TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**FINAL REPORT**

**SOFTWARE ENGINEERING**

**FINAL REPORT**

*Supervisor*: **Phạm Thái Kỳ Trung**

*Student*: **Thái Thành Gia Bảo-519K0064**

**Kiều Nguyễn Xuân Thiện-519V0045**

Class **: 19K50301**

Course  **: 23**

**THÀNH PHỐ HỒ CHÍ MINH, YEAR 2021**

**Executive Summary**

**Table of Contents**

**1. Introduction**

**1.1. Purpose and Scope**

**1.2. Product Overview**

**(including capabilities, scenarios for using the product, etc.)**

**1.3. Structure of the Document**

**1.4. Terms, Acronyms, and Abbreviations**

**2. Project Management Plan 2.1. Project Organization**

**2.2. Lifecycle Model Used**

**2.3. Risk Analysis**

**2.4. Hardware and Software Resource Requirements (Do not forget to describe what new software or hardware each team member learned during the project)**

**2.5. Deliverables and Schedule**

**2.6. Monitoring, Reporting, and Controlling Mechanisms**

**2.7. Professional Standards**

**2.8. Evidence all the artifacts have been placed under configuration management**

**2.9. Impact of the project on individuals and organizations (Include a description of what impact your project will have on individuals and society)**

**3. Requirement Specifications**

**3.1. Stakeholders for the system**

**3.2. Use case model**

**3.2.1. Graphical use case model**

**3.2.2. Textual Description for each use case**

**3.3. Functional requirements**

**3.4. Non-functional requirements**

**4. Architecture**

**4.1. Architectural style(s) used**

**4.2. Architectural model**

**4.3. Technology, software, and hardware used**

**4.4. Rationale for your architectural style and model**

**5. Design**

**5.1. Database design**

**5.2. Static model – class diagrams**

**5.3. Dynamic model – sequence diagrams**

**5.4. Rationale for your detailed design model**

**5.5. Traceability from requirements to detailed design model**

**6. Test Plan**

**6.1. Requirements/specifications-based system level test cases**

**6.2. Traceability of test cases to use cases**

**6.3. Techniques used for test generation**

**6.4. Assessment of the goodness of your test suite (Which metrics were used for such assessment?)**

**7. Demo**

**7.1. Database**

**7.2. Source code**

**7.3. Testing**

**List of Figures**

**List of Tables**

1. Introduction
   1. Purpose and Scope

This software will act as the main shopping website for the store or company, mainly to be the website were customers will be able to select and choose what they want to buy, put them in a shopping cart and choose which payment method they want to have.

1.2. Product Overview

(including capabilities, scenarios for using the product, etc.)

This is a software to be able to set up a system of buying through online means as well as notifying delivery for the people placing orders online.

More so as mentioned above for the customers they can buy what they want from the website and put it into their shopping cart.

The customers can choose their method to pay like using credit cards, paypal, zalo, momo, etc.

After the payment, the system will send the payment info to the bank and the delivery information to the operators so they can send the orders to delivery

This system will also allow the operators to see the stock, prices and values as well as able to update any of those categories on demand.

This system will be able to send the sales reports and monthly revenue to the accountants assigned and the executives.

1.3. Structure of the Document

This document will go through the main functions as well as show graphs on how this system works. Starting with how this project was organized, and showing in depth on how this software works, and what processes are done with it.

1.4. Terms, Acronyms, and Abbreviations

There will not be any acronyms.

2. Project Management Plan

2.1. Project Organization

Coder: Kiều Nguyễn Xuân Thiện

Manager and Analyst: Thái Thành Gia Bảo

2.2. Lifecycle Model Used

The life cycle model used is the general life cycle model used for software development.

Demonstrated here.





2.2.1. Planning

This stage was just planning and discussion between members and what steps should be taken.

2.2.2. Analysis

This stage would include me, and analyze what the requirements needed to have.

2.2.3 Design

The design analysis and what the software and program should look like.

2.2.4 Implementation

The Implementation phase, is when we try to develop the software according to the analysis stage.

2.2.5. Testing and integration

Testing the software and if it’s not up to standards, we return to the implementation phase, and once it’s ready, we’ll teach it to the users and how they use the software through the use case diagrams.

