S/W Design And Architecture

Assignment 3 – S/W Design Documentation

525910 Hanmoi Choi

1 PREFACE 3

2 OVERALL 3

2.1 MVC FLOW 3
2.2 PACKAGE DIAGRAM 4
2.3 ACTIVITI DIAGRAM 4
2.3.1 ENTIRE SYSTEM 4
2.3.2 ADMIN 5
2.3.3 SELLER 6
2.3.4 USER 7

3 DESCRIPTION 8

3.1 PRESENTATION LAYER 8
3.1.1 FRAMEWORK 8
3.1.2 PATTERN 8
3.1.3 CLASS DIAGRAM 8
3.2 CONTROLLER LAYER 9
3.2.1 FRAMEWORK 9
3.2.2 PATTERN 9
3.2.3 CLASS DIAGRAM 9
3.3 DOMAIN LOGIC LAYER (BUSINESS LOGIC LAYER) 10
3.3.1 FRAMEWORK 10
3.3.2 PATTERN 10
3.3.3 CLASS DIAGRAM 11
3.4 DATA SOURCE LAYER (REPOSITORY LAYER) 12
3.4.1 FRAMEWORK 12
3.4.2 PATTERN 12
3.4.3 CLASS DIAGRAM 12
3.4.4 DATABASE ENTITY RELATION DIAGRAM 13

4 SEQUENCE DIAGRAMS 14

4.1 COMMON MODULE 14
4.1.1 SIGN UP USER (BOTH A BUYER AND A SELLER) 14
4.1.2 MODIFY USER INFORMATION (BOTH BUYER AND SELLER) 15
4.1.3 SEARCH ITEMS 16
4.1.4 LOG OUT 16
4.2 ADMIN MODULE 17
4.2.1 DELETE A SELLER 17
4.2.2 GENERATE REPORT 17

4.3 SELLER MODULE **18**
4.3.1 ADD AN ITEM 18
4.3.2 DELETE AN ITEM 18
4.3.3 CHECK AN ORDER 19
4.4 BUYER MODULE **20**
4.4.1 PLACE AN ORDER 20
4.4.2 CHECK AN ORDER 21

1 Preface

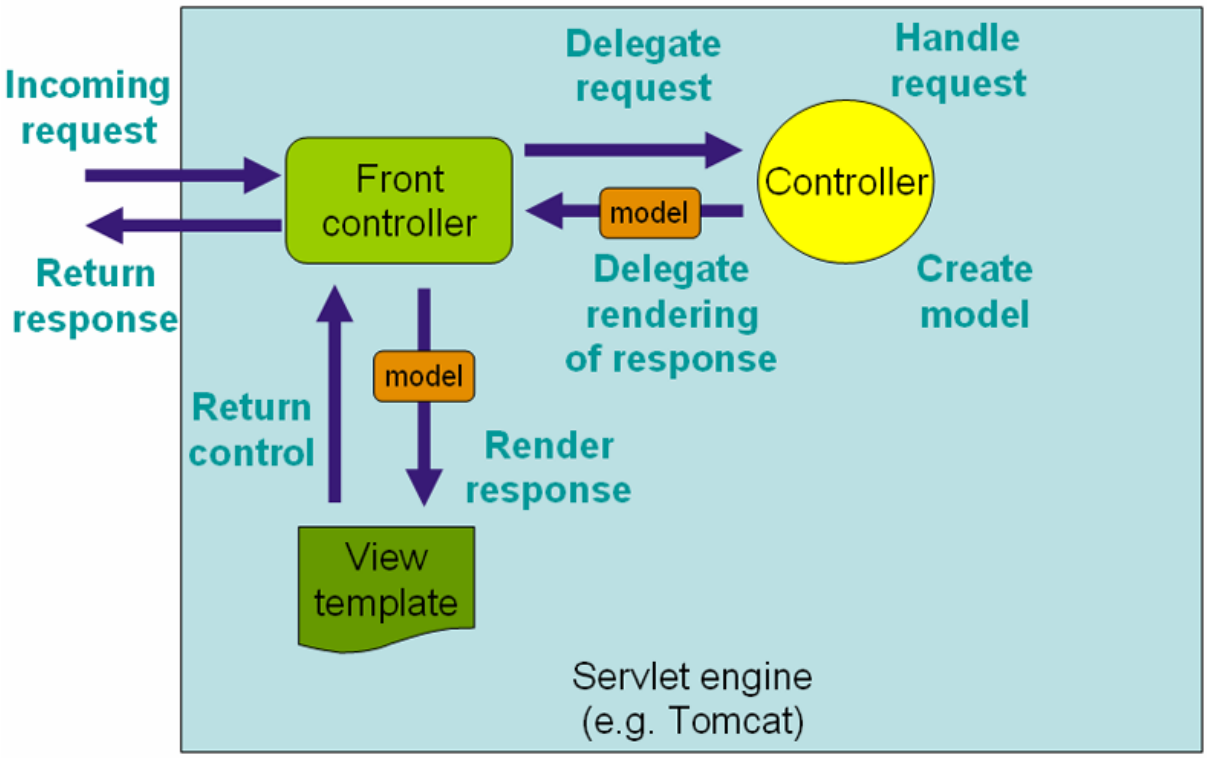
The aim of the document is to provide developers the detailed information of how to implement 'Bigsale.com' website in terms of software engineering.

The document provides various UML diagrams including activity diagrams, class diagrams, sequence diagrams, package diagrams and database ER diagram. Also it explains which patterns are used to elaborate the design.

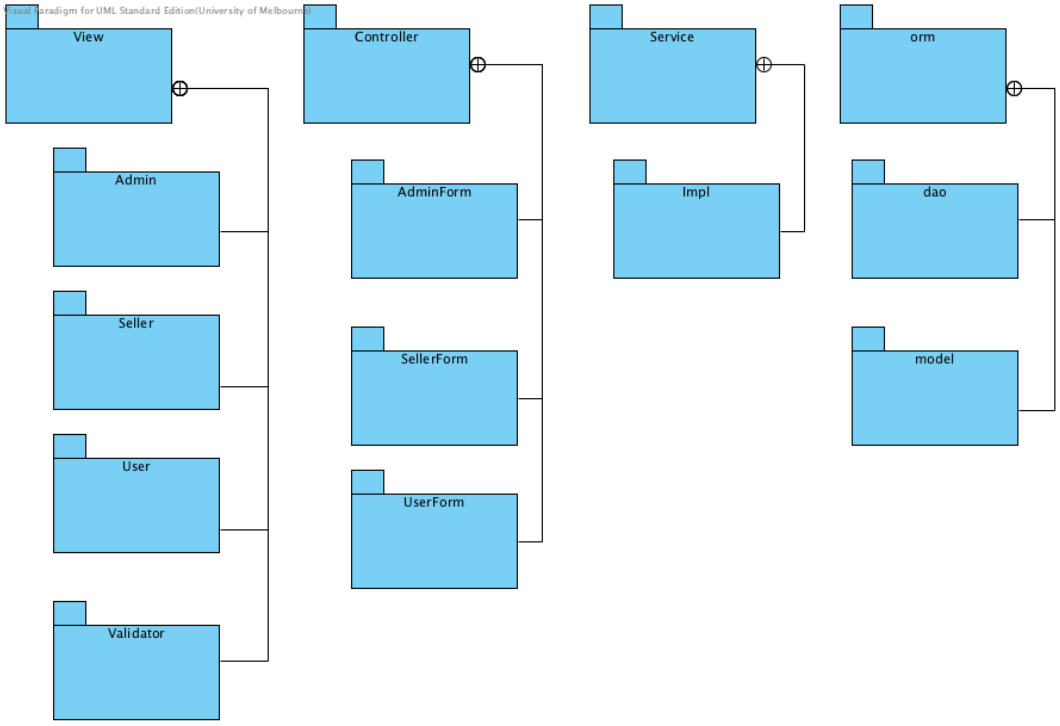
The document is divided into big three parts in accordance to MVC pattern, view, controller, and Model (aka Repository).

