

FC6P01 Project

Final Report

‘Mart’ - An Online Food Distribution Network system - A web based Food Delivery System

Name: Chamith Chathuka Wickramarathna

ID Number: 16033954

Date: 02 April 2017

First Supervisor: Ms. Theja Nadeeshani Perera Ilanperuma

Second Supervisor: Mr. Thilina Ranathunga

# Declaration

**Module: FC6P01 Deadline: 30/05/2017**

**Module Leader: Mr Nishan Sembacuttiaratchy Student ID: 16033954**

PLAGIARISM

You are reminded that there exist regulations concerning plagiarism. Extracts from these regulations are printed below. Please sign below to say that you have read and understand these extracts:

(signature:)\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date: 24/06/17

This header sheet should be attached to the work you submit. No work will be accepted without it.

Extracts from University *Regulations* onCheating, Plagiarism and Collusion

Section 2.3: "The following broad types of offence can be identified and are provided as indicative examples…

1. Cheating: including taking unauthorised material into an examination; consulting unauthorised material outside the examination hall during the examination; obtaining an unseen examination paper in advance of the examination; copying from another examinee; using an unauthorised calculator during the examination or storing unauthorised material in the memory of a programmable calculator which is taken into the examination; copying coursework.
2. Falsifying data in experimental results.
3. Personation, where a substitute takes an examination or test on behalf of the candidate. Both candidate and substitute may be guilty of an offence under these Regulations.
4. Bribery or attempted bribery of a person thought to have some influence on the candidate's assessment.
5. Collusion to present joint work as the work solely of one individual.
6. Plagiarism, where the work or ideas of another are presented as the candidate's own.
7. Other conduct calculated to secure an advantage on assessment.

(viii) Assisting in any of the above.

Some notes on what this means for students:

1. Copying another student's work is an offence, whether from a copy on paper or from a computer file, and in whatever form the intellectual property being copied takes, including text, mathematical notation and computer programs.

2. Taking extracts from published sources *without attribution* is an offence. To quote ideas, sometimes using extracts, is generally to be encouraged. Quoting ideas is achieved by stating an author's argument and attributing it, perhaps by quoting, immediately in the text, his or her name and year of publication, e.g. "e = mc2 (Einstein 1905)". A *references* section at the end of your work should then list all such references in alphabetical order of authors' surnames. (There are variations on this referencing system which your tutors may prefer you to use.) If you wish to quote a paragraph or so from published work then indent the quotation on both left and right margins, using an italic font where practicable, and introduce the quotation with an attribution.

# Abstract

This Report is prepared to address A web based Food Delivery System solution for restaurants. Aims to conquer and redefine the current distribution network in Sri Lanka by delivering almost anything you need straight to your doorstep within approximately 30-60 minutes. This system will help for small restaurants and groceries which currently do not have an online system and own product delivery system. Partnering with this system will help those restaurants and groceries to expand their sales and market share. At initial stage of the document gives the reader clear idea about the background of the problem. Afterwards this gives an idea of the proposed system and how to overcome from the problem with given solution. Then this document provides and idea about a system construction process and some screen shots from the current process of the system. At last this report gives list of references used to prepare the document for further reference.

Acknowledgement

I thank to Mrs Mahesha Thejani and Mrs Theja Nadeeshani, for guiding me to do the project and supervising me to to the end of the project with their suggestions.

Also my thank goes to Mr. Thilina Ranathunga for giving we support from the technical side and introducing new technologies used and also sharing his experience with using those technologies.

I also thank to Mr. Nishan Sembacuttiaratchy for conducting us with several project guidance sessions and sharing techniques for writing documentation which really helped me when I was preparing the documentation.

Finally, My thank goes to my classmates who shared their ideas on my project and helped me to develop the documentation

Contents

[Declaration 2](#_Toc478871699)

[Abstract 3](#_Toc478871700)

[Introduction 5](#_Toc478871701)

[Background 6](#_Toc478871702)

[Work Completed 8](#_Toc478871703)

[Organization, According to the Gantt chart mentioned in previous page so far author was able to completed most of the tasks up to date. 9](#_Toc478871704)

[Feasibility Study 9](#_Toc478871705)

[Requirement Specification 11](#_Toc478871706)

[Class Diagram 13](#_Toc478871707)

[Entity Relationship Diagram 19](#_Toc478871708)

[Further Work 20](#_Toc478871709)

[Bibliography 21](#_Toc478871710)

# Introduction

Today, with busy life style of the people, and with the development of new technology people always trying to save their time and put that effort to do something useful or even they spent more time with their family members. These technologies based systems and devices are helping to improve the efficiency and do the traditional task in new innovative methods. This new technology supports individuals to improve their life style and business to increase their sales and earn revenue with expanding market share. (von Breitenbuch, 2004)

