**Great Outdoor Documentation**



**Team Members :**

|  |  |
| --- | --- |
| Team 1 | Team 2 |
| Aniket Sammadar(Scrum Master) | Agnibha Chandra(Scrum Master) |
| Charu Arora | Shalu Panwar |
| Anshu Kumar | Ayushi Dixit |
| Rintu Mazumder | Kunal Roychoudhury |
| Azhar Hussain | Sujit Kumar |
| Aman Raj | Vikas Singh |

Capgemini India Technology Services

Plot 72 & 73, EPIP Zone, Vijayanagar, KIADB Export Promotion Industrial Area, Whitefield, Bengaluru, Karnataka 560066

Table of Contents

1. Introduction ------------------------------------------------------------------------------------
2. Overview-----------------------------------------------------------------------------------------
3. Team Member Work Allocation-------------------------------------------------------------
4. Epic & Stories -----------------------------------------------------------------------------------
5. Use Cases ----------------------------------------------------------------------------------------
   * 1. Product Management System -----------------------------------------------------------
6. Add a Product ---------------------------------------------------------------------
7. Edit a Product ---------------------------------------------------------------------
8. Delete a Product------------------------------------------------------------------
9. Add a Product Master------------------------------------------------------------
10. Search a Product------------------------------------------------------------------
11. Filter a Product -------------------------------------------------------------------
    * 1. Address Management System-----------------------------------------------------------

A) Add an Address--------------------------------------------------------------------

B) Update an Address----------------------------------------------------------------

C) Delete an Address-----------------------------------------------------------------

* + 1. Wishlist Management System -----------------------------------------------------------

A) Add a Product ---------------------------------------------------------------------

B) View Wishlist ---------------------------------------------------------------------

iv)Adding Item To cart------------------------------------------------------------------------

v)Placing the Order---------------------------------------------------------------------------

vi) Cancel Management System-------------------------------------------------------------

A) Cancel an Order--------------------------------------------------------------------

B) Cancel a Product-------------------------------------------------------------------

vii) Shelf Time Report------------------------------------------------------------------------

viii)User Management System-----------------------------------------------------------

A) User Registration--------------------------------------------------------------------

B) User Login----------------------------------------------------------------

C) User Logout-----------------------------------------------------------------

ix) Retailer Inventory Management System-----------------------------------------------

A) View Inventory

B) Update Receive Time

C) Update Sale Time

x) Go Admin Report Management System

A) Revenue Report

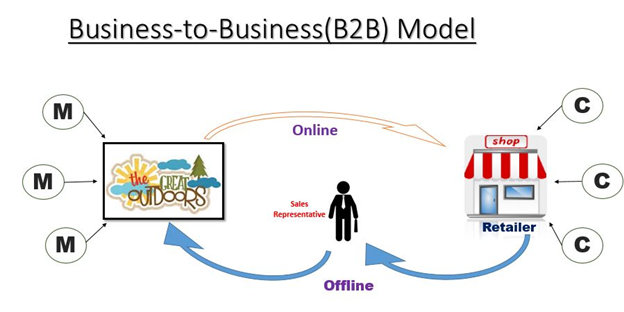
B) Growth Report

xi) Delivery Time Report --------------------------------------------------------------------

1. Class Diagram ----------------------------------------------------------------------------------
2. ER Diagram--------------------------------------------------------------------------------------
3. Test Case Scenario -----------------------------------------------------------------------------
4. BDD------------------------------------------------------------------------------------------------
5. Conclusion----------------------------------------------------------------------------------------

1. Introduction

Great Outdoors (GO) is an electronic distributor of outdoor products. GO sources the products from various manufacturers and supplies it to Retailers across various countries.



GO primarily deals with *5 Product Categories*:

1. Camping Equipment

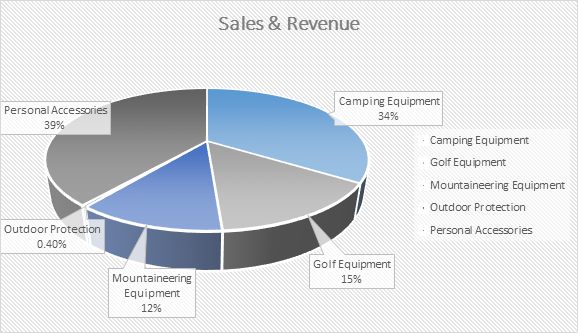
2. Golf Equipment

3. Mountaineering Equipment

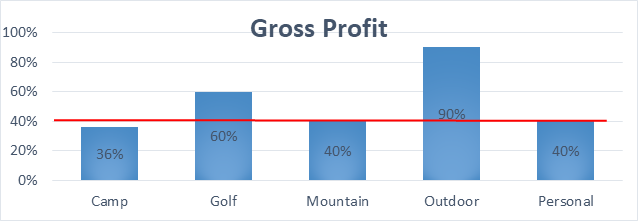
4. Outdoor Protection

5. Personal Accessories

*Contribution to Topline*



*Contribution to Bottom-line*



2.Overview

Great Outdoors have the following functional requirements:

* Order Management
* Change Revenue Mix
* Return Management
* Retail Store Inventory

The model followed was an agile model. Agile SDLC model is a combination of iterative and incremental process models with focus on process adaptability and customer satisfaction by rapid delivery of working software product. Agile Methods break the product into small incremental builds. These builds are provided in iterations.

The project had eight sprints to undergo. Every sprint had incremental learning. The first sprint had design and BDD/TDD implementations to display followed by core java and collection implementation in the second sprint. The database was static made through collections to enhance and have a practical knowledge on the collections framework. The third sprint had JDBC implementation in the middle layer. The UI was not yet done so we had a presentation layer in core java itself to test our functionalities. The next sprint bought in the UI and the BDD approach in Angular followed by Cucumber and Selenium testing. The next sprint got the entire user interface ready with Rest services and JPA implemented. Spring Framework was introduced to us in the next sprint with Rest implementation incorporated into Spring because of its feasibility. The last sprint got Spring converted to Spring Boot along with Logger implementation and herein the entire application is ready to serve a shopping site with the major functionalities.

4. Epic & Stories

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Epic | Stories | As a/an | I want to | So that .. |
| Customer | Register | User | Register into the system | A new user is added to the database. |
| Login | User | Login to the system | I’m logged in to use my functionalities. |
| Search Hotels | User | Search for Hotel Rooms | To find a room suitable as per my requirements. |
| Book Hotels | User | Book Hotel rooms | I can book my preferred room. |
| View Booking Transactions | User | View My booking transactions | I can see my current booking status & Booking transaction history. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Epic | Stories | As a/an | I want to | So that .. |
| Hotel Employee | Login | Hotel Employee | Login to the system using the credentials as provided by Hotel Booking company. | I’m logged in to use my functionalities. |
| Book Rooms | Hotel Employee | Book rooms of my own hotel. | I can book rooms of my hotel if there is an offline Customer. |
| View Booking Transactions | Hotel Employee | View my hotel’s booking transactions | I can see which rooms are booked for which dates, and which rooms are available for customers. |
| Enable & Disable Rooms | Hotel Employee | Toggle the availability of rooms in my hotel. | I can make certain rooms available or unavailable for online booking. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Epic | Stories | As a/an | I want to | So that .. |
| Admin | Login | Admin | Login to the system | I’m logged in to use my functionalities. |
| Add Hotel  Delete Hotel | Admin  Admin | Add Hotels  Delete Hotels | New Hotels can be added to system’s database.  Hotels can be removed from the database. |
| Modify Hotel | Admin | Modify Hotels | I can update hotel details such as description. |
| Add Rooms | Admin | Add rooms | I can add new rooms in a hotel |
| Delete Rooms | Admin | Delete rooms | I can delete rooms from a hotel |
| Modify Room | Admin | Modify Room details | Update room details such as Price per night. |
| Update Offers | Admin | Update Offers on hotels and rooms | I can attract more customers |
| View Reports | Admin | View various booking reports | I can view list of available hotels, bookings of specific hotels, and bookings for specific dates |

5. Use Cases

5.1 Product Management System

Overview

Product is one of the most important aspects of any e-commerce website. Great Outdoors follows a Business to Business model. Great Outdoor is currently focusing on 5 categories of products like camping, mountaineering, outdoors, personal and golf. Our main aim is to make a user friendly product management system where operations can be performed with one click. We tried to improvise the update option in such a way that the update form will be pre field with the data of the product such that the user has to only update the field user wants. Our Software is also made highly secured using authentication guard of angular. Product Management System can only be accessed with proper credentials of a product master.

Prerequisite

User must be logged in as product master to perform the add product, edit product and delete product functionalities.

User must be logged in as admin to add product master.

Filter Product, Search Product and view Product functionalities can be performed by any user.

Non Functional Requirement:

