

|  |
| --- |
| Project Document |
| Shipping Helper System for Customers and Deliverer |

|  |  |
| --- | --- |
| **Group 2** | |
| **Group Members** | Truong Vu Binh – Student1267009  Nguyen Van Dong - Student1220013 |
| **Supervisor** | Mr. Kien Luong Trung |
| **Project code** |  |

- Hanoi, 12/2022 -

**Table of Contents**

[Acknowledgement 3](#_Toc121340184)

[Definition and Acronyms 3](#_Toc121340185)

[1 Project Introduction 4](#_Toc121340186)

[1.1 Project Information 4](#_Toc121340187)

[1.2 Project Team 4](#_Toc121340188)

[1.3 Product Background 4](#_Toc121340189)

[1.4 The existing system 4](#_Toc121340190)

[1.5 The proposed system 4](#_Toc121340191)

[1.6 Project Scope & Limitations 5](#_Toc121340192)

[2 Project Management Plan 6](#_Toc121340193)

[2.1 The purpose and target of a project 6](#_Toc121340194)

[2.2 Management Approach 6](#_Toc121340195)

[2.3 Responsibility Assignments 7](#_Toc121340196)

[2.4 Project Communications 7](#_Toc121340197)

[2.5 Project Work Plan 7](#_Toc121340198)

[2.6 Configuration Management 7](#_Toc121340199)

[3 Software Requirement Specification 9](#_Toc121340200)

[3.1 User Requirements 9](#_Toc121340201)

[3.2 Functional Requirements 12](#_Toc121340202)

[3.3 Non-Functional Requirements 12](#_Toc121340203)

[4 Software Design Description 14](#_Toc121340204)

[4.1 System Design 14](#_Toc121340205)

[4.2 Detailed Design 15](#_Toc121340206)

[5 Software Testing Documentation 16](#_Toc121340207)

[5.1 Scope of Testing 16](#_Toc121340208)

[5.2 Test Plan 16](#_Toc121340209)

[5.3 Test Cases 16](#_Toc121340210)

[6 Release Package & User Guides 16](#_Toc121340211)

[6.1 1. Deliverable Package 16](#_Toc121340212)

[6.2 2. Installation Guides 16](#_Toc121340213)

[6.3 3. User Manual 17](#_Toc121340214)

# Acknowledgement

During the time of studying, we have well provided and imparted for all the necessary and highly valuable of professional knowledge by our teacher in Aptech, we have our faithful, careful style and creative mind in studying in the future.

Project is a good opportunity to apply appropriate for our knowledge and subjects. While performing project, we have actual and valuable experience. After one term, we were getting all one’s thoughts together and labor at a task of project. During executing this project, we have received a lot of helping from our guider, teachers.

Especially, we would thank to our teacher- Mr: Kien Luong Trung - for his enthusiastic, thoughtful, and careful guidance.

# Definition and Acronyms

|  |  |
| --- | --- |
| **Acronym** | **Definition** |
| BA | Business Analysis |
| BR | Business Rule |
| ERD | Entity Relationship Diagram |
| GUI | Graphical User Interface |
| PM | Project Manager |
| SDD | Software Design Description |
| SPMP | Software Project Management Plan |
| SRS | Software Requirement Specification |
| UC | Use Case |

# Project Introduction

## Project Information

This is the project report documentation, as well as the overall documentation for the 4th semester project, written and provided by the 2nd group of class C2005L.

This project is to create a customer-to-customer friendly platform for shipments across Vietnam.

## Project Team

Our team consist of 2 members: Me, Nguyen Van Dong, who oversees this documentation, as well as Truong Vu Binh, who is the main developer of the project.

Mr. Luong Trung Kien is our supervisor for the project, who provide us support and guidance throughout the course of the project.

## Product Background

In Vietnam, the rise of e-commerce has begun in late 2010s with companies like Lazada and Shopee are the first to join the fields. As the pandemic strike, the demand for such services significantly increases. However, the companies’ services are catered around business-to-customer model instead of customer-to-customer. The same can also be said for international business such as Amazon, although their business are a combination of business-to-customer and customer-to-customer service. Furthermore, most if not all business tends to focus on relationships between customers, while the middle-man roles tend to be ignored.

