

System and Software Architecture Description (SSAD)

FlowerSeeker

Team 05

Name	Roles
Eder Figueroa	Primary Role: Project Manager/ Implementer. Secondary Role: Tester.
Sophia Wu	Primary Role: Tester/ Trainer Secondary Role: Implementer
Doris Lam	Primary Role: IIV&Ver/ Quality Focal Point/ Implementer. Secondary Role: Tester.
Hiram Garcia	Primary Role: IIV&Ver/ Quality Focal Point/ Implementer. Secondary Role: Tester.

04/01/2013

Version History

Date	Author	Version	Changes made	Rationale
10/15/12	YB, ZC	1.0	<ul style="list-style-type: none"> Add Section 1 and Section 2.1.1 – 2.1.3 	<ul style="list-style-type: none"> Initial Draft for FlowerSeeker SSAD document
10/22/12	YB, ZC	2.0	<ul style="list-style-type: none"> Add section 2.1, 2.1.4, 2.2 	<ul style="list-style-type: none"> Add some needed new sections
10/25/12	YB	2.1	<ul style="list-style-type: none"> Add section 2.1.2 	<ul style="list-style-type: none"> Complete section 2
11/04/12	YB	3.0	<ul style="list-style-type: none"> Revise 2.1.1, 2.1.2, 2.1.3 	<ul style="list-style-type: none"> Correction for document
11/26/12	YB	4.0	<ul style="list-style-type: none"> Fix bugs, Add section 3, section 4 	<ul style="list-style-type: none"> Correct some mistake in previous versions, and add two new sections
12/05/12	YB,SW	5.0	<ul style="list-style-type: none"> Revise section 3 	<ul style="list-style-type: none"> Edit section 3 to Technology-Specific System Design and added some diagrams
12/10/12	YB,SW	6.0	<ul style="list-style-type: none"> Fix minor bugs 	<ul style="list-style-type: none"> Correct some mistakes in previous version.
12/15/12	YB	6.1	<ul style="list-style-type: none"> Fix minor bugs 	<ul style="list-style-type: none"> Correct some mistakes in previous version
2/4/13	EF	6.2	<ul style="list-style-type: none"> Updated diagrams 	<ul style="list-style-type: none"> Remove Struts FW references and replace with Spring FW Artifact diagram revised and updated for a more solid design After new incorporation of java frameworks and removal of hibernate and struts, frameworks sections was updated
2/20/13	EF	6.3	<ul style="list-style-type: none"> Updated diagrams 	<ul style="list-style-type: none"> Diagrams (artifacts model and sequence diagrams) updated after the ARB session by TAs and Instructors reviews
4/01/2013	EF	6.8	<ul style="list-style-type: none"> Updated use case Diagram 	<ul style="list-style-type: none"> After TA revision, the system behavior was revised. All general purpose ambiguous uses cases were fixed

Table of Contents

System and Software Architecture Description (SSAD)	i
Version History	ii
Table of Contents	iii
Table of Tables	iv
Table of Figures	v
1. Introduction	1
1.1 Purpose of the SSAD	1
1.2 Status of the SSAD	1
2. System Analysis	2
2.1 System Analysis Overview	2
2.2 System Analysis Rationale	19
3. Technology-Specific System Design	21
3.1 Design Overview	21
3.2 Design Rationale	35
4. Architectural Styles, Patterns and Frameworks	36

Table of Tables

<i>Table 1: Actors Summary.....</i>	<i>3</i>
<i>Table 2: Artifacts and Information Summary</i>	<i>4</i>
<i>Table 3: Hardware Component Description</i>	<i>23</i>
<i>Table 4: Software Component Description.....</i>	<i>23</i>
<i>Table 5: Florist Management Class Description.....</i>	<i>24</i>
<i>Table 6: Customer Management Class Description</i>	<i>25</i>
<i>Table 7: Product Management Class Description.....</i>	<i>27</i>
<i>Table 8: Review Management Class Description</i>	<i>28</i>
<i>Table 9: Coupon Management Class Description.....</i>	<i>29</i>
<i>Table 10: Payment Management Class Description</i>	<i>30</i>
<i>Table 11: Architectural Styles, Patterns, and Frameworks.....</i>	<i>36</i>

-

Table of Figures

<i>Figure 1: System Context Diagram</i>	<i>2</i>
<i>Figure 2: Artifacts and Information Diagram.....</i>	<i>4</i>
<i>Figure 3: Process Diagram</i>	<i>5</i>
<i>Figure 4: Hardware Component Class Diagram.....</i>	<i>21</i>
<i>Figure 5: Software Component Class Diagram</i>	<i>22</i>
<i>Figure 6: Deployment Diagram.....</i>	<i>22</i>
<i>Figure 7: Supporting Software Component Class Diagram.....</i>	<i>23</i>
<i>Figure 8: Florist Management Class Diagram</i>	<i>24</i>
<i>Figure 9: Customer Management Class Diagram.....</i>	<i>25</i>
<i>Figure 10: Product Management Class Diagram</i>	<i>27</i>
<i>Figure 11: Review Management Class Diagram.....</i>	<i>28</i>
<i>Figure 12: Coupon Management Class Diagram</i>	<i>29</i>
<i>Figure 13: Payment Management Class Diagram.....</i>	<i>30</i>
<i>Figure 14: Add Product Sequence Diagram</i>	<i>31</i>
<i>Figure 15: Florist Registration Sequence Diagram.....</i>	<i>31</i>
<i>Figure 16: List Added Product Sequence Diagram.....</i>	<i>32</i>
<i>Figure 17: Search by name Sequence Diagram</i>	<i>32</i>
<i>Figure 18: Search by price Sequence Diagram.....</i>	<i>33</i>
<i>Figure 20: Search by Occasion Sequence Diagram.....</i>	<i>33</i>
<i>Figure 21: Payment Sequence Diagram.....</i>	<i>34</i>
<i>Figure 22: Search by Location Sequence Diagram.....</i>	<i>34</i>

1. Introduction

1.1 Purpose of the SSAD

The purpose of this SSAD is to record the results of the object-oriented analysis and design of our FlowerSeeker system. The programmers and developers of our team can use this SSAD as reference to the architecture of the system that we are developing. During the development of this system, our team will be faithful to the structure of the system described in SSAD.

1.2 Status of the SSAD

This version we included redesign on artifacts diagram and class diagrams and updated on sequence diagrams based on technology revision, TA, IIV&V, Instructor and client feedback.

2. System Analysis

2.1 System Analysis Overview

The purpose of our system *Flower Seeker*, which is being developed now, is to provide an online market for flower customers and florist (flower seller). On *Flower Seeker*, customers can search for flowers based on style, price, florist rating, occasion and so on. On the other hand, it also provides a place for florist to easily upload the information of their flowers and sell them on web. Florist can purchase the membership design of flowers and make their own designs. In a word, *Flower Seeker* let the customers and florists get an opportunity to communicate with each other directly and build their business relationship based on that.

2.1.1 System Context

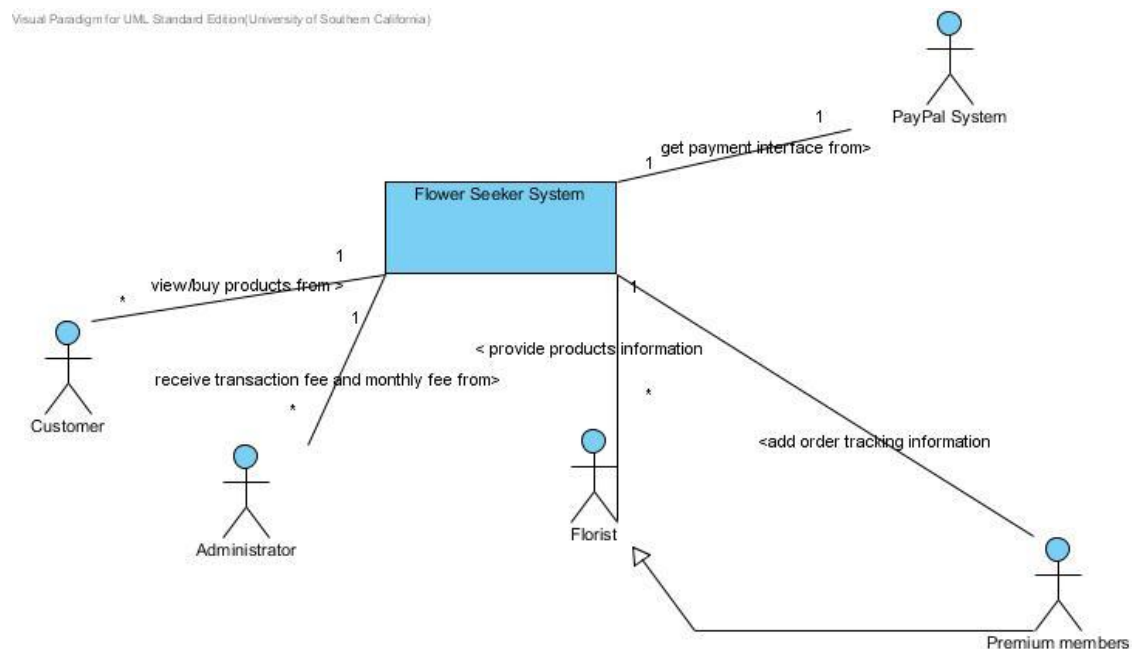


Figure 1: System Context Diagram

Table 1: Actors Summary

Actor	Description	Responsibilities
Florist	Users who sell flowers on this website	Provide detailed information of their products/flowers to Customer Ships order in limited time period once order is placed Generate coupons
Customer	Users who buy flowers from Florists on this website	Make Orders Reviewed and Ranked the product he/she bought(Not Mandatory)
Administrator	An executive/managerial staff of this website. The administrator provides all the predefined stuffs including: Predefined Occasions, licensed Design.	Provide Predefined Occasions Provide licensed Design Monitor System situation and block bad users
Premium member	Florist who pays monthly fee to FlowerSeeker website owner	Provide order tracking information
Paypal	Separate System being built by Paypal, with which the Flower Seeker Payment System must interact	Provide the interface for Flower Seeker System to give Flower Seeker System add to cart and payment system.

