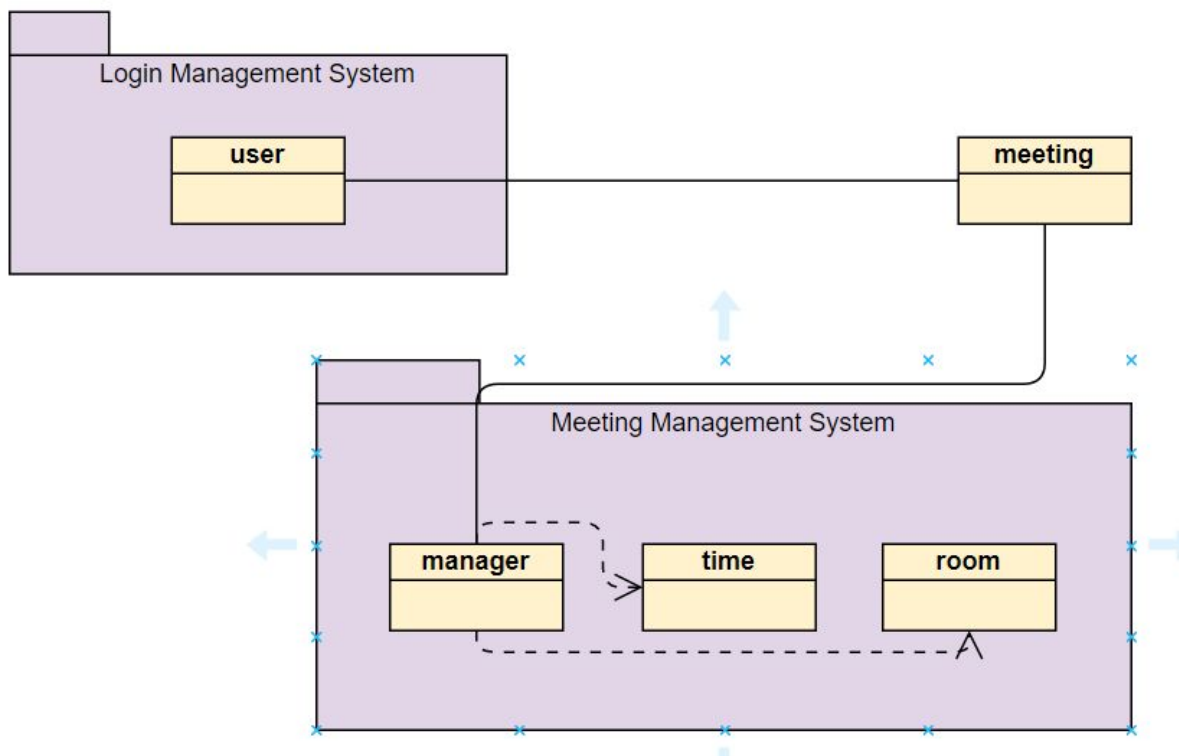
5.2.1 Class Diagram

5.2.1.1 Model related class

Visual Paradigm Online Express Edition



5.2.2 Package Diagram



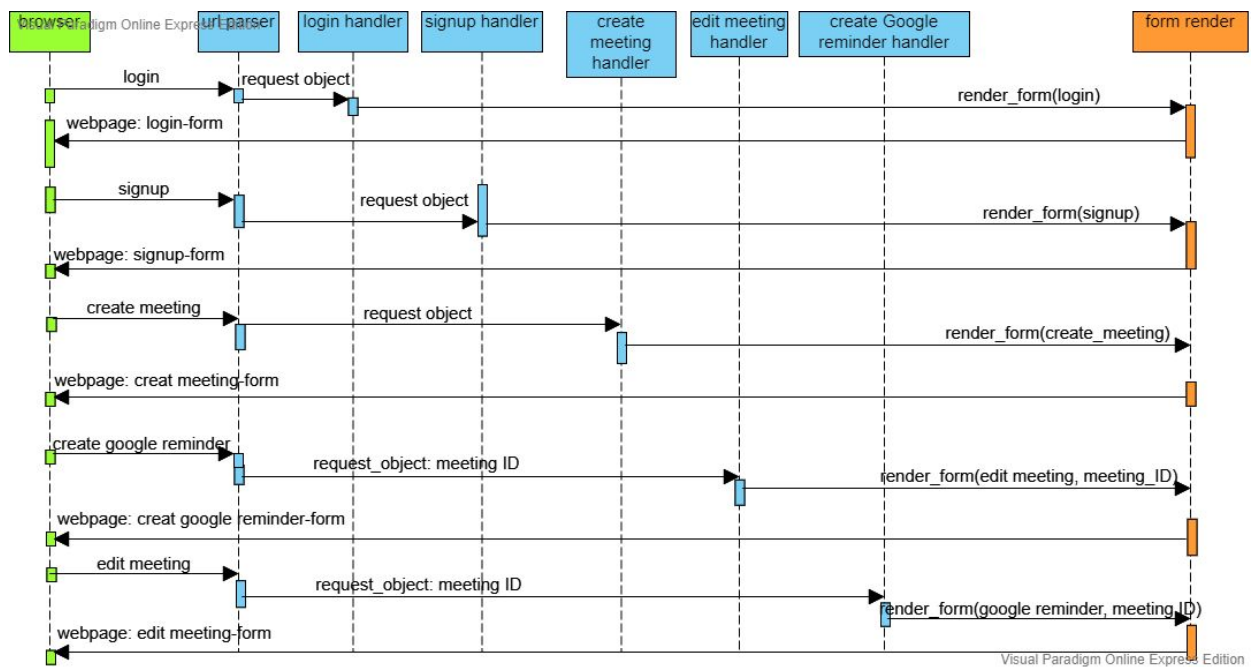
6. Process View

Because there are so many form related operation in this application so we use an abstract form creator relate to a data model to avoid handwrite all of the form. Also, this help us to keep the consistency between the data type of form and database.