The ecommerce can be defined as a modern business methodology that can be address then needs of organizations, merchants and consumers to cut the costs while improving the quality of goods and services and increasing the speed of service diversity by using the internet. (Meier and Stormer, 2009)

This system providing service for Merchants and customers. This system is providing SaaS for its end users. SaaS is a simply software which is delivered from s server remote location your desktop it is an application hosted on Remote server. Software uses its own interfaces, designed to provide a special characteristics for the software. Some of the SaaS applications require to pay a subscription fee from its end users, but the proposed system only taking a small commission fee from the server, SaaS application requires active internet connection to connect to this web site so if there is no internet connection, there will be no application for client to connect, that is the main disadvantage of this system, but currently almost everyone has access to internet this will be a not a problem.(Wohl, 2008)

Online ordering of foods represents 30% of daily takeaway ordering according to Cowen and Company Research Report and it is now rapidly growing with the due to its convenience, correctness of ordering and no waiting queues. Studies show about 69% of customers using their mobile devices for online food ordering.

As the first step off the document ‘background’ section author gives clear idea about the current difficulties faced by the customers face when doing traditional take away shopping and difficulties faced by small restaurant and grocery owners when delivering food to customers and expanding their business. How to overcome for these problems by analysing those problems and over solution will help sellers to overcome from it. In addition to those parts, this section will provide details of the technologies that the system is using and the reason for selecting those technologies. It will also give a concise idea on implementation of the system.

Next section of the document will provide information on the work completed up to now. This report will be given tasks which were addressed to pre given Gantt chart in the project proposal and feasibility study, Wire frame design and some of the UML diagrams related to the project and the ER diagram of the project.

# Background

Mart is food delivery network which connects merchants and customers. Small shop owners and Restaurant owners must spend additional investments to provide a delivery service for its customers, which is big barrier for the business to grow and reach its target. Most of the small restaurants or shops do not have their own web site for their customers to reach.

When we consider form the customer’s perspective Customers have to visit the physical place to buy their favourite food or purchase food items for their needs. It is a time consuming and stressful work this generation.

Mart gives a better solution for by bridging these two types of customers by giving their platform for their delivery needs, marketing needs and sales expansion needs from the merchant side and giving quick delivery at lower cost from the client side.

Currently “Mart” is operated by a phone system and manually takes the order over the phone and then contacts the nearest delivery person and provides delivery details to him again over the phone and receiving the cash on delivery method. Customers need to provide their address every time they place and order. After purchasing the ordered items from the customer mentioned restaurant or shop.

Since this process make much more errors at the delivery stage since everything is done over the phone customers are complaining for wrong orders and that make the process more complex and financial loss for the company. Increased phone charges directly affecting to the profit of ‘Mart’.

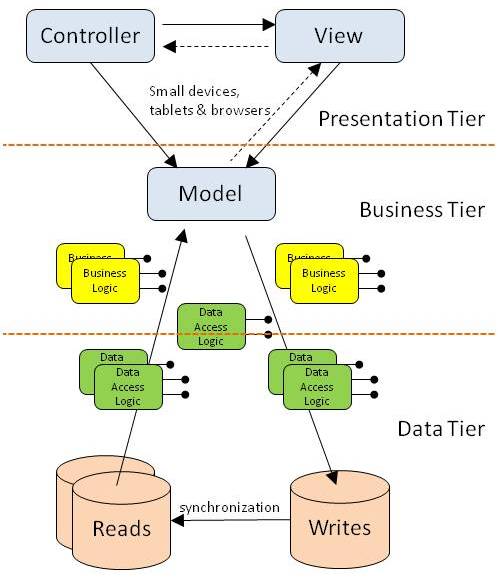
To overcome from these problems Author suggested an online platform for Mart and where gives the small restaurant and shop owners to maintain their profiles by partnering with ‘ Mart’ and after registering with systems customers can order and their favourite food products and grocery items from their devices, after placing and order Restaurant owner automatically gets the notification, and if that restaurant or the grocery request delivery method by ‘Mart’ Mart agent will pick the order from the restaurant or grocery with short amount of time and deliver for the customer provided address.

Since this a ecommerce site this site revenue generation is done by getting 5% commission from the total value of the each order customer placed through this system, also ‘Mart’ standard delivery charges will be applied for the customers who select ‘Mart Delivery’ option.

Developer used Spring Boot frame work to develop the backend of the web application, for the Database development developer used MySQL 5.5 Community Edition server. Hibernate 5 is used to object relational mapping. Materize CSS is an Open Source responsive User interface designing.

Initial system of the system developed using as a combination of a Classic Waterfall Model and the Prototyping Model as the development model.

Architecture of the system developed as MVC implemented 3 tier architecture. Use of hybrid is MVC pattern do not describe how to best design and data access and how to manage the complexity that occur in the system. (Rawsthorne, Rawsthorne and profile, 2017)



(Anon, 2017)

# Mission

The mission of this project to build an efficient and free User Friendly Online shopping platform.