2.1.2 Artifacts & Information

Visual Paradigm for UML, Standard Edition (University of Southern California)

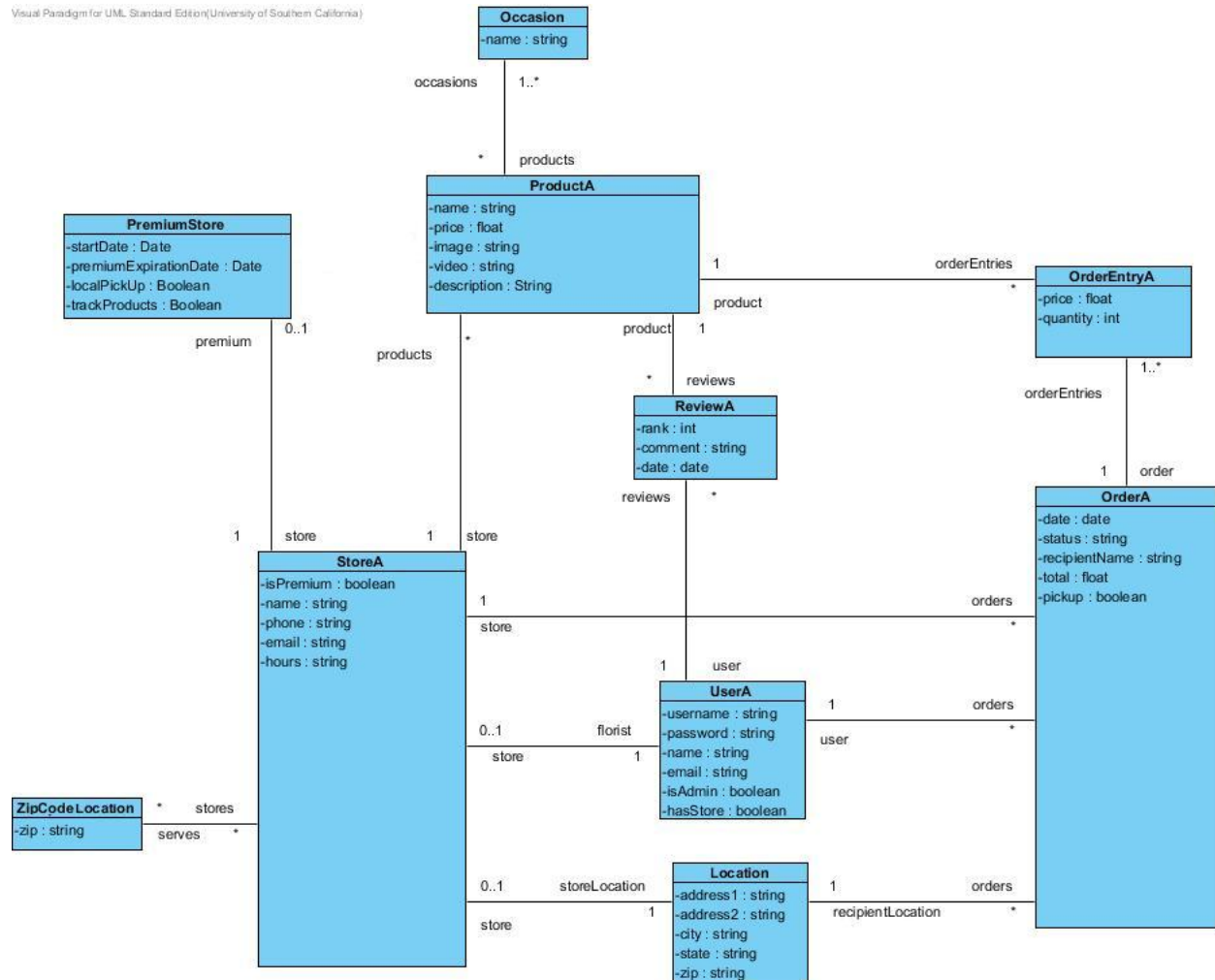


Figure 2: Artifacts and Information Diagram

Table 2: Artifacts and Information Summary

Artifacts	Information
Occasion	This table contains all information regarding the special occasion a product can have
ProductA	This entity represents a flower product in flower seeker system
PremiumStore	This table contains all information regarding the premium account for a premium florist
OrderEntryA	This entity represents all information of a final order after a product is purchased
Review	This entity contains the information needed for the rank and review on each products

Artifacts	Information
OrderA	This entity represents an order when one or many products are bought
Store	This entity represents each florist information
UserA	This entity represents each user information
zipCode	This entity contains all the destination zip codes where a store can deliver products
Location	This entity represents the physical location for each store registered

2.1.3 Behavior

Visual Paradigm for UML, Standard Edition (University of Southern California)

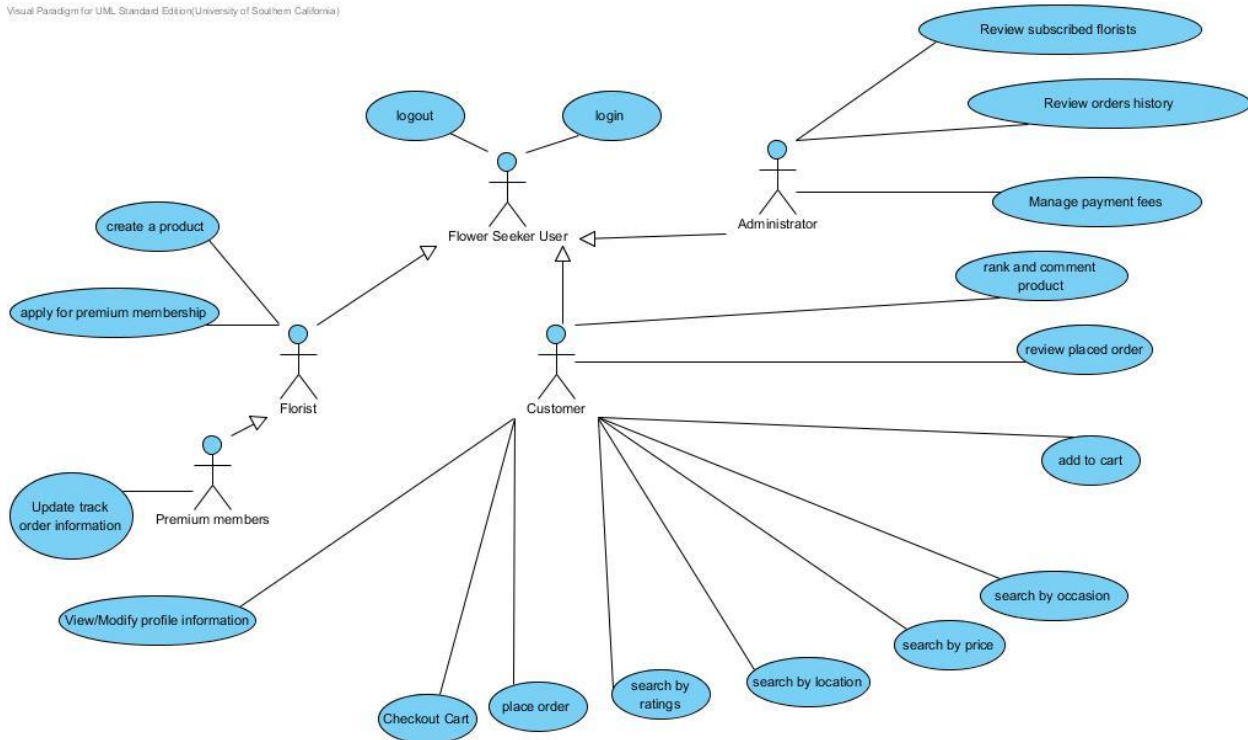


Figure 3: Process Diagram

2.1.3.1 Capability Login

2.1.3.1.1 Use Case 1

Process Description: Login

Identifier	UC-1: Login
Purpose	To determine whether the person logging into the system can be authenticated. If yes, then it need to figure out the role of the logged in person such as florist or buyer

Requirements	None
Development Risks	None
Pre-conditions	The user is currently on Login page
Post-conditions	User is authorized; the user will be given access to system operations appropriate to his/her role. Otherwise, the user is denied.

Typical Course of Action: login successful

Seq#	Actor's Action	System's Response
1	enter user name and password and click login button	
2		Send the username and password to the flower seeker login checking system. System validates entries and check for existing user.
3		Redirect the user to the corresponding role's main page

Exceptional Course of Action: login failed

Seq#	Actor's Action	System's Response
1-2	Same as typical course actions 1-2	
3		System checks: entries for whether username exists and incorrect password username and password.
4		Enables notification label "Wrong username, Wrong password" for the login page

2.1.3.1.2 Use Case 2

Process Description: logout

Identifier	UC-2: Log out
Purpose	Log out of current user and remove the sessions of current user's.
Requirements	None
Development Risks	None
Pre-conditions	A user has logged in and currently on any page.
Post-conditions	The user logouts the system

Typical Course of Action: logout successful

Seq#	Actor's Action	System's Response
1	click log out button	
2		Remove the session of current user, then redirect the page to flowerseeker home page

2.1.3.2 Capability Modify a product**2.1.3.2.1 Use Case 4****Process Description: modify product**

Identifier	UC-4: create a product
Purpose	Allow florist to modify their existing products so that customers can see it and purchase them.
Requirements	WC_1407 WC_1444 WC_1453 WC_1458 WC_1459 WC_1460
Development Risks	None
Pre-conditions	Florists have logged in and on the product management page
Post-conditions	Florists have the updated description of the products uploaded.

Typical Course of Action: finish uploaded

Seq#	Actor's Action	System's Response
1	Modifies any of the following product: -Description -Name -Price Selects: -Product image -Occasion	
2		Update the product description into the database System redirects to Florist profile page

Alternate Course of Action: Incomplete product description

Seq#	Actor's Action	System's Response
1	Leave any of the following product empty: -Description -Name -Price Selects: -Product image -Occasion	
2		System validates that all fields are non-empty. Refresh florist product page and shows what missed field must be complete.