So to speak, every form is created from a defined data model and render with webpage renderer.

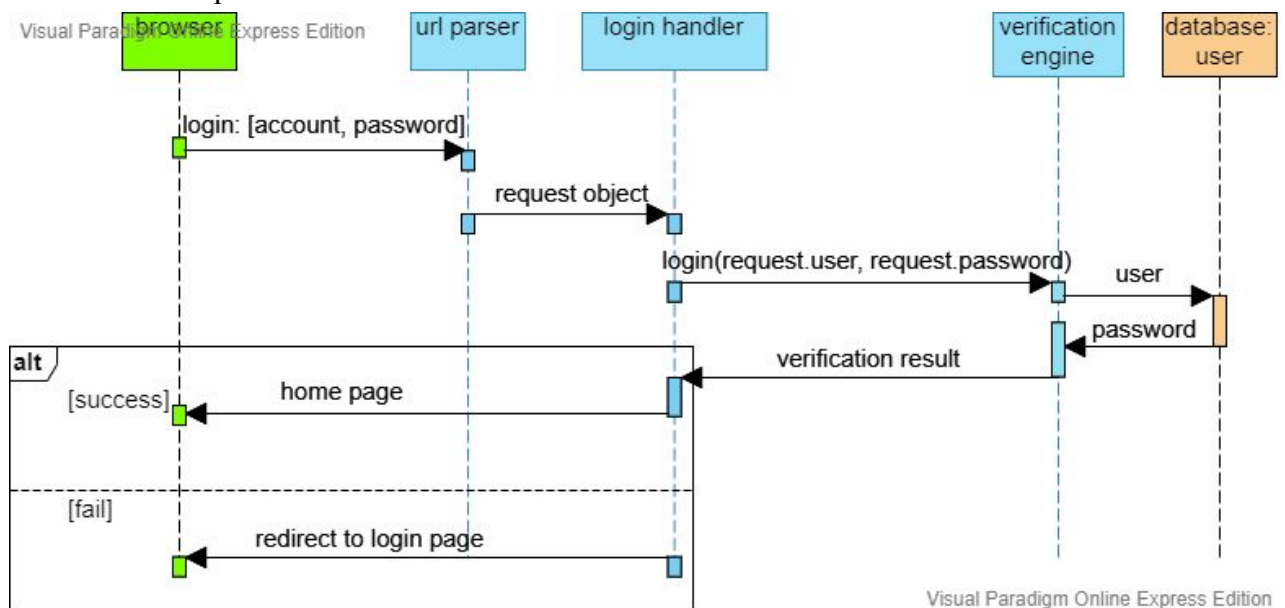
6.1 form create process

Each operation involving form follows a unified form render process. Process like login, signup, create meeting, create Google calendar reminder, edit meeting are all related to form.