With that in mind, we came together to create an application with an idea in mind, a customer-to-customer friendly solution that emphasis not only the customers, but also the middleman in the business relationship.

## The existing system

As previously stated, most companies’ businesses models focus on relationships between customers, while leaving the middleman behind too little to no importance.

Take Shopee for instance, the app has been in Vietnam for a long time. As such, they provide a huge catalogue of services and products for their customers, while giving an option for distributors/participated owners to distribute/promote/sell and take orders from customers. Once the order is completed, the customer would then retrieve their order within 2 to 3 days. But that begs a question, is there a business entity, or other middleman that does all the delivery? For Shoppe in particular, they have a system of deliverers, where they perform and finish the tasks in days. This, in fact, leaves the middleman/shipping merchants role redundant, with little to no importance emphasis on them.

## The proposed system

And that leads us to this system. Putting all the businesses and logistical aside, this system is a webapp catered for shipping merchants and customers, which aims to provide a much easier and more approachable access to deliverance rather than traditional postal delivery systems.

The system, instead of having the shipping merchants on the passive, where they are forced to deliver on an assigned path, shipping merchants can now take the jobs of their choice, at their own pace, and allow them to tackle the delivery as they deemed the most optimized for them.

## Project Scope & Limitations

Due to time constraint and limitations, the final product is far from what originally planned for the completed product. There would be a big need of revamping the current UI, as well as certain functionality to ensure they work as intended. Furthermore, some functions, while currently in functional state, need polishing and fine-tuned to improve user experience, as well as giving a better presentation

In this version, we have completed basic functionality, which have passed internal testing for basic usage. We hope the application is usable with a good UI and user experience, as well as provide good experience for not only the customer, but also the delivery merchants

# Project Management Plan

## The purpose and target of a project

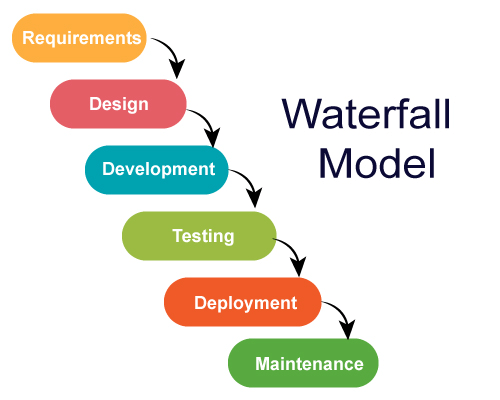
### Project Objectives

This project aims to provide a functional web application, with fundamental features and a functional system. Along with that, we hoped to learn new knowledge about .NET web developing, and how a web application could be developed

But most importantly, we wish to broaden our understanding of C# programming language, Web development, as well as database management and additional knowledge we picked up while creating the app.

## Management Approach

### Project Process



The above image is what describe the “Waterfall Model”, a common model used in various projects. Our working process shares the same model, except for the “Design” phase, as none of the team member are well-versed in such field, and training to catch up with the design basic would require time. Due to limited and conflicting time and schedule, we decided to proceed without the Design phase.

### Quality Management

To ensure the project overall flow, we agreed upon using proper C# coding convention, as well as separating our codebase into various layers for easier maintenance. We also use GitHub to manage the project codebase, where all features’ implementations are separate into different branches and being reviewed by each other before merging to the main branch, where the branch will also be deleted should the merge is completed. We also used Discord to communicate during the project.

## Responsibility Assignments

## Project Communications

The team uses Discord, an instant messaging and social platform as our primary means of communication throughout the project. We also use GitHub for managing codebase and collaborating on the development process.

## Project Work Plan



## Configuration Management

### Document Management

This document was written in Microsoft Word

Some diagrams used in this document are created using diagram.io, a free web application for creating simple yet elegant and informative diagrams.

### Source Code Management

All Source code are managed/archived by Truong Vu Binh. He created and provided the project repository, as well as managing changes to the source codes by reviewing and approving pull requests (PR). Due to the time constraint and scheduling, we decided to use this method, as this would help us avoid code conflicting and ensure everyone shared the same codebase.