2.1.3.3 Capability Create a product**2.1.3.3.1 Use Case 3****Process Description: create a product**

Identifier	UC-3: create a product
Purpose	Allow florist to create their products by themselves and shows it on the web so that customers can see it and purchase them.
Requirements	WC_1407 WC_1444 WC_1453 WC_1458 WC_1459 WC_1460
Development Risks	None
Pre-conditions	Florists have logged in and on the product management page
Post-conditions	Florists have the description of the products uploaded.

Typical Course of Action: finish uploaded

Seq#	Actor's Action	System's Response
1	Fills product: -Description -Name -Price Selects: -Product image -Occasion	
2		Place the product description into the database and shows it on the page with the product when ever visited.

		System redirects to Florist profile page
--	--	--

Alternate Course of Action: Incomplete product description

Seq#	Actor's Action	System's Response
1	Fills product: -Description -Name -Price Selects: -Product image -Occasion	
2		System validates that all fields are non-empty. Refresh florist product page and shows what missed field must be complete.

2.1.3.4 Capability Apply for premium membership

2.1.3.4.1 Use Case 5

Process Description: apply for premium membership

Identifier	UC-5: apply for premium membership
Purpose	Regular users can apply for premium membership in order to get premium features
Requirements	WC_1466
Development Risks	None
Pre-conditions	Florist has logged in is currently on the manage account page.
Post-conditions	Redirect to manage account page.

Typical Course of Action

Seq#	Actor's Action	System's Response
1	click the "update to premium" button	
2		Displays update to premium page
3	Fill required information and click "submit" button	

4		System saves the info in database and display Profile page
---	--	--

2.1.3.5 Capability View Customers' Info

2.1.3.5.1 Use Case 6

Process Description: View Modify/Profile Information

Identifier	UC-6: View Customers' Information
Purpose	Florist can view his customers' contact information
Requirements	WC_1461
Development Risks	None
Pre-conditions	Florist has logged in and is currently on account management page
Post-conditions	Show all the customers' information on screen.

Typical Course of Action

Seq#	Actor's Action	System's Response
1	click "Modify profile information" button	
2		System shows all customers contact information on each corresponding field.
	User modifies any information field, and press update button	
		System updates new information. Redirect to profile page

Typical Course of Action

Seq#	Actor's Action	System's Response
1	click "Modify profile information" button	
2		System shows all customers contact information on each corresponding field.

	User modifies any information field, and leaves empty information fields. Press update button	
		System refresh page and activate label “All fields must be complete”

2.1.3.6 Capability Tracking Order

2.1.3.6.1 Use Case 7

Process Description: tracking order

Identifier	UC-7: Tracking order
Purpose	Premium member can allow their customers to track orders.
Requirements	WC_1466
Development Risks	None
Pre-conditions	Premium member has logged in and is currently on review order page.
Post-conditions	Order information is updated.

Typical Course of Action

Seq#	Actor's Action	System's Response
1	click the “add phase” button	
2		System redirects to the review page
3	Add the order information and click “submit” button	
4		System adds information in database and redirect to review order page.

2.1.3.7 Capability Order Management

2.1.3.7.1 Use Case 9

Process Description: add to cart

Identifier	UC-9: Customer: Add to cart
Purpose	Virtualizing a shopping cart to give buyers the ability to add products to shopping cart and allow buyers pay all the selected products in shopping cart together.

Requirements	WC_1446 WC_1408 WC_1400
Development Risks	None
Pre-conditions	Customer has logged in and is currently on any product information page
Post-conditions	Product is added to cart

Typical Course of Action: add to cart Successful

Seq#	Actor's Action	System's Response
1	Click 'Add to cart' Button	
		System refresh cart page section reflecting product quantity and price changes.

2.1.3.7.2 Use Case 10

Process Description: Place Order

Identifier	UC-10: Place Order
Purpose	Finish Order
Requirements	WC_1446 WC_1408 WC_1400
Development Risks	None
Pre-conditions	Customer has logged in and is currently on the payment page
Post-conditions	Order is placed.

Typical Course of Action: Place Order Successful

Seq#	Actor's Action	System's Response
1	Inputs shipping information	
2	Click 'Place Order' Button	
3		System verifies shipping information. System generates communication credentials to PayPal.
4		System redirects payment, and product information to PayPal. System receives PayPal Answer

5		System records approved order. Redirect to Order Successful Display Page
---	--	---

Alternate Course of Action: Wrong shipping information

Seq#	Actor's Action	System's Response
1	Input shipping information	
2	Click 'Place Order' Button	
3		System verifies shipping information.
4		Refresh page and activate "Review incorrect information"

Alternate Course of Action: Place Order Failure

Seq#	Actor's Action	System's Response
1	Input shipping information	
2	Click 'Place Order' Button	
3		System verifies shipping information. System generates communication credentials to PayPal.
4		System redirects payment, and product information to PayPal. System receives unsuccessful payment from PayPal
5		User Refresh Page activating label PayPal unsuccessful payment label message

2.1.3.8 Capability Search

2.1.3.8.1 Use Case 11

Process Description: Search by ratings

Identifier	UC-11: Search by ratings
Purpose	Find the items that customers want by ratings
Requirements	WC_1446

Development Risks	None
Pre-conditions	Customer has logged in and is currently on the search page
Post-conditions	Display all the items related to the rating stars

Typical Course of Action: Search Successful

Seq#	Actor's Action	System's Response
1	Click 'ratings' Button	
2		Display 1-5 stars on the screen
3	Click corresponding star link	
4		Send stars to System Search module
5		Search the items with rating at least the number of stars, return the results and display them.

2.1.3.8.2 Use Case 12

Process Description: Search by location

Identifier	UC-12: Search by location
Purpose	Find the items with specific location
Requirements	WC_1398
Development Risks	None
Pre-conditions	Customer has logged in and is currently on the search page
Post-conditions	Display all the items with the input location information

Typical Course of Action: Search Successful

Seq#	Actor's Action	System's Response
1	Click 'location' Button	
2		Display location input table
3	Fill in the location information.	
4		Check the completeness of input
5		Send location information to System
6		search the items with the same location, return the results and display them

Alternate Course of Action: Search Failure

Seq#	Actor's Action	System's Response
1	Click 'location' Button	

2		Display location input table
3	Fill in the location information.	
4		Check the completeness of input
5		remind user which part he/she is missing in the input table

2.1.3.8.3 Use Case 13

Process Description: Search by price

Identifier	UC-13: Search by price
Purpose	Find the items that customers want by price
Requirements	WC_1445
Development Risks	None
Pre-conditions	Customer has logged in and is currently on the search page
Post-conditions	Display all the items related to the price

Typical Course of Action: Search Successful

Seq#	Actor's Action	System's Response
1	Click 'price' Button	
2		Display several options on the screen
3	Click corresponding price link	
4		Send price to System Search module
5		Search the items with price at most this price, return results and show on screen

2.1.3.8.4 Use Case 14

Process Description: Search by occasion

Identifier	UC-14: Search by occasion
Purpose	Find the items that customers want by occasion
Requirements	WC_1445
Development Risks	None
Pre-conditions	Customer has logged in and is currently on the search page
Post-conditions	Display all the items related to the occasion

Typical Course of Action: Search Successful

Seq#	Actor's Action	System's Response
1	Click 'occasion' Button	
2		Display several options on the screen
3	Click corresponding option link	
4		Send occasion to System Search module
5		Search the items with this occasion, return results and show on screen

2.1.3.9 Capability order management**2.1.3.9.1 Use Case 15****Process Description: review placed order**

Identifier	UC-15: review placed order
Purpose	Show the customer a list of his/her orders placed before.
Requirements	WC_1406
Development Risks	None
Pre-conditions	Customer has logged in and is currently on the profile page
Post-conditions	The orders placed by this customer is showed on current web page

Typical Course of Action: show order history successful

Seq#	Actor's Action	System's Response
1	click the show order history button	
2		System retrieves the corresponding information from the database. Then shows it to the webpage of user

Alternate Course of Action: show order history failure

Seq#	Actor's Action	System's Response
1	click the show history button	
2		System have problem to get information from database or shows it on web. Then show a dialog box to ask user try it again. redirected to previous page

2.1.3.9.2 Use Case 18

Process Description: review all the subscribed florists

Identifier	UC-18: review order history
Purpose	Show who are subscribing premium membership
Requirements	WC_1467
Development Risks	None
Pre-conditions	Admin has logged in
Post-conditions	Show all the order on the page

Typical Course of Action: Order history

Seq#	Actor's Action	System's Response
1	Click 'See order histories' link	
2		System lists all orders on groups of max 50 items per screen.

2.1.3.10 Capability Rank and Review

2.1.3.10.1 Use Case 16

Process Description: Rank and Review on florist

Identifier	UC-16: Review Scoring on florist
Purpose	Allow Customers to rank and review on florists
Requirements	WC_1451 WC_1452
Development Risks	None
Pre-conditions	Customer has logged in and is currently on any page of product that he has bought before.
Post-conditions	Review and rank star show on this product page

Typical Course of Action: Rank and Review on florist successful

Seq#	Actor's Action	System's Response
1	write comments in the comment textbook and click 1-5 star, click the 'submit' button	
2		Check the completeness, calculate the

		overall rank of this product
3		Show rank and review on this product's page

Alternate Course of Action: rank and review failure

Seq#	Actor's Action	System's Response
1	write comments in the comment textbook and click 1-5 star, click the 'submit' button	
2		Check the completeness; calculate the overall rank of this product again. redirected to previous page
3		Stay on the previous page and show what information are missing in the required fields

2.1.3.11 Capability Upgrade

2.1.3.11.1 Use Case 17

Process Description: Upgrade regular user to premium member

Identifier	UC-17: Upgrade regular user to premium member
Purpose	Upgrade regular user to premium member
Requirements	WC_1466
Development Risks	None
Pre-conditions	Admin has logged in and is currently on the grand premium member page
Post-conditions	Some regular members are changed to premium members

Typical Course of Action: Upgrade successful

Seq#	Actor's Action	System's Response
1	Click 'show requests' button	
2		System shows all the regular users that applied for premium membership but did not get grant.
3	Select regular users who applied for the premium membership and click "grant" button	
4		System change those regular users to

		premium members
--	--	-----------------

2.1.3.12 Capability Order Management

2.1.3.13 Capability View premium members' Info

2.1.3.13.1 Use Case 19

Process Description: review all the subscribed florists

Identifier	UC-18: : review all the subscribed florists
Purpose	Show who are subscribing premium membership
Requirements	WC_1467
Development Risks	None
Pre-conditions	Admin has logged in
Post-conditions	Show all the subscribed florists on the page

Typical Course of Action: Rank and Review on florist successful

Seq#	Actor's Action	System's Response
1	Click 'show subscribed members' button	
2		System shows all the premium members on the screen.