2 Overall

2.1 MVC Flow

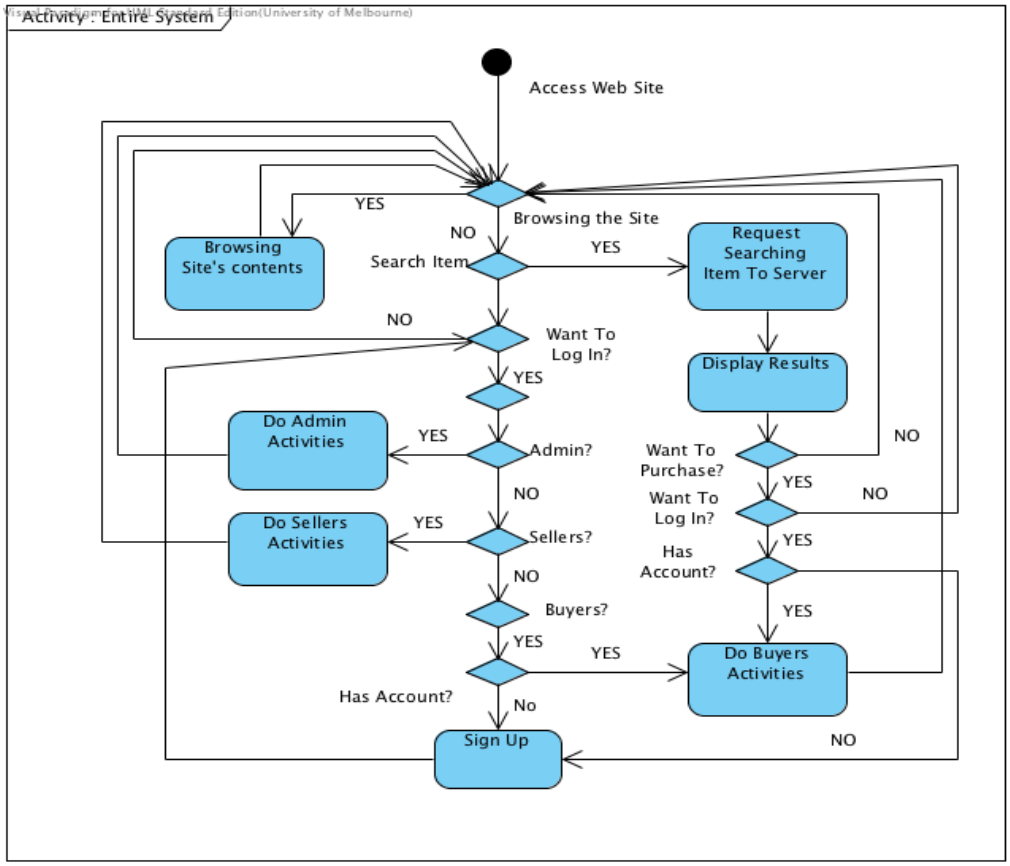


2.2 Package Diagram

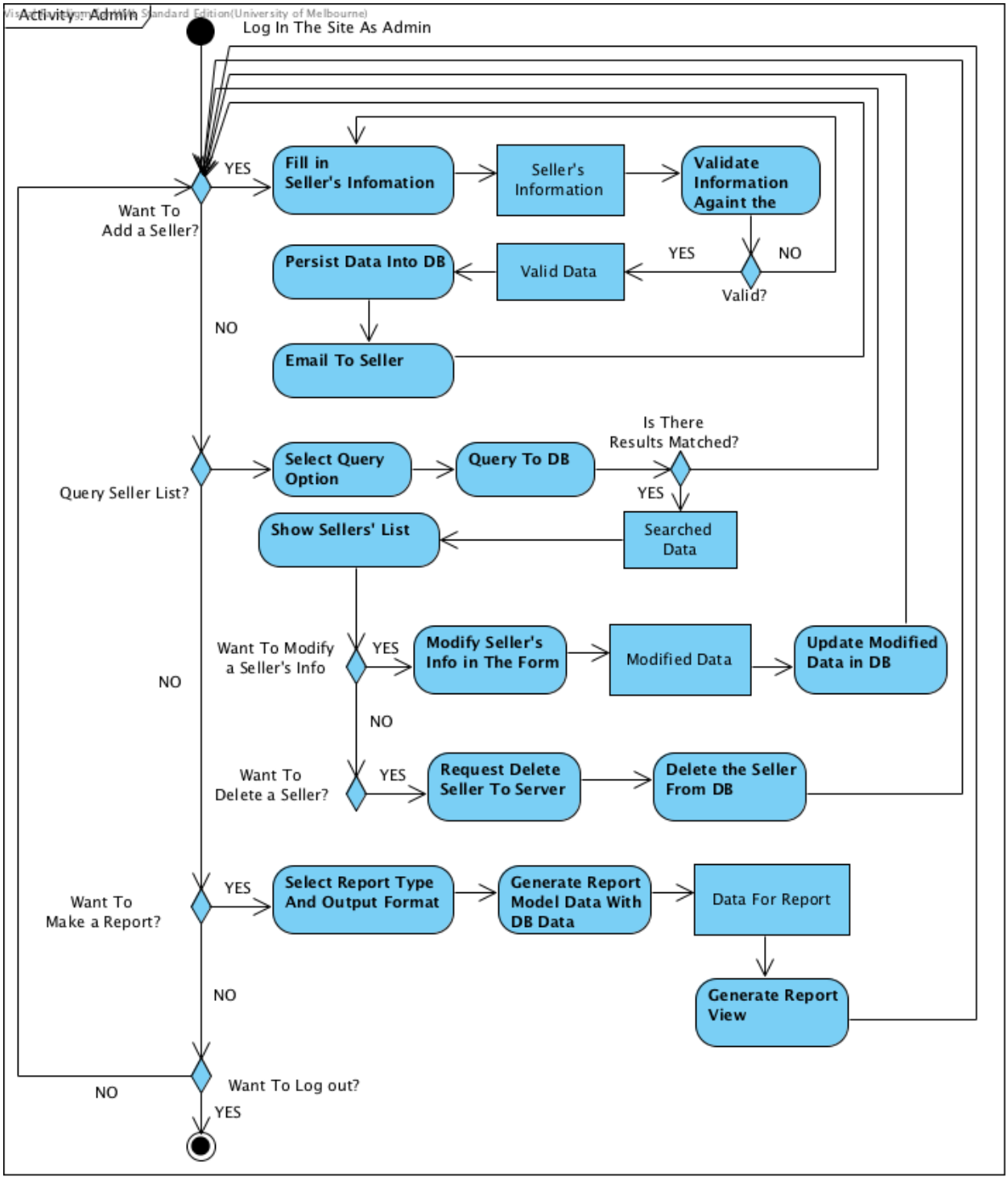


2.3 Activity Diagram

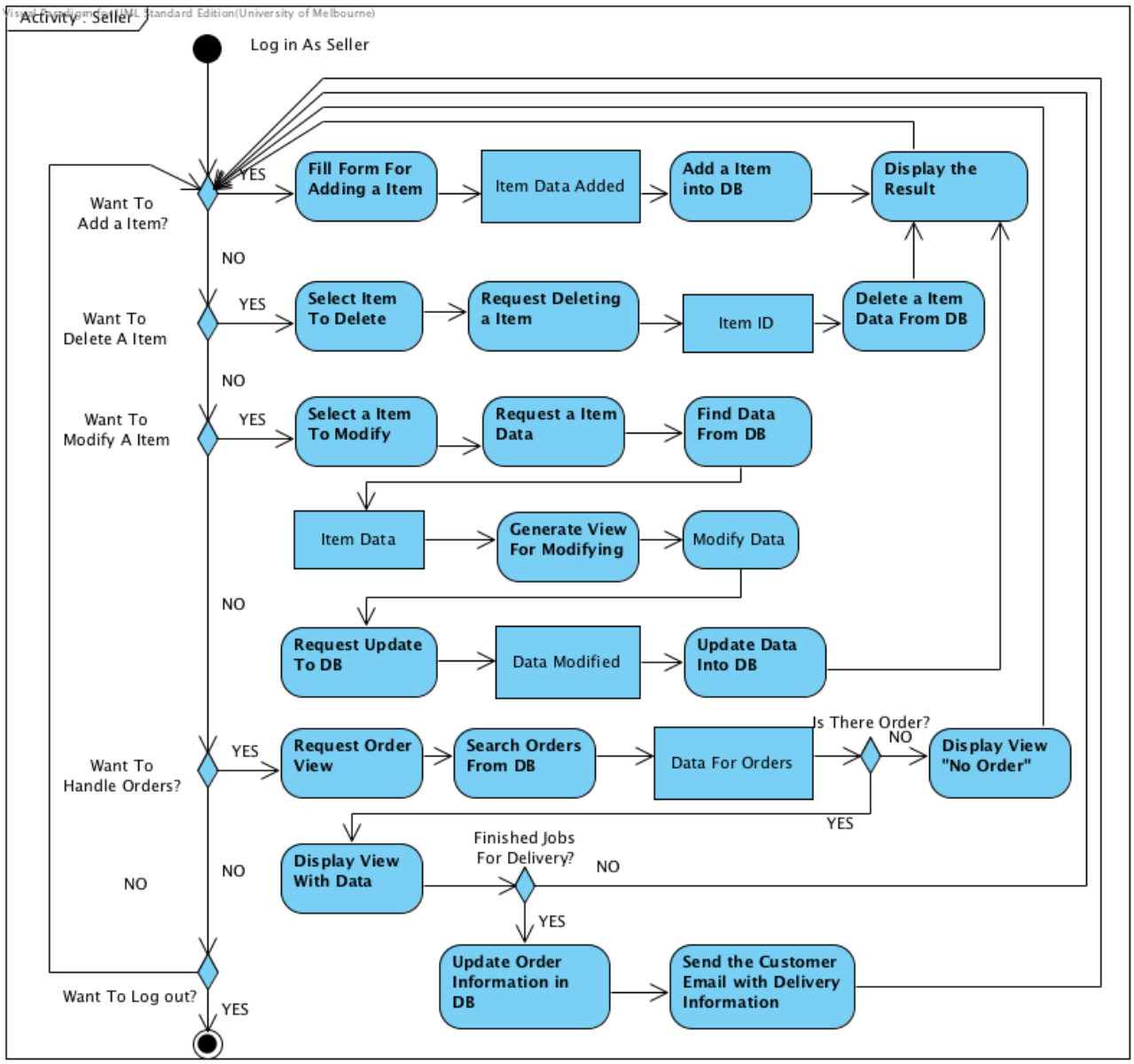
2.3.1 Entire system



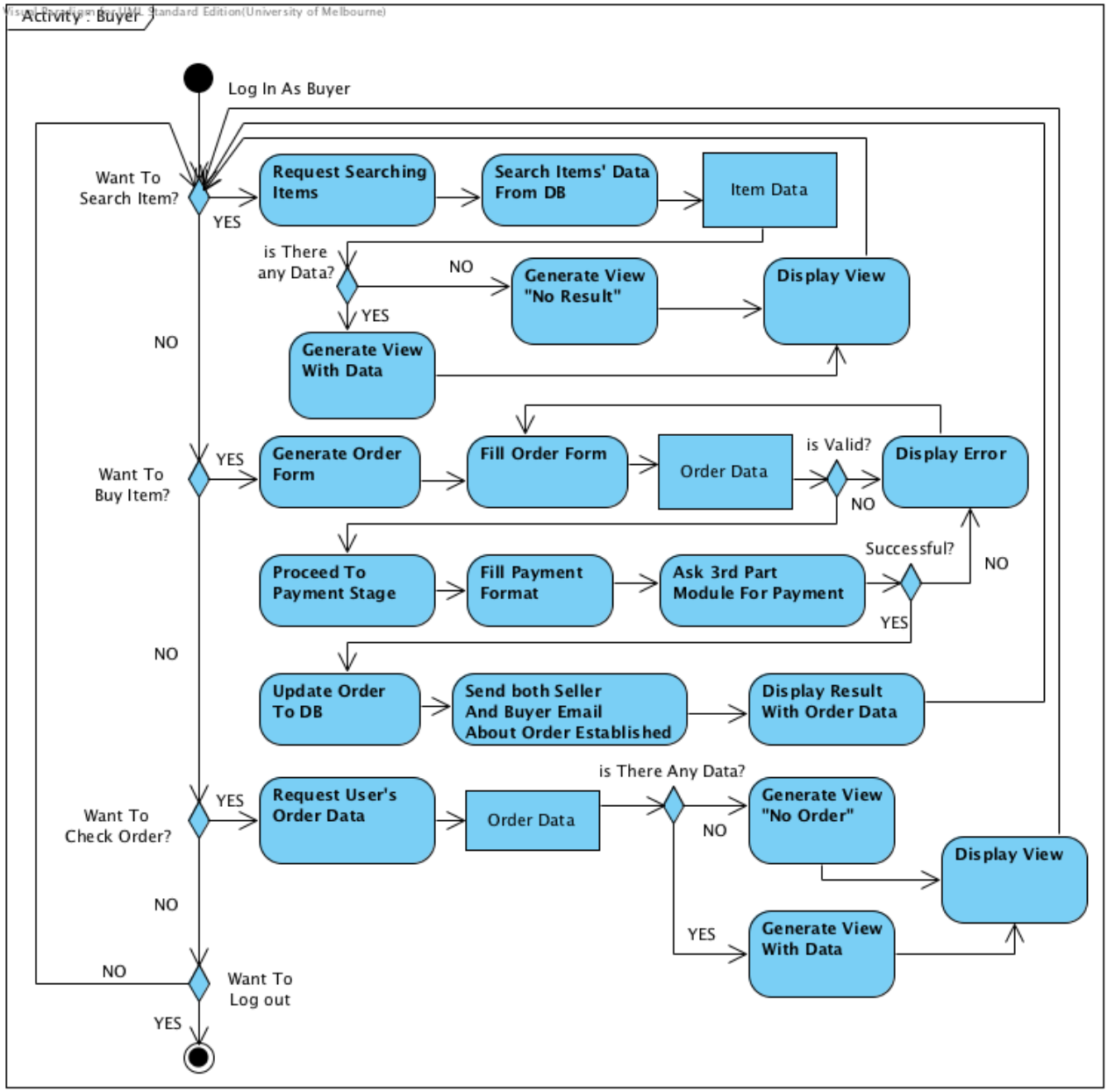
2.3.2 Admin


Activity: Admin
Log In The Site As Admin
Want To Add a Seller?
YES
Fill in Seller's Infomation
Seller's Information
Validate Information Againt the
Valid?
YES
NO
Valid Data
Persist Data Into DB
NO
Email To Seller
Query Seller List?
Select Query Option
Query To DB
Is There Results Matched?
YES
Searched Data
Show Sellers' List
Want To Modify a Seller's Info
YES
Modify Seller's Info in The Form
Modified Data
Update Modified Data in DB
NO
Want To Delete a Seller?
YES
Request Delete Seller To Server
Delete the Seller From DB
NO
Want To Make a Report?
YES
Select Report Type And Output Format
Generate Report Model Data With DB Data
Data For Report
Generate Report View
NO
Want To Log out?
NO
YES

2.3.3 Seller


Activity : Seller
Log in As Seller
Want To Add a Item?
YES
Fill Form For Adding a Item
Item Data Added
Add a Item into DB
Display the Result
NO
Want To Delete A Item
YES
Select Item To Delete
Request Deleting a Item
Item ID
Delete a Item Data From DB
NO
Want To Modify A Item
YES
Select a Item To Modify
Request a Item Data
Find Data From DB
Item Data
Generate View For Modifying
Modify Data
NO
Request Update To DB
Data Modified
Update Data Into DB
Want To Handle Orders?
YES
Request Order View
Search Orders From DB
Data For Orders
Is There Order?
NO
Display View "No Order"
YES
NO
Display View With Data
Finished Jobs For Delivery?
NO
YES
NO
Update Order Information in DB
Send the Customer Email with Delivery Information
Want To Log out?
YES

