

|  |
| --- |
| SellNBye  E- Commerce System |
| Sri Lanka Institute of Information Technology  Student Number Name  IT17029278 - Uswatte U.L.N.P.  IT15056894 - I.S.M.Dissanayake  IT16043060 - Mudunkotuwa M.A.S.  IT16234062 - Rajapakshe D.D.S.  IT17053938 - Weerasinghe K.V.A.N.R |
| 2019 May 19  Group : Y3S1.07(IT).1  Group ID : N3B-G1 |

Year – 3rd year 1st sem

Campus – SLIIT Malabe

Subject – PAF

Contents

[Abstract 3](#_Toc9188026)

[1. Introduction 4](#_Toc9188027)

[2. Objectives 5](#_Toc9188028)

[3. Procedure(Requirement Gathering & Analysis) 7](#_Toc9188029)

4. Class Diagram……………………………………………………………………………………………………..8

[5. ER Diagram 9](#_Toc9188031)

6. UseCase Diagram………………………………………………………………………………………………10

[6. Onion Diagram 11](#_Toc9188032)

7. Activity Diagrams….…………………………………………………………………………………………..12

8. Technologies……………………………………………………………………………………………………..17

9. References……………..…………………………………………………………………………………………..19

# 

# Abstract

This project is an E-Commerce system and is an online system, also known as an Electronic commerce. Basically, refers to selling and buying products and services over the Internet. It has revolutionized the concept of modern business and changed the idea of locating on a certain territory and focusing on a particular customer audience. By transforming classic commerce into ecommerce, the Internet made it possible to grow any business more rapidly and get substantial profits with minimal initial investments. As a solution, web Application comes up with an over the network.

Now our project is an E- Commerce system therefore we decide to develop web application using Java sprint Boot Framework. In traditional web application built in three parts: a user interface, a database, and a server-side application. This server-side application is called a monolith, which is further divided into 3 layers — presentation, business layer, and data layer. The entire code is maintained in the same codebase. In order for the code to work, it is deployed as a single unit. Any small change requires the entire application to be built and deployed. As a solution, we used Microservices application Architecture billed an over project. Microservices truly independent therefore they not shared database, independently developed, tested, deployed, monitored, and scaled. These can be even developed in different programming languages. We used a several frameworks, that are MySQL Hibernate as an Object-Oriented Database, Layer Architecture, Maven Dependency management tool, Eureka API, RESTful API, Bootstrap for interphase, Ajax jQuery and JavaScript. . The source code will be created in the software IntelliJ IDEA (The Java IDE.)

The proposal provides a useful insight about the project carried out as a whole. It provides details on the requirement specification, analysis, application modeling, design, testing and implementation future scope and limitation of the application development. The importance of this system to organization is Efficient data access, Restricting unauthorized access and Concurrent access.

The software helps them maintain day to day transaction in computer and it could lead business to its success.

# 1. Introduction

SellNBye is an e-commerce system, which provides a platform for the registered sellers to advertise their products and registered buyers to order and purchase them. The engineering team has decided to implement the system based on micro-services architecture, using RESTful communication.

Things to do ---- :

* Identify possible web services to be implemented as micro-services (at least 5 – service per team member)
* Design the architecture for the SellNBye system, indicating the identified web services
* Design the API for each web service
* Design the class diagram per web service (use styles and patterns like MVC)
* Identify the DB requirements, then design the DB
* Select the tools (methods, technologies, frameworks, libraries, plugins, IDEs, etc…) to develop
* each web service and justify the selection
* Develop the system and test (using test clients) - Web services may intercommunicate to
* complete the overall features
* Each service should perform CRUD operations on DB via relevant features

Clickable link to the public VCS repo

<https://github.com/DilrukshiRajapakshe/PAF-Ecommerce/tree/master>

# 2. Objectives

Modern business and changed the idea of locating on a certain territory and focusing on a particular customer audience. By transforming classic commerce into ecommerce, the Internet made it possible to grow any business more rapidly and get substantial profits with minimal initial investments. As a solution, E-Commerce web Application comes up with an over the network.