2.2.6. Maintenance

This is the section where we teach the users how to maintain the program, and how to update the program once the requester asks for updates to the system.

2.3. Risk Analysis

This would include human errors in trying interact with the system, errors and damages from outside interferences.

This will also include glitches in the system, things that may mess with systems memory and or/ changes to the software that makes the data incorrect.

2.4. Hardware and Software Resource Requirements (Do not forget to describe what new software or hardware each team member learned during the project)

Hardware requirement: Windows 7 or higher

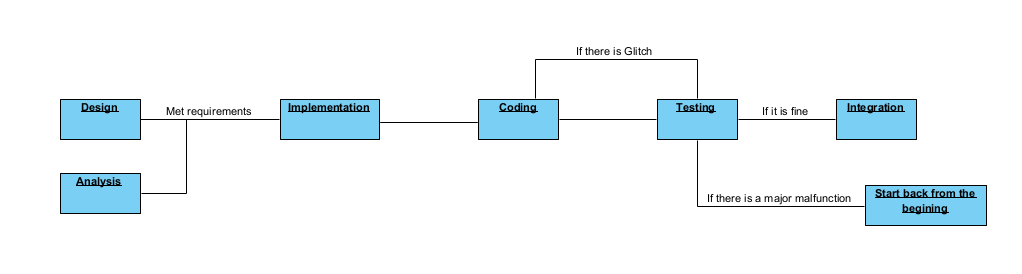
Software requirement: Web support of Internet Explorer 5.0 or higher, Windows chrome, or firefox

Database: Microsoft SQL Server

Client Framework: HTML or Javascript

2.5. Deliverables and Schedule

This depends on how the project is done, if there are any problems, the project will take more time, based on this chart.



2.6. Monitoring, Reporting, and Controlling Mechanisms

For the Project:

For the project we need to focus on the design and analysis, communication here is important as any miscommunication about the requirements will mess up the scope and size of the project, as well as make a bad call.

This will also call for the coding, communications between teams will be important, therefore, this will rely on most of the design and analysis team, making the right plans and descisions and project managers to make the right calls and observe the progress.

For the Program:

Monitoring would come to the employees.

With reports, employees and users should report to the maintenance teams.

How to control this system is via the App on the phone or through the website on the computer. Customers can log in through the website to make their purchases by entering their credit information and delivery information.

Employees should be able to manage the website and transfer the note to the delivery section, which will notify the delivery employees to pick up their package and deliver.

Employees will be able to use the computer the check up and update the stock and prices.

For the project, quirks and risks may arise on working with the project and we will have to start over and over again, making the project time take longer and taking up time of everyone.

2.7. Professional Standards

The standards for this program should be up to demand and the users should be able to use it and manage it with proper ease.

For the project, most people will be required to be responsible for their own part, an issue in the planning phase will be on the fault of the planning and design team.

The Fault of the code and glitches will fall on the shoulders of the coding and implementation teams.

2.8. Evidence all the artifacts have been placed under configuration management

2.9. Impact of the project on individuals and organizations (Include a description of what impact your project will have on individuals and society)

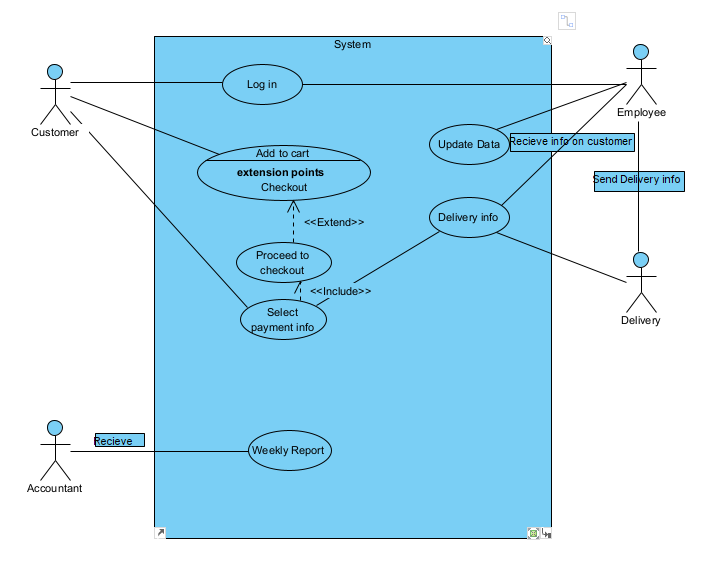
For the individual, this will not affect much on them for this is just another website.

