

|  |
| --- |
| SellNBye  E- Commerce System |
| Sri Lanka Institute of Information Technology  Student Number Name  IT16234062 - Rajapakshe D.D.S. |
| 2019 May 19  Group : Y3S1.07(IT).1  Group ID : N3B-G1 |

Year – 3rd year 1st sem

Campus – SLIIT Malabe

Subject – PAF

# 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.

# Table of Contents

Contents

[Abstract 3](#_Toc9188026)

[1. Introduction 4](#_Toc9188027)

[2. Objectives 5](#_Toc9188028)

[3. Procedure(Requirement Gathering & Analysis) 6](#_Toc9188029)

4. Functional and Non- Functional…………………………………………………………………………..7

5. Product Backlog…………………………………………………………………………………………………8

6. Use case scenario……………………………………………………………………………………………….9

7. UseCase Diagram………………………………………………………………………………………………10

8. Activity Diagram……………………………………………………………………………………………….11

9. Class Diagram……………………………………………………………………………………………………12

10. Activity Diagrams….………………………………………………………………………………………..13

11. Technologies…………………………………………………………………………………………………..14

12. References…………..…………………………………………………………………………………………..15

# 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

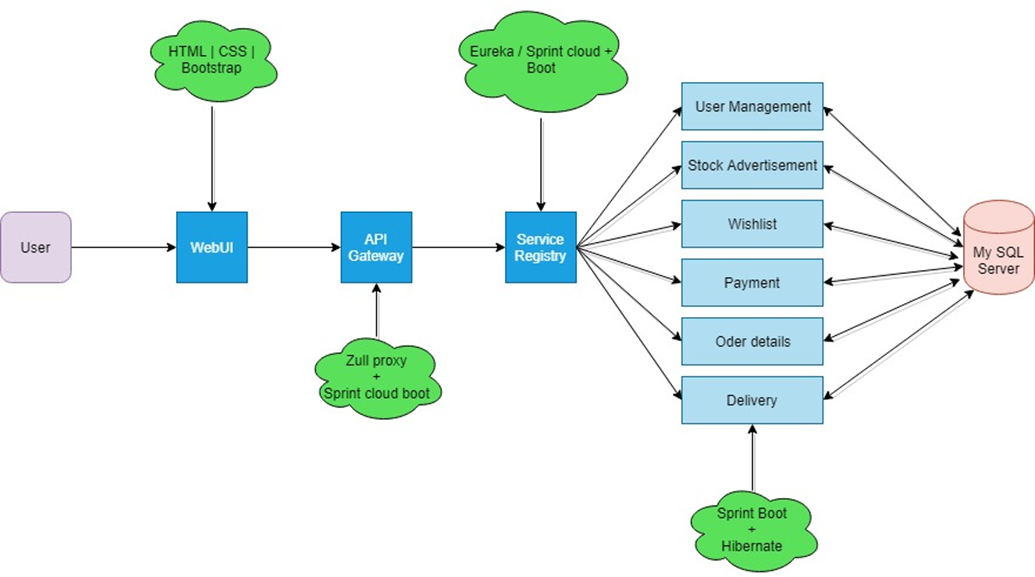
# 2. Objectives

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.

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 informations are Insert into table.

Microservice Architecture Diagram



# 3. Procedure

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, I am identified an my Microservices. I gather an information visiting most fames E sit. I am identified what are the crud function and how there are designing their UI.I want to know about what is a Microservice, Sprint boot API, what is an ORDB(JPA, Hibernate), RESTful API, Eureka server. Then I start to gathe an information using internet. After I can identify an my service . My service is payment. It has two table payment and orderDetail.

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 Mapping Our ER diagram , Use case diagram, Activity diagram

Implementation

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

Testing

I use a postman to test my backend it correctly is working or no.