2.3.4 User


Activity : Buyer
Log In As Buyer
Want To Search Item?
YES
Request Searching Items
Search Items' Data From DB
Item Data
is There any Data?
NO
Generate View "No Result"
Display View
YES
Generate View With Data
NO
Want To Buy Item?
YES
Generate Order Form
Fill Order Form
Order Data
is Valid?
Display Error
NO
Proceed To Payment Stage
Fill Payment Format
Ask 3rd Part Module For Payment
Successful?
NO
YES
NO
Update Order To DB
Send both Seller And Buyer Email About Order Established
Display Result With Order Data
Want To Check Order?
YES
Request User's Order Data
Order Data
is There Any Data?
NO
Generate View "No Order"
Display View
NO
YES
Generate View With Data
NO
Want To Log out
YES

3 Description

3.1 Presentation Layer

3.1.1 Framework

a. Java Server Page (JSP)
b. Apache Tiles 2
c. Spring MVC

3.1.2 Pattern

a. Template view

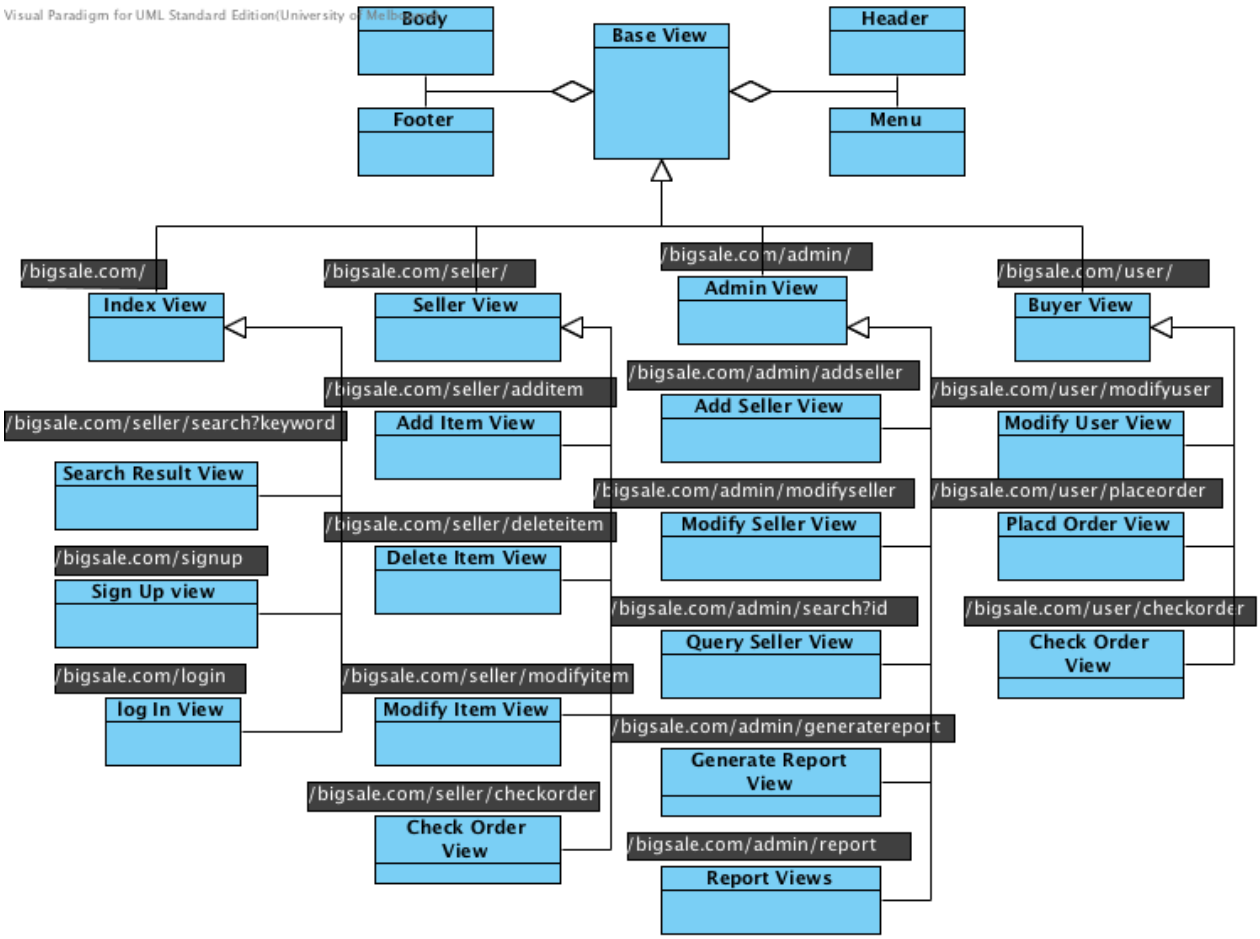
JSP allows developers to separate model data from view template. And tiles2 enables each template JSPs to be reusable. By using tiles2 framework developers create a big frame for each view pages. For example, bigsale.com's big frame can divide into a header, a menu, a body and footer. And each JSP view is one of the parts.

b. Front Controller

Spring Framework provides the front controller to handle every request from clients and delegates each request to specific a controller in accordance to configuration of XML files or annotations.

I will choose these patterns even though I am supposed to implement my own presentation layer framework. I could reuse my view codes and collaboration can be much easier due to less coupling among controllers. Each developer could implement their own code without concerning other developers' code.