2.1.4 Modes of Operation

FlowerSeeker system will operate in only one mode, so there is no additional information concerning modes of operations provided.

2.2 System Analysis Rationale

The following is the Analysis Rationale section for the FlowerSeeker System project:

FlowerSeeker Users - these users include florists, customers, and administrator. These users require authentication in order to access the system with various permissions and functionalities based on the type of user.

There is 1 external system actor that interface with the Flower Seeker System. It is:

Google wallet system – it provides payment system but requires a monthly fee which we can accept

3. Technology-Specific System Design

3.1 Design Overview

3.1.1 System Structure

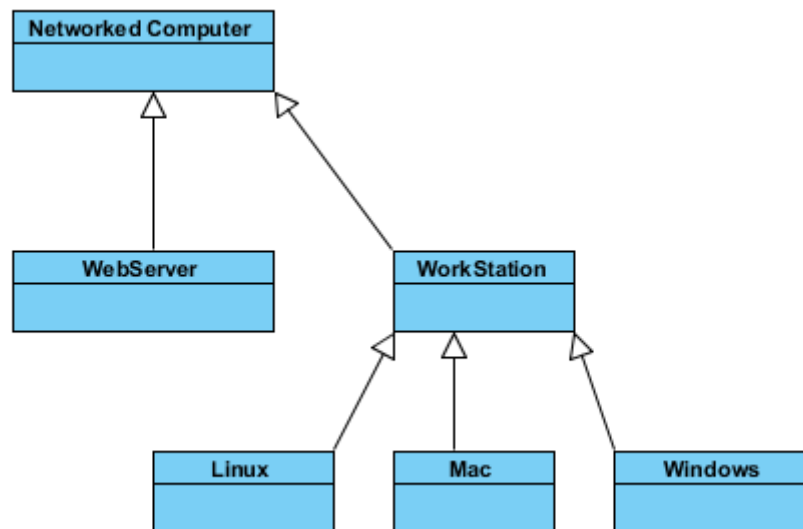


Figure 4: Hardware Component Class Diagram

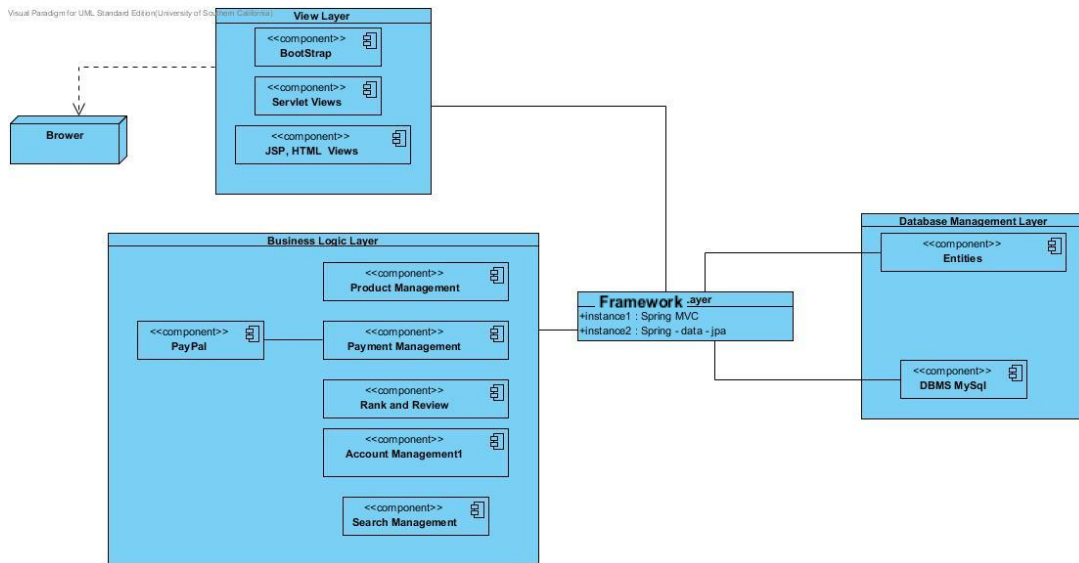


Figure 5: Software Component Class Diagram

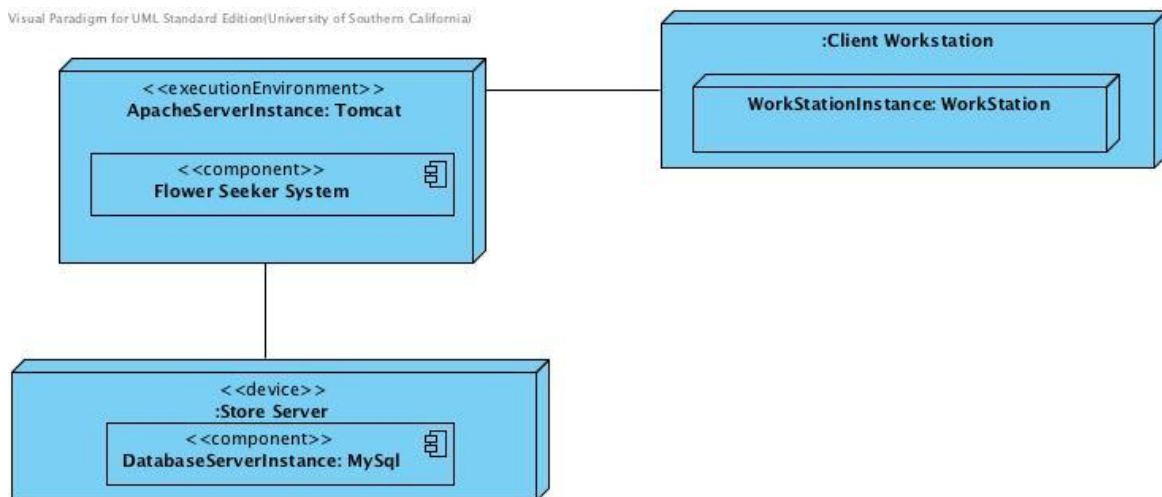


Figure 6: Deployment Diagram

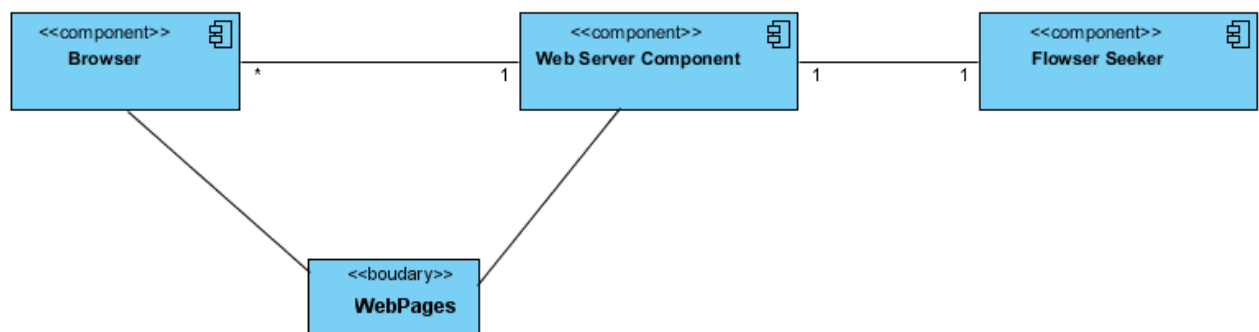


Figure 7: Supporting Software Component Class Diagram**Table 3: Hardware Component Description**

Hardware Component	Description
Web server	FlowerSeeker System deployed on it
Work station	Users' computer in which they use browser to visit this system
Linux	Users' computer with Linux OS
Windows	Users' computer with Windows OS
Mac	Users' computer with Mac OS
Networked Computer	Networked Computer

Table 4: Software Component Description

Software Component	Description
Presentation Layer	Serve as GUI of this system, mapping users' actions to business logic layer's actions to process, show the data received from the business logic layer.
Business logic layer	Manage entire business logic of the system
Data Management Layer	It contains all the tables in the database. Every Entity Class corresponds to one Table in database. DAO contains all the operating class for those entities.
Framework Layer	Provides instances to main object to all business and data management layers within the system

