City University of Hong Kong

Department of Computer Science

CS3343 (A) Software Engineering Practice

2014/15

Analysis and Design Report

Student Names	Student ID
HO, Wai Kit	53144248
WONG, Chung Man	53145233
YIU, Yiu Yeung	53144144
Kong , Tsz Kit	53143798
Lau, Kam Yu	53144170
So, Chun Hei	53144525

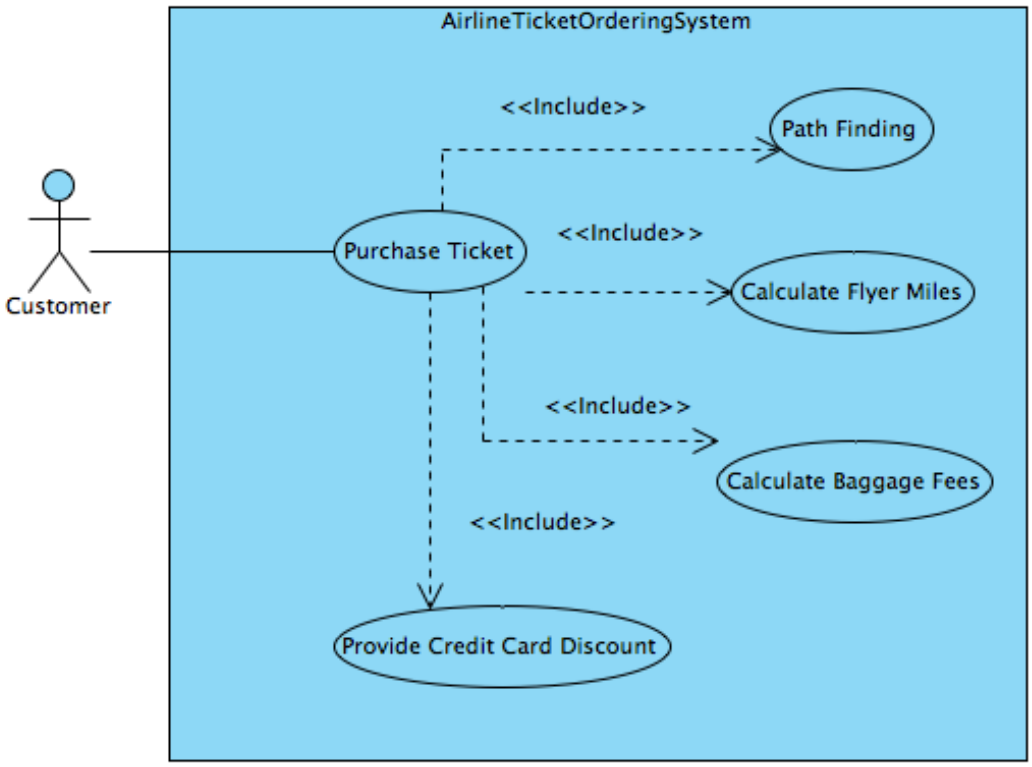
Table of Contents

1. Introduction

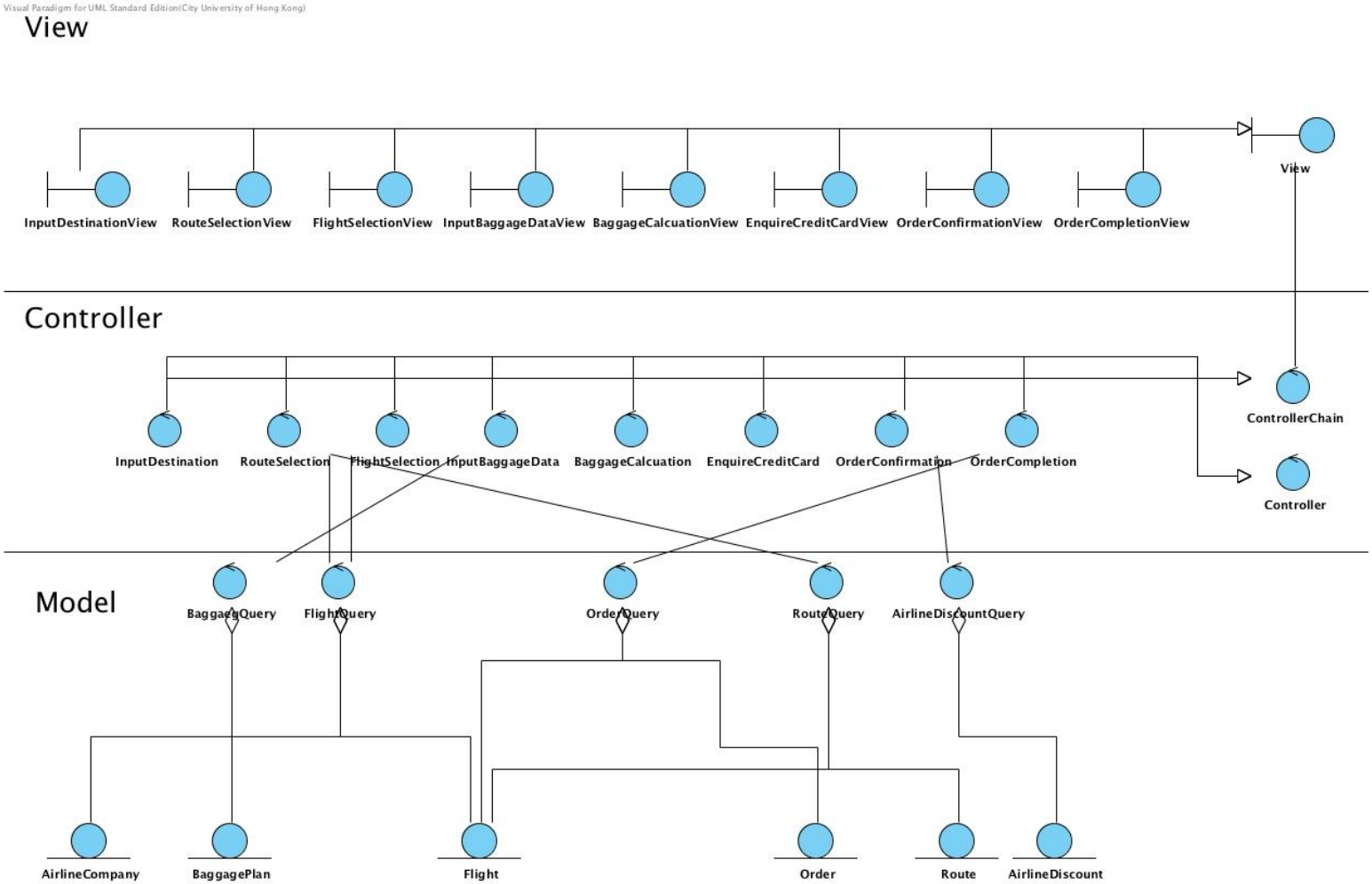
The aim of this project is to provide a useful and easy way for the customer to purchase the airline ticket with a smart method. Ecommerce is a popular market in the service sector. Airline Company should provide a clever system to handle all kind of stuff.

For example: better path finding and suggestion for the flight routes, baggage fees calculation automatically, credit card discount handling and flyer miles calculation. With such kind of functions, customer can enjoy the service and order the ticket within just few minutes.

2. Use Case Diagram



3. Analysis Class



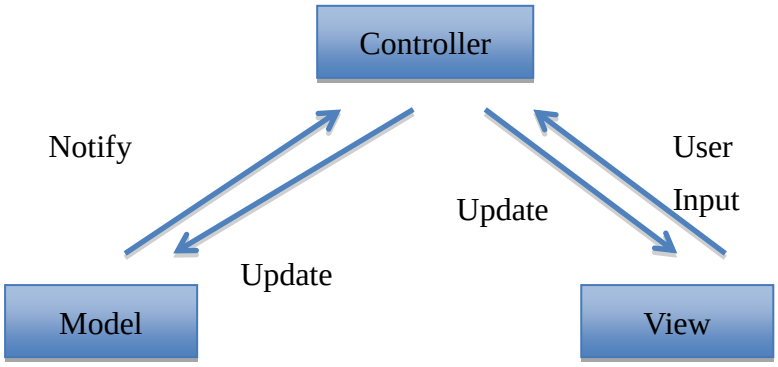
4. Domain Modeling

Domain Name	Domain Class	Responsibility
InputDestinationView	Boundary	To ask user to input destination
RouteSelectionView	Boundary	To ask user to select route
FlightSelectionView	Boundary	To show the selected flight
InputBaggageDataView	Boundary	To ask user to input baggage
BaggageFeeCalculationView	Boundary	To tell user the baggage fee
EnquireCreditCardView	Boundary	To ask user to input credit card
OrderConfirmationView	Boundary	To ask user to confirm the order
OrderCompletionView	Boundary	To tell user that the order is completed
Route	Entity	To store route data
Flight	Entity	To store flight data
BaggagePlan	Entity	To store Baggage data
CreditCard	Entity	To store credit card data
Order	Entity	To store order data
InputDestinationController	Controller	To control the view
RouteSelectionController	Controller	To get the routes
FlightSelectionController	Controller	To control the view
InputBaggageDataController	Controller	To control the view
BaggageFeeCalculationController	Controller	To calculate the baggage fee
EnquireCreditCardController	Controller	To control the view
OrderConfirmationController	Controller	To summarize the final amount
OrderCompletionController	Controller	To control the view

