Flight Management System Design

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Issue** | **Description** | **Author** |
| 10/22/2018 | Set up environment | Set up and build structure source code  Install and config Git  Design and build up business model and class diagram, do annotation validation | Bon Nguyen,  Nghia Nguyen,  Hiep Nguyen |
| 10/23/2018 | Define Template view | Using Tiles Resolver for template view | Hiep Nguyen |
| 10/23/2018 | Login/ Logout | Implement Authentication and Authorization for Admin | Hiep Nguyen |
| 10/23/2018 | Airport Management | Create/ Edit Airport | Nghia Nguyen |
| 10/23/2018 | Multi-language | English/ Vietnamese, using interceptor | Nghia Nguyen |
| 10/23/2018 | Airplane Management | RESTFul/AJAX create/ update airplane | Bon Nguyen |
| 10/23/2018 | Exception Handler | Catch all the exception using @ControllerAdvice | Bon Nguyen |
| 10/24/2018 | Booking and View Booking | Using Spring Web Flow and @Query | Hiep Nguyen |
| 10/24/2018 | Airline Management | Create / Edit Airline | Hiep Nguyen |
| 10/24/2018 | Flight Management | Create / Edit Flight | Nghia Nguyen |
| 10/24/2018 | Custom Validation,  Custom Formatter | Custom validation and formatter for creating new flight | Nghia Nguyen |
| 10/24/2018 | Chat Support  NoticeInterceptor | Implement WebSocket/SockJS for chatting between customer and supporter  Implement NoticeInterceptor for publish for special blurb message to user | Bon Nguyen |
|  |  |  |  |

1. **Table of Contents**

[**Introduction**](#_7rdifxxo9txq) **4**

[Purpose](#_uqe3a04mttu2) 4

[**Requirements**](#_v8bf8kzf9hq2) **4**

[**High Level Design**](#_2et92p0) **4**

[**Detailed Design**](#_i4nj0dwfkpv) **6**

[Static Diagram](#_dpckvuz6zf3x) 7

[**Issues, Risk and Dependencies**](#_cxgh8ibfrjdc) **7**

[**Technologies**](#_yl777gr21re1) **7**

[**Future Considerations**](#_rrms1as910et) **8**

[**References**](#_ohcaimfcqrvb) **8**

# Introduction

We are going to design and implement an online Flight Management and Booking (FMB) project. This is an engineering proof of concept. It is not a planned marketable project.

The goal is to exercise the Spring MVC technology according to the Web Application Architecture (WAA) course guidelines to validate its use in future projects.

## Purpose

Online FMB is a web application which aims to provide users the ability to view and make a ticket booking for a flight online. Once the booking is successful booked, the user will receive a confirmation code [auto generated by the system]. The user then can use this code to view their booking information at anytime by providing the system the confirmation code for security check.

The system also provides back-office features for system administration purpose. These features are provided for authorized persons only, who have to provide username and password for authentication & authorization at the first time accessing the system

# Requirements

The main requirement for this project is

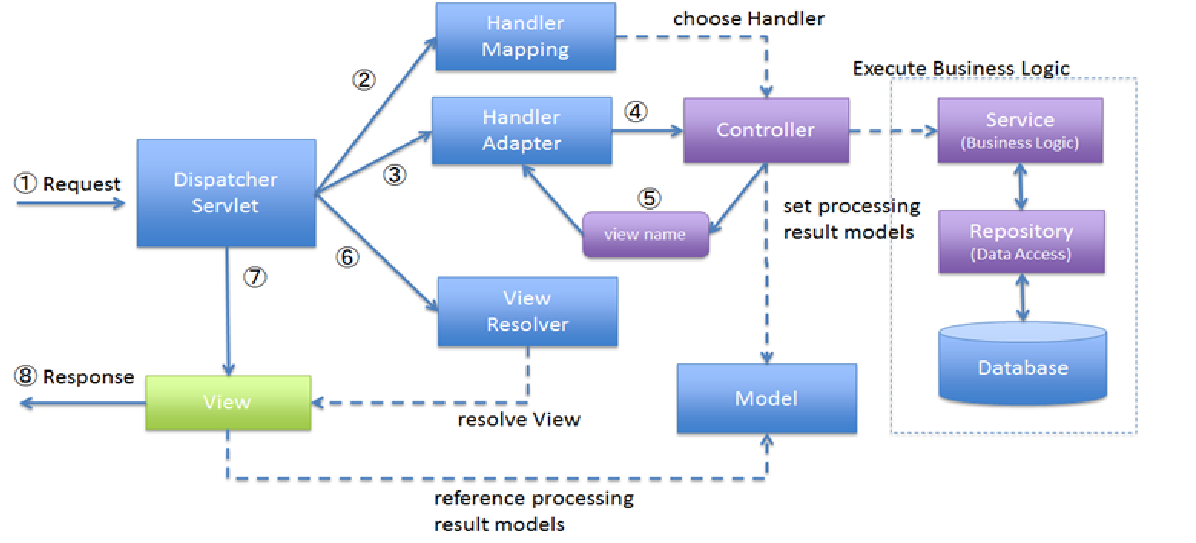
* For User
  + The ability to View scheduled flights - The system should present to the user scheduled flight in the system. So the user can select for booking a ticket
  + The ability to book a ticket for a selected flight - The system should allow user to book ticket for a flight when selected
  + The ability to call for support from system supporter - The should provide user a function that allows user to call for support from system supporter when necessary
* For Admin Portal
  + Create and Update Airline, Airport and Airplane - The system should provide administrator the ability to add new or update existing Airline, Airport and Airplane
  + Schedule new flight - The system should provide administrator the ability to schedule for a new flight
  + Update schedule of existing flights - The system should provide administrator the ability to update the schedule of existing flight
  + Online chat to support user - The system should provide administrator the function that they can online chat to support user when necessary