3.1.3 Class Diagram



3.2 Controller Layer

3.2.1 Framework

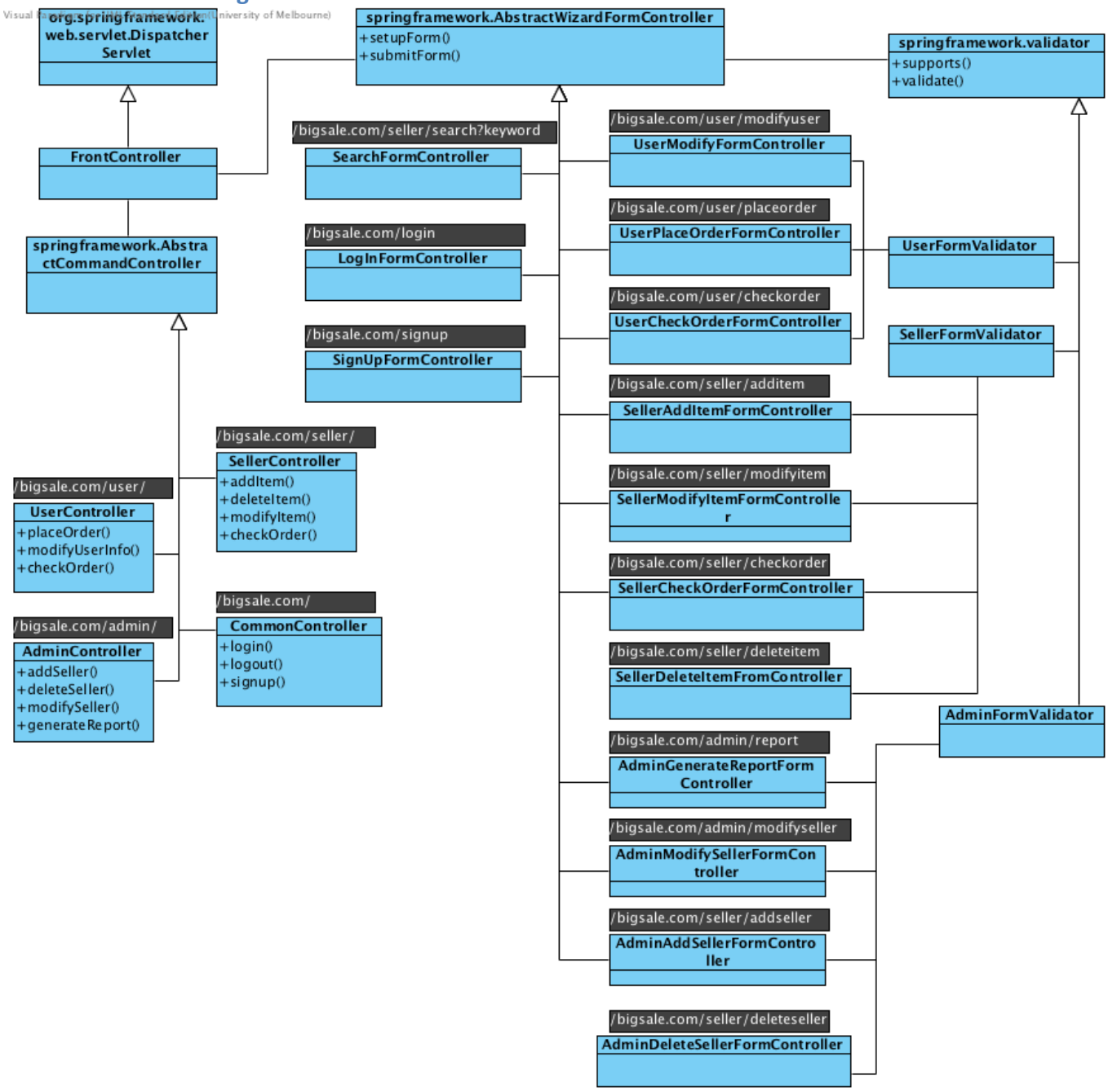
a. Spring MVC

3.2.2 Pattern

a. Command Pattern

For validation view, Spring framework's validator interface is used. Real implementations are injected automatically by Spring framework in accordance to configuration.

3.2.3 Class diagram



3.3 Domain Logic Layer (Business Logic Layer)

3.3.1 Framework

Spring MVC
Hibernate

3.3.2 Pattern

a. Visitor Pattern

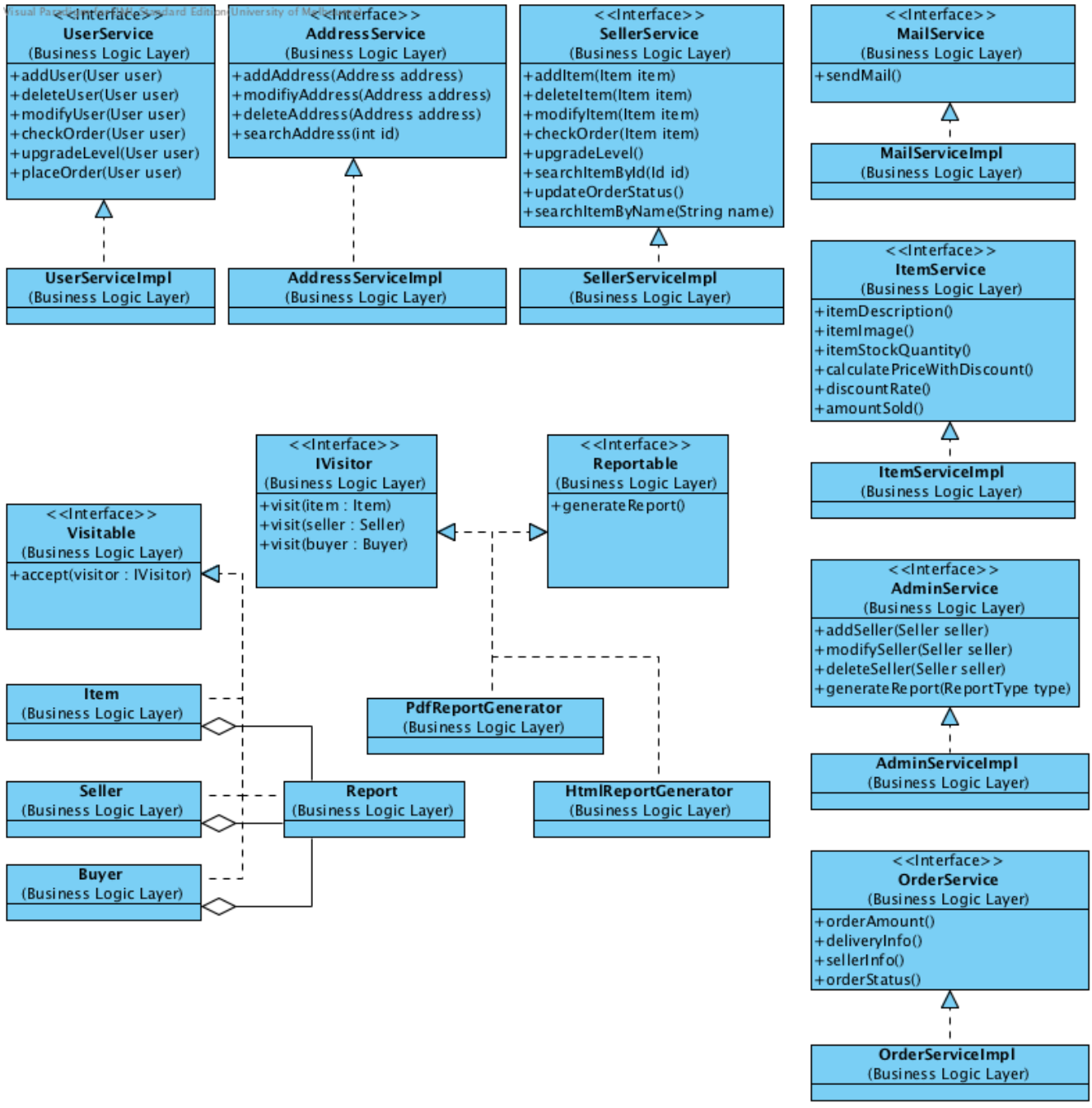
In order generator report, visitor pattern is used. This pattern allows developer to separator report data and report generator module.

b. Domain Model Pattern

Hibernate provides this pattern in really easy way. Each domain class could map to a physical database map. With simple configurations developers could use this pattern.

I will choose domain model pattern for domain logic layer when I have to select my own J2EE pattern because only this pattern can manage well the endless change of clients' requirement.