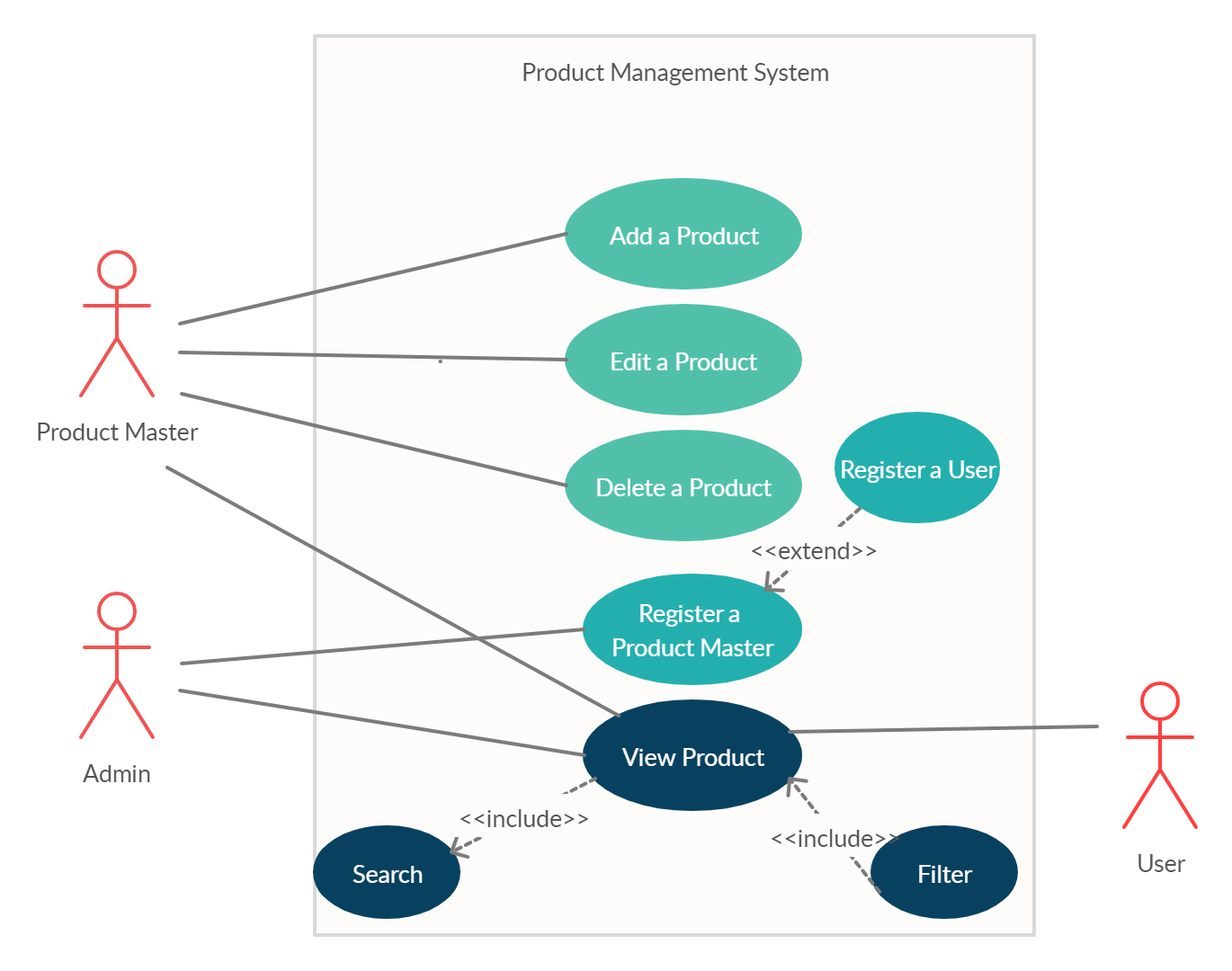
All modules are show designed that the result will be obtained within 10 seconds of request. Proper Loading indicator has been instantiated. Hence once a request been asked no one can further request anything. Authentication has been given a higher priority. All form validations are properly maintained. To make it more user friendly proper alert messages are shown wherever required. For instance if someone click on any other tab while filling up a form, our software will show a pop up alert – “all the data will be discarded” before navigation to the clicked tab. Pop message will also be shown during any action like delete add or edit product. Any changes in database will only be performed if user gives a confirmation to those pop up warnings.

Designed and Implemented By:

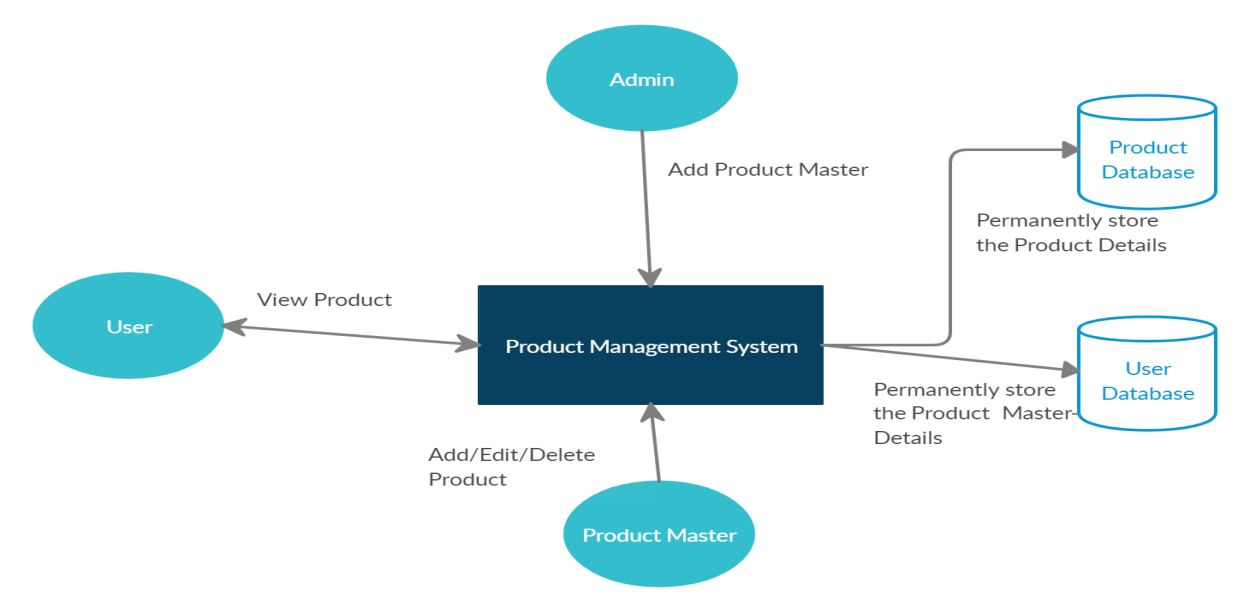
Agnibha Chandra ( Id : 46002001)

Designation : Senior Analyst/ A5

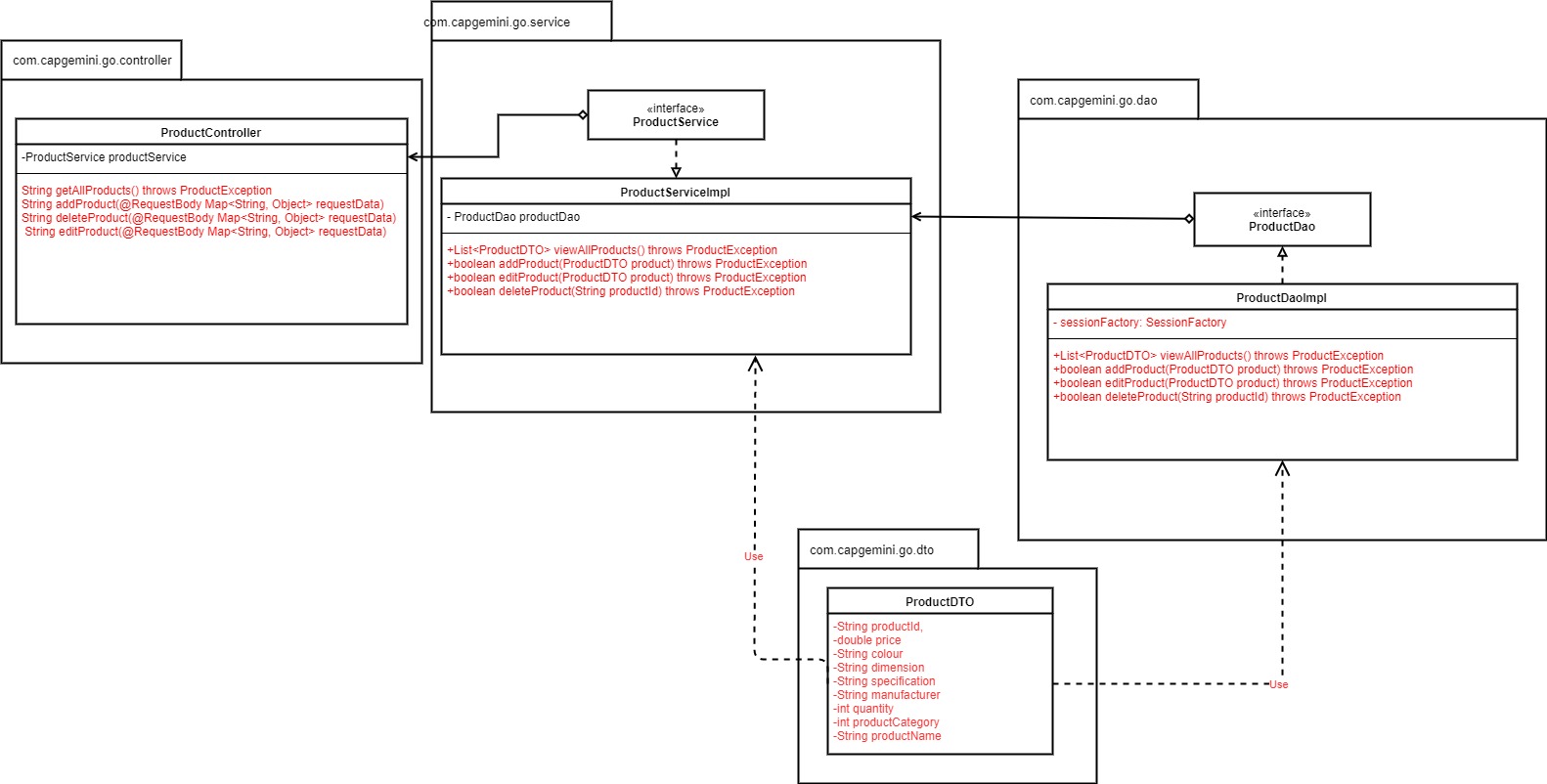
Use Case Diagram for Product Management System



Product Management Data Flow Diagram Level 0



Class Diagram for Product Management



a) Add a Product

This module has been designed to add a product in the database. This can be only accessed by Product Master. The form is designed with proper validation.

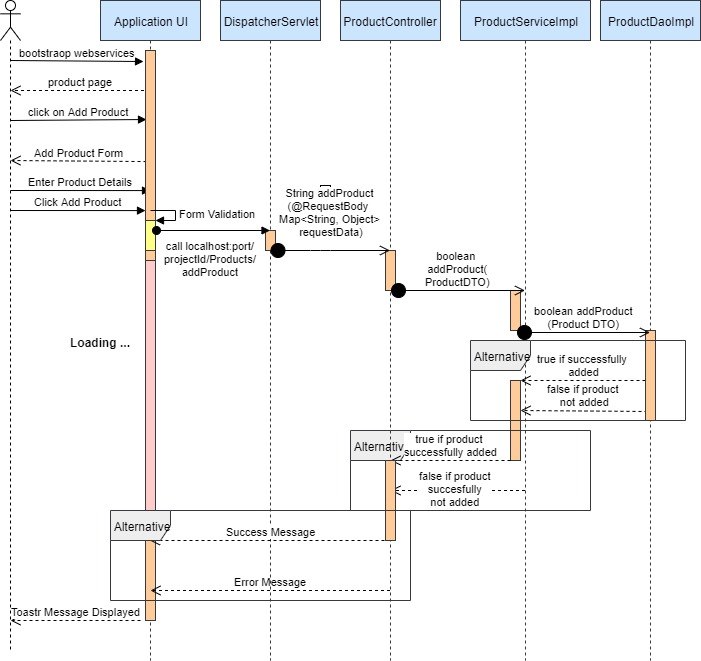
Validation:

Product Id: product id should be alphanumeric and must start with an alphabet

Images: Should be in .jpg format

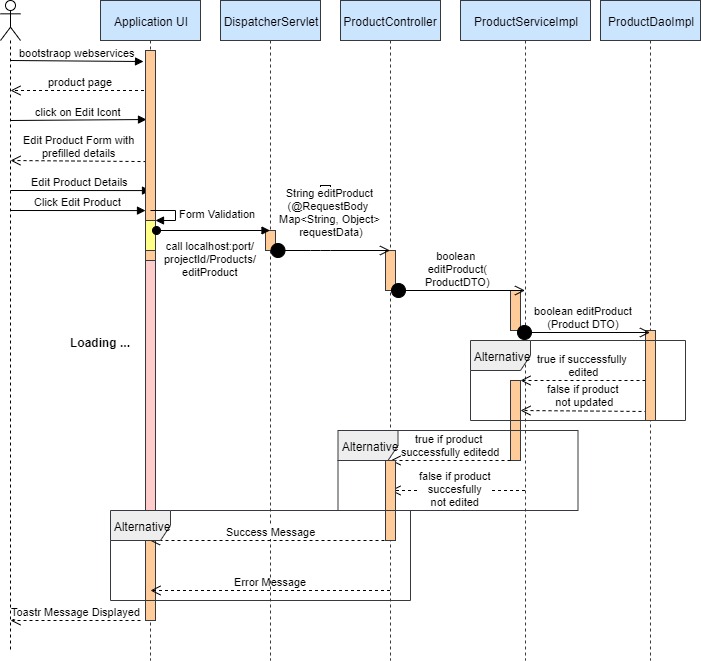
All fields are required.