### Tools & Infrastructures

Our team ues the following tools and software to help aid the development process:

* C# Programming Language
* .NET 6 SDK
* **Visual Studio:** Our go to option when programming .NET applications. It has a plethora of tools to help us quickly get started on the project server-related functions, as well as databases.
* **Visual Studio Code:** Visual Studio is a great IDE, but it lacks supports for HTML and front-end developments in general, as such, we switch to Visual Studio Code as our code editor when we are developing the View aspect of the project.
* **Azure Data Studio:** Azure Data Studio is a free database management software that allow us to quickly glance at our project database, as well as helping us write queries faster and align well with the SQL coding convention thanks to its autocompletion feature.

# Software Requirement Specification

Our web application is created with 3 user groups in mind, each with a scoped set of permissions and features available to them. These are:

**For Shipping Merchants/Driver/Deliverer:**

* View the list of available offers posted by shop merchants
* View detail of a selected offer
* Accept an offer in listing
* Update the offer status after accepting an offer
* Edit their profile
* Update their password

**For Shop Merchants:**

* View the list of offers posted by the user
* Create a new Offer
* Update and Delete offers if the offer is not being accepted
* Edit their profile
* Update their password
* Report a shipping merchant offense

**For Admin:**

* Manage the list of all offers
* Manage user accounts
* Reply to customer messages
* View list of reported shipping merchants

## User Requirements

|  |  |
| --- | --- |
| **Use Case ID** | UC-01 |
| **Use Case Name** | Sign into the application |
| **Actors** | Guest |
| **Description** | User signs in to the application |
| **Pre-condition** | User is connected to the internet |
| **Post Condition** | User is signed in and redirect user to the dashboard |
| **Trigger** | User wishes to take an offer or post an offer |
| **Main flow** | |  |  |  | | --- | --- | --- | | No | Actor | Action | | 1 | Guest | Access the ShipLink home page | | 2 | System | Display the home page | | 3 | Guest | Tap/click on Sign In/Join Us link at the top of the page | | 4 | System | Display the sign in page | | 5 | Guest | Fill in their username and password then tap/click on Sign in | | 6 | System | Authenticate the user using the given credentials | | 7 | System | Display error if invalid, else redirect user to the dashboard | |
| **Alternative Flow** | N/A |
| **Exception Flow** | N/A |
| **Business Flow** | N/A |

|  |  |
| --- | --- |
| **Use Case ID** | UC-02 |
| **Use Case Name** | Sign out of the application |
| **Actors** | Registered User |
| **Description** | User signs out of the application |
| **Pre-condition** | User is connected to the internet |
| **Post Condition** | User is signed out and redirect user to the home page |
| **Trigger** | User wishes to sign out oof the application |
| **Main flow** | |  |  |  | | --- | --- | --- | | No | Actor | Action | | 1 | Guest | Access the ShipLink page | | 2 | System | Display the dashboard | | 3 | Guest | Tap/click on Sign Out at the left side of the dashboard | | 4 | System | Sign the user out of the system | | 5 | System | Rediect user to the home page | |
| **Alternative Flow** | N/A |
| **Exception Flow** | N/A |
| **Business Flow** | N/A |

|  |  |
| --- | --- |
| **Use Case ID** | UC-04 |
| **Use Case Name** | Accept a posted offer |
| **Actors** | Registered User |
| **Description** | User wants to accept an offer on the dashboard |
| **Pre-condition** | User is connected to the internet and has signed in to the application as a shipping merchant |
| **Post Condition** | Offer is saved to their Accepted list |
| **Trigger** | Shipping merchant wished to accept an offer |
| **Main flow** | |  |  |  | | --- | --- | --- | | No | Actor | Action | | 1 | User | Access the ShipLink home page | | 2 | System | Display the dashboard page if user is signed in | | 3 | User | Tap/click on Detail button on the offer listing | | 4 | System | Display the Offer Detail page | | 5 | User | Check Offer detail, then tap/click Accept Offer | | 6 | System | Acknowledge the acceptance and save the offer to the shipping merchant Accepted List | | 7 | System | Display a success message when finished | |
| **Alternative Flow** | N/A |
| **Exception Flow** | N/A |
| **Business Flow** | N/A |
|  |  |