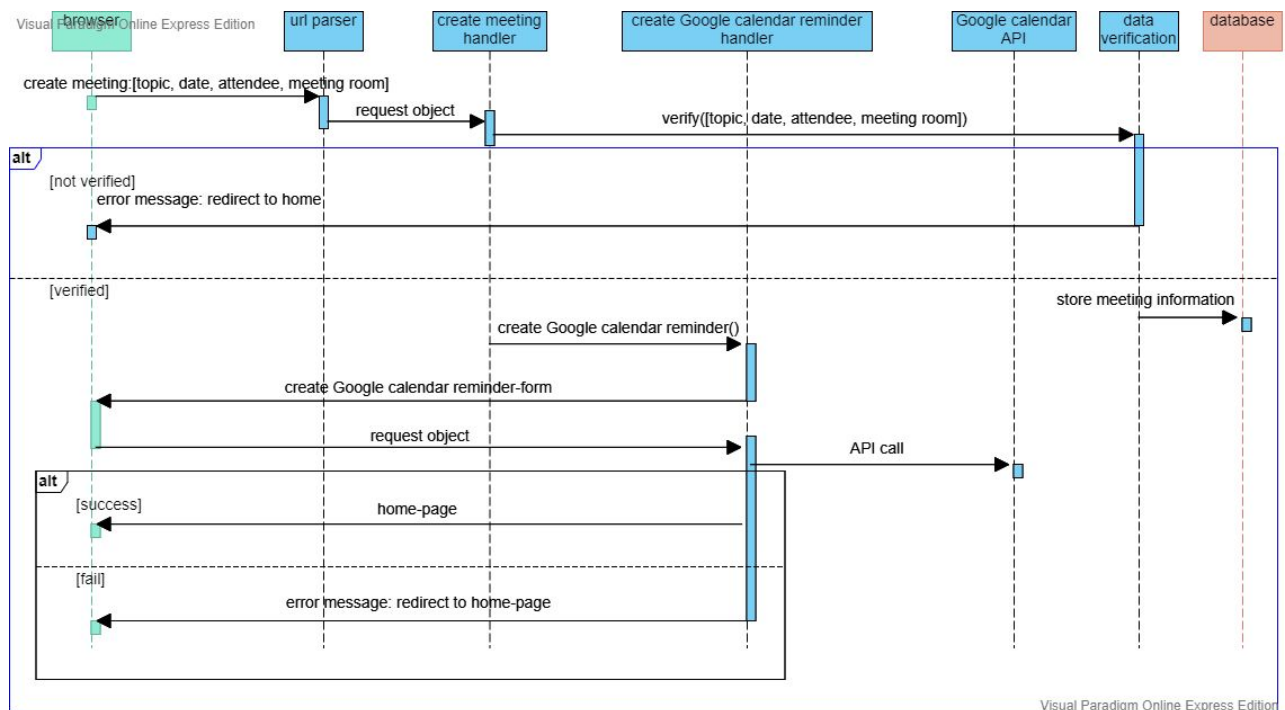


6.2 login process

Since the password stored in database is encrypted, we use a verification engine to decrypting and verified the password.