3.1.2 Design Classes

3.1.2.1 Account Management Classes

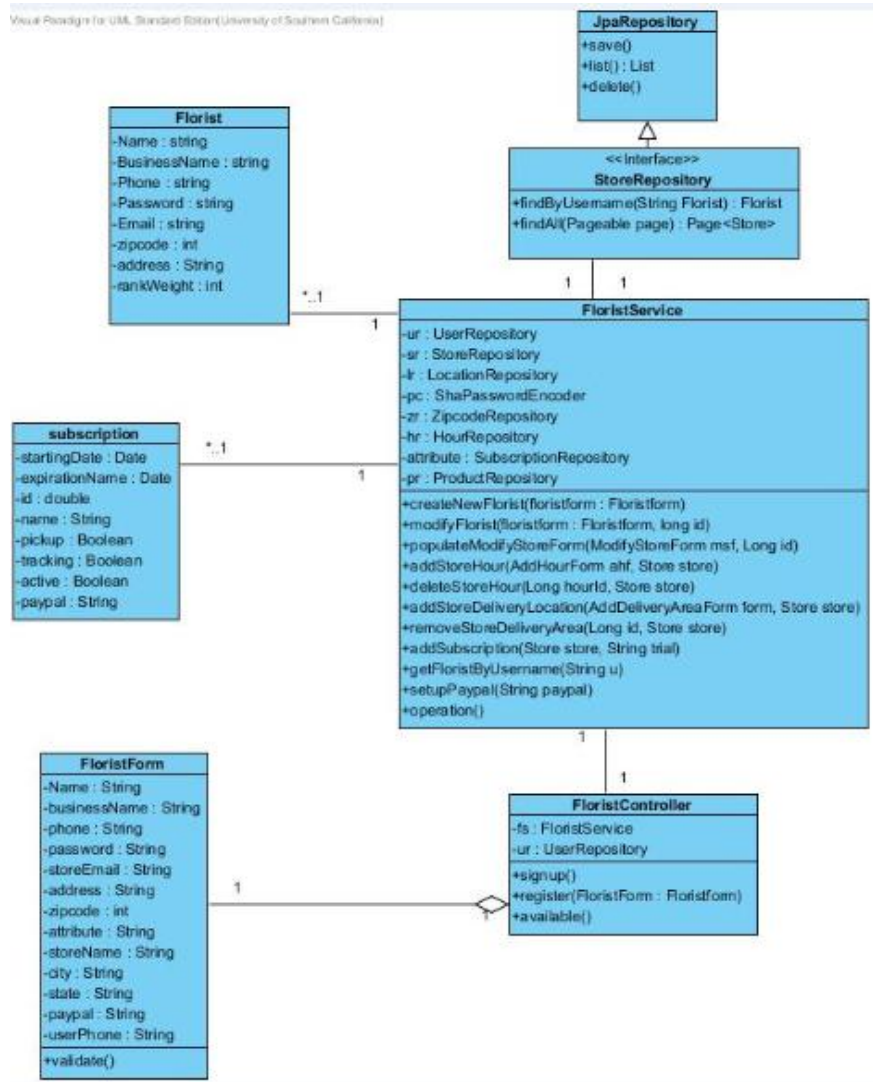


Figure 8: Florist Management Class Diagram

Table 5: Florist Management Class Description

Class	Type	Description
Florist	Entity	It contains florist data model
FloristForm	Controller	Contains the information in a form format used to render GUI in JSP
FloristController	Controller	Control events between GUI and FloristService
FloristService	Controller	Florist business logic class
FloristRepository	Controller	Interface
JpaRepository	Controller	Allows to use java JPA for serialization of Florist model
PremiumUser	Entity	Contains Florist premium entity model

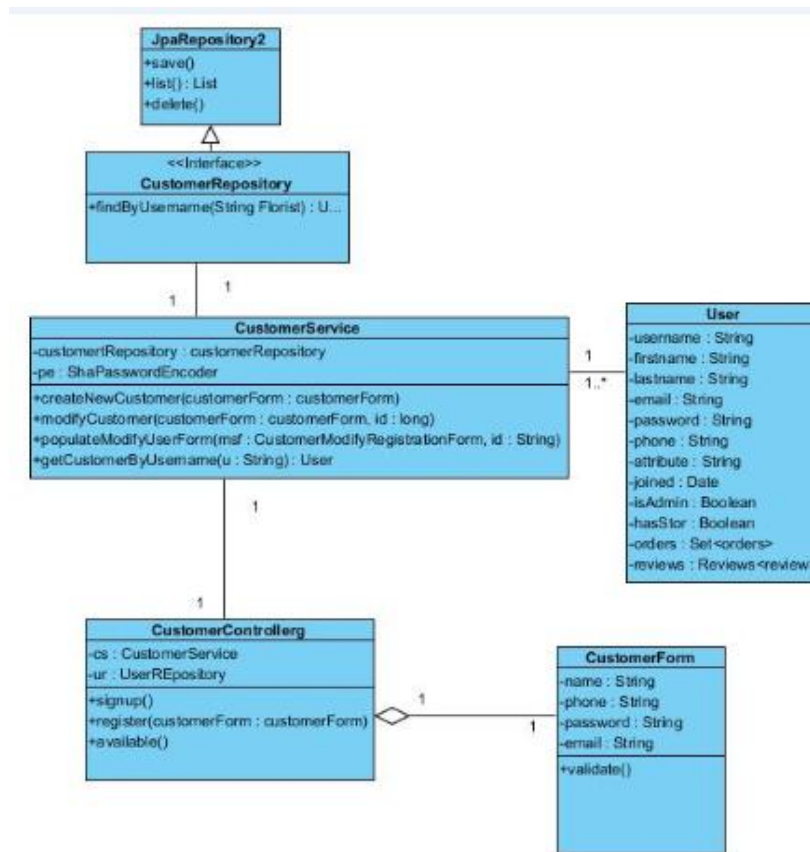


Figure 9: Customer Management Class Diagram

Table 6: Customer Management Class Description

Class	Type	Description
Customer	Entity	It contains Customer data model
Customer Form	Controller	Contains the information in a form format used to render GUI in JSP
Customer Controller	Controller	Control events between GUI and Customer Service
CustomerService	Controller	Customer business logic class
Customer Repository	Controller	Interface
JpaRepository	Controller	Allows to use java JPA for serialization of Customer model

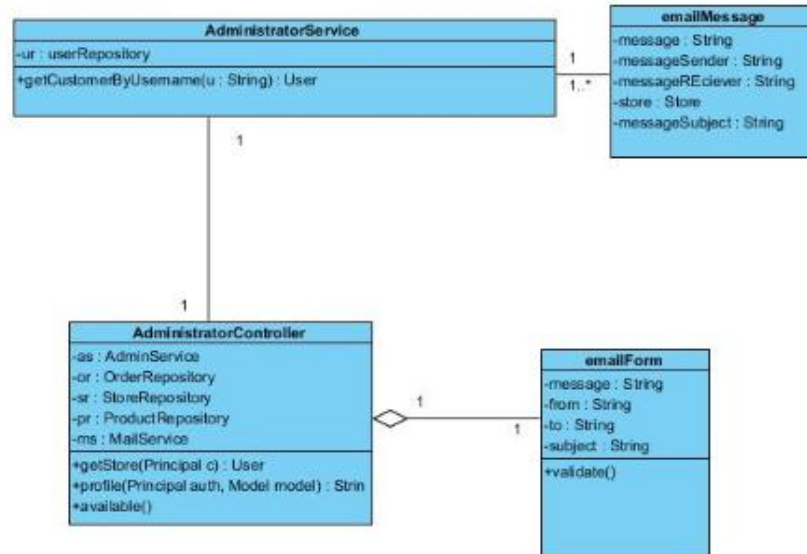


Figure 10: Administrator Management Class Diagram

Table 7: Administrator Management Class Description

Class	Type	Description
emailMessage	Entity	It contains Administrators email data model
emailForm	Controller	Contains the information in a form format used to render GUI in JSP
AdministratorController	Controller	Control events between GUI and AdministratorService
administratorService	Controller	Administrator business logic classes

Visual Paradigm for UML, Standard Edition (University of Southern California)

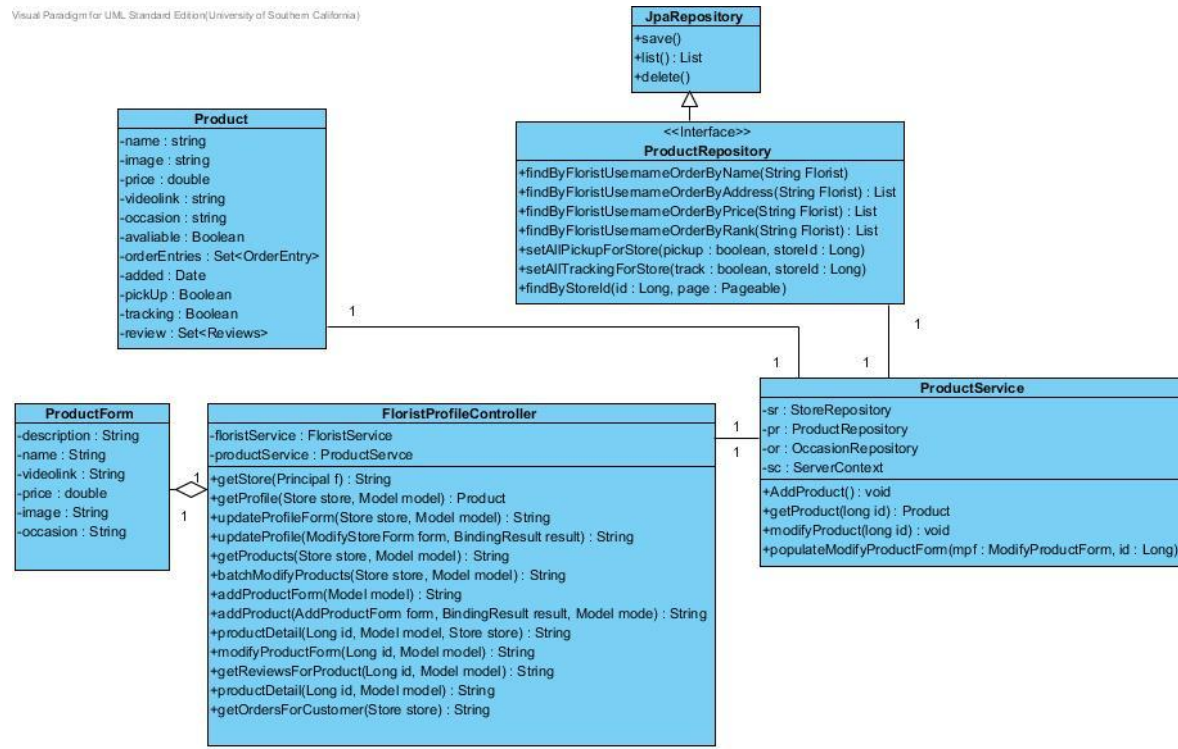


Figure 11: Product Management Class Diagram

Table 8: Product Management Class Description

Class	Type	Description
Product	Entity	It contains Product data model
Product Form	Controller	Contains the information in a form format used to render GUI in JSP
Product Controller	Controller	Control events between GUI and Product Service
Product Service	Controller	Product business logic class
Product Repository	Controller	Interface
JpaRepository	Controller	Allows to use java JPA for serialization of Product model

Visual Paradigm for UML, Standard Edition (University of Southern California)