Sequence Diagram for Adding a Product



* 1. Edit a Product

It has been observed that during manual entry for a product any individual can often make mistakes in entering data. Hence there should always be a scope to update the data which has been once entered. Thus it is our client requirement to introduce an edit option. Hence we have implemented an user friendly edit option for our product management system. Just on clicking an edit icon over a particular product the edit form will pop up. Now we gather from user review that if they have to refill the entire product form then there might be a chance of making more mistakes. It may happen that on correcting a particular data user have entered some wrong data for other fields. Hence our software is designed in such a way that on clicking the edit option, the form will be already prefilled with data. User has to only make changes where it requires. Moreover we have been restricted to make any further changes in the product ID field otherwise it will be treated as a completely new product. On changing the quantity filed it will be treated as the no of new items we want to add for that particular product.

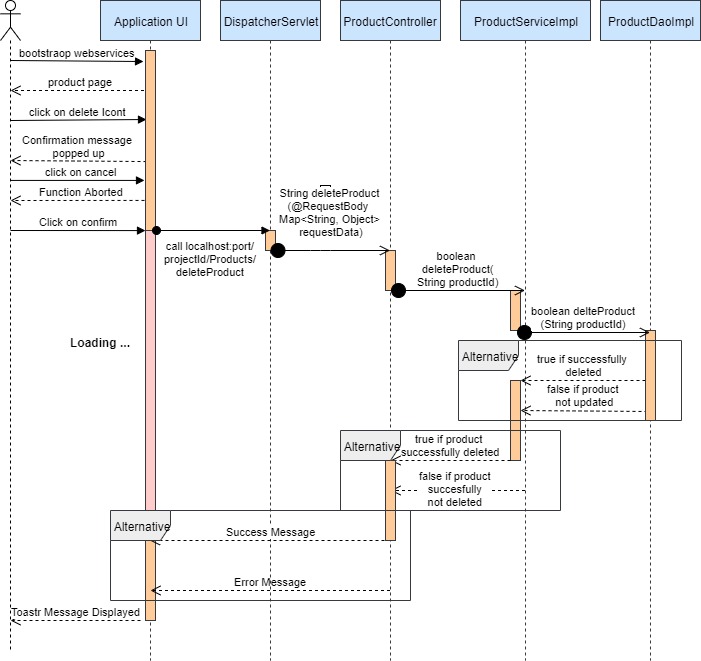


Sequence Diagram for Editing a Product

* 1. Delete a Product

User/ Product Master needs to click a button to delete a product. A confirmation warning will pop up. On clicking on the confirm message the product will be deleted. It has been found through several researches that there can be nothing more important than data. Hence no one wants to lose any data which has been one generated. Thus we have implemented soft delete operation for deleting a product. It will only change the product present status as false and will not show up in product page. However it will be always recorded in the database.

Sequence Diagram for Deleting a Product



* 1. Add a Product Master

We have extended the functionality of registering a user to implement this module. It is very similar to registering a regular user. Only difference lies in role mapping. We have defined a user category field in user database where it has been mapped to a product master category. Our registration function has been well designed show that the password entered is properly encrypted. We have also designed an user friendly registration form with several validations.

**User Id:** Should be alphanumeric and must starts with a letter

**Password :** Should contains at least one small letter, one capital letter, one digit, one special character like @,#,! and must be at least 8 characters long.

**Re-enter Password :** Should match with the previous password to prevent any mistype.

**Phone Number and email :** email should be in proper email format and phone number must be 10 digits long

Sequence Diagram for Adding a Product Master

This has been only a wrapper over registering user ref . 5.2.a

* 1. Search a Product (Implemented in Front End)

The entire model has been designed in such a way that we can enter any keyword related to the product to find the product. Keyword such as brand name, product name or description feature can be used. The entire functionality has been designed in Angular.

f) Filter Product (Implemented in Front End)

We have provided a toggled filter button which open ups to a filter bar. Many customers are brand specific as well as everyone has a specific budget during shopping. Thus, we provide a filter option with price range and brand . We have also provided an additional sort feature which will sort the product by price. Normally product will be sorted by name.

Project Progress Status (Product Management System)

|  |  |  |  |
| --- | --- | --- | --- |
| Sprint# | Task Assigned | Status | Remark |
| Sprint 1 | Creating UML Diagrams, Defining Test Cases and Sequence Diagrams | Epic and Stories written. Use Case Diagram and Sequence Diagram defined | Update the sequence diagram, include the life block in sequence diagram. Develop Knowledge about micro services architecture |
| Sprint 2 | Implement the test cases using Junit, Implement the modules with core java implementation. Use Java Collection API for data storage (non-persistence) | Modules Implemented using Java. Junit test cases are written and successfully tested.  Sequence Diagrams are modified according to the previous sprint feedback | 3-layer architecture is not properly designed. Write more test case scenario. Properly comment the code. Code Convention is not upto industry standard. Presentation layer is not implemented |
| Sprint 3 | Implement 3-tier architecture. Link the business logic with Database using JDBC connection | Database is designed as per ER  Diagram.  DAO, Service and Presentation layer is properly implanted. All Validations are done in presentation layer.  Code is properly commented. Inline comments and redundant codes are removed. JDBC connections are done.  Logger is being implemented. | All ok |
| Sprint 4 | Design the front end with Html , CSS, Bootstrap | Front end pages are designed with HTML, CSS and Bootstrap. JSP and servlet are being used to connect the front end with backend Java modules. | Pages are not made responsive for mobile. Proper use of bootstrap is missing. Unity in colour scheme is missing.  Additional advise : To make the software more dynamic. Toaster message should be implemented |
| Sprint 5 | Replace the presentation layer with Angular client App and write the BDD test Cases using cucumber. | Angular app is designed. Pages are made mobile responsive according to the previous sprint feedback.  Colour scheme is done uniform across all pages.  Entire project is made dynamic. Toastr messege have been implemented  Jersey as well as Servlet Technology is being explored to link the front-end with backend | Proper Scrum Model is not followed. Every Individual is being asked to assigned a single module. JDBC connections are not closed properly. Proper Documentation is missing. |
| Sprint 6 & Sprint 7 | Replace JDBC connection with JPA Hibernate. Implement Sprint MVC | JDBC is being replaced with JPA Hibernate API. Connection open and close is being managed by session and transaction management of Hibernate. Singleton design is being done for session and transaction management is done using Spring bean. Dependency injection is implemented using Auto wiring. Started working on updated documentation.  We have properly divided the modules. | Proper Documentation required. Add proper validation for product id and product image. Write Spring test cases.  Additional Advise:  To implement audit trail |
| Sprint 8 | Spring Boot Implementation | Documentation Started | NA |

5.2 Address Management System

Overview

Addresses provide a means of physically locating a building. They are used in identifying buildings as the end points of a [postal system](https://en.wikipedia.org/wiki/Mail). Great Outdoors follows a Business to Business model. Address Management system is very user friendly, where retailer can add new address and also the same retailer can add new addresses to existing retailer account. For unique identity we are creating individual address identity for individual address. Retailer can also update the existing address and also delete the address from its account. Our Software is also made highly secured using authentication guard of angular. Address Management System can only be accessed with proper credentials of a retailer.

Prerequisite

User must be logged in as retailer to perform the add address, update address and delete address functionalities.

Non Functional Requirement:

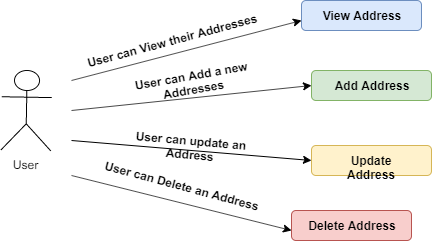
All modules are show designed that the result will be obtained within 10 seconds of request. Proper Loading indicator has been instantiated. Hence once a request been asked no one can further request anything. Authentication has been given a higher priority. All form validations are properly maintained. To make it more user-friendly proper alert messages are shown wherever required. For instance if someone click on any other tab while filling up a form, our software will show a pop up alert – “all the data will be discarded” before navigation to the clicked tab. Pop message will also be shown during any action like delete add or edit product. Any changes in database will only be performed if user gives a confirmation to those pop-up warnings.