3.3.3 Class Diagram



3.4 Data Source Layer (Repository Layer)

3.4.1 Framework

Spring MVC
Hibernate

3.4.2 Pattern

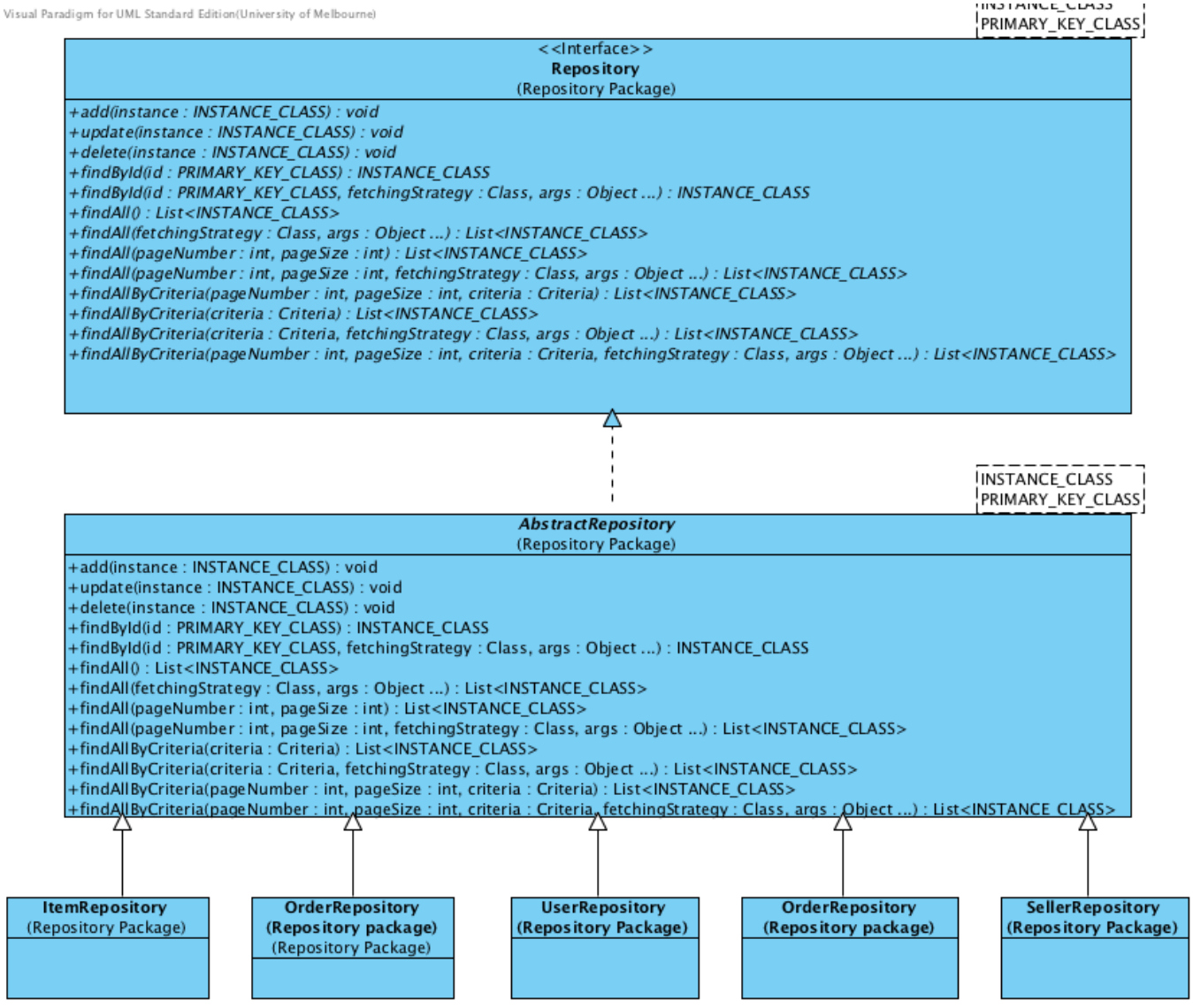
a. Data Mapper Pattern

Hibernate generates Data Value Objects (DVOs) and maps to Tables. This pattern works well with Domain Model Pattern.

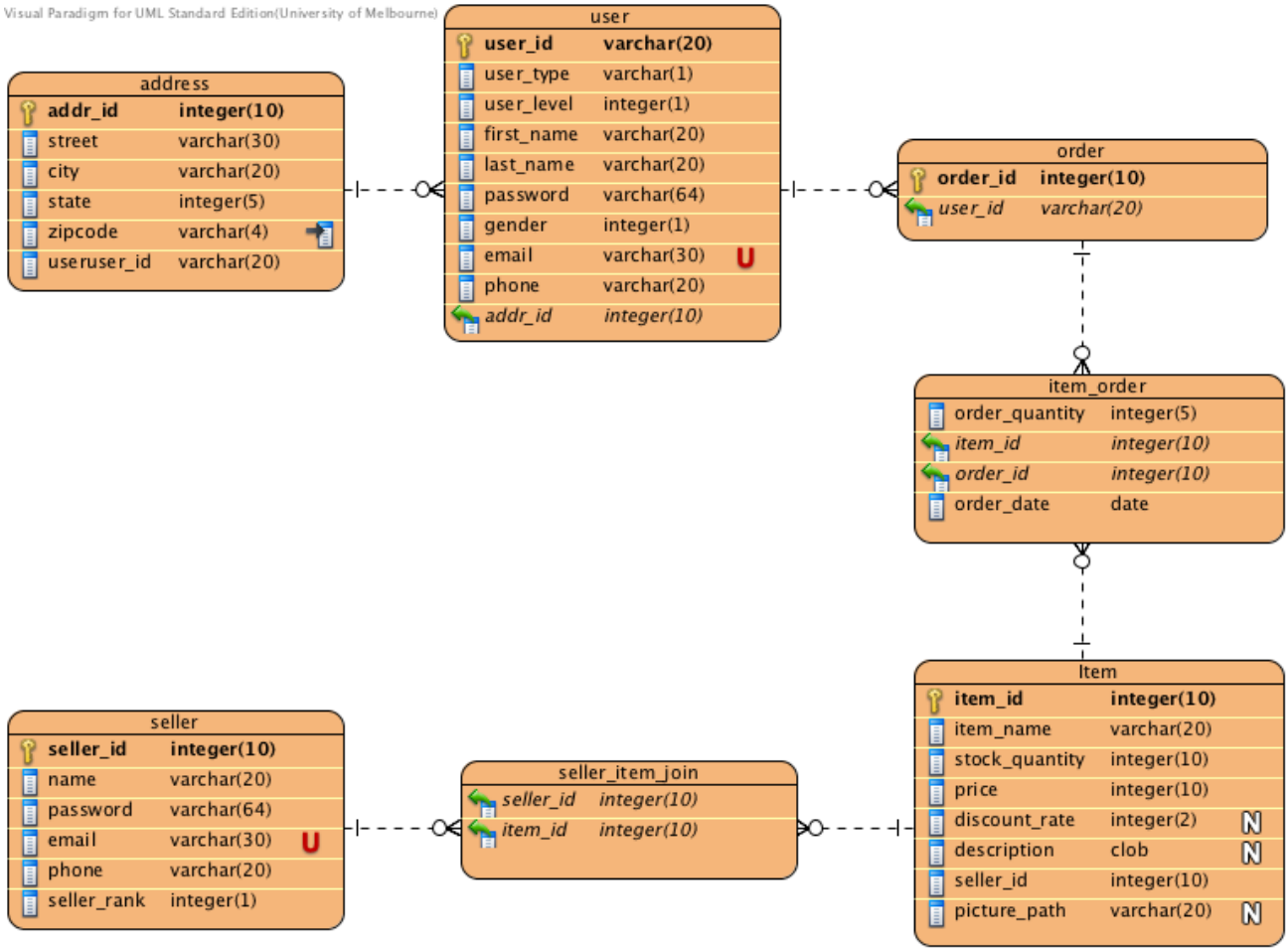
b. Lazy Load Pattern

Hibernate provide proxy object of the real database connection in default.