The Management levels are listed below:

1. User Management :

When a person wants to register for the system, if that person is an admin, he/she need to register as an admin. If that person is a customer, he/she can register as a customer and if later they want to sell item in the system they can register as a seller too.

1. Stock Advertisement Management :

In this system only Registered User can sell products. They can “Add Items”, “Delete Items”, “View added Items” and “Update added Items” of the system. Any user can search for items of the stock. And nonregistered user can see those added items through the home page.

1. Wish List Management :

After searching the products, customer wants to add interest items to a list before purchase.

Wish list is a collection of desired products saved by customer to their user account, signifying interest without immediate intent to purchase. So here ,give this feature to them with adding items to the Wishlist , check-out and delete items from the list and also share s it’s link with others. It helps to the customer; choose the best item he wants and buy it.

1. Payment Management :

When a customer done inserting item into cart, Customer decide to purchase an item from the company. That moment calculates a Full payment Bill and after validating Credit Card Info that information is Insert into table.

1. Delivery Management :

When the customer pressed the confirm payment button the interface redirected to the address editing form. In that interface appearing the customer's address that he has entered when has registering. Or the customer can update the address or can change the address that he wants to deliver the goods.

Microservice Architecture Diagram

A close up of a map

Description automatically generated

# 3. Requirement Gathering & Analysis

We are planning to implement a web application system for the Electronic company called “SellNBye” The system should be implemented for the whole company, which is a wide scope to handle.

First, we identified an Our Microservices. We want to know about what is a Microservice, Sprint boot API, what is an ORDB(JPA, Hibernate), RESTful API, Eureka server. Then we are gathering an information using internet. After we can identify an out service. That are:

* User Management
* Stock Advertisement Management
* Wish List Management
* Payment Management
* Delivery Management

Designing of the System

We designing our UI using Html Boostrafe WE study most fames E-commerce web side and we develop our system Front end . First, we Are mapping Our ER diagram , Use case diagram, Activity diagram

Implementation

Once a Finish Designing our UI, we start our project back end implementation Using Java language.

Testing

We use a postman to test our backend it correctly is working or no

Project Plan (Gantt Chart)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Tasks | 2019 | | | | | | |
| Week1 | Week2 | Week3 | Week4 | Week5 | Week6 | Week7 |
| Requirements Gathering |  |  |  |  |  |  |  |
| Design document |  |  |  |  |  |  |  |
| Coding |  |  |  |  |  |  |  |
| Final Product & Report |  |  |  |  |  |  |  |