Designed and Implemented By:

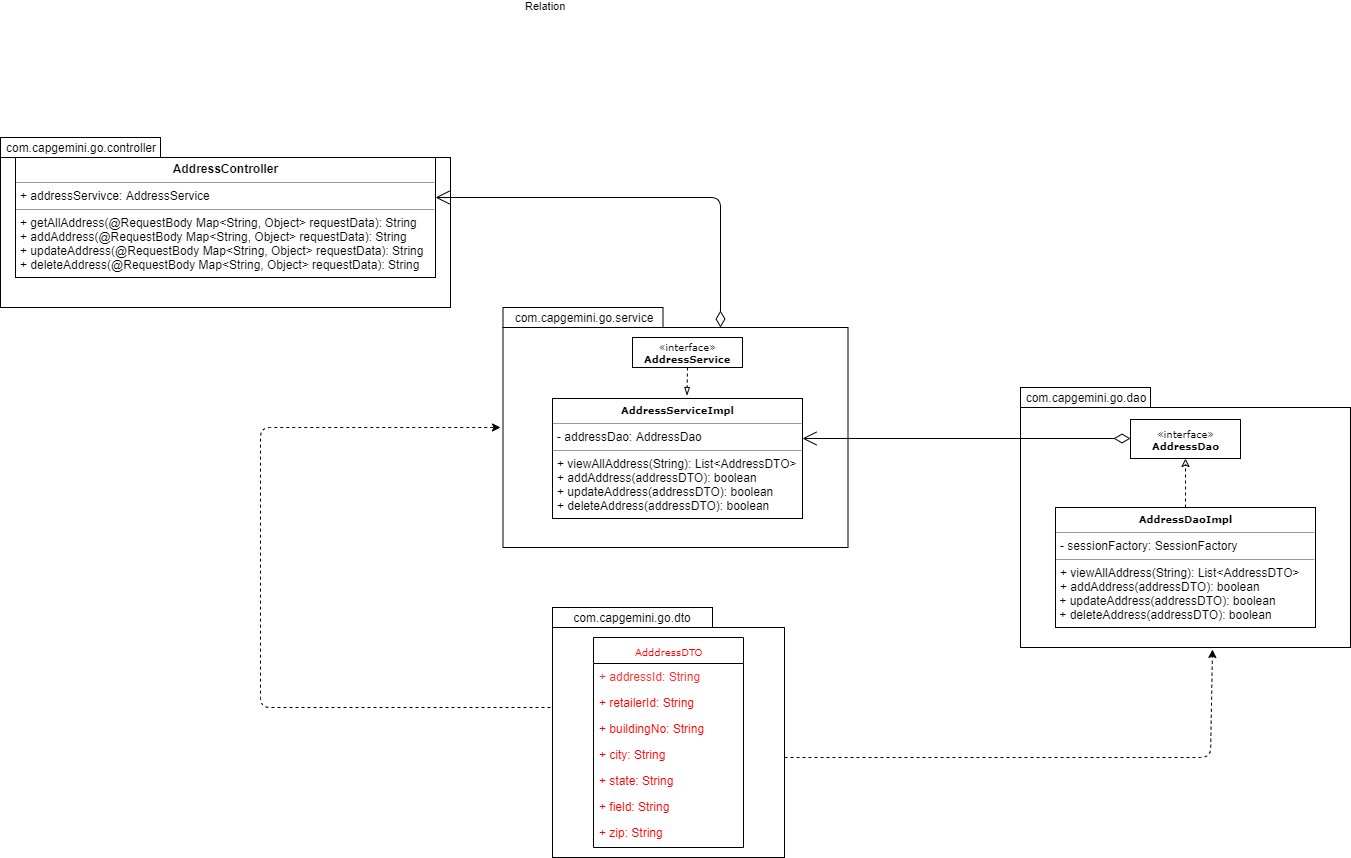
Ayushi Dixit (Id : 46003312)

Designation : Senior Analyst/ A5

Use Case Diagram for Address Management System



Class Diagram for Product Management



a) Add an Address

This module has been designed to add a address in the database. This can be only accessed by Retailer. The form is designed with proper validation. The address ID is being generated automatically by our system so that each address have their own unique identity.

Validation:

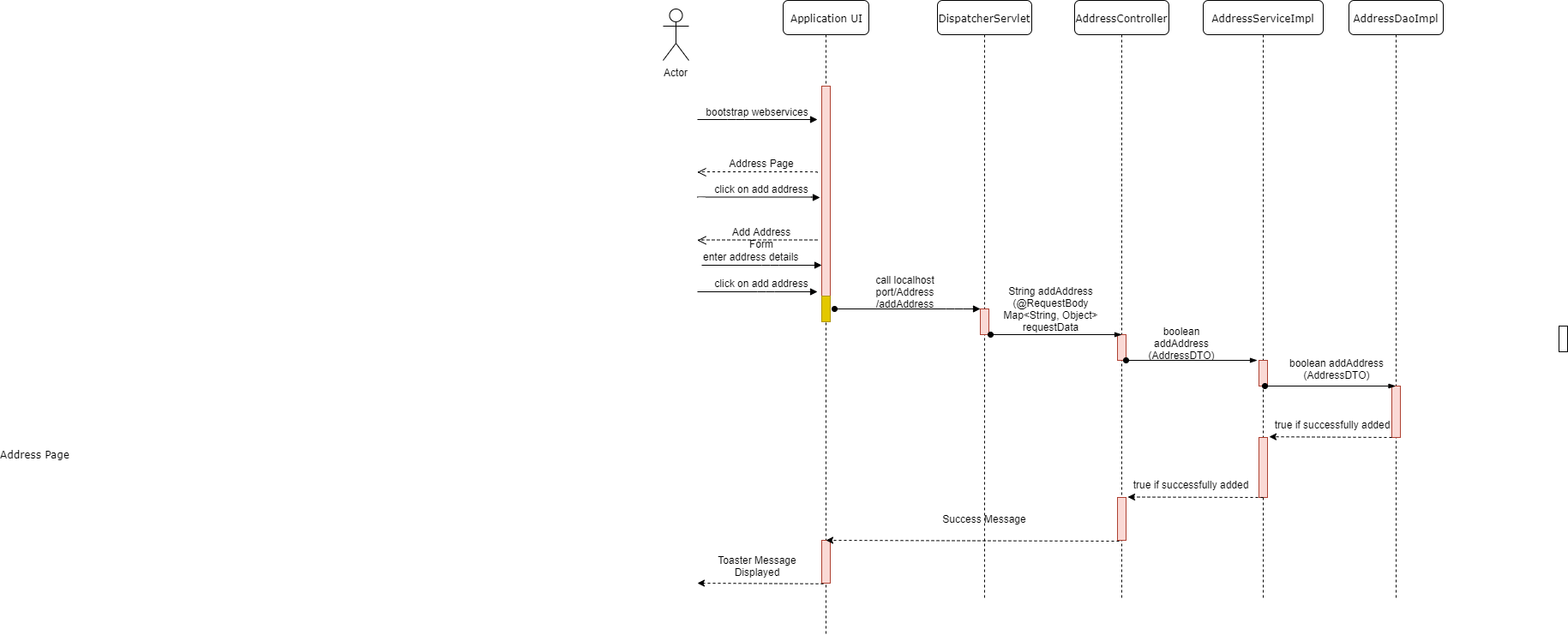
City: city name should be correct

State: state name should be correct

Country: country name should be correct

All fields are required.

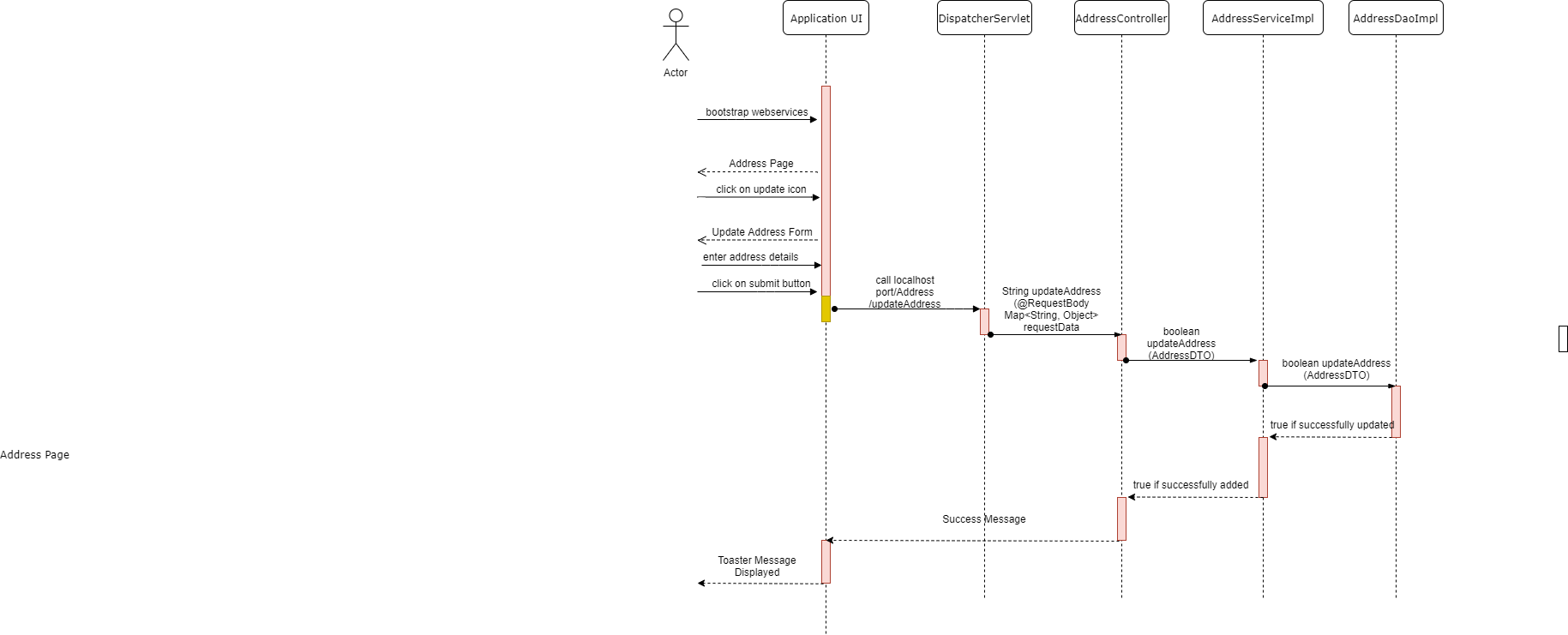
Sequence Diagram for Adding an Address



b) Update an Address

There should always be a scope to update the data which has been once entered. Thus, it is our client requirement to introduce an update option. Hence, we have implemented an user friendly update option for our address management system. Just on clicking an update icon over a particular address the update form will pop up. Hence our software is designed in such a way that on clicking the update option, the form will be already prefilled with data. User has to only make changes where it requires. More over we have been restricted to make any further changes in the address ID field.