# High Level Design

Below are main entities designed for this project in order to satisfy project’s requirements

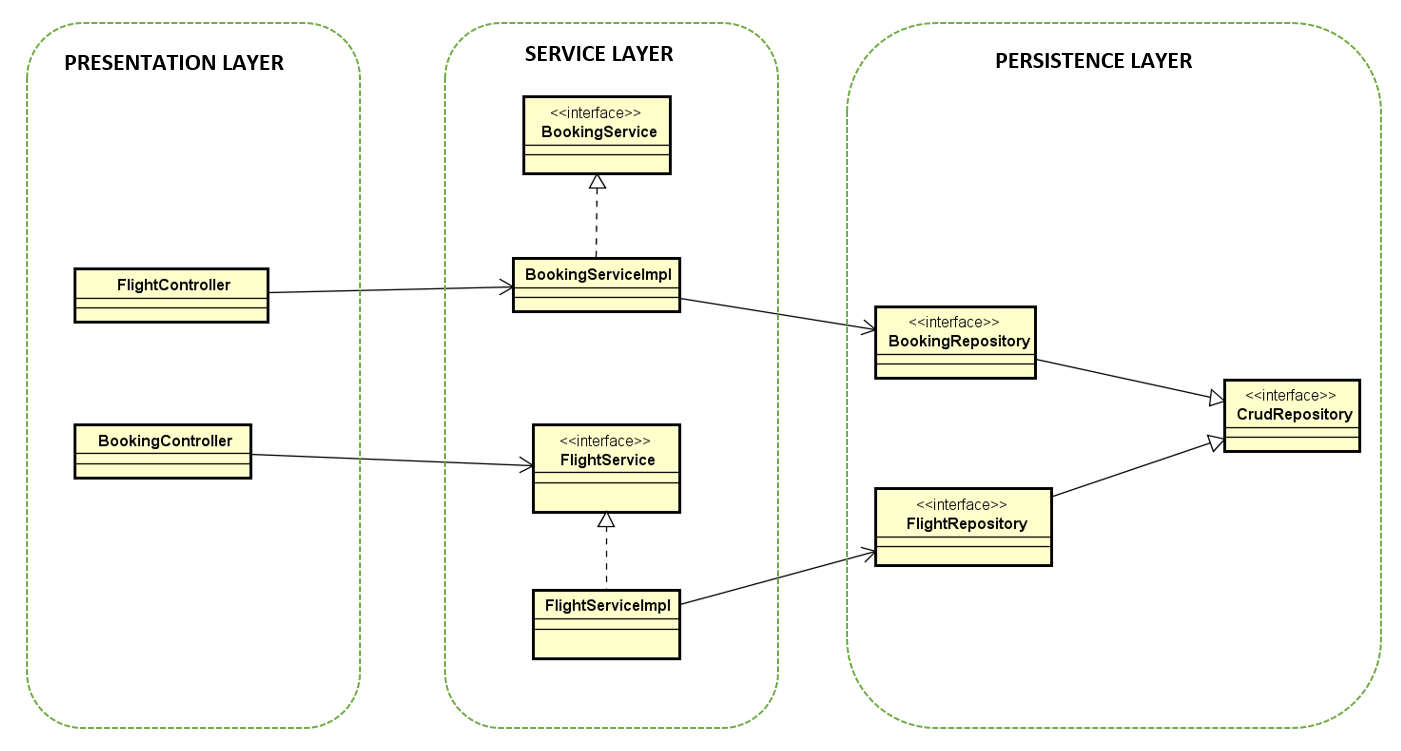
* Airline - This entity contains all information of an airline (e.g. airline name...)
* Airplane - This entity contains all information of an airplane (e.g. serial number, model, capacity...)
* Airport - This entity contains all information of an airport (e.g. airport code, name, city, country...)
* Passenger - This entity contains all information of a passenger who are booking ticket for flights (e.g. passenger first name, last name, passport...)
* Booking - This entity contains all information of booked ticket (e.g. which flight is booked, passenger who book the ticket, booking date, confirmation code...)
* Flight - This entity contains all information of a flight (e.g. flight number, departure date/time, arrival date/time, departure airport, arrival airport, which airplane to be used, from which airline...)