|  |  |
| --- | --- |
| **Use Case ID** | UC-05 |
| **Use Case Name** | Post an Offer |
| **Actors** | Registered User |
| **Description** | User post an offer to the application |
| **Pre-condition** | User is connected to the internet, and it’s signed in as a shop merchant |
| **Post Condition** | Offer is saved to the database |
| **Trigger** | User post an offer to the application |
| **Main flow** | |  |  |  | | --- | --- | --- | | No | Actor | Action | | 1 | Guest | Access the ShipLink home page | | 2 | System | Display the dashboard page if user is signed in | | 3 | Guest | Tap/click on New Offer link at the left side of the dashboard | | 4 | System | Display the New Offer form page | | 5 | Guest | Fill in the destination address, price, note to shipping merchants, as well as uploading an image of the package, package description and tap Create Offer | | 6 | System | Save the information to the database | | 7 | System | Redirect back to the Posted Offer list | |
| **Alternative Flow** | N/A |
| **Exception Flow** | N/A |
| **Business Flow** | N/A |

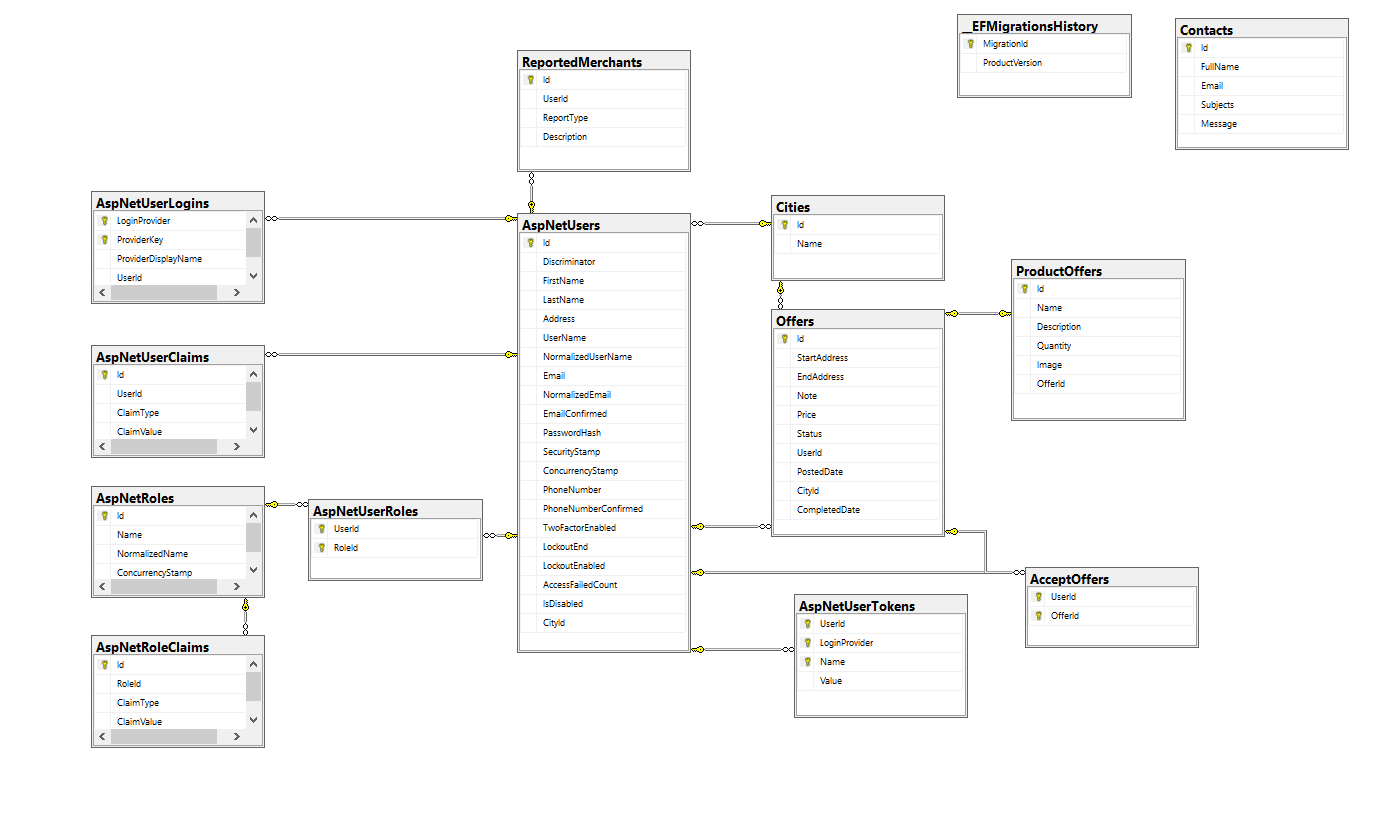
|  |  |
| --- | --- |
| **Use Case ID** | UC-06 |
| **Use Case Name** | Manage offers |
| **Actors** | Admin |
| **Description** | Manage all offers |
| **Pre-condition** | User is connected to the internet and it’s signed in as admin |
| **Post Condition** | System display all offers in the system |
| **Trigger** | Admin wants to see the list of all posted offers |
| **Main flow** | |  |  |  | | --- | --- | --- | | No | Actor | Action | | 1 | Guest | Access the ShipLink home page | | 2 | System | Display the home page | | 3 | Guest | Tap/click on Sign In/Join Us link at the top of the page | | 4 | System | Display the sign in page | | 5 | Guest | Fill in their username and password then tap/click on Sign in | | 6 | System | Authenticate the user using the given credentials | | 7 | System | Display error if invalid, else redirect user to the admin dashboard, show the offer list | |
| **Alternative Flow** | N/A |
| **Exception Flow** | N/A |
| **Business Flow** | N/A |