For society, this will help a store or market to help them have a better and more easy way to track and make deliveries.

3. Requirement Specifications

3.1. Stakeholders for the system

3.2. Use case model

3.2.1. Graphical use case model 

3.2.2. Textual Description for each use case

Log in would be the use case for when the customer logs into their customer profile, and for the employee would be for when they log into their employee profile.

Customers profile would allow the customer to browse the website store and add items to the cart, once the customer has finished with the selection of items to the cart, they can proceed to checkout where the customer can enter their payment method.

Once finished, the system will send out the delivery info to the employees, in which they will send to the delivery to notify the delivery staff about a new order being made.

The employees can check on and update the info in the system at any time, check up on stock and track sales.

The delivery staff will have access to their delivery info at any time.

For every week or month, the accounting staff will be sent a report on sales to keep the revenue in check.

3.3. Functional requirements

Compatibility with older OS

Improve P2P so that the connection is better

Incentive compatibility for everyone

Inter-domain support

A functional UI

Functional and working website for both parties.

3.4. Non-functional requirements

Easy deployment

A nice looking UI

4. Architecture

4.1. Architectural style(s) used 

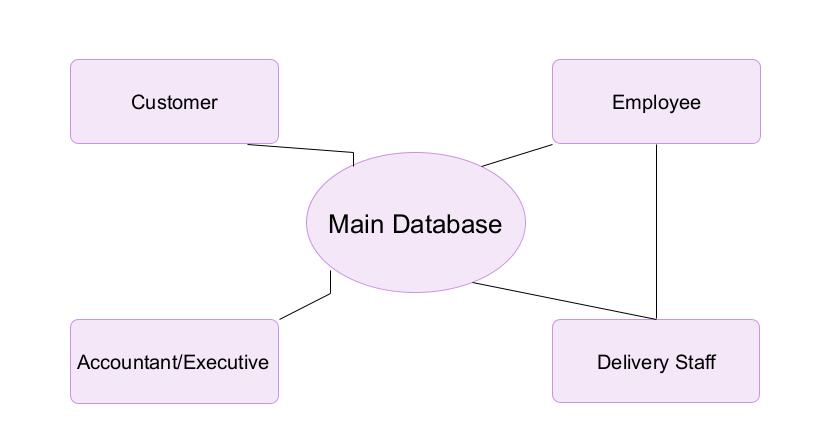
We used the waterfall model as our plan for developing the system, step by step, it is simple and steady. It is also a safe way to build the program without breaking.

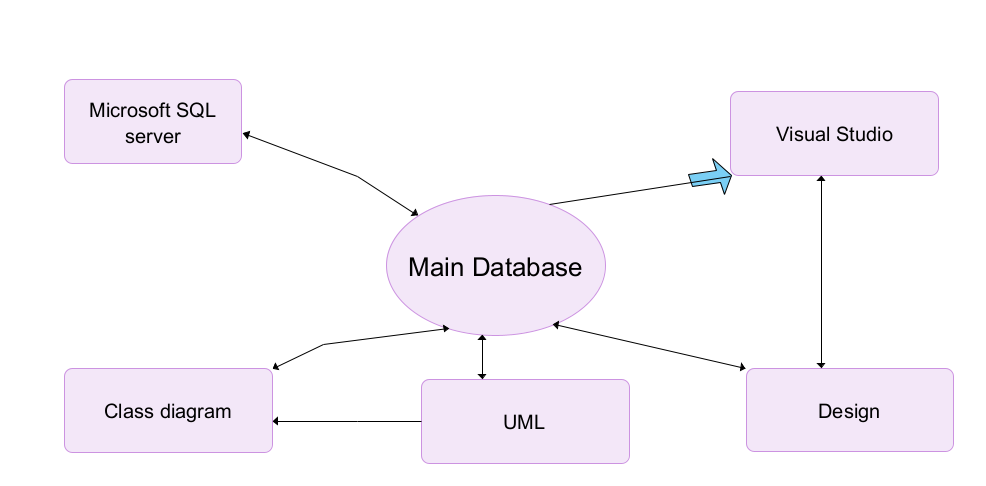
Data Centered Architecture:

Using the main database to store the main data and most of the information on customers, delivery, stock and most of the resources of the company.