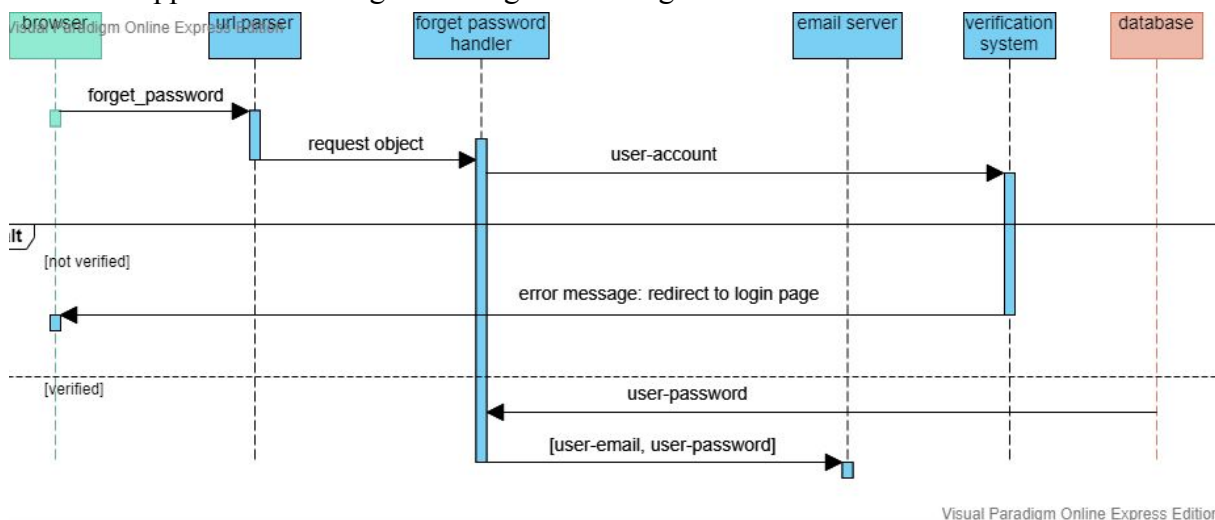


6.3 create meeting process



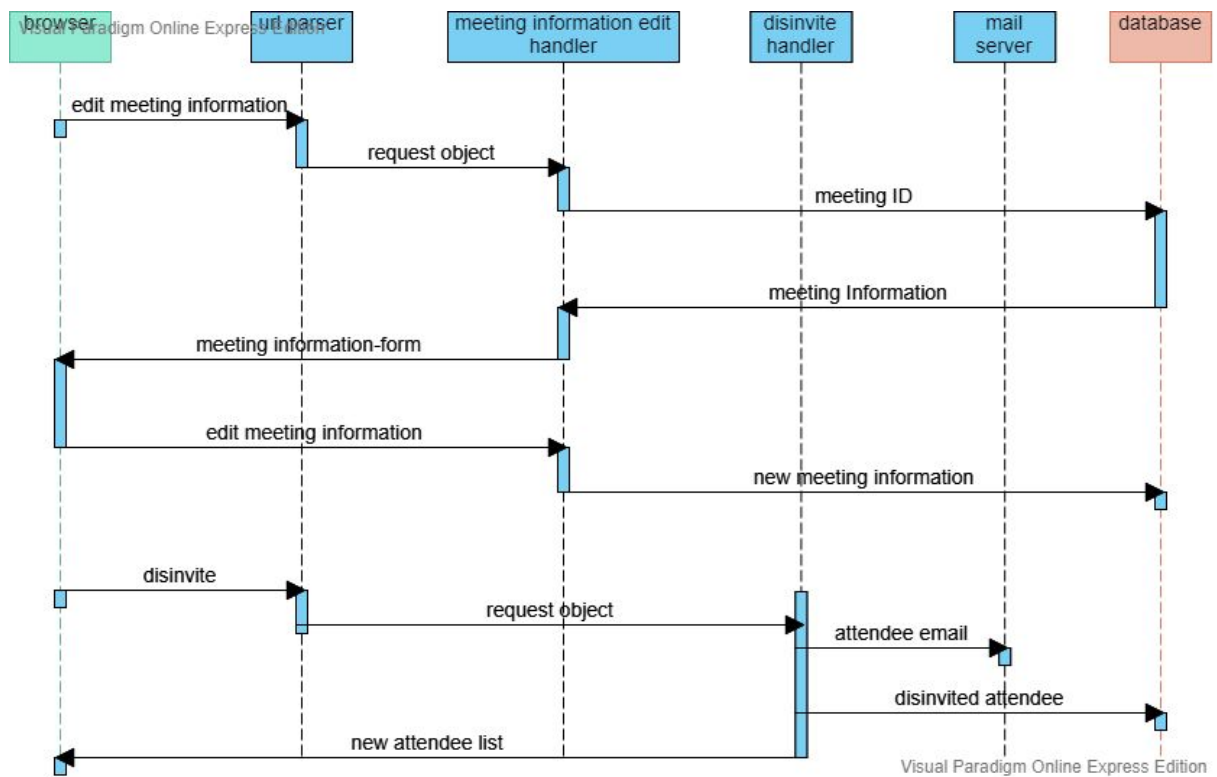
6.4 forget password recovery process

when a user forgets his/her password, we would send him/her his original password. In the sense of security, this method is not best, because the user's email account may be hijacked or eavesdropped. But this is good enough for testing.