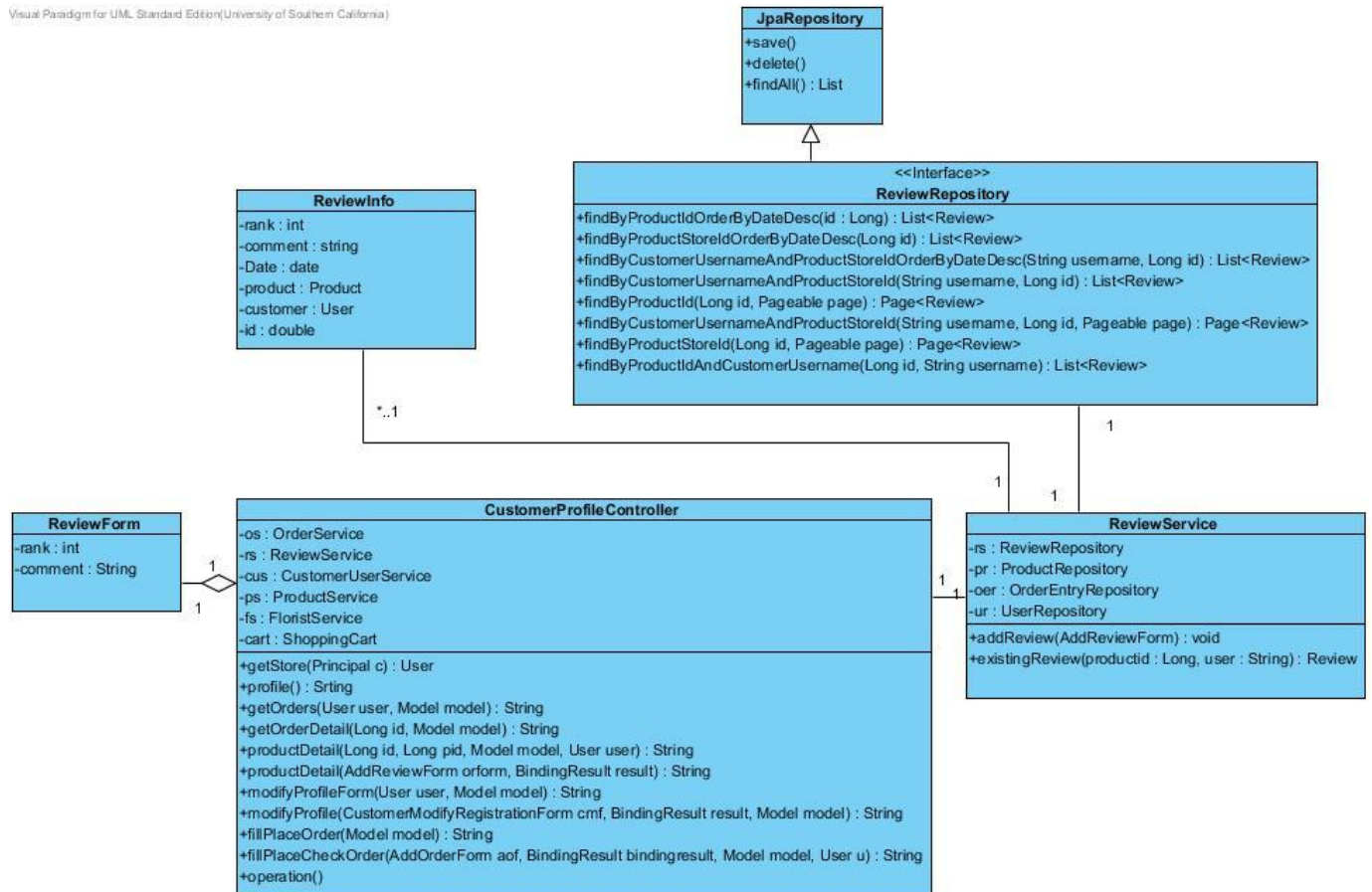


Figure 12: Review Management Class Diagram

Table 9: Review Management Class Description

Class	Type	Description
Review	Entity	It contains Review data model
Review Form	Controller	Contains the information in a form format used to render GUI in JSP
ReviewController	Controller	Control events between GUI and Review Services
ReviewService	Controller	Review business logic class
ReviewRepository	Controller	Interface
JpaRepository	Controller	Allows to use java JPA for serialization of Product model

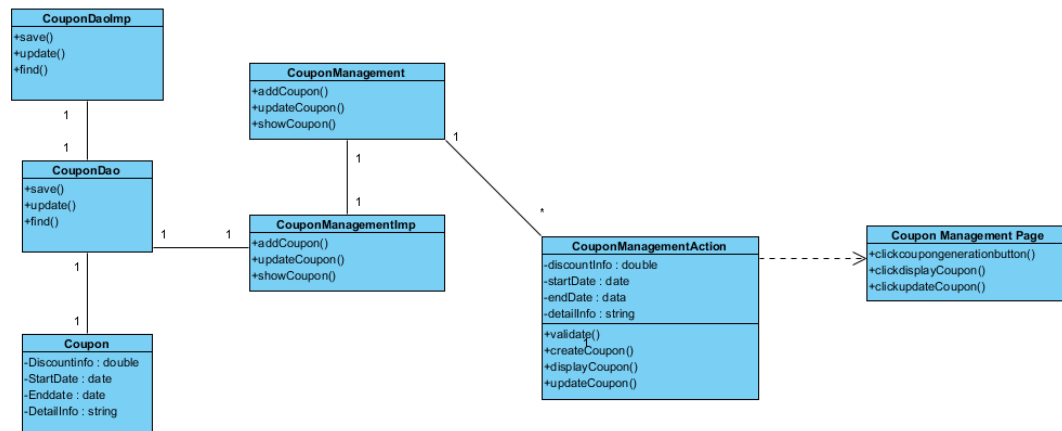


Figure 13: Coupon Management Class Diagram

Table 10: Coupon Management Class Description

Class	Type	Description
Coupon	Entity	It contains Coupon's data
Coupon Dao	Controller	Interface
Coupon DaoImp	Controller	Implements Product Dao
CouponManagementImp	Controller	Implements ProductManagement
CouponManagementAction	Controller	Validate inputs and call on corresponding services.
CouponManagement	Controller	Interface
CouponManagementPage	Boundary	Web GUI

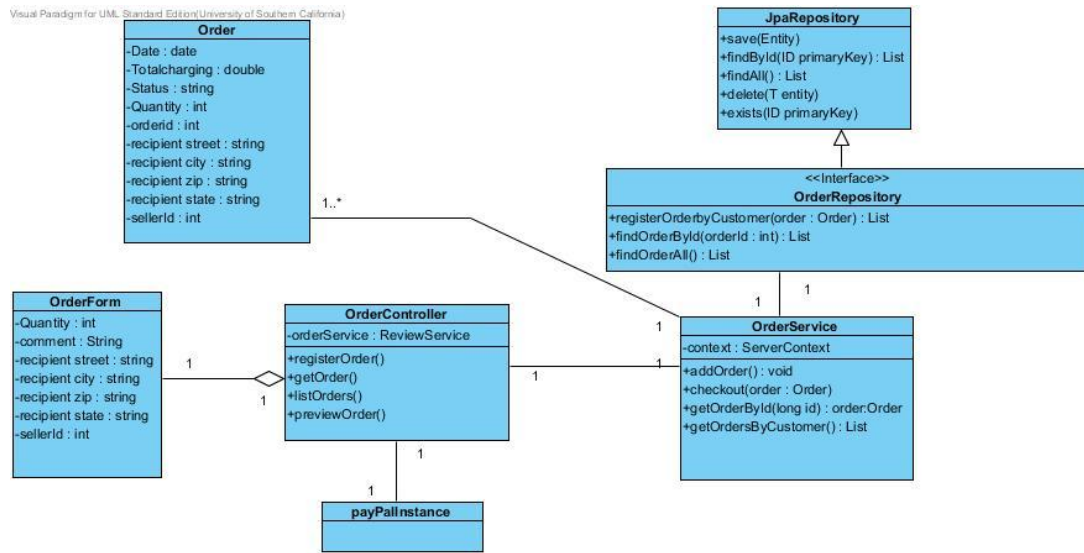


Figure 14: Payment Management Class Diagram

Table 11: Payment Management Class Description

Class	Type	Description
Order	Entity	It contains Order data model
Order Form	Controller	Contains the information in a form format used to render GUI in JSP
Order Controller	Controller	Control events between GUI and Order Services
Order Service	Controller	Order business logic class
Order Repository	Controller	Interface
JpaRepository	Controller	Allows to use java JPA for serialization of Order model
GoogleWalletInstance	Controller	NDI that manages checkout process