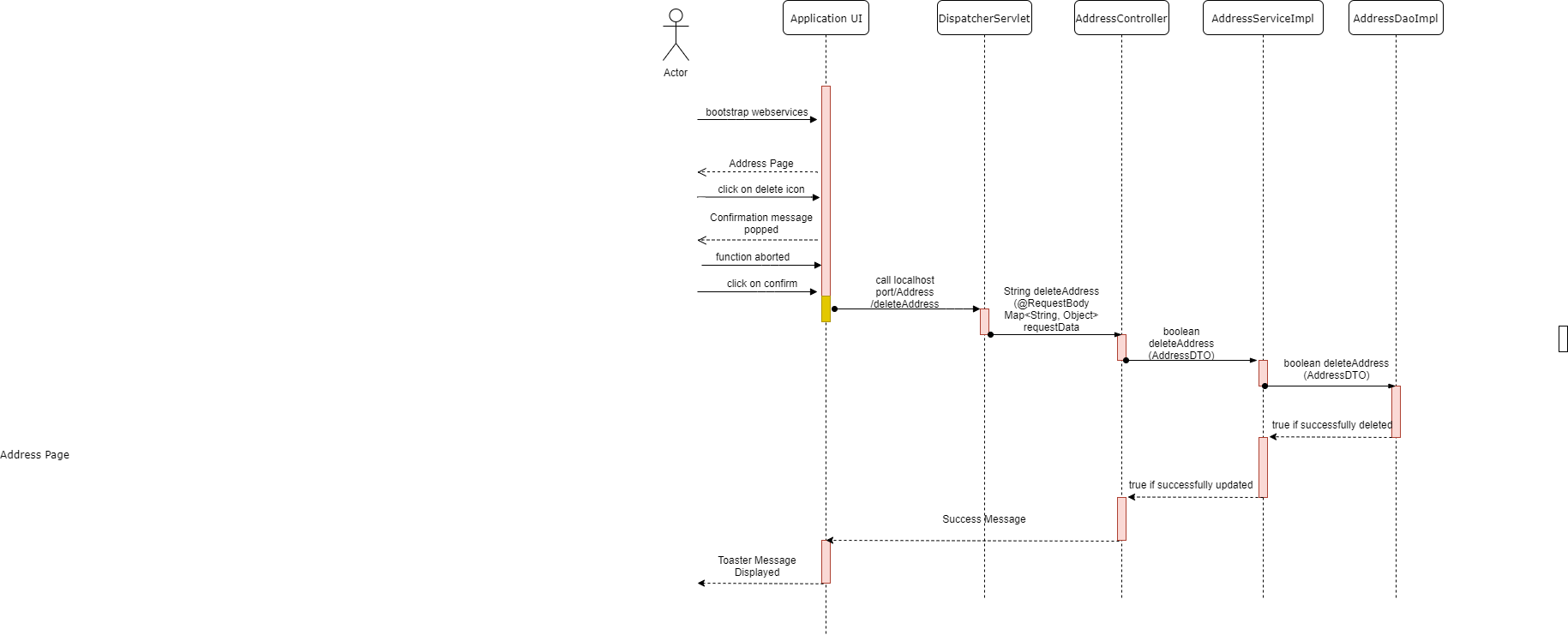
Sequence Diagram for Updating an Address



c) Delete an Address

User needs to click a button to delete an Address. A confirmation warning will pop up. On clicking on the confirm message the address will be deleted.

Sequence Diagram for Deleting an Address



5.3 Admin Module

Overview

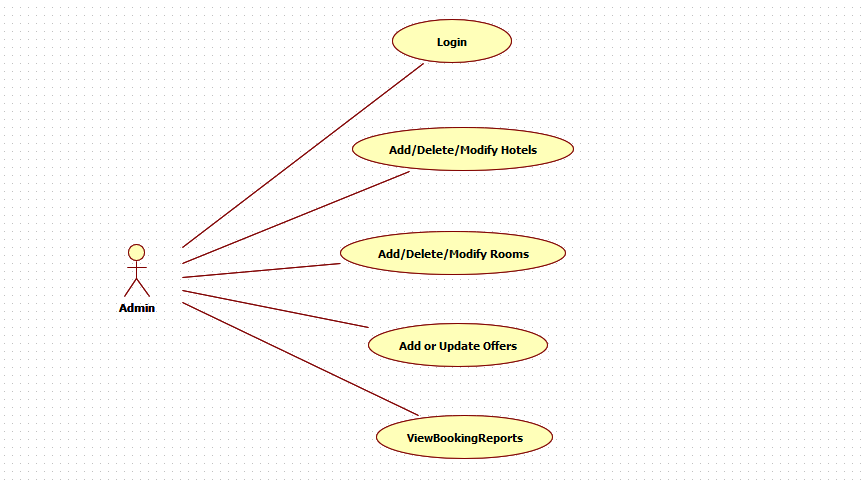
Admin module is the core in any online system as admin has the full control over the system. Logging-in, Adding Hotels, deleting hotels, adding rooms, deleting rooms, updating offers, generating reports like list of hotels, bookings of specific hotels, bookings of specific dates come under this module. Our software uses authorization so that only authorized admin can use the admin functionalities, i.e Role Mapping.

Designed and Implemented By:

Shubham Pal ( Id : 46022266)

Designation : Senior Analyst/ A5

Use Case Diagram for Admin :



* Admin Login

The form uses authorization mechanism, so that only the authorized user can perform admin functionalities.

* Add / Delete/ Modify Hotels

The “Add Hotel” functionality is designed to add new Hotels into the System Database.

The “Delete Hotel” functionality is designed to delete an existing hotel from the System Database.

The “Modify Hotel” functionality is designed to update hotel details such a Hotel Description, Rooms etc.

* Add/ Delete/ Modify Rooms

The “Add Room” functionality is designed to add new rooms in an existing hotel.

The “Delete Room” functionality is designed to delete a room from an existing hotel.

The “Modify Room” functionality is designed to update room details such as price per night etc.

* Update offers

This functionality is designed to add or update offers on hotel and room prices.

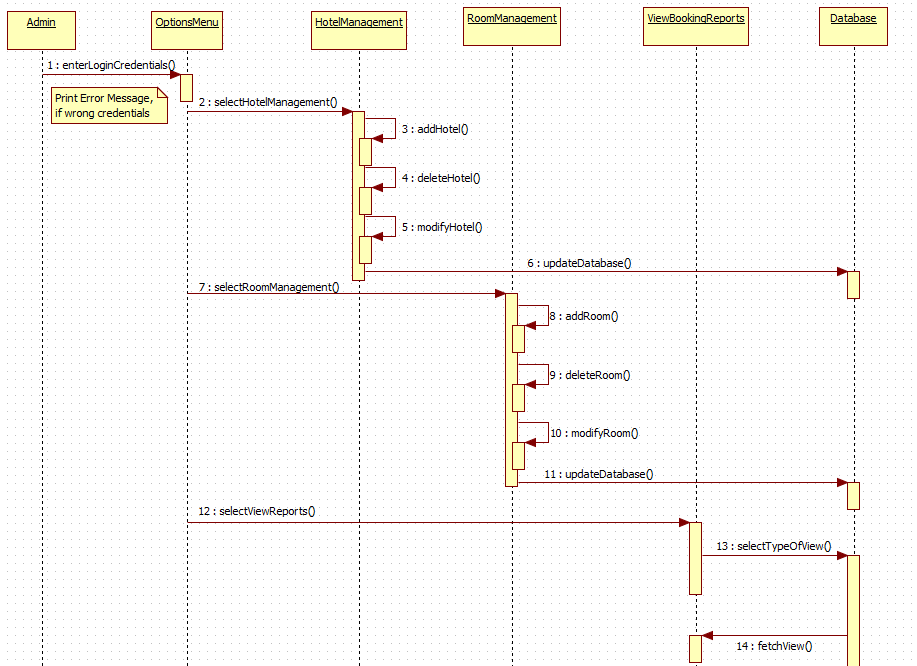
* View Booking Reports

This functionality is designed to generate various type of reports, namely,

List of hotels

Bookings of specific Hotels

Bookings of specific dates

Sequence Diagram Admin :

b) User Logout

This module has been designed to Log-Out an already logged-in user.

***Pre-Requisite***:

User should already be logged-in.

Sequence Diagram for User Logout

5.4 Wishlist Management System

Overview

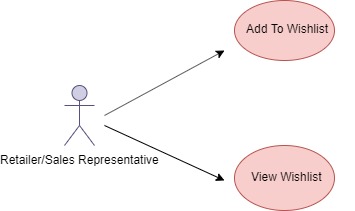
Wishlist management is one of the unique features of this application. This feature can be used by either **retailer** (who is the key customer of our services) and **sales representative** (who makes online entry of all offline orders). This feature enables user to build a list of their favourite products and keep it separately. Every user can have their different list of products. To make it simple we have given option to our user to add their favourites in their wish list by just clicking in heart icon on the product while on home page.

Designed and Implemented By:

Shalu Panwar ( Id: 46001525)

Designation : Associate Consultant

Use Case Diagram for Wishlist Management System



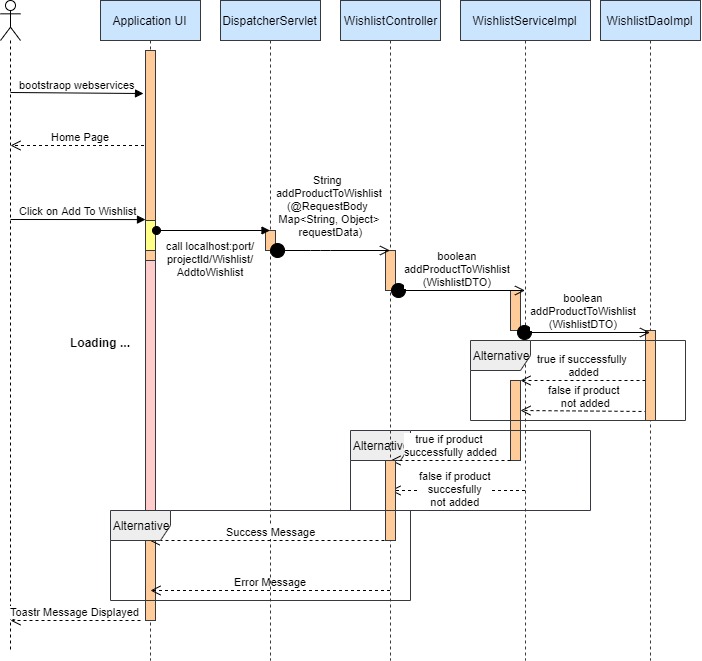
a) Add Product to Wishlist

This function is used by user to add the product in their Wishlist.

***Pre-Requisite***:

User should already be logged-in and on home page.

Sequence Diagram



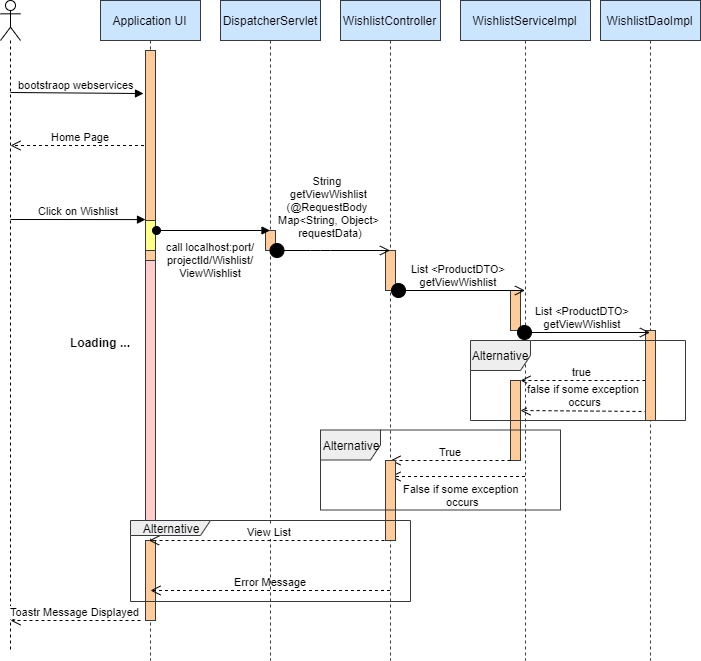
b) View Wishlist

This module has been designed to view Wishlist of user.