3.4.3 Class Diagram



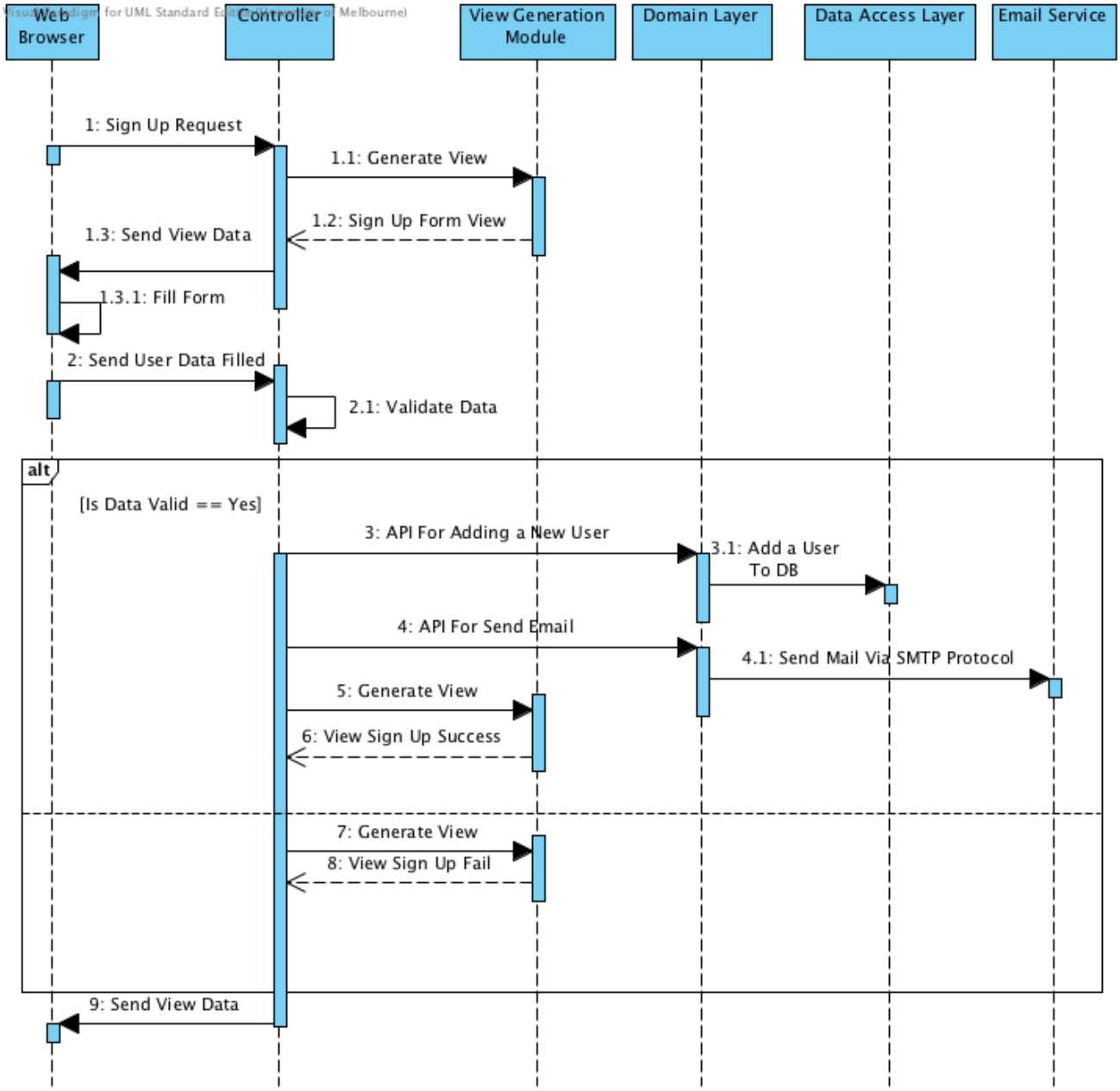
3.4.4 Database Entity Relation Diagram



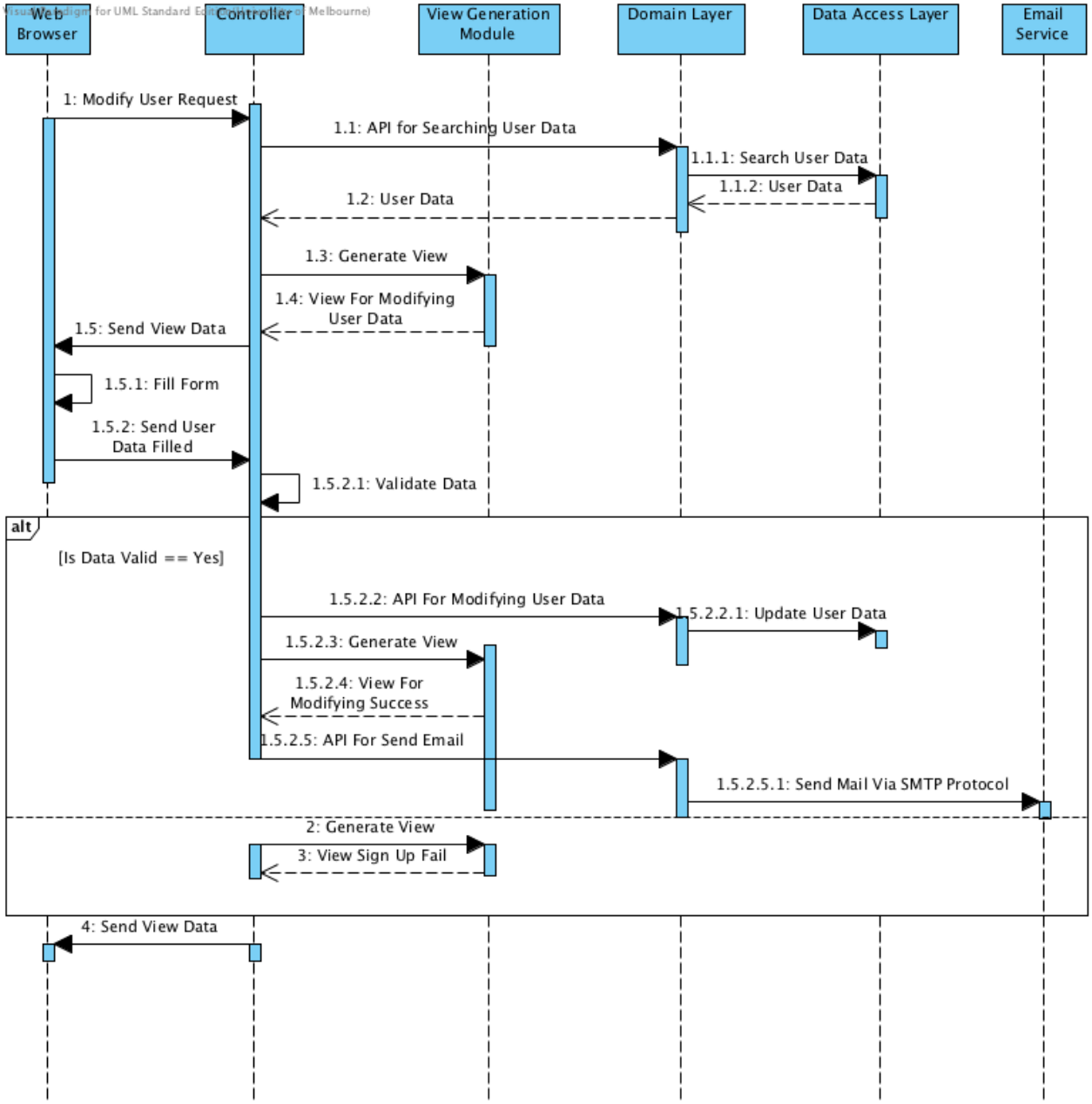
4 Sequence Diagrams

4.1 Common Module

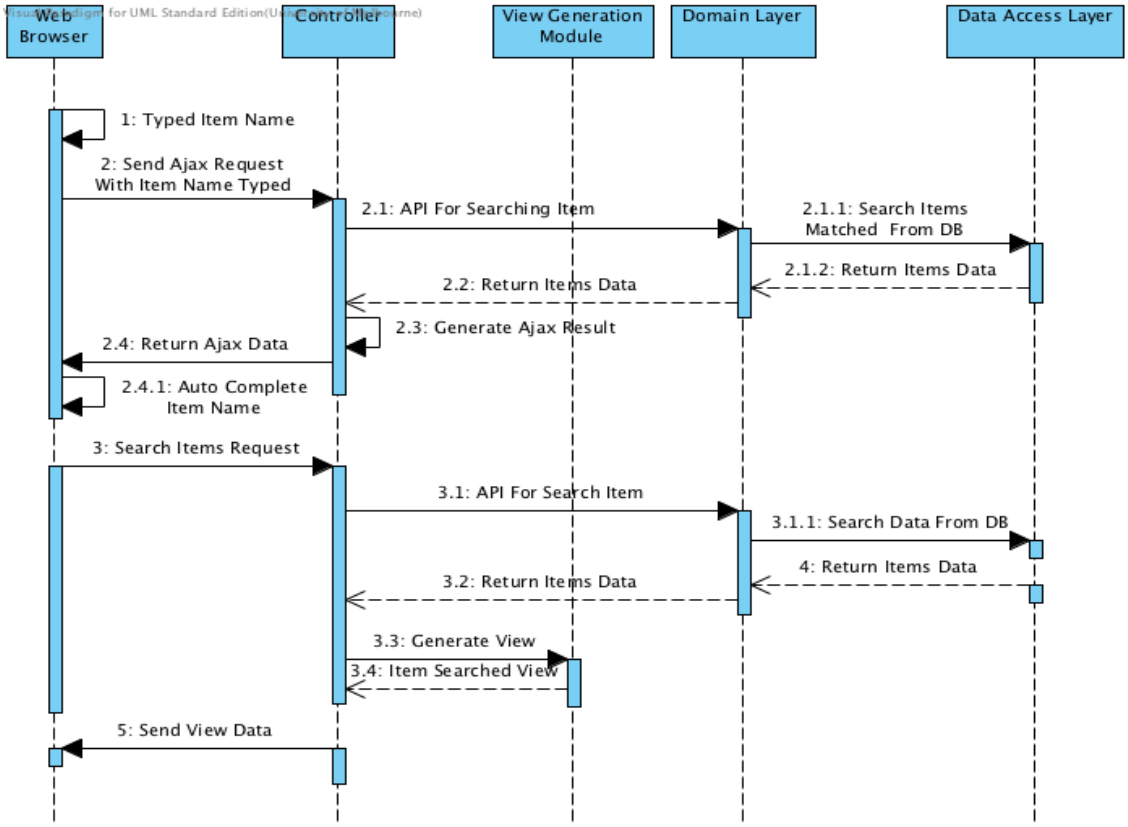
4.1.1 Sign Up User (Both a buyer and a seller)



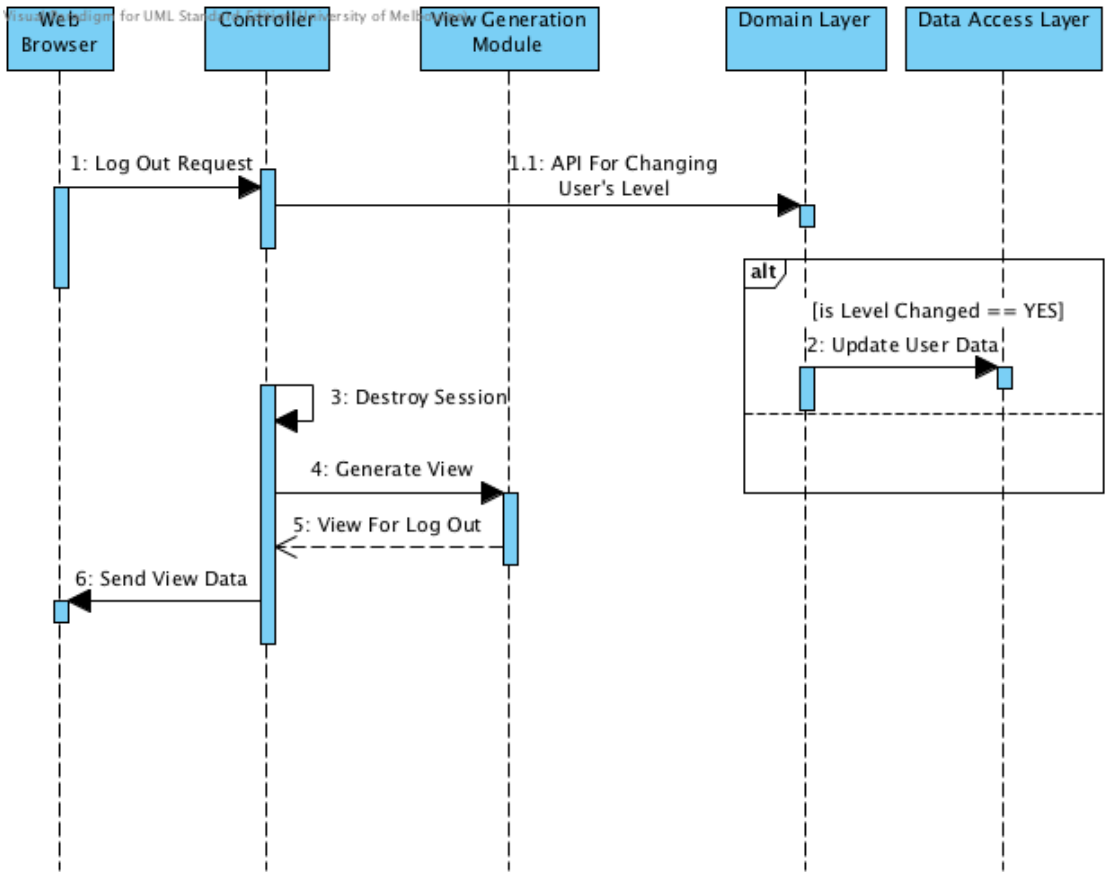
4.1.2 Modify User Information (Both Buyer and Seller)