|  |  |
| --- | --- |
| **Use Case ID** | UC-07 |
| **Use Case Name** | Manage users |
| **Actors** | Admin |
| **Description** | Manage registered users in the system |
| **Pre-condition** | User is connected to the internet and it’s signed in as an admin |
| **Post Condition** | Admin can manage users |
| **Trigger** | Admin wishes to see a list of registered user information |
| **Main flow** | |  |  |  | | --- | --- | --- | | No | Actor | Action | | 1 | Guest | Access the ShipLink home page | | 2 | System | Display the home page | | 3 | Guest | Tap/click on Sign In/Join Us link at the top of the page | | 4 | System | Display the sign in page | | 5 | Guest | Fill in their username and password then tap/click on Sign in | | 6 | System | Authenticate the user using the given credentials | | 7 | System | Display error if invalid, else redirect user to the admin dashboard, | | 8 | Admin | Tap Manage User at the left side of the dashboard | | 9 | System | Display the list of users | |
| **Alternative Flow** | N/A |
| **Exception Flow** | N/A |
| **Business Flow** | N/A |

|  |  |
| --- | --- |
| **Use Case ID** | UC-07 |
| **Use Case Name** | Register as a shipping merchant |
| **Actors** | Guest, Admin |
| **Description** | User register as a shipping merchant |
| **Pre-condition** | User is connected to the internet |
| **Post Condition** | User is registered and redirected to the dashboard |
| **Trigger** | User wishes to partake to the system as a driver |
| **Main flow** | |  |  |  | | --- | --- | --- | | No | Actor | Action | | 1 | Guest | Access the ShipLink home page | | 2 | System | Display the home page | | 3 | Guest | Tap/click on Sign In/Join Us link at the top of the page | | 4 | System | Display the sign in page | | 5 | Guest | Tap the Create a new account link | | 6 | System | Display the Register form | | 7 | System | On account type, user select account type to be “Shipping Merchant”, then provide their First Name, Last Name, Email Address, Phone Number, and tap Register | | 8 | System | Save the information to the database, redirect user to the Dashboard | |
| **Alternative Flow** | N/A |
| **Exception Flow** | N/A |
| **Business Flow** | N/A |

## Functional Requirements

### System Functional Overview

**

## Non-Functional Requirements

### Reliability

We make sure that:

* All forms have proper validation put in
* All basic functionality works properly and reliable

### Availability

As this project is a web application, it is available to almost any modern device with a web browser.