***Pre-Requisite***:

User should already be logged-in and clicks on Wishlist on navbar.

Sequence Diagram



5.5) Retailer Add Item to Cart System

Overview

In an e-commerce site, a retailer or a customer chooses different products to place order for. For that, he needs to add the products to his cart. So, cart plays a very important role for placing the order in the e-commerce sites. We have tried to make the process of adding the item to cart an easier one so that the customer doesn’t find any difficulties or complexities in adding the item to the cart. The product will be added to the cart only when the credential of the retailer is correct.

Prerequisite

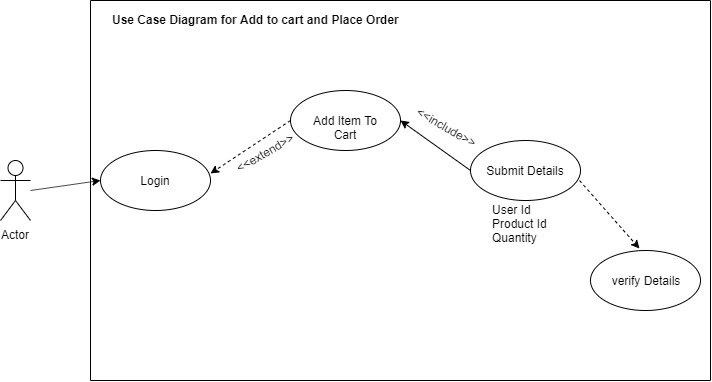
User must be logged in as retailer to perform the add item to cart functionality. He/she has to give his/her correct credential to add the item into the cart.

Designed and Implemented By:

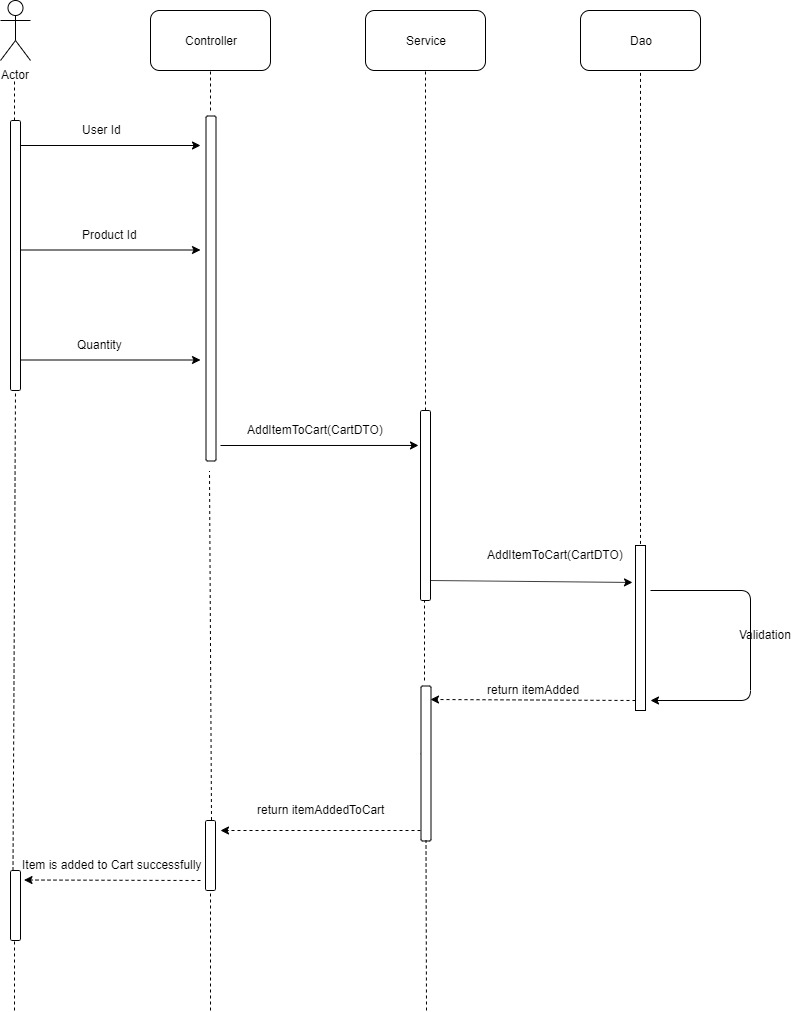
Azhar Hussain ( Id : 46001961)

Designation : Senior Analyst/ A5

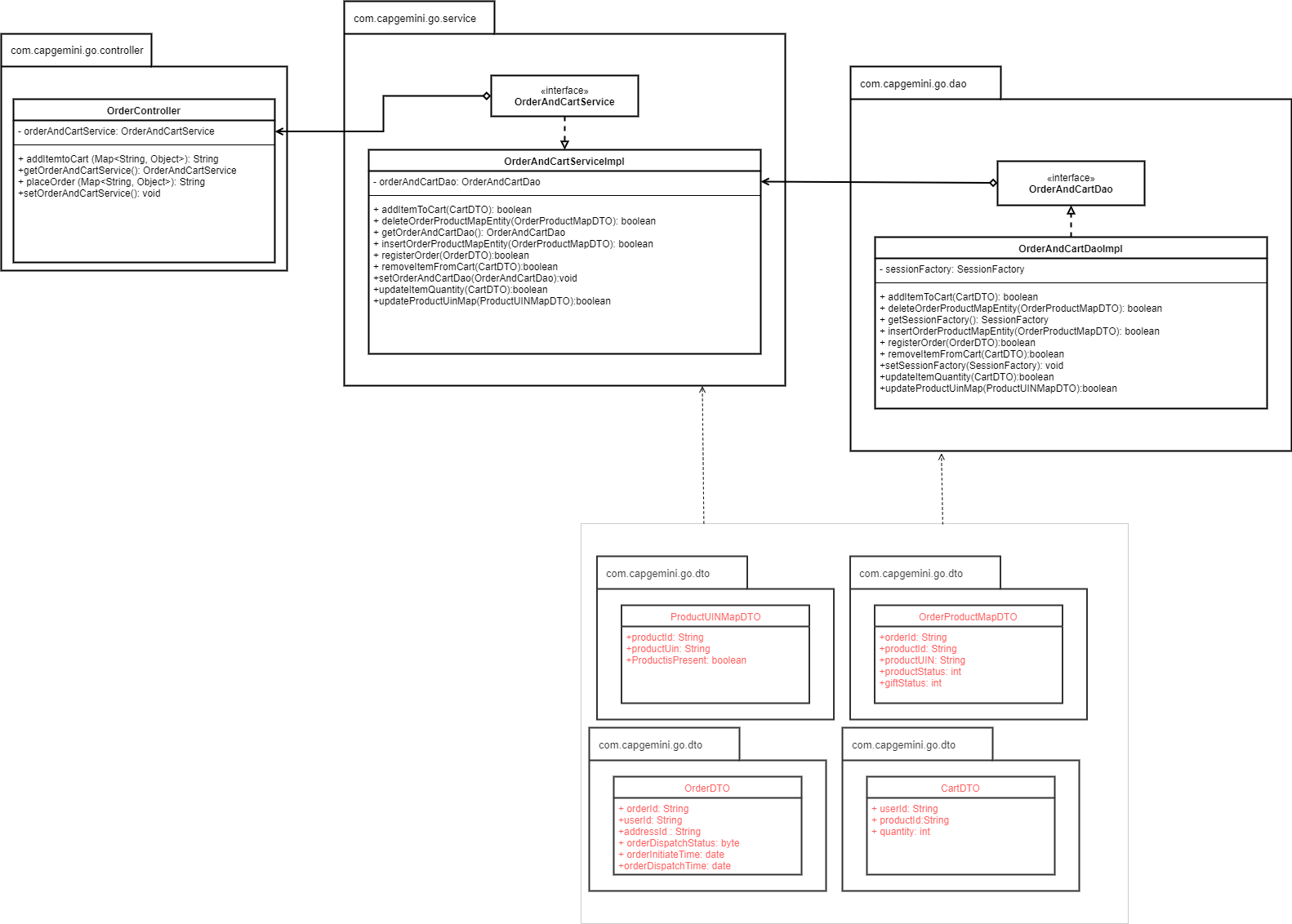
Use Case Diagram for Add Item to Cart System



Sequence Diagram for Adding Item to cart



Class Diagram for Adding Item to cart



Project Progress Status (Adding Item to Cart System)

|  |  |  |  |
| --- | --- | --- | --- |
| **Sprint#** | **Task Assigned** | **Status** | **Remark** |
| **Sprint 1** | Creating UML Diagrams, Defining Test Cases and Sequence Diagrams | Epic and Stories written. Use Case Diagram and Sequence Diagram defined | Update the sequence diagram, include the life block in sequence diagram. |
| **Sprint 2** | Implement the test cases using Junit, Implement the modules with core java implementation. Use Java Collection API for data storage (non-persistence) | Modules Implemented using Java. Junit test cases are written and successfully tested.  Sequence Diagrams are modified according to the previous sprint feedback | 3-layer architecture is not properly designed.  Give a correct message for the retailer. Write more test case scenario. Properly comment the code. Code Convention is not up to industry standard. Presentation layer is not implemented |
| **Sprint 3** | Implement 3-tier architecture. Link the business logic with Database using JDBC connection | Database is designed as per ER  Diagram.  DAO, Service and Presentation layer is properly implanted. All Validations are done in presentation layer.  Code is properly commented. Inline comments and redundant codes are removed. JDBC connections are done.  Logger is being implemented. | Code is correct. Improve the concepts of the syntax written. |
| **Sprint 4** | Design the front end with Html , CSS, Bootstrap | Front end pages are designed with HTML, CSS and Bootstrap. JSP and servlet are being used to connect the front end with backend Java modules. | Pages are not made responsive for mobile. Proper use of bootstrap is missing. Unity in colour scheme is missing.    Additional advice: To make the software more dynamic. Toaster message should be implemented |
| **Sprint 5** | Replace the presentation layer with Angular client App and write the BDD test Cases using cucumber. | Angular app is designed. Pages are made mobile responsive according to the previous sprint feedback.  Colour scheme is done uniform across all pages.  Entire project is made dynamic. Toastr messege have been implemented  Jersey as well as Servlet Technology is being explored to link the front-end with backend | Proper Scrum Model is not followed. Every Individual is being asked to assign a single module. Concepts of different syntax of angular is not there. JDBC connections are not closed properly. Proper Documentation is missing. |
| **Sprint 6 & Sprint 7** | Replace JDBC connection with JPA Hibernate. Implement Sprint MVC | JDBC is being replaced with JPA Hibernate API. Connection open and close is being managed by session and transaction management of Hibernate. Singleton design is being done for session and transaction management is done using Spring bean. Dependency injection is implemented using Auto wiring. Started working on updated documentation.  We have properly divided the modules. | Proper Documentation required. Add proper validation for product id and product image. Write Spring test cases.      Additional Advise:  To implement audit trail |
| **Sprint 8** | Spring Boot Implementation | Documentation Started | NA |