3.1.3 Process Realization

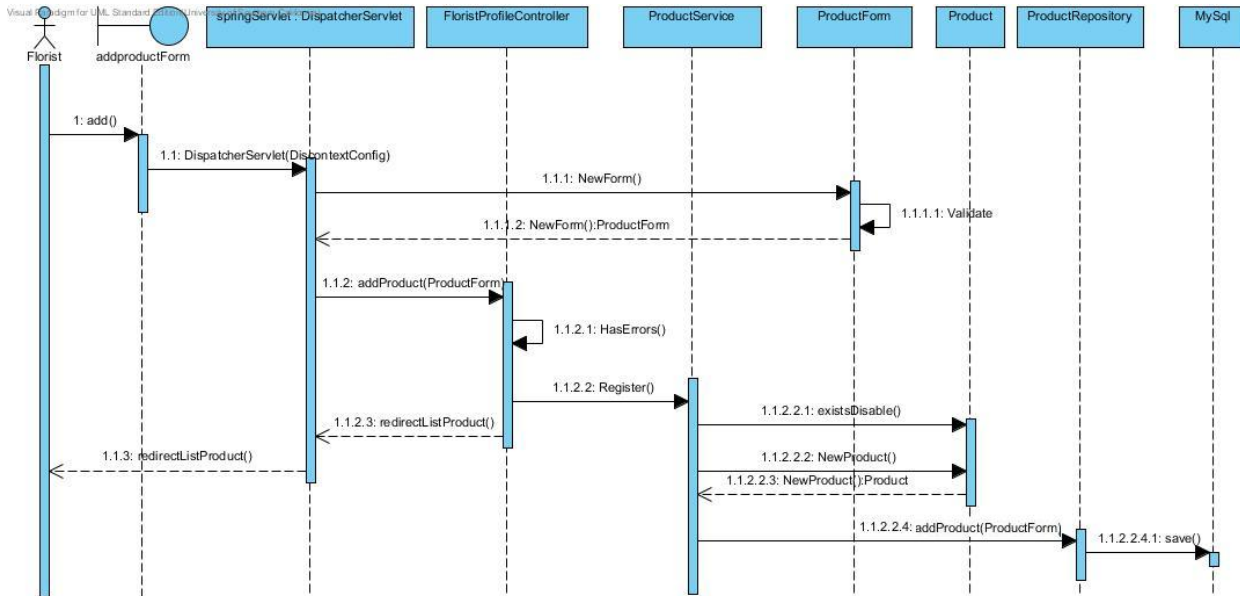


Figure 15: Add Product Sequence Diagram

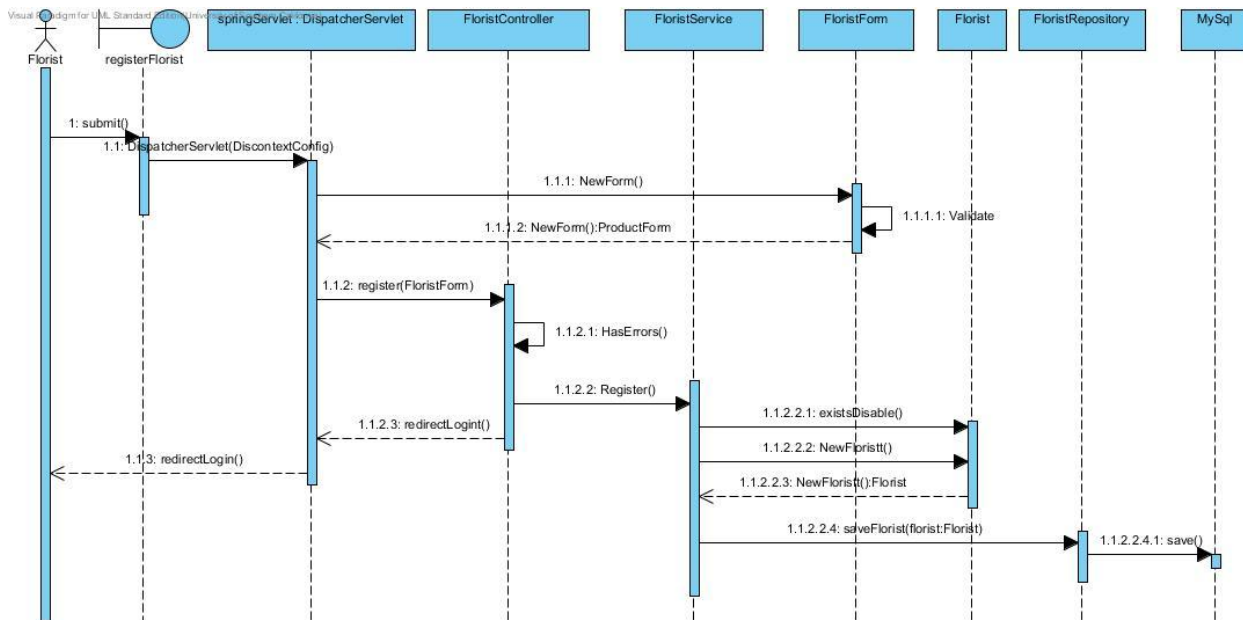


Figure 16: Florist Registration Sequence Diagram

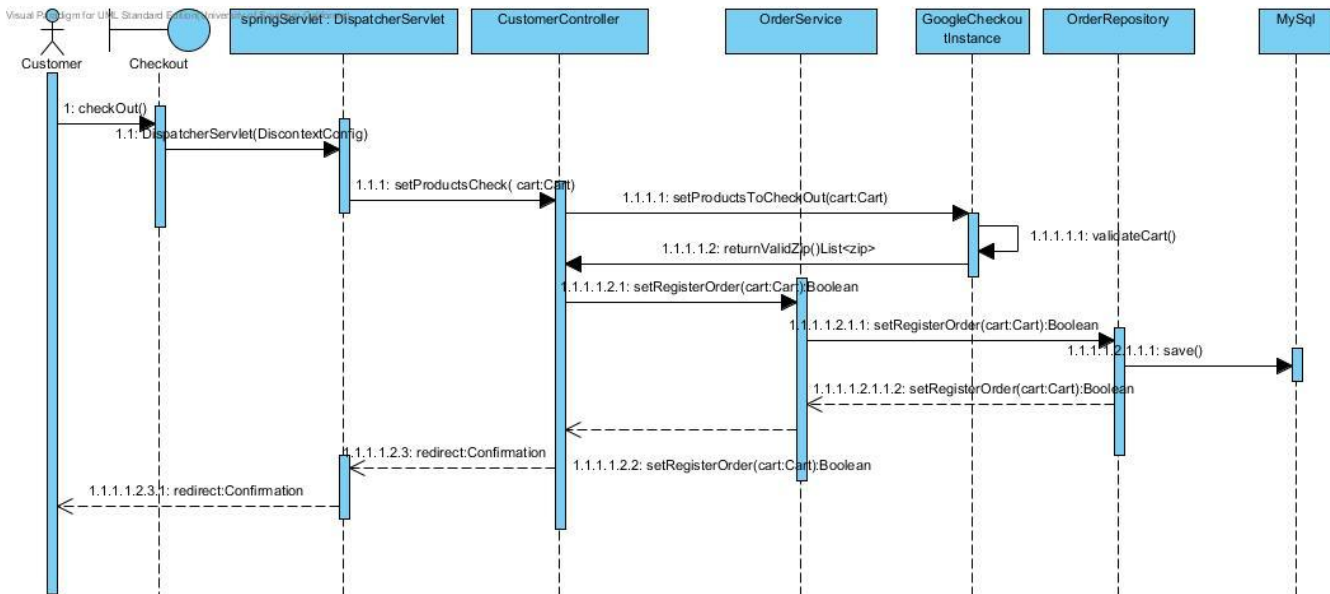


Figure 17: List Added Product Sequence Diagram

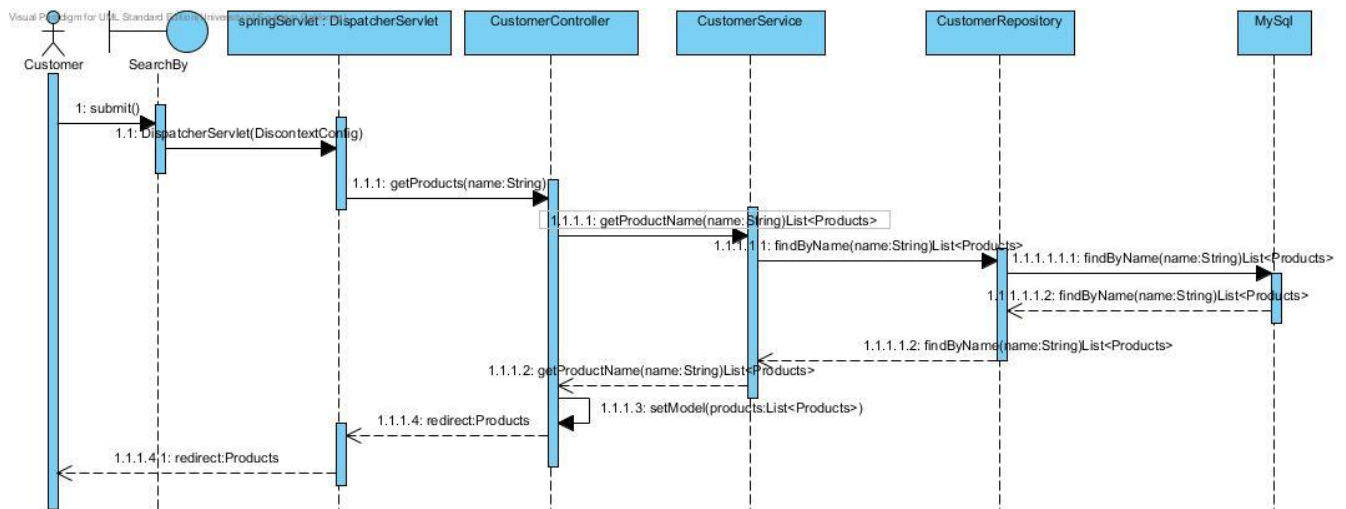


Figure 18: Search by name Sequence Diagram

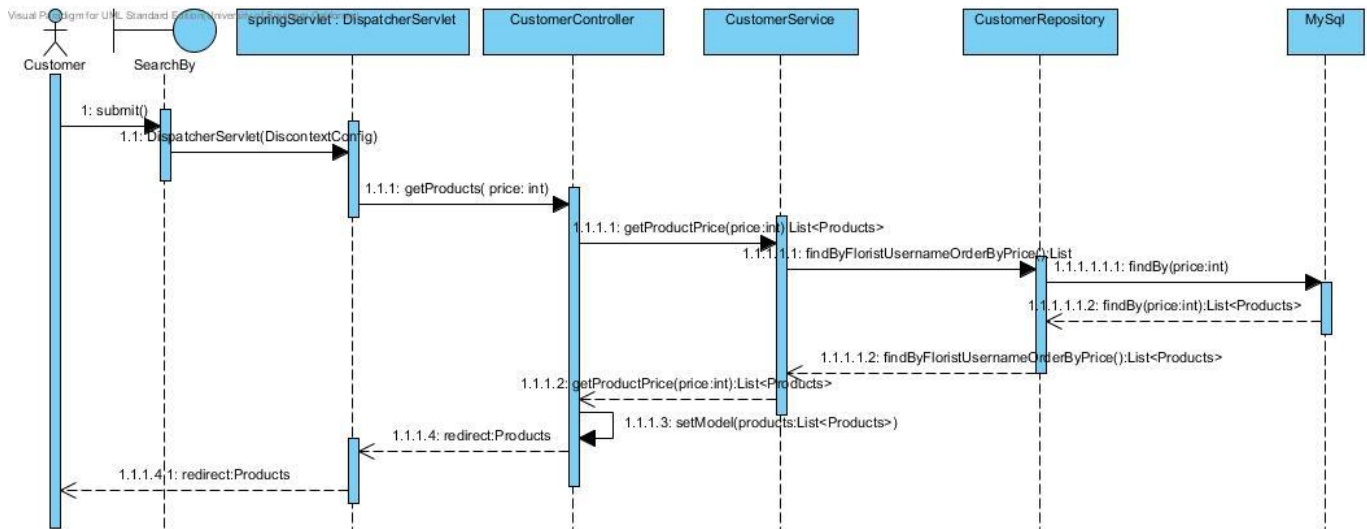


Figure 19: Search by price Sequence Diagram

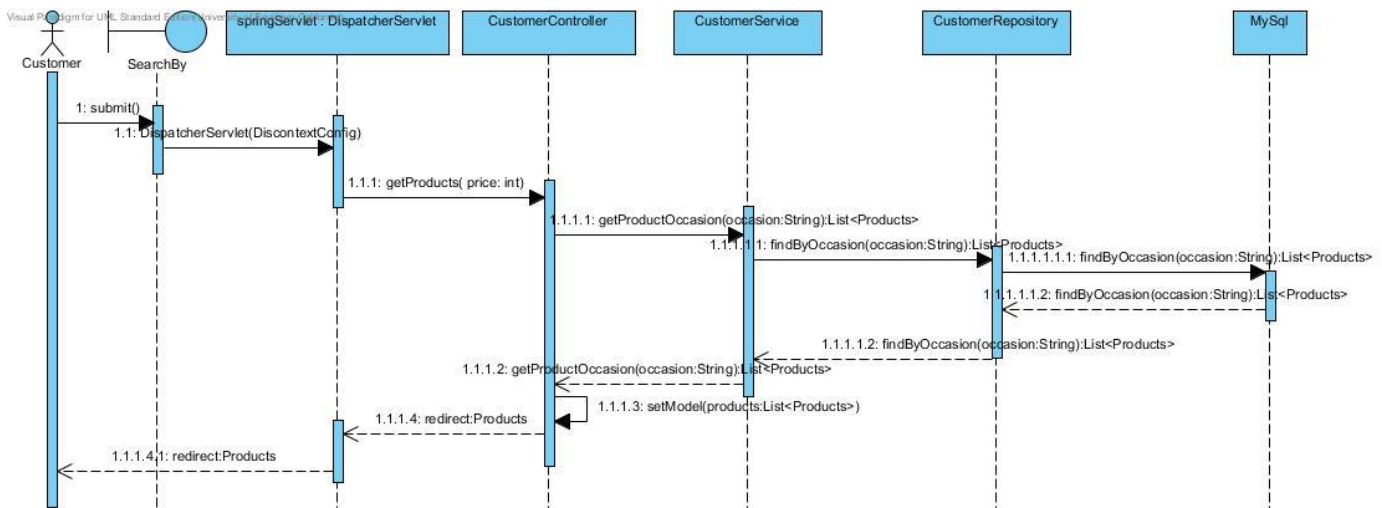


Figure 20: Search by Occasion Sequence Diagram

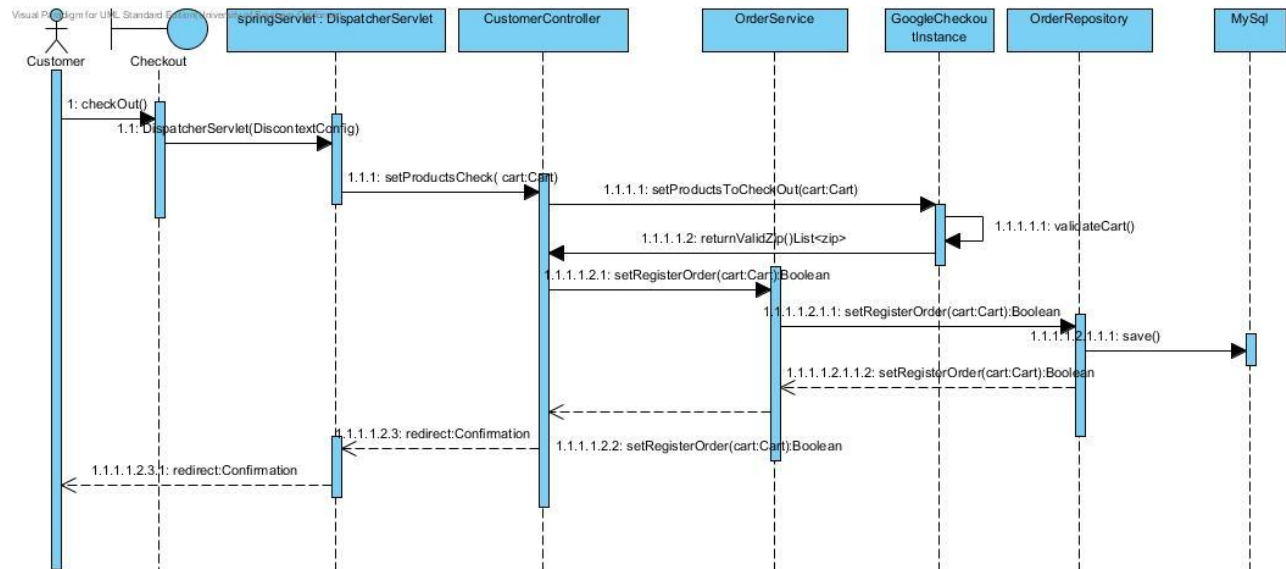


Figure 21: Payment Sequence Diagram

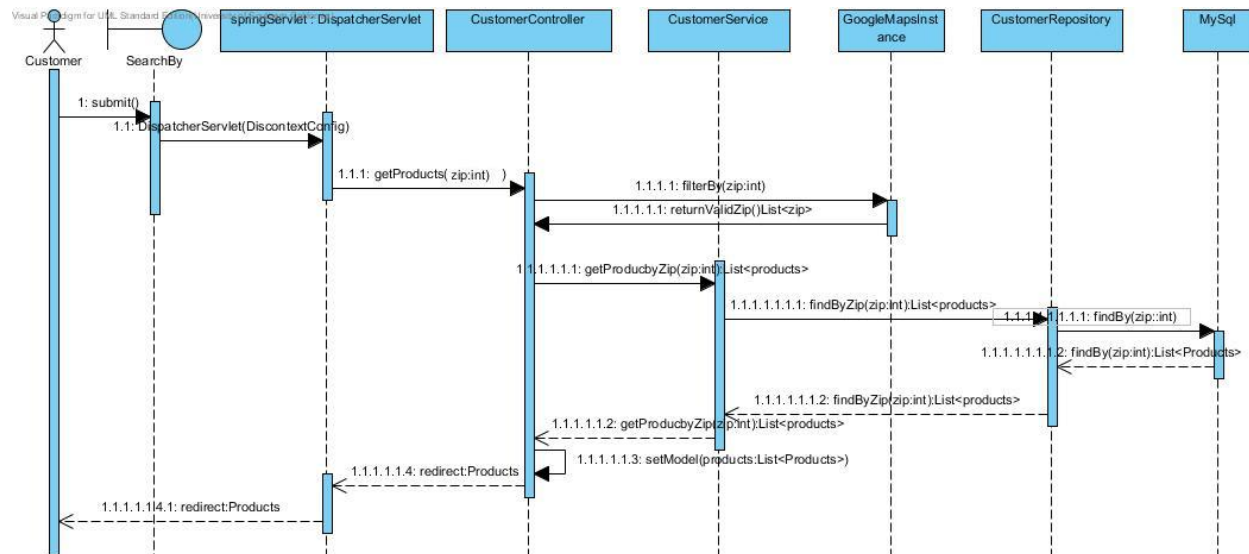


Figure 22: Search by Location Sequence Diagram

3.2 Design Rationale

Our System is based on J2EE Project. It has three layers.

Presentation Layer (View): includes all the boundaries (pages) and view form classes.

Business Logic Layer (Control): includes all the controllers and various logic algorithms procedures. Validate user's input and mapping user's input to different operations.

Data Management Layer (Model): includes all the entities. Every Entity corresponds to one table in Database.

Frame - Work Layer: this layer is in charge of all the main objects creation, injection and database mapping objects and connections between the whole system layers.

4. Architectural Styles, Patterns and Frameworks

Table 12: Architectural Styles, Patterns, and Frameworks

Name	Description	Benefits, Costs, and Limitations
Browser-Server Framework	Our project is a web-project. Client does not need to install any software, just need to have a browser on computer and connect to internet	<p>Benefits:</p> <ul style="list-style-type: none"> (1) Redeploy project is very easy. (2) Support multiple platforms. <p>Limits:</p> <ul style="list-style-type: none"> (1) Some features cannot be built because of the band width.