5. Factoring in design constraint

MVCPattern

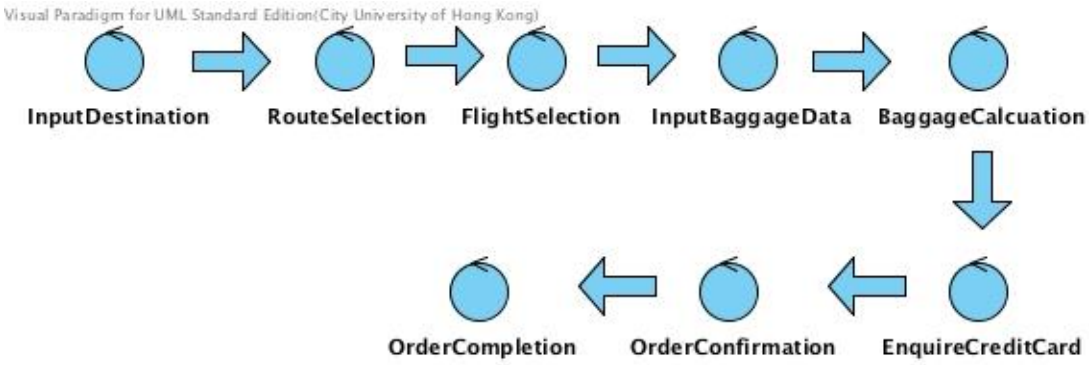
We applied MVC design pattern on Airline Ticket Ordering in order to have high maintainability of our system. By applying MVC pattern, we spliced the system into three part - Model, View and controller.

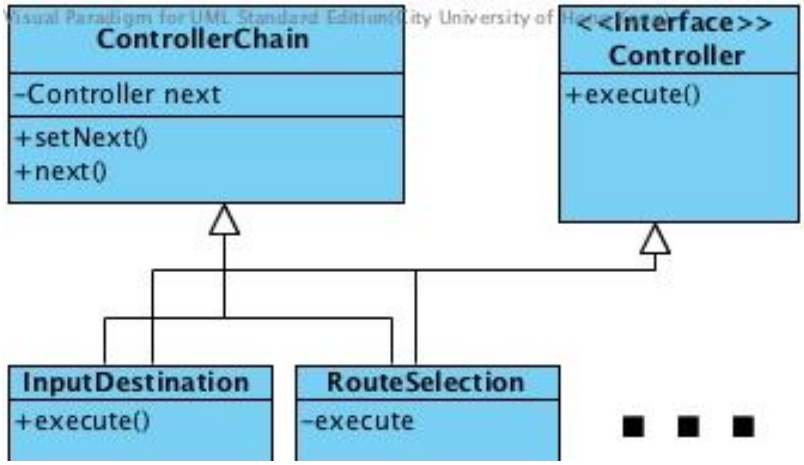


User will input data and get the output from view. Controller can get the user input from View and control the output for the View. Also, controller is response for update those data in model.

Developers can easier to maintain the system and understand those codes after apply MVC pattern.

Chain Pattern

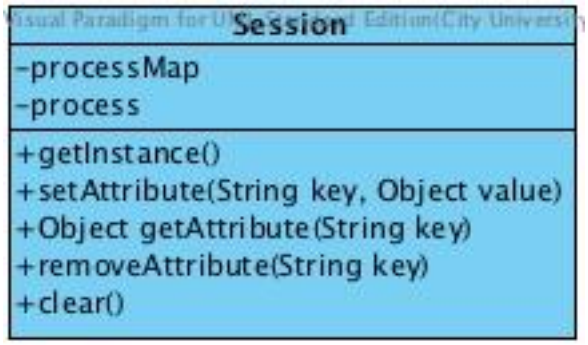




In our system, user can buy air tickets step by step. So we decided to apply chain Pattern. Current controller will handle the action that it responsible for. After a controller execute, it will execute the next controller.

This pattern allows an object to send a request without knowing which object will receiver and execute it.