4.1.3 Search Items

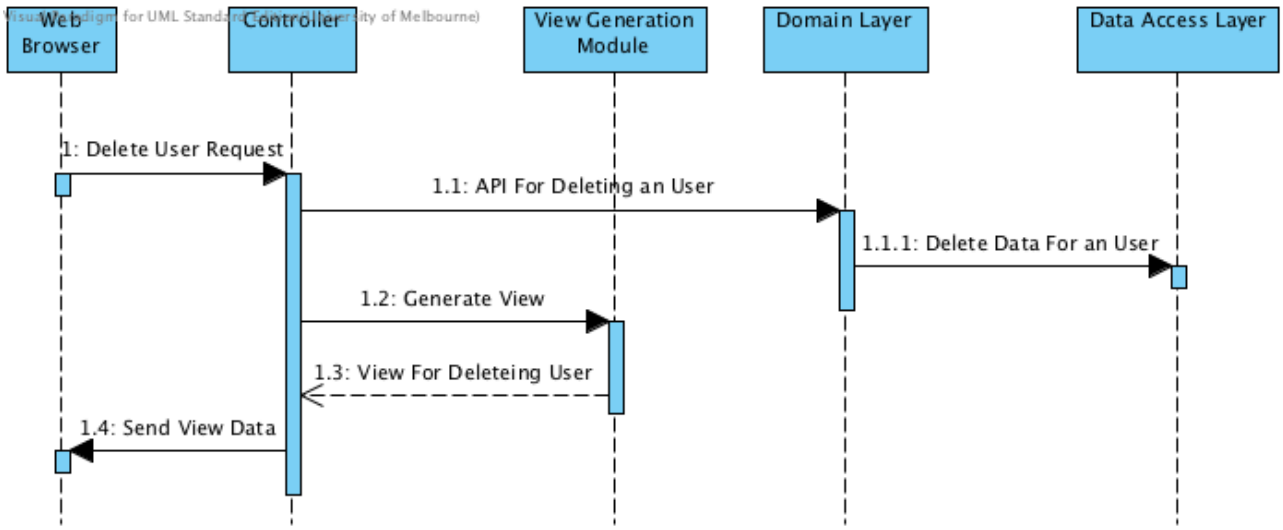


4.1.4 Log out

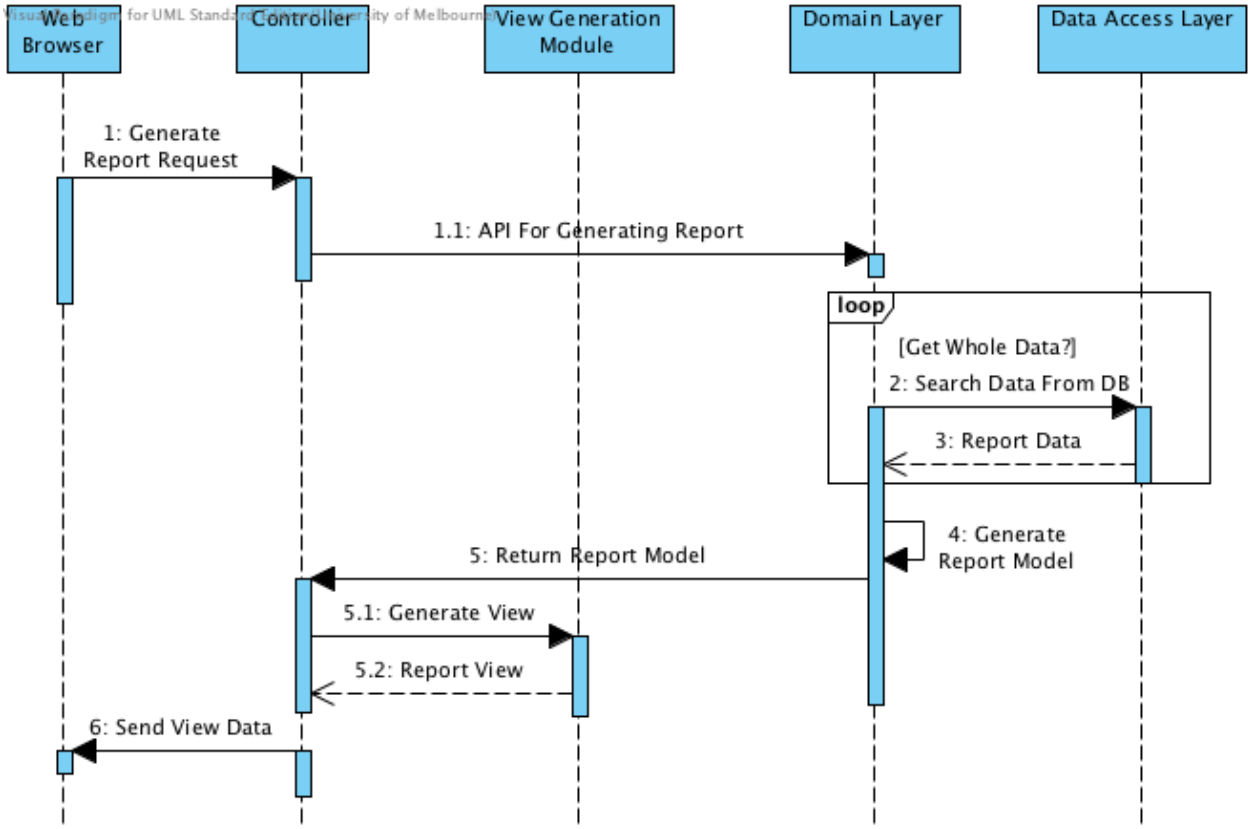


4.2 Admin Module

4.2.1 Delete a Seller

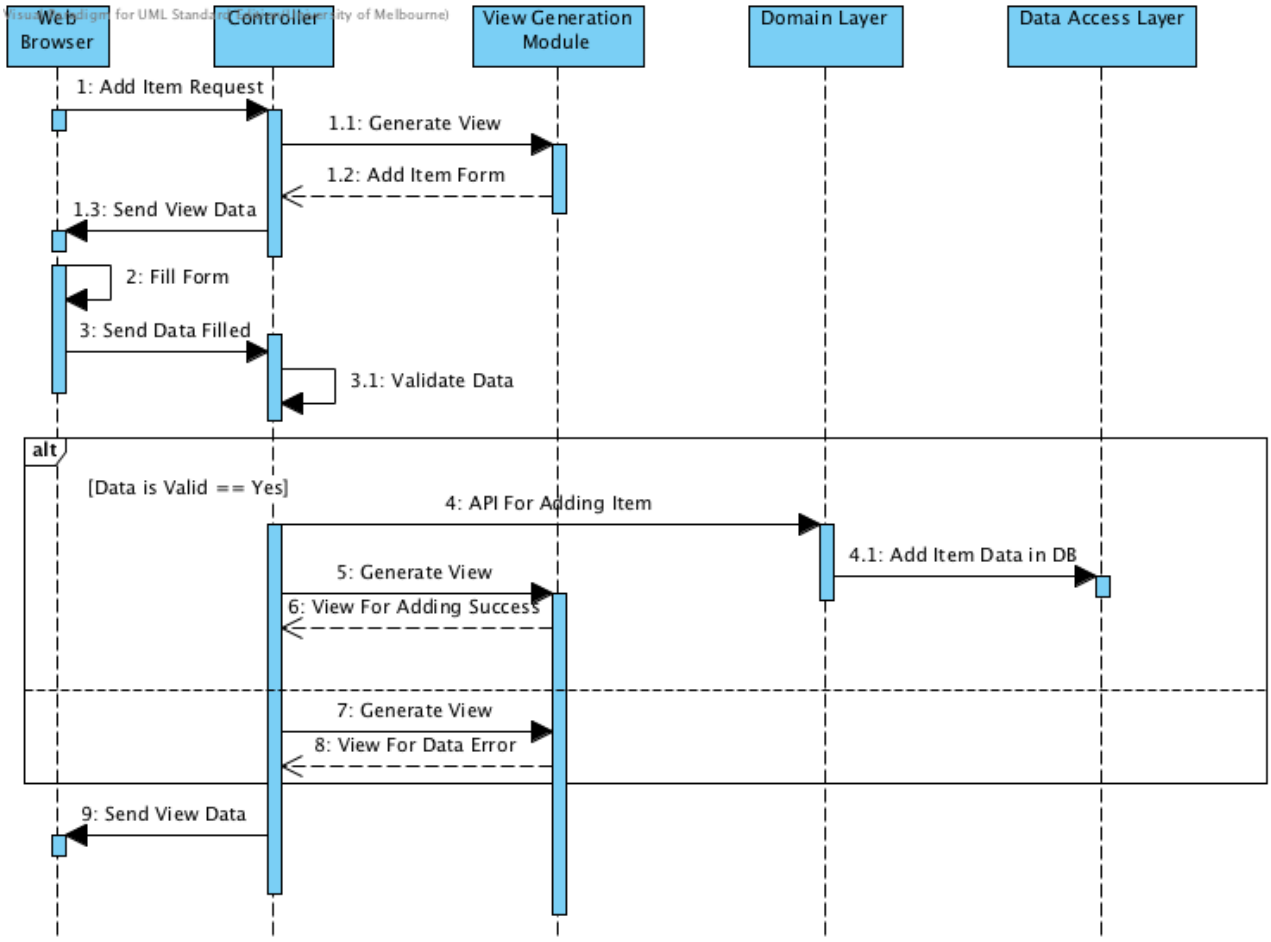


4.2.2 Generate Report

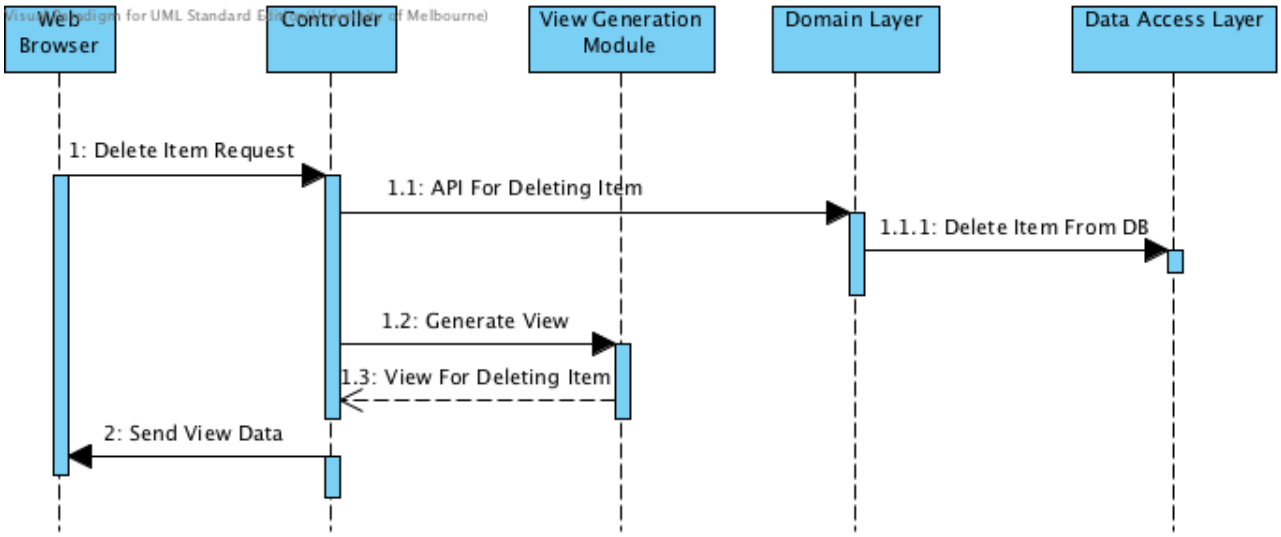