Google Wallet	Google wallet checkout provides a defined API that handles all payment processes in a secure manner	<p>Benefits:</p> <ul style="list-style-type: none"> 1) provides Secure, testable and available payment module for the system 2) Allows multiple payment methods <p>Limits:</p> <ul style="list-style-type: none"> 1) Users need to create a account with Google <p>Cost: 1.9% per transaction</p>
JPA	JPA enforces object oriented programming style. Allowing communication between objects and databases by replacing direct persistence communication between object-persistence entities	<p>Benefits:</p> <ul style="list-style-type: none"> 1) Implementation of database management objects becomes easier and less coupled <p>Limits:</p> <ul style="list-style-type: none"> 1) JPA only deal with objects, if the system needs to retrieve simple data from a table, the entire object must be constructed and passed <p>Cost: Free</p>
MVC	Object oriented design pattern. This pattern allows separating the logic, with the system view. Injecting all object oriented pattern benefits for a scalable project	<p>Benefits:</p> <ul style="list-style-type: none"> 1) Allows system scalability 2) Enhance developers system domain knowledge

		Limits: No known for FlowerSeeker Cost: Free
SpringFramework	Framework that helps to build robust enterprise java applications, it provides different components for specific task such as database management. Free of charge and lightweight.	Benefits: Allows to decouple object on the system and provides several useful framework for database and view layers Cost: Free of charge, but needs high learning curve from developers