Singleton Pattern

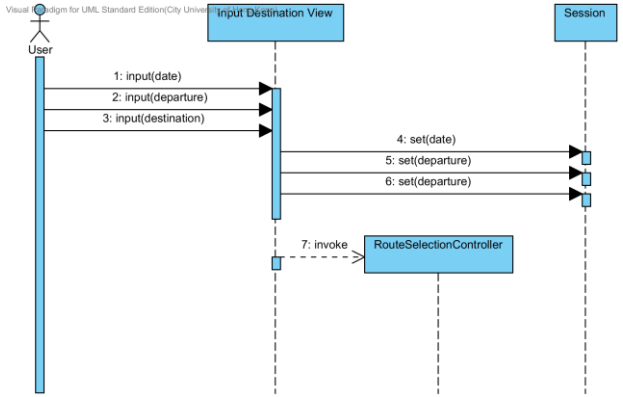


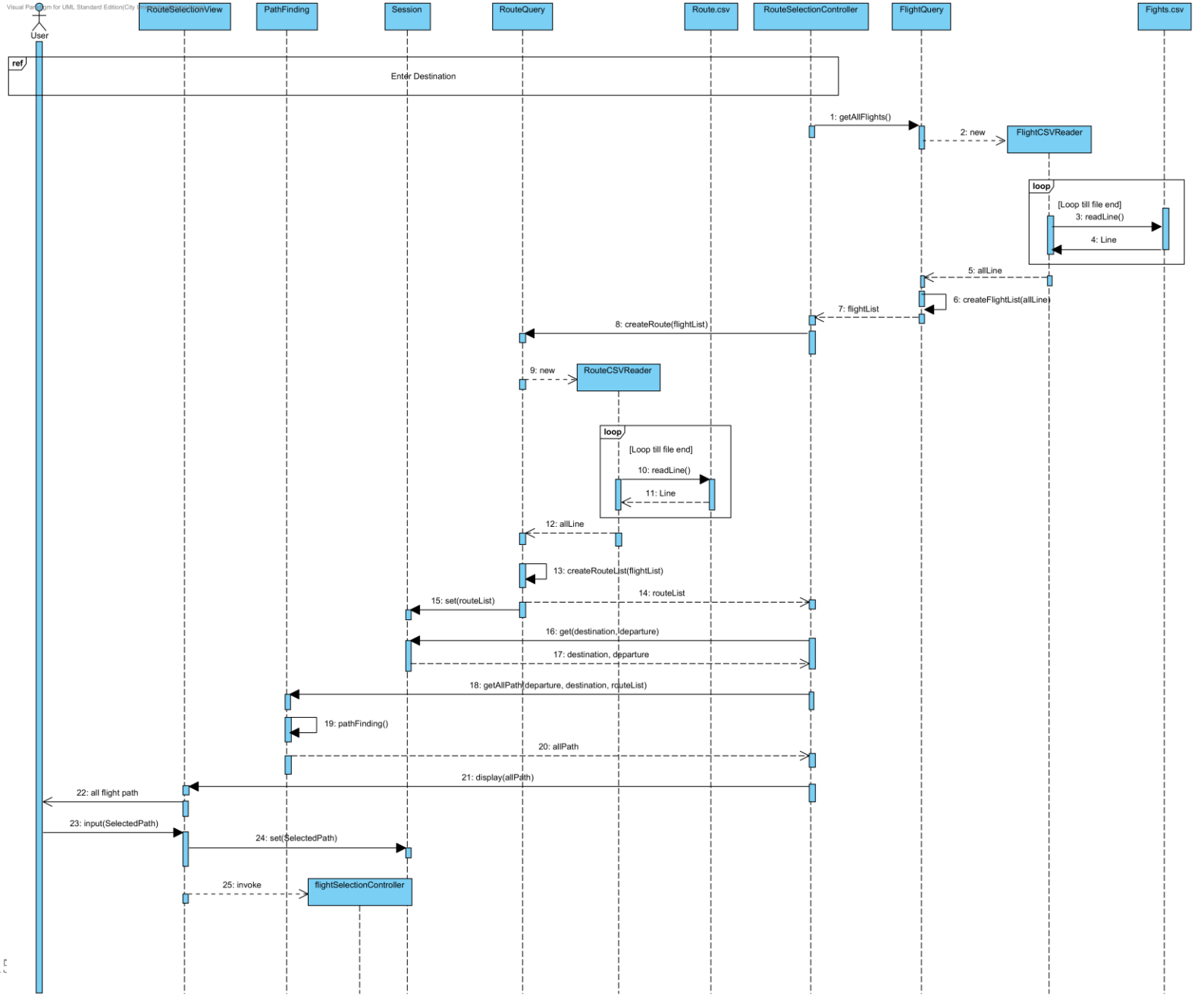
In this project, we need store data between those controllers and views before user complete the buying process. We created a singleton class call session. For this class, it only has one Instance. We store all the information temporariness in the instance of this class.

To see the large version of class diagram, you can visit https://github.com/kitho/CS3343/blob/master/doc/AnalysisAndDesignReport/ClassDiagram.jpg

6. Sequence Diagram

Input DestinationRoute Selection





Baggage Calculation