# 3. Functional and Non- Functional

Non- Functional

* Security – it is important to specify the level of security that should be met such as the OWASP top 10
* Privacy – meeting basic requirements for GDPR
* Scalability and performance – ensuring that the system can scale to meet expected traffic and order volume at normal and peak times
* Speed of key user journeys – defining how long each step with key user journeys will take
* Speed of web services – defining how long web services will take to provide a response
* Accessibility – ensuring that the platform meets the basic accessibility standards throughout
* Documentation – ensuring that the platform is sufficiently documented
* Quality – even the best e-commerce platforms can be used badly so you should insist that code is developed to a good quality standard
* Extensibility – ensuring that the platform is extended in such a way to make future development feasible
* Data integrity and retention – defining how long data should be stored and how the integrity of data is maintained
* Testing – defining how unit testing will be built into the solution
* Compatibility – ensuring that the platform can be easily integrated with 3rd party systems
* Search – defining how quickly the system will return search results
* Availability – defining the agreed uptime of the platform under normal conditions
* Infrastructure – defining the infrastructure performance thresholds (CPU and memory usage)

Functional

* Responsive (Must work on Mobile, Tablets and Desktops)
* Search function for the website
* Easy to update site content, add products, remove products, edit products, write blogs
* Catered prices for wholesale and end customer
* Email marketing sign up for newsletters
* Website built with best SEO practice
* Website Forms
  + Contact us form
  + Online Enquiry Form
  + Associated workflows for notifications to staff

4. Product Backlog

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ID | AS a | I want to be able to | So that | Priority | Status | Estimated hours |
| 1 | Register user | Add items into cart | I can purchase it | 1 | Done | 3hr |
| 2 | Register user | remove Item in cart | So I can update my cart | 3 | Done | 3hr |
| 3 | Register user | Update Item in cart | I can update my cart | 4 | Done | 4hr |
| 4 | Register user | Search Item in cart | I can see item in my cart | 5 | Done | 4hr |
| 5 | Register user | View Item in cart | I can see what the items are | 6 | Done | 2hr |
| 6 | Register user | Pay the item | I can Purchase my item in cart | 2 | Done | 4hr |
| 7 | Register user | View my Purchase history | See my Purchase history | 7 | Done | 2hr |

5. Use case scenario

|  |  |
| --- | --- |
| Number | 3 |
| Use case name | Payment |
| Summary | When the Register user wants to pay Items online. |
| Priority | 8 |
| Preconditions | User has selected the what item he wants to buy |
| Post conditions | * User selected the item. * User paid items. |
| Primary Actors | Register user |
| Trigger | User has insert items into cart by using online. |

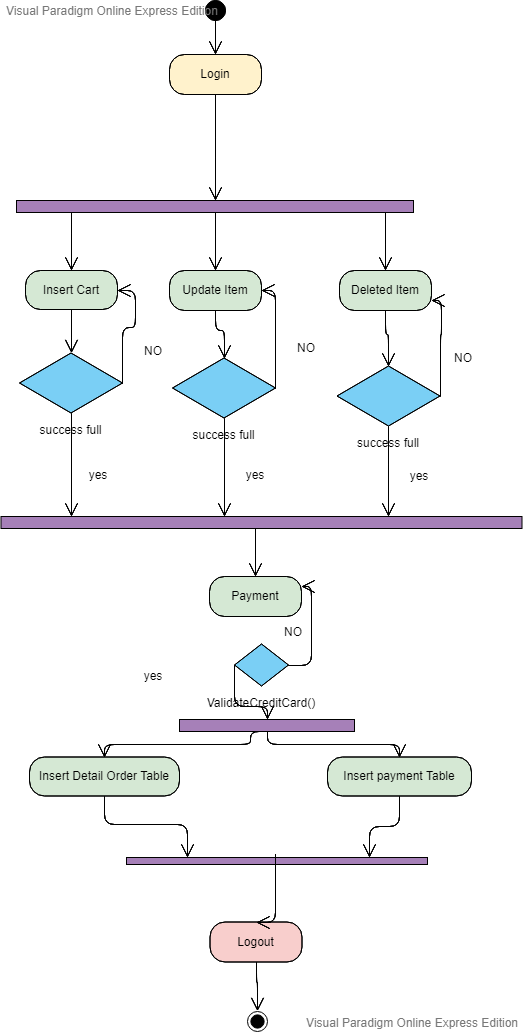
|  |  |  |
| --- | --- | --- |
| Main scenario | Step | Action |
|  | 1 | User search items. |
|  | 2 | User search items details. |
|  | 3 | User chooses the items. |
|  | 4 | User will be insert to item into cart |
|  | 5 | User view a Cart |
|  | 5 | System displays cost. |
|  | 6 | User fills find cost details. |
|  | 7 | User pays their payments with their payment options. |
|  | 8 | System checks payment details. |
|  | 9 | System confirms payment details. |
|  | 11 | System display welcome message. |
|  | 12 | User will exit the system. |
| Extensions | 9a | If incorrect payment details, system will show error message. |
|  | 9b | System shows option that re-enter payment details. |
|  | 9c | System exit option. |
| Open issues | 1 | Should the system ask if the user want to select Items in agin? |