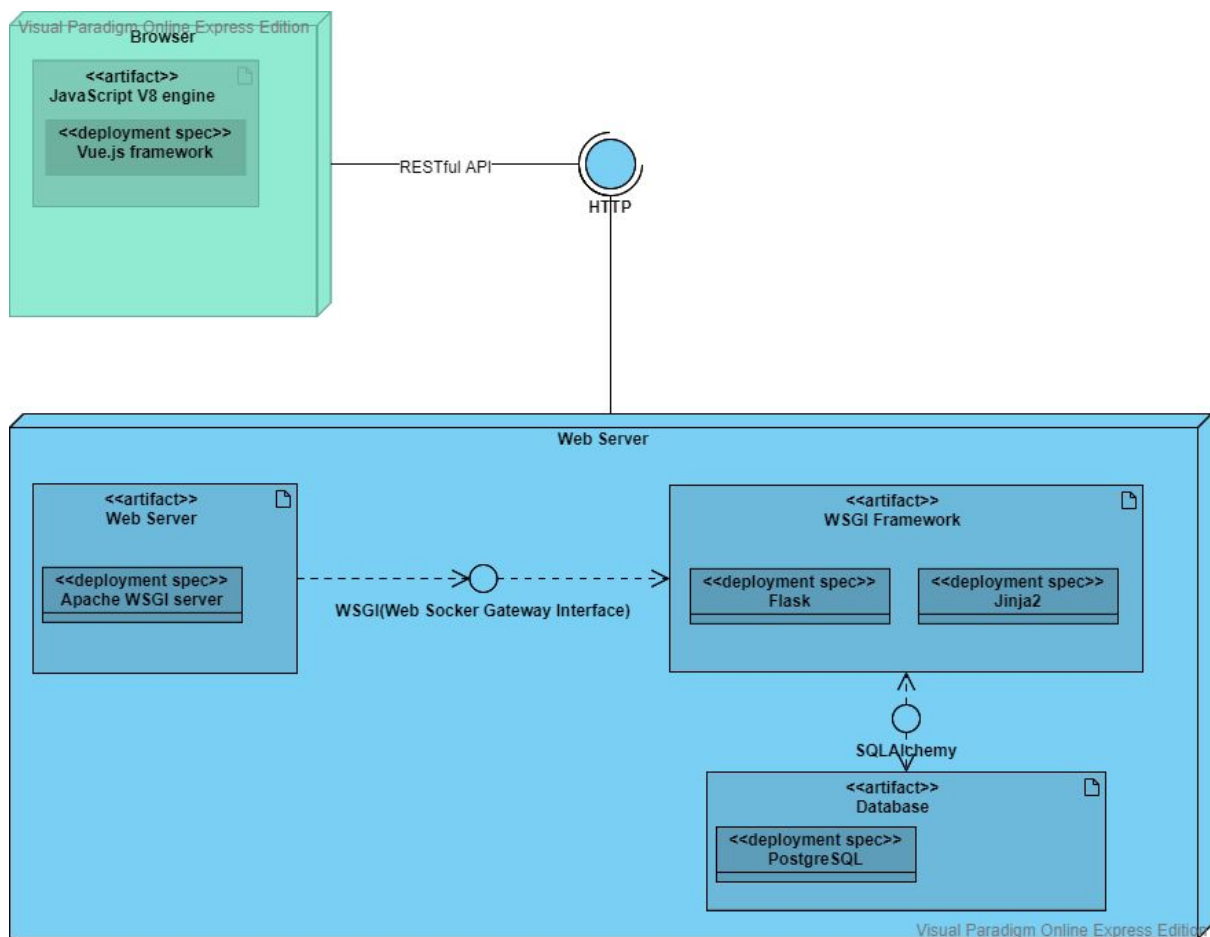


6.5 edit meeting information process

This process involves changing the information of a meeting and disinviting some of the attendees. When disinviting the attendees, the web page shouldn't wait until the background operation is done, it should keep responding to the user until the server sends back an event to interrupt, which is known as asynchronous request.