4. Class Diagram

A close up of a map

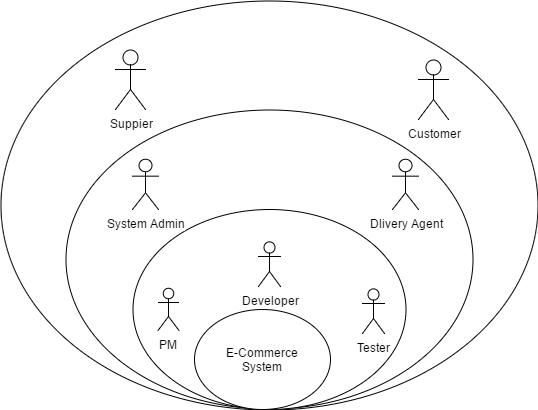
Description automatically generated

5. ER Diagram

A close up of a logo

Description automatically generated

6. Onion diagram



# UseCase Diagram

# 

# 8. Activity Diagrams

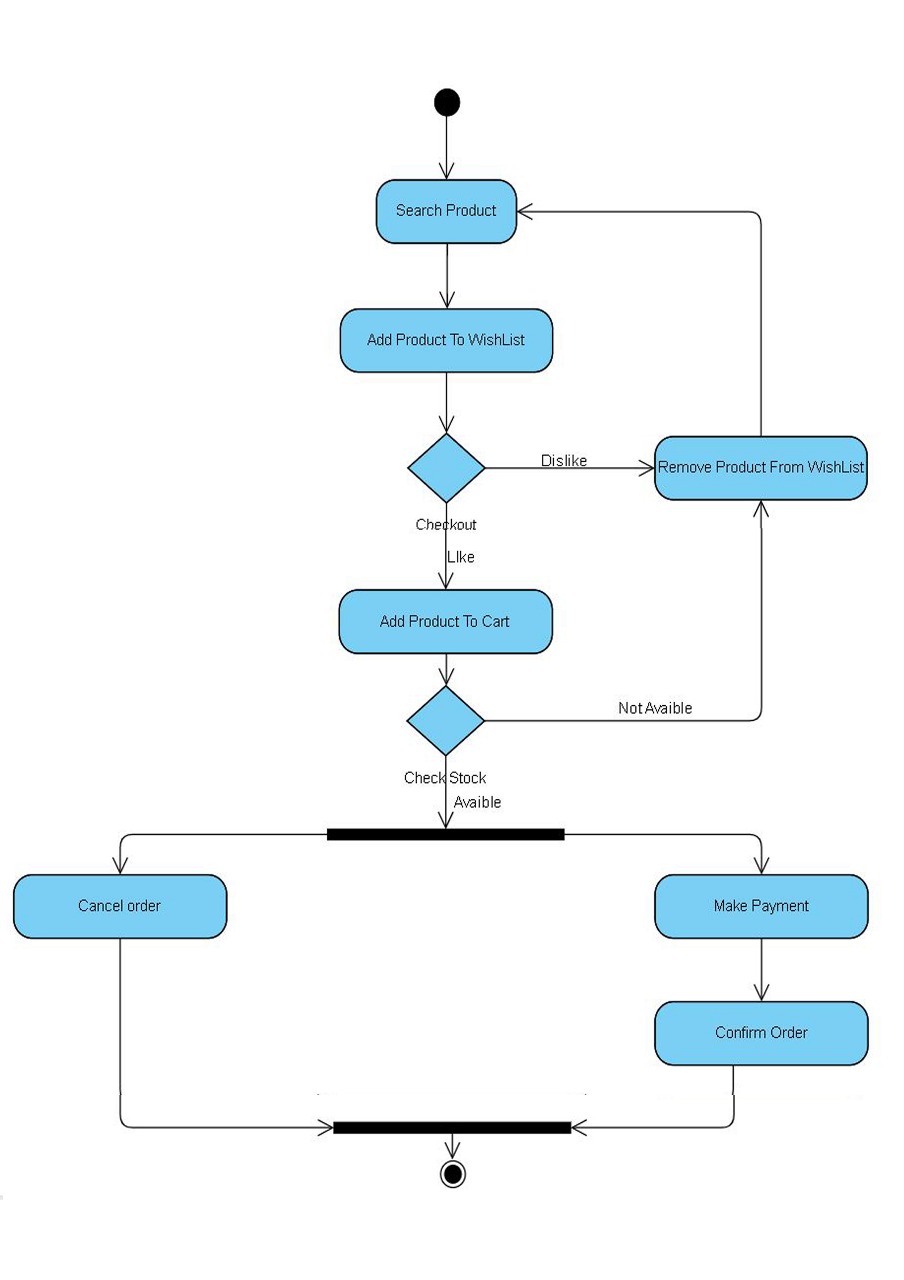
# 1. Profile Management:

# 

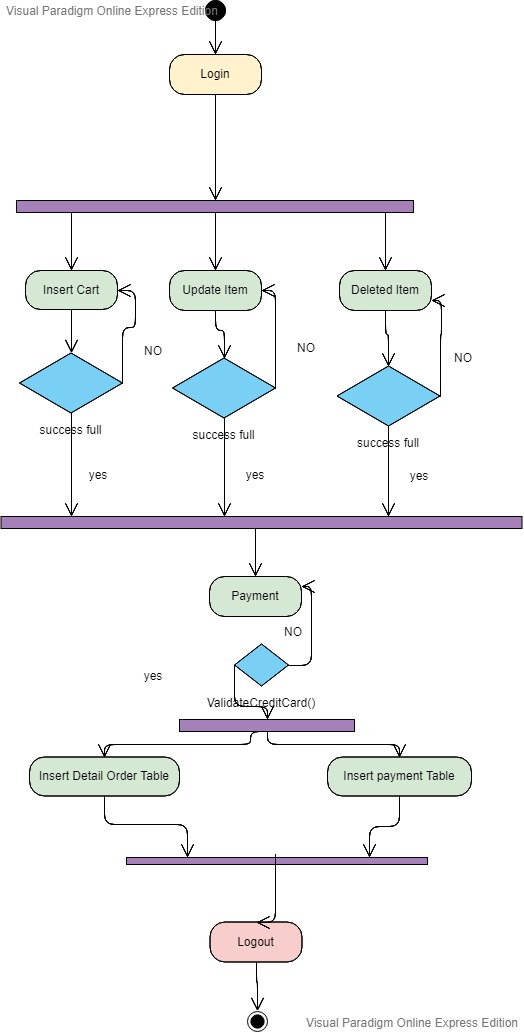
# 2. Stock Advertisement Management:

# 

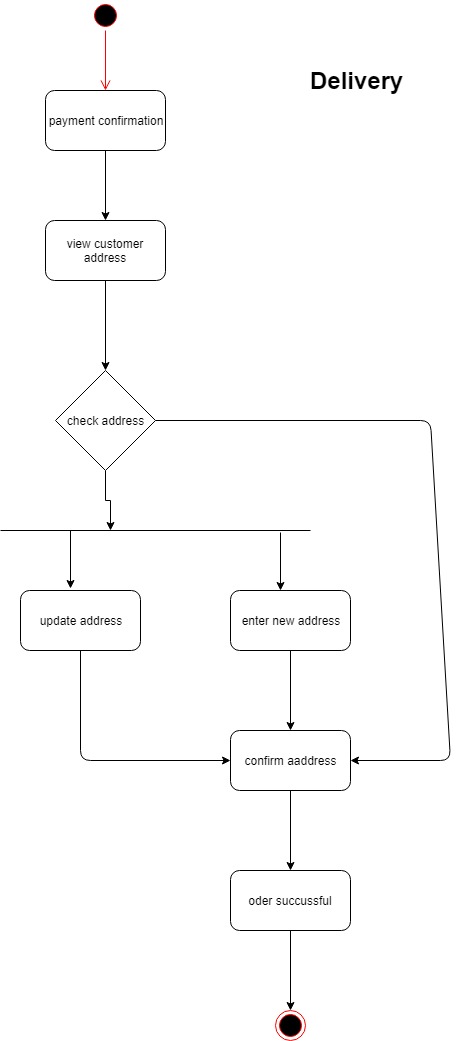
3. Wish List Management:



4. Payment Management:



5. Delivery Management:



# 8.Technologies

HTML

HTML stands for Hyper Text Markup Language. HTML describes the structure of Web pages using markup. HTML elements are the building blocks of HTML pages. HTML elements are represented by tags. HTML tags label pieces of content such as "heading", "paragraph", "table", and so on. Browsers do not display the HTML tags, but use them to render the content of the page.

Bootstrap

Bootstrap is the most popular CSS Framework for developing responsive and mobile-first websites. Bootstrap 4 is the newest version of Bootstrap.

AJAX

AJAX stands for Asynchronous JavaScript and XML. AJAX is a new technique for creating better, faster, and more interactive web applications with the help of XML, HTML, CSS, and Java Script. Ajax uses XHTML for content, CSS for presentation, along with Document Object Model and JavaScript for dynamic content display.

Hibernate

Hibernate is a high-performance Object/Relational persistence and query service, which is licensed under the open source GNU Lesser General Public License (LGPL) and is free to download. Hibernate not only takes care of the mapping from Java classes to database tables (and from Java data types to SQL data types), but also provides data query and retrieval facilities. This tutorial will teach you how to use Hibernate to develop your database based web applications in simple and easy steps.