This project is designed based on Spring MVC model, below is its high level design diagram

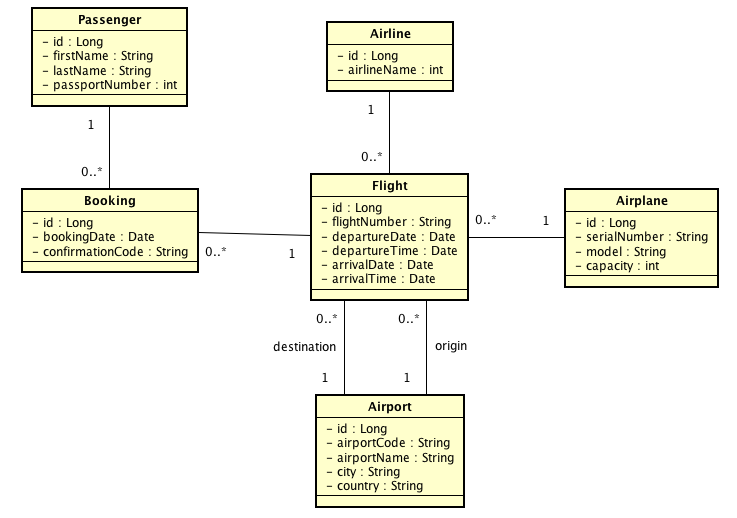


**Figure 1** **High Level Design Diagram**

# Detailed Design



## Static Diagram



**Figure 2 Static Diagram**

# Issues, Risk and Dependencies

Our target for this project is using the studied technologies of Spring MVC as much as possible. We had a trouble with integrating them. When we did on homework or demos, they are very easy to config. However, it is hard to connect and config them as a whole part. It requires harmony such as same versions, the order, and location of each part, grouping parts which have the same layer.

Because we want to spread out this project by using many technologies. Therefore, we do not have enough time to go in depth for each part. For that reason, there are some risks such as SQL injection, XSS that still exists somewhere in our application.

We used Git for managing our code. We got a lot of conflicts when members committed code. After this project, we get an experience for it. We should develop each feature on different branches. Whenever a part is done and passed the test, it will be merged to the main branch.

# Technologies

Follow the project’s guideline for the technologies used, in this project, we tried to apply as much as possible technology studied in the course and listed below

* Annotation
* Data binding
* Validation and Custom Validation
* Custom Formatter
* Custom Interceptor - Used to implement to broadcast Special blurb message to user when they visit the site
* Internationalization - Support for 2 languages; English and Vietnamese
* Exception Handling - Use controller advice to globally handle exception in application
* REST/Ajax two way; error handling - as for demo, it used only for adding new Airplane
* Security; Authentication & Authorization - support only for admin user when access to back-office website
* Web Sockets - Used to implement Online support function
* Web Flow - Used to implement flight booking process
* Tiles - Apply for all pages in the website. There are two tiles defined separately for User and admin at back-office
* Persistence in database [JPA]
* Database MySQL

# Future Considerations

* Business Domain:
  + Customer registration - The project could extends to allow customer registration
  + Promotion - The project could provide promotion functionality
  + Multiple customer online support - The project could extend to support for multiple user online support at the same time
  + Online checking - The project could provide online checking functionality
  + Payment - The project could provide payment functionality
  + Search for flight - The project could provide searching functionality
* Technologies
  + Mobile app - Port project into mobile application to provide user the ability to access from mobile devices
  + Spring Cloud and Microservices: distributed config, service discovery, routing & load balancing

# References

* Amuthan Ganeshan, Spring MVC Beginner’s Guide Second Edition, 2016.
* Homework and demos in this course.
* Google Search