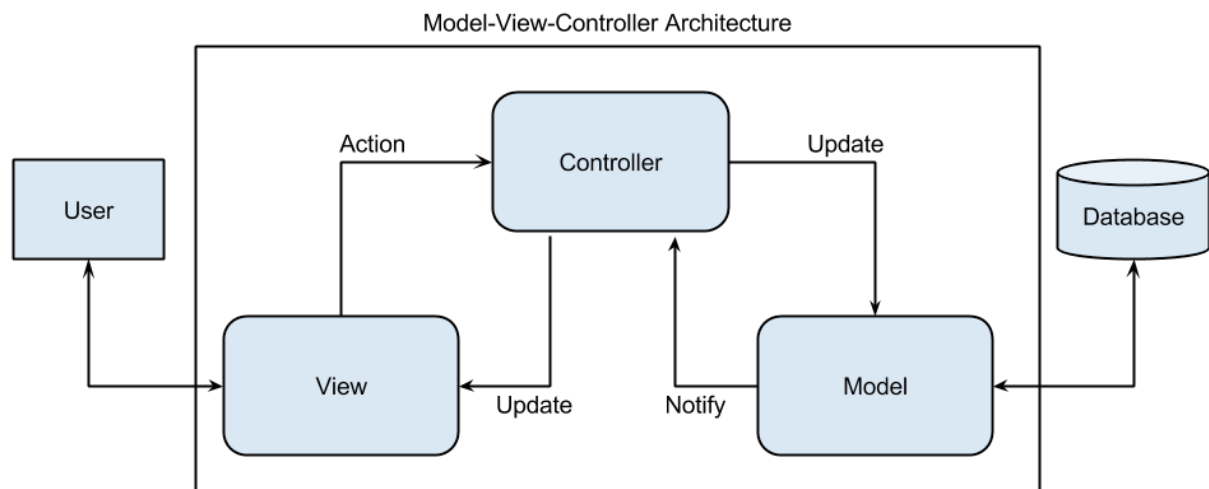
7. Deployment View



We use client-server model in this service, a SPA(Single Page Application) would transfer to user and the client node communicate with server with RESTful API. On the server side we separate the service into three layer: web server, application layer and database layer. Web server play a role between client and application, application layer handle all of the API functionality and bussiness logic such as arrange meeting room for each meeting. Database would be completely seperated from application for security and responsibility decoupling issues.

8. Implementation View

8.1 Overview



Virtual Room Reservation System is based on Flask , therefore still follows the MVC patterns. The advantages of using MVC architecture is that the modification never affects the entire model and faster web application development process.

View ,which is client layer, is a part of MVC architecture.Client layer as the name implies that is for user. User can make Reservation System on this section.and the actions that user made will handle by Controller ,which is business layer. Business layer connect with data layer and client layer. It can pass data from Database to view,and vice versa.

8.2 Layers

Subject Subsection	Admin Subsection
Create subject Delete subject View subject	User account creation User account authentication

Layering architecture is based on use case classes. Subject subsection is responsible for tasks and links creation, modification, deletion, and so on. Administration subsection is responsible for user account creation, authentication, and login.

9. Size and Performance

The size of the software is still not calculated, but it is in the range 100-200MB. The MYSQL database will be hosted in another hosting space.

The client computers need to have a web browser in order to access the system. All the functionalities will be processed at the backend.

9.1 Volumes

1. Reservation System registered individual customer : about 10
2. Reservation System corporate customers if non-networked and not-shared : about 10
3. Reservation System corporate customers if accounts enable separate IDs : about 30
4. Reservation System maximum reservation task : about 30
5. Reservation System maximum invitation: about 30

9.2 Performance

1. Time from click the “建立會議” to success build new meeting to database: less than 10 seconds req.
2. Time from click the “取消會議” to success delete the meeting to database: less than 10 seconds req.
3. The system responds to any request under standard database and web server script timeouts (30 seconds)
4. system performance can depend on available hardware, network and internet connection capabilities
5. If the file size increases by N bytes, latency increases within acceptable range.
6. If the users increases by M, latency increases within acceptable range.
7. If the network connection latency increases, software would convert to offline mode if latency > threshold.

10. Quality

10.1 Security

All passwords for Reservation System User will be encrypted with high security. In collaboration mode, the admin or manager can authorize other users to access the Reservation System with different level of authorization.

10.2 Reliability

We will make sure every Reservation System is well protected and modified without losing consistency and data integrity.

10.3 Performance

The system responds to any request under standard database and web server script timeouts (5 seconds), also system performance can depend on available hardware, network and internet connection capabilities. Actual performance can be determined only after the system is deployed and tested.

10.4 Portability and reuse

Our Virtual Reserve Room is designed as a complete and standalone product, it's possible to extend the product, since our functionalities are very well structured and layered.