SASS

SASS (Syntactically Awesome Stylesheet) is a CSS pre-processor, which helps to reduce repetition with CSS and saves time. It is more stable and powerful CSS extension language that describes the style of document structurally. This tutorial covers the basics of SASS.

Microservice

microservice architecture - is an architectural style that structures an application as a collection of services that are

* Highly maintainable and testable
* Loosely coupled
* Independently deployable
* Organized around business capabilities.

The microservice architecture enables the continuous delivery/deployment of large, complex applications. It also enables an organization to evolve its technology stack.

Eureka API

It is RESTful API. It is used to register the java sprint boot Maven Microservice. Eureka has a server and client component. A service registry with Eureka Server and a Discovery client with Eureka Client. Eureka is a REST (Representational State Transfer) based service that is primarily used in the AWS cloud for locating services for the purpose of load balancing and failover of middle-tier servers.

Sprint Boot

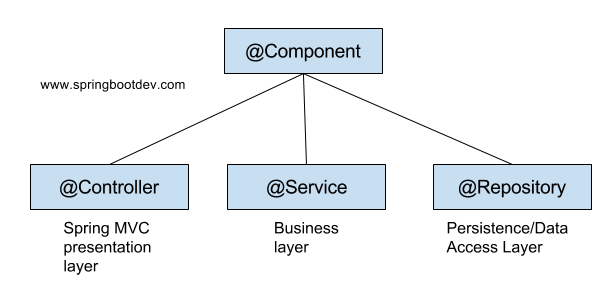
Spring makes it easy to create Java enterprise applications. It provides everything you need to embrace the Java language in an enterprise environment, with support for Groovy and Kotlin as alternative languages on the JVM, and with the flexibility to create many kinds of architectures depending on an application’s needs. As of Spring Framework 5.1, Spring requires JDK 8+ (Java SE 8+) and provides out-of-the-box support for JDK 11 LTS. Spring supports a wide range of application scenarios

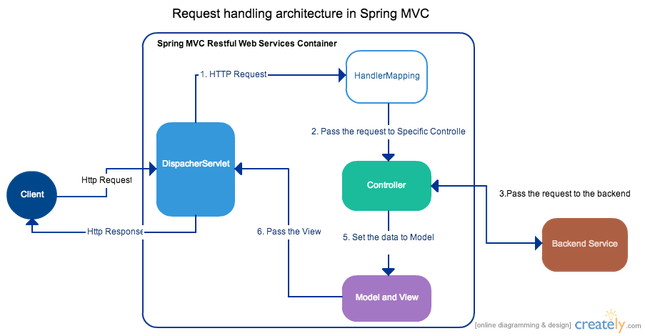
RESTful API

An API is the messenger that takes a request, tells a system what you want to do and then returns the response back to you. [More detail refer my blog Article .](https://dilrukshirajapakshepersonal.blogspot.com/2019/03/rest-full-web-application.html)

MVC Layer architecture

It is Enterprise application Architecture. It has a several layer. In a project requirement can change a layers. Ower project have a 7th layers.

* Entity layer-DB Entity
* DTO Layer-Interspace variable
* Repository– crud function (db headline)
* Service – crud function
* Controller – event headline (URL headline)
* Main – main thread, hibernate session, transaction
* Advice – try cash headline
* 



9. References

https://www.w3schools.com/htmL/html\_intro.asp

https://www.w3schools.com/whatis/whatis\_html.asp

https://www.w3schools.com/bootstrap4/default.asp

https://www.w3schools.com/js/js\_ajax\_intro.asp

https://www.w3schools.com/whatis/whatis\_ajax.asp

https://www.tutorialspoint.com/ajax/what\_is\_ajax.htm

https://www.tutorialspoint.com/hibernate/index.htm

https://www.tutorialspoint.com/hibernate/hibernate\_overview.htm

<https://www.tutorialspoint.com/hibernate/hibernate_architecture.htm>

<https://www.tutorialspoint.com/sass/sass_syntax.htm>

<https://dzone.com/articles/why-microservices>

https://microservices.io/

https://docs.spring.io/spring/docs/5.1.7.RELEASE/spring-framework-reference/overview.html#overview