4.3 Seller Module

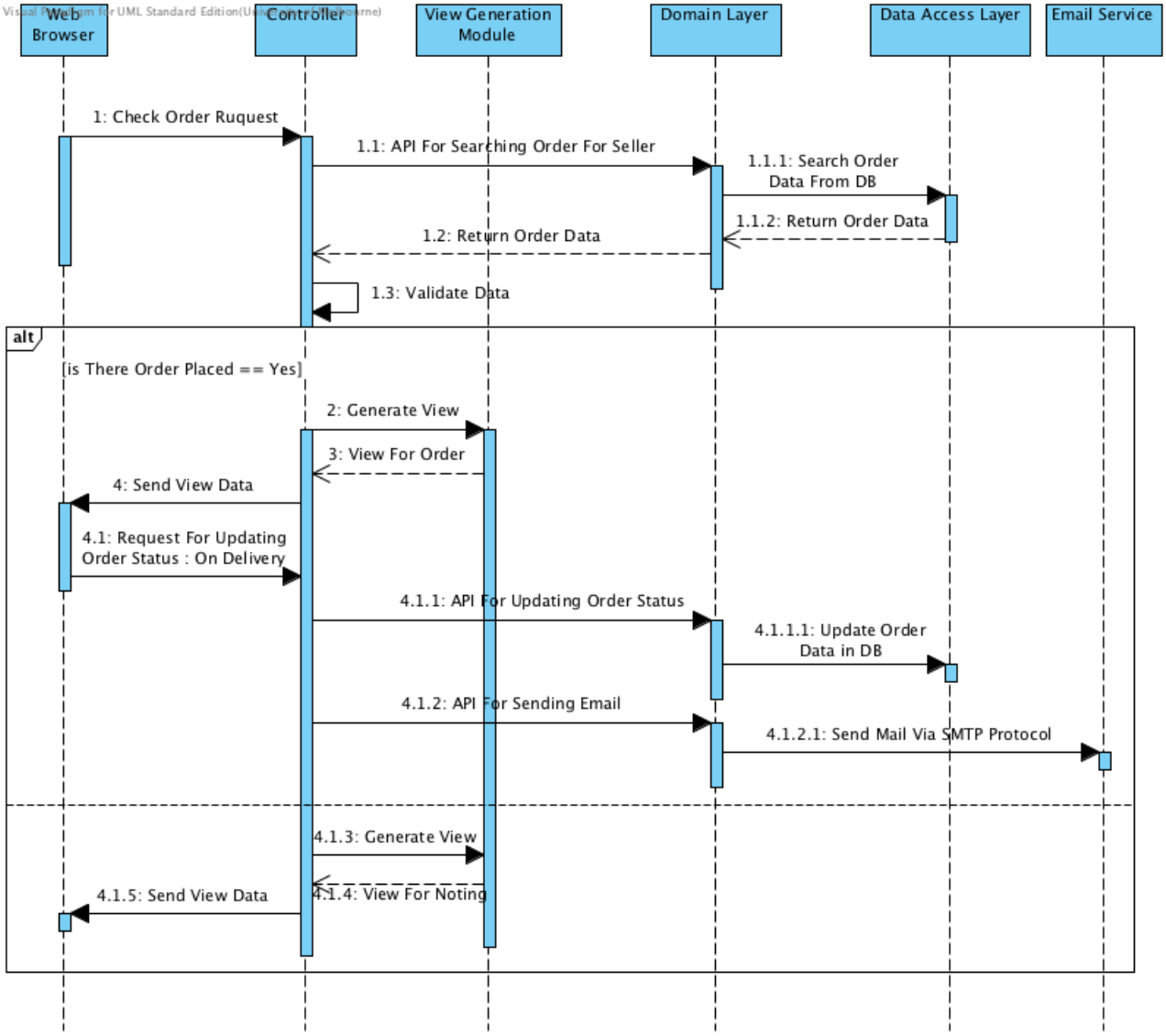
4.3.1 Add an Item



4.3.2 Delete an Item

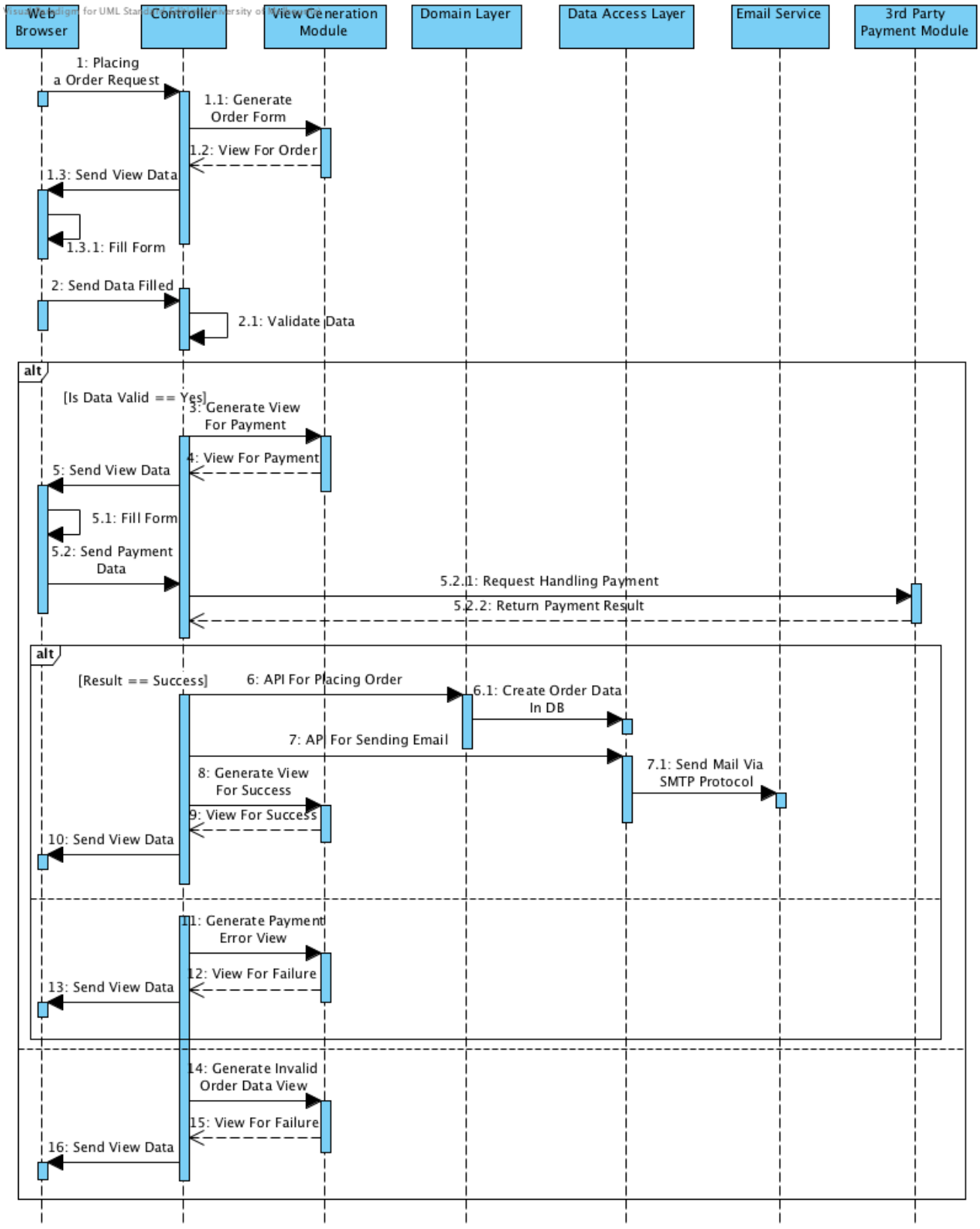


4.3.3 Check an Order



4.4 Buyer Module

4.4.1 Place an order



4.4.2 Check an order