The main database can be accessed by customers, employees, delivery staff, accountants and executives in a different way.

4.2. Architectural model





4.3. Technology, software, and hardware used

Hardware used: HP Laptop.

Software used: Visual Studio 2020, Visual Paradigm, Microsoft SQL server Manager 2017

4.4. Rationale for your architectural style and model

The architectural style used is the most simple and straightforward method to get the results we needed.

Therefore, we are able to make it quickly and efficiently, while also having little flaws and glitches as it is the perfect architectural method for this system.

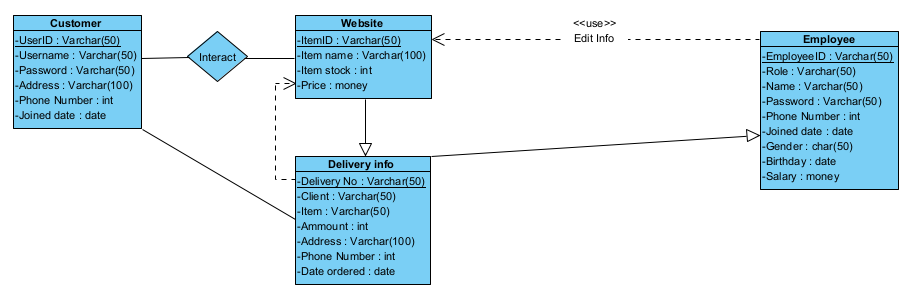
5. Design

5.1. Database design

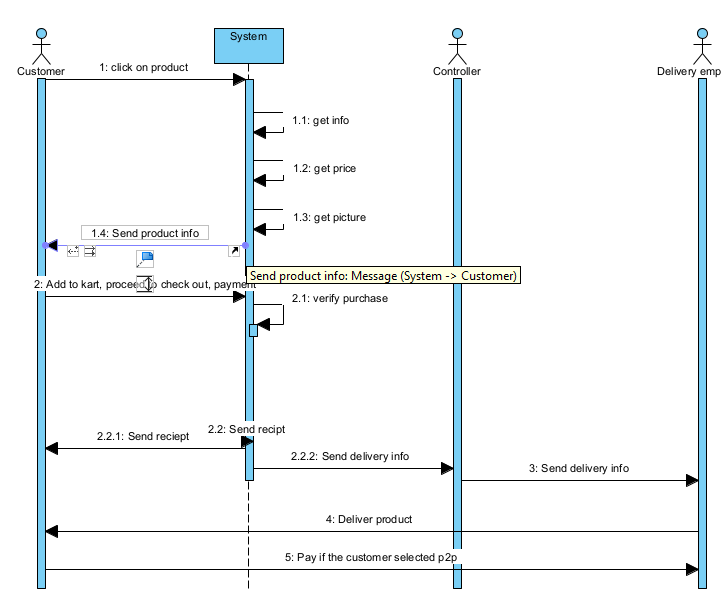
This will contain the main data on the system, such as Customers, delivery info, the website info and the employee info.

Each will have their own different info regarding their table.

5.2. Static model – class diagrams



5.3. Dynamic model – sequence diagrams



5.4. Rationale for your detailed design model

This design model was used for its simplicity and efficiency, as it is the most simple and perfect design for the team.

5.5. Traceability from requirements to detailed design model

6. Test Plan

6.1. Requirements/specifications-based system level test cases

Based on the functional requirements and what the requester wants from the system.

6.2. Traceability of test cases to use cases

6.3. Techniques used for test generation

For each generation of the software, it will be run and tested on it what requirements it has fulfilled, and which part of it is fully functional

It will also be tested for bugs and glitches in the system, if there are any bugs or glitches, the system will keep being re-tested to root out and fix bugs.

6.4. Assessment of the goodness of your test suite (Which metrics were used for such assessment?)

7. Demo

7.1. Database

7.2. Source code

7.3. Testing