7. Use case Diagram

A close up of a map

Description automatically generated

8. Activity Diagram



9. Class Diagram

A screenshot of a social media post

Description automatically generated

10. 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

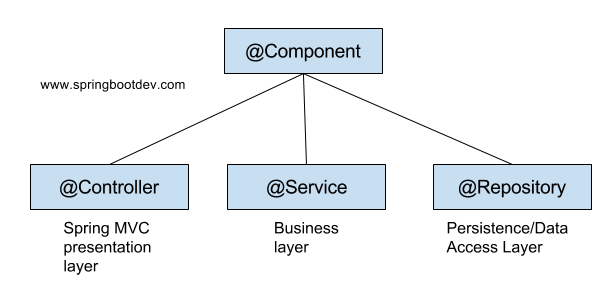
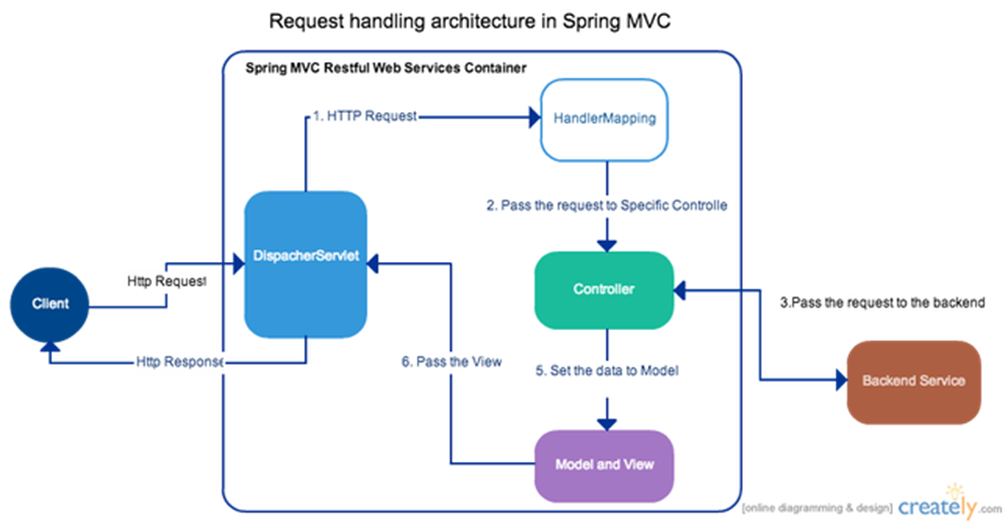
RESTful API

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://www.blogger.com/blogger.g?rinli=1&pli=1&blogID=6862655794791762273" \l "editor/target=post;postID=4817492079252675374;onPublishedMenu=allposts;onClosedMenu=allposts;postNum=2;src=postname)

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 csh headline



# MySQL – Database creation

# Postman – back end testing

# IntelliJ ide – java developer ide (multiple language can run)

# Tomcat – server

# 13. Testing

# A screenshot of a computer screen Description automatically generatedA screenshot of a computer Description automatically generatedA screenshot of a computer Description automatically generatedA screenshot of a social media post Description automatically generated

# 12. 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