5.7) Shelf Time Report

Designed and Implemented By:

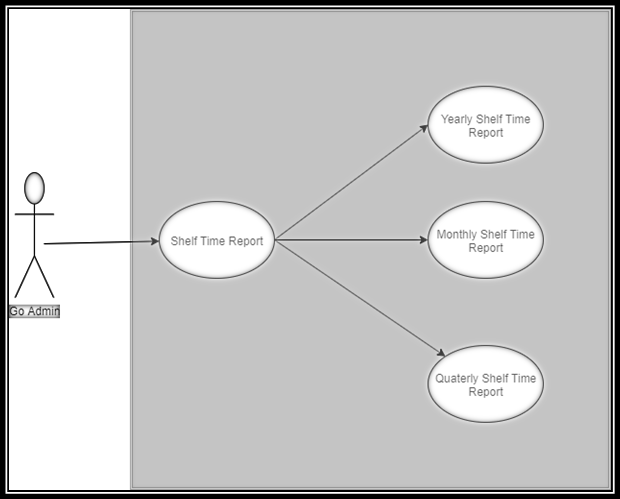
Vikas Singh ( Id : 46001755)

Designation : Senior Analyst/ A5

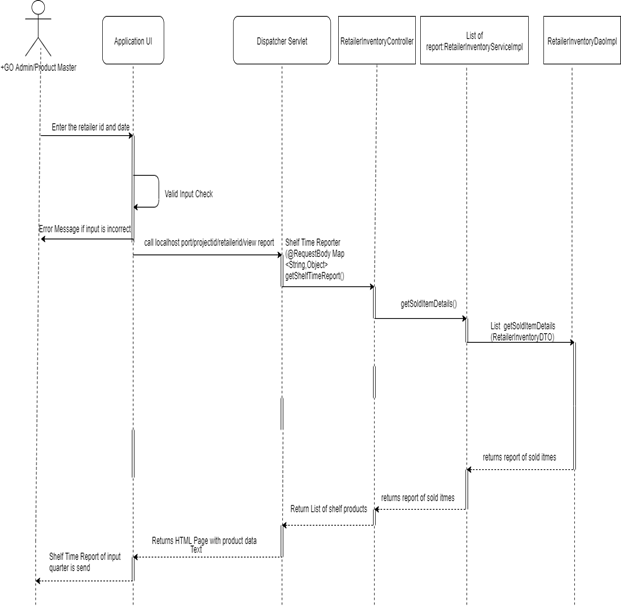
Shelf Time Report:

This module is designed to get the shelf time report of each item. Shelf Time is the total time period of each item for which it has been in the inventory. It is basically calculated by taking the difference between the received timestamp of an item and the sold timestamp of the given item. This functionality is accessed by the Go Admin to get the updated report of all sold items for each retailer. The user has to enter the retailer Id and time for which month, quarter or year he wants to get the report.

Use Case Diagram for Shelf Time Report



Sequence Diagram for Shelf Time Report



5.10 Go Admin Reports Management System

Overview

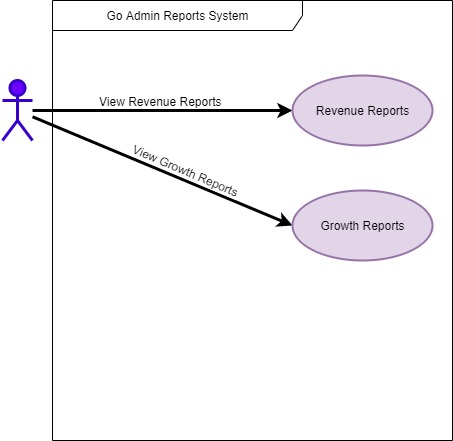
Reports of a company are the main tool for tracking and analysing performance and overall business health. A well-documented report will reveal nuances related to core business functions, while opening opportunities to improve and make market gains. In this module we are presenting revenue reports of different products. We are also providing month to month, quarter to quarter and year to year growth reports of the company to analyse its performance in the market. We have also provided search, sort and pagination for every generated tables.

Designed and Implemented By:

Rintu Mazumder ( Id : 46001757)

Designation : Senior Analyst

Use Case Diagram for Wishlist Management System



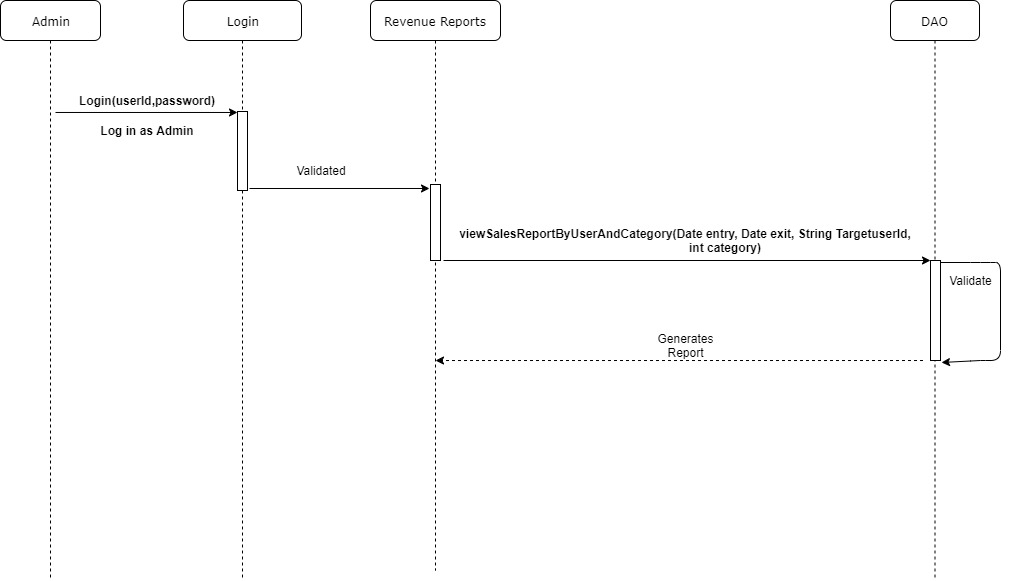
a) View Revenue Reports

This function is used by admin to view Revenue Reports of various users and products within a particular period.

***Pre-Requisite***:

Admin should already be logged-in and reports page.

Sequence Diagram



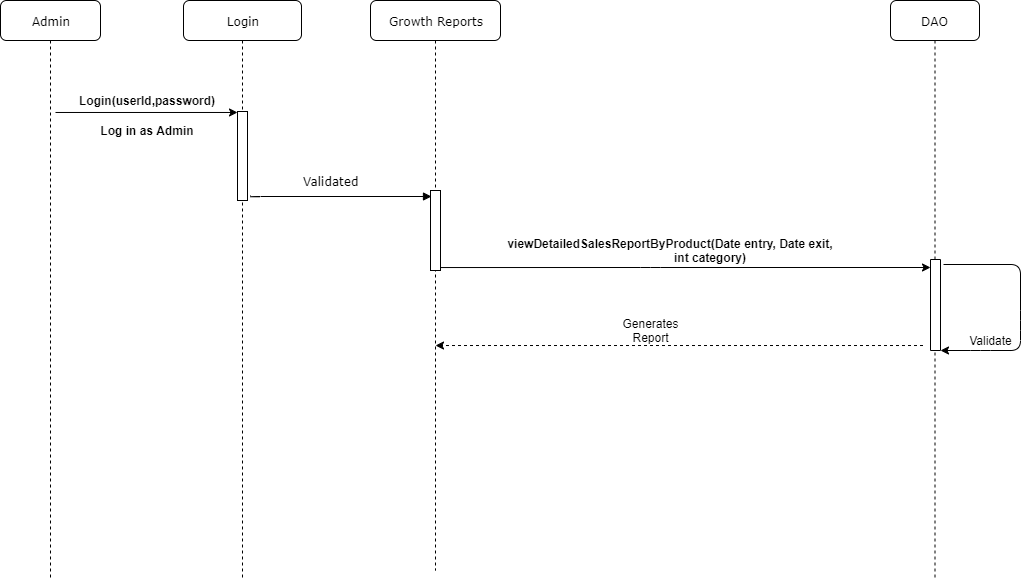
b) View Growth Reports

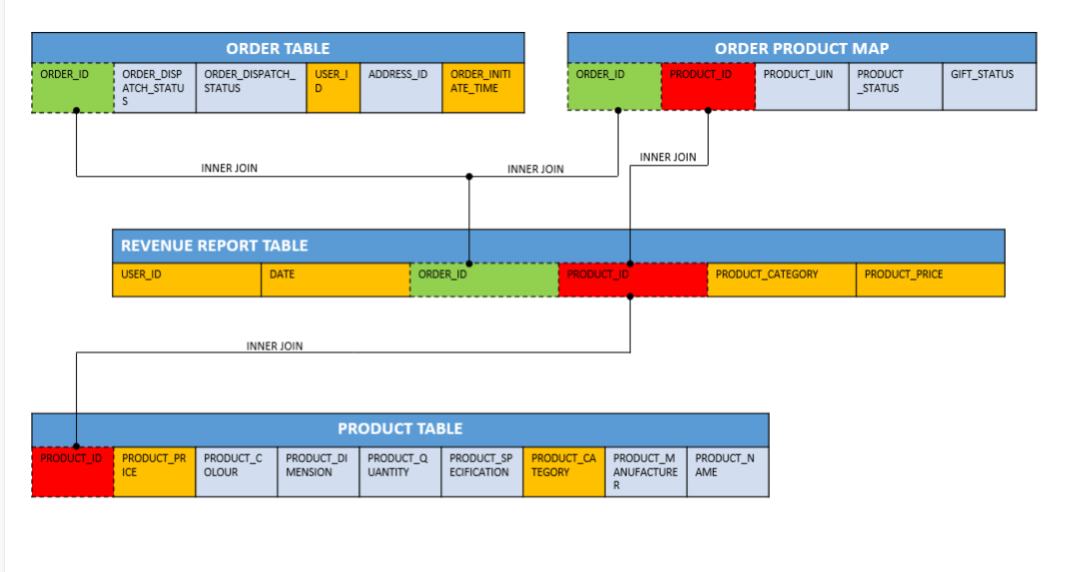
This function is used by admin to view Growth Reports of various products within a particular period.