### Security

All functionalities have proper authentication method implemented and secured

### Maintainability

The codebase is separated into different layers, allowing easier maintainability, as well as improve readability for future should there be a need to implement additional functions, or making modification to existing codebase

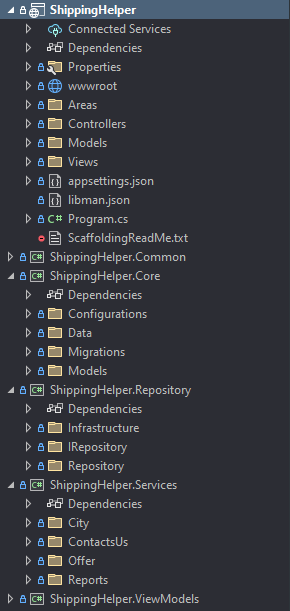
### Portability

The project is a web application, as such, the app can be used by either mobile devices, tablets, and/or personal computers.

# Software Design Description

## System Design

### System Architecture



Our project following a hybrid of MVC and N-tier architecture. Whereas:

* **ShippingHelper**: Our primary web application. This project strictly follows the MVC design architecture and the presentation of the application. All controllers will execute necessary actions by calling the respective services.
* **ShippingHelper.Common**: This layer contains extension methods, and other utilities method for various purposes (for instance: File Uploading, generate friendly date/time)
* **ShippingHelper.Core:** This layer contains the Database configurations, the application database models, and all database modification history
* **ShippingHelper.Repository:** This layer containsmethodsused to interact with the database, this layer includes a base repository that all other repositories are implementing, as well as the Unit of Work class for calling the repository method of choice.
* **ShippingHelper.Services:** This layer contains calls for each respective repository methods. Controller Action Methods would consume these services, which in turns, call the repository method.
* **ShippingHelper.ViewModels:** This layer contains models to be used in the web application. Since we do not want to directly tamper with the database model, we used these View Models as a means of data collections from forms, where then the View Models data will then be converted to their respective database models.

## Project Design

### Sign in to the system

Diagram

Description automatically generated

### Add a new Offer

Diagram

Description automatically generated

### Accept a new Offer

Diagram

Description automatically generated

### Edit an offer

Diagram

Description automatically generated

# Software Testing Documentation

## Scope of Testing

When conduct our testing, we only test the repository layer of our application, using NUnit testing framework, we wrote test cases for methods that returns data to us. As such, for other operation, such as Creating, Updating, or Deleting, they are all manually tested by the person responsible for testing on our team.

## Test Plan

Our test cases are written using the following tools:

* **NUnit testing framework**: a popular testing framework that has built-in support in our IDE
* **Moq**: a mocking library allow us to create mock data to test our repository functionality without ever having to directly interact to the project database

All test cases are white box tests, and conducted in an isolated environment, meaning, the test cases are performed by using mock data, with a mock-up database context required for all the repository to work.

# Release Package & User Guides

## Deliverable Package

| **No.** | **Deliverable Item** | **Description** |
| --- | --- | --- |
| 1 | Schedule/Task Tracking | Provided in both printed and digital form |
| 2 | Source Codes | Source Code fully available on GitHub repository |
| 3 | Database Script(s) | Given in project GitHub repository |
| 4 | Final Report Document | Provided in both printed and digital form |
| 5 | Slide | Provided in project repository |

## Installation Guides

### System Requirements

|  |  |
| --- | --- |
| **Server** | **Client** |
| 1GHz Dual-core CPU or higher | 1GHz Dual-core CPU or higher |
| SQL Server 2017 or later | Any web browser with latest update |
| .NET 6 Runtime | Windows 7 or later |
| Windows 8.1 or later | At least 1GB of RAM or more |
| IIS Role or feature enabled | 10GB of free hard disk space |
| 20GB of free hard disk space | Internet connection |
| At least 2GB of RAM or more |  |

### Installation Instruction

**For Server:**

* Enable Internet Information Services (IIS) found in Optional Features in Windows
* Install latest version of .NET Runtime, SQL Server
* Add the web app into IIS
* Query the database using the given SQL Scripts to have the database required for the webpage to work