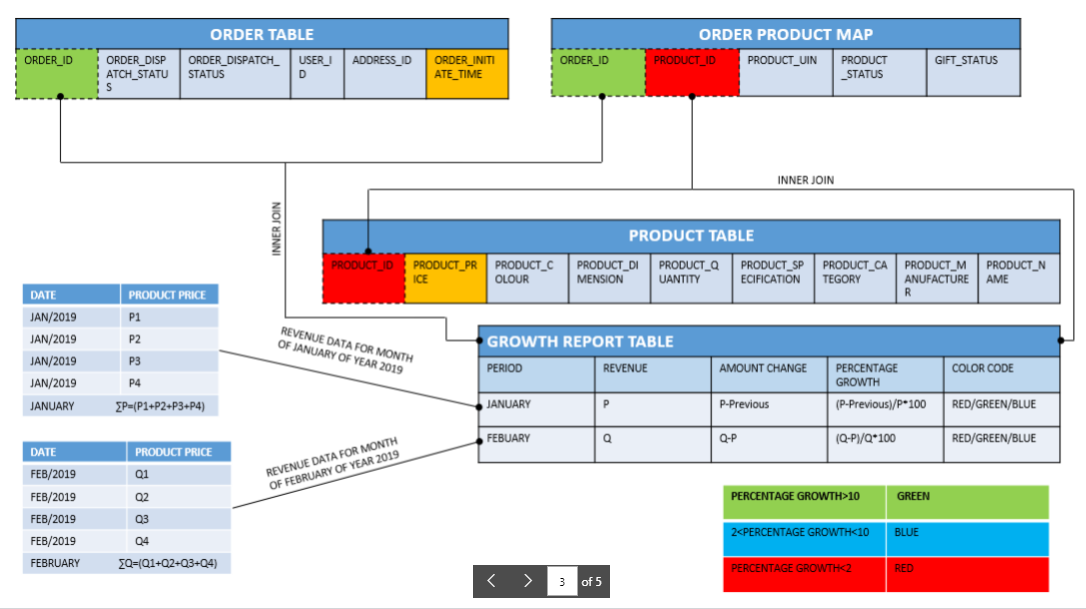
***Pre-Requisite***:

Admin should already be logged-in and reports page.

Sequence Diagram







5.11) Delivery Time Report:

Overview:

Reports of a company are the main tool for tracking and analysing performance and overall business health. Here we are calculating the delivery time report by finding the time difference between the product dispatch timestamp and product receive time stamp. In this module we are presenting delivery time report for tracking the time taken to reach the retailer from GO warehouse. We are also providing item level delivery time, product category level delivery time and outlier category level delivery time to analyse which category taken maximum and minimum time in delivering the item. We have also provided the calculated delivery time for each item.

Prerequisite

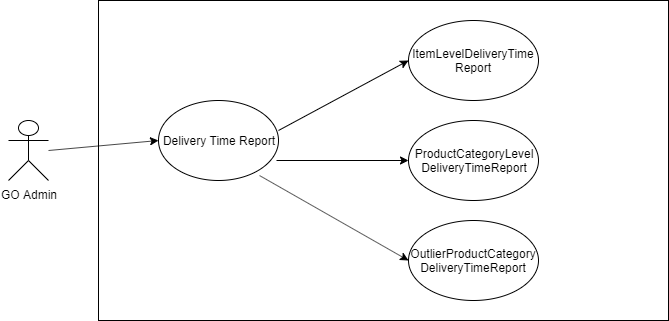
User must be logged in as GO admin to track the delivery time report functionality. He/she has to give his/her correct credential to enter the admin page.

Designed and Implemented By:

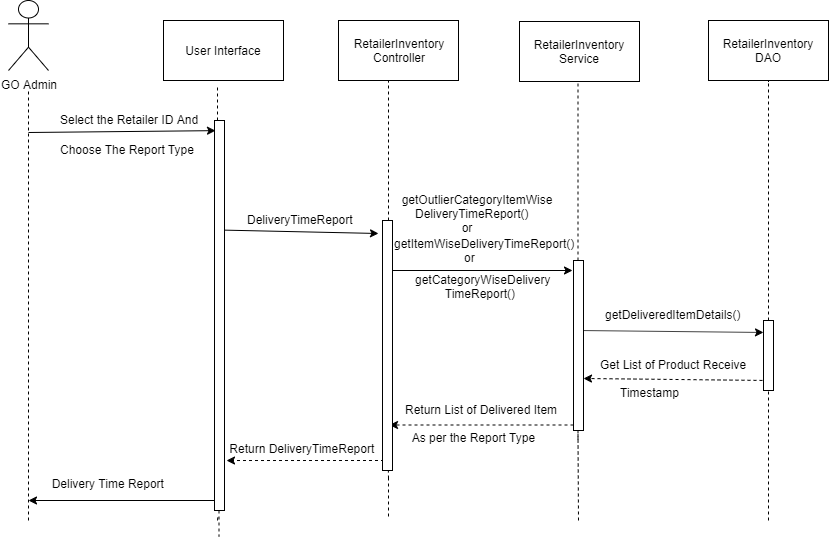
Sujit Kumar (Id: 46001549)

Designation: Senior Analyst

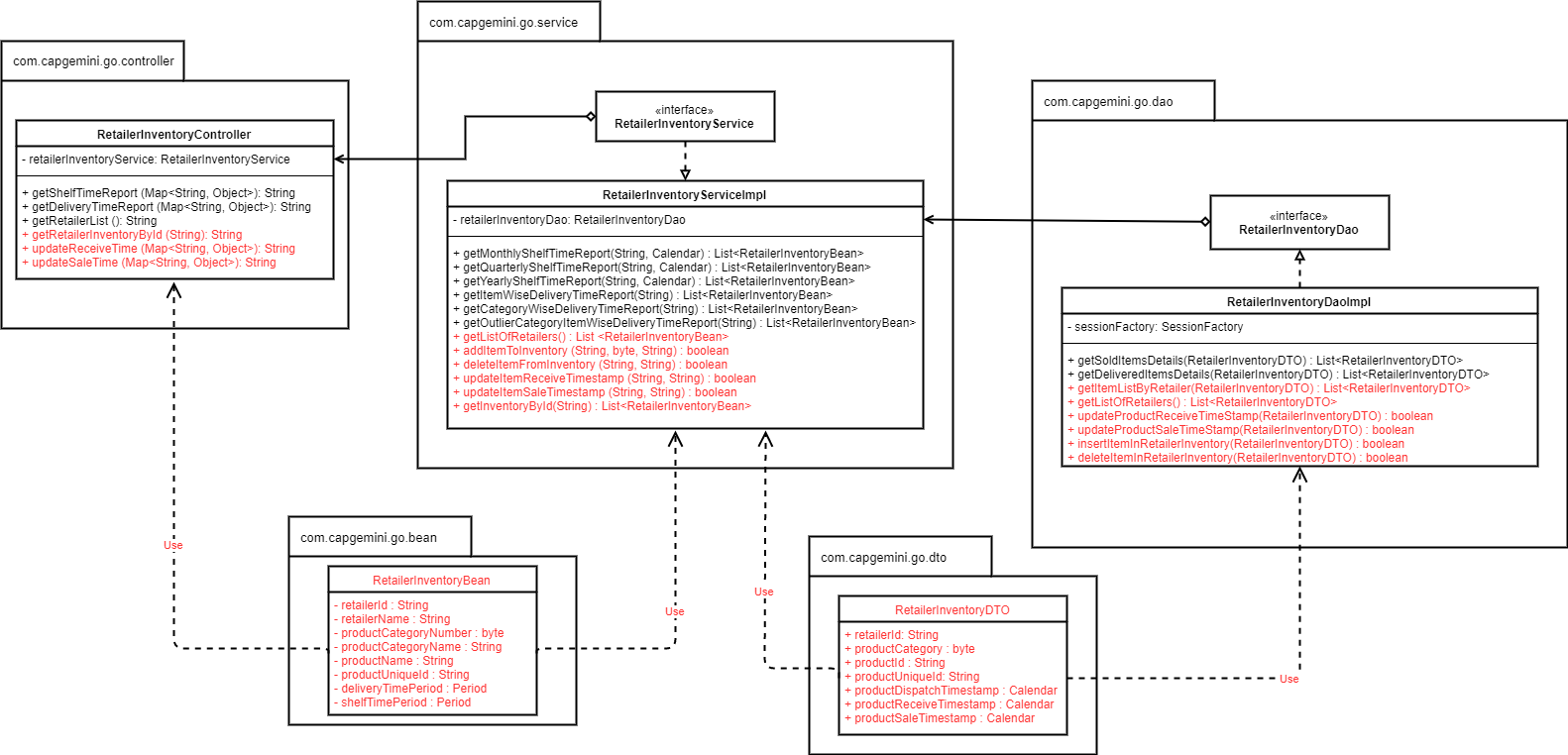
Use Case diagram:



Sequence Diagram:



Class Diagram for retailer inventory:



10. System Requirements

Below is a list of the minimum Hardware and Software requirements to access Great Outdoors website.

**Operating System:**

* Windows 7 and above.
* Mac OSX 10.8, 10.9, 10.10 or 10.11
* Android 3 and onwards.

**Hardware:**

* Processor (CPU) with 2 gigahertz (GHz) frequency or above
* A minimum of 4 GB of RAM
* Monitor Resolution 1024 X 768 or higher (For better view)
* A minimum of 5 GB of available space on the hard disk
* Internet Connection Broadband (high-speed) Internet connection with a speed of 2 Mbps or higher
* Keyboard and a Mouse or some other compatible pointing device

**Browsers:**

* Chrome\* 58+
* Microsoft Edge\* 20+
* Mozilla Firefox 40+
* Internet Explorer 11+ (Windows only)
* Safari 6+ (MacOS only)
* Android\* 3+

*\**Google Chrome version 42+ and Microsoft Edge do not support NPAPI-type plug-ins, including Java plug-ins and many media browser plug-in.

*Users using unsupported browsers may experience issues submitting forms, placing orders, purchasing, updating details and transaction management threads.*

**Browser Configuration:**

Your browser must be configured as follows:

* **Strongly Recommended--**add [www.greatoutdoors.com](http://www.greatoutdoors.com/) to trusted sites.
* JavaScript must be enabled
* CORS must be configured properly
* Cookies must be enabled.
* Pop-up windows must be enabled.

**Software:**

* Java — to view and interact with all available blackboard applications.
* Eclipse — Eclipse workbench was used to run JDK (write, compile and run the code).
* Visual Studio Code — for writing codes for frontend using angular, VS Code was used as a workbench.
* Apache Tomcat — it was used as a server for hosting the website.