**Objectives**

Provide common online platform for small restaurant owners as well as field giants to reach out their customers.

Give end customers easy and delightful service.

Provide delivery service for reasonable price restaurants does not own their own delivery service.

# Background of the problem

This section describes the problems people face when they need to order food from the restaurants and the problems faced by the small restaurants to reach their customers, since the target merchant audience of the Mart platform is small scale restaurants, which has one branch.

Firstly, consider about the traditional restaurant, restaurant can’t keep his shop open all 24 hours to receive orders. This is narrowing sales or the customer.

Most of the restaurants give phone ordering facility customers, this traditional system has several disadvantages, customers must wait some minutes to place the order, sometimes customer should wait while telephone operator pick the order it that phone system is managed by PABX. It will also cost money while waiting on the phone, more customer wait lead to customer dissatisfaction. Separate person might need to handle the orders placed through phone. There is a high probability of miscommunications when taking the ordering over the phone and taking down the wrong order.

Menu management is not an easy task in traditional restaurant they must work on, Customers need to always call and ask for available items in traditional restaurants.

Expansion of business and reaching to new customers, with traditional restaurant owners they have invest huge sum of money to open a new branch of restaurant. When considering the cost for new place, recruit new staff etc.)

# Solution

As a solution for the above-mentioned problems author developed an online food ordering platform. Targeting small restaurant owners but common for all restaurants.

This is how mart shopping platform provide solutions for the above problems.

Mart is an Online Shopping platform so for the restaurant owners who register with mart able to open their own store and accept their orders any time, through the mart shopping platform.

End customers can register with few simple steps for the first time and directly order the food based on the Realtime availability checking and the checking the latest menus provided by the restaurant owners. 24/7. 0% mistakes when placing the order if the customer ordered correct food.

Menu management is never being easy, even restaurant owners can check the best-selling products Realtime and focus on those products and change the menu items daily.

Since this is online system restaurant owners can reach to large amount of audience with zero investment cost since the mart shopping platform is free for all merchants. Restaurants which do not have their own delivery service Mart provides delivery service for a small fee.

Mart shopping platform also provides merchants a dashboard to check the important restaurant metrics in real time. Mart shopping platform keeps track of the data and restaurant owners can identify their top customers and improve the customer experience providing target market campaign.

Existing web applications.

Speedee.lk is similar web applications author found searching the internet. Speedee.lk site also offer their sites with products and restaurant pages.

Features

Users able to search products by category.

Users can explore the products that are listed in the restaurant owners page.

Can order the food.

Customers can rate food.

Features not available with the system

Products are not well organized.

Ex- Customer cannot identify searching a food and the restaurant make the food.

No clear merchant side integration.

Site user interface is not a user friendly one.

Quickee.lk

This site is exact similar site like speedee.lk and more developed and eye catching user interface.

Disadvantages.

Users can not pre-order foods if the sop is closed.

Menu lists are not getting updated.

No clear merchant side integration.

Design

Software process model is an abstract representation of software process. It is a structured set of activities required to develop a software system. There are many software models available, but author chose prototyping model for the development of Mart online platform.

Because classical water fall model since this is web based system may not be able to completely defined all terms of the appearance and operation of the web interface, and therefore a prototyping approach would be more practical than attempting to completely define the requirements as in a waterfall approach. This enabled developer to understand the customer requirements at early development stage, also it helped to get feedback from the customers of the Mart system.

Protoyping model consists of few steps

[](http://istqbexamcertification.com/wp-content/uploads/2012/01/Prototype-model.jpg)

Figure 1 Prototyping Model

Identify Initial requirements

In this step, the business analyst decided what the software will be able to do. The business analyst considers who the user is and what will likely be and what the user will want from the product.

Develop Initial Prototype

The developer will consider the requirements proposed by the publisher and begin to put together a model of what the final system will look like.

Review

Once the prototype developed business analyst has a chance of to see what the product might look like and whether the prototype matching the system requirements specification. End user also able to do suggestions for improvement.

Revise

The final step of the process is to revise the prototype based on the feedback of the business analyst or the testers.

(study.com 2017)

There are several types of prototyping also available. Developer used Evolutionary Prototyping method. First version of the Mart developed with minimal functionality. Only the well understood requirements are included in the first prototype. (tutorialspoint,2017)

Work Breakdown Structure of the Mart System of the Project

Work breakdown Structure (WBS) is a model of project work that should be done in a structured project in a hierarchical structure. They form the basis for organization and coordination of the project (Miller, 2009)

Figure 2 WBS -Mart

## Feasibility Study

What is expected from a feasibility study is that verifying the system is worth implementing and whether it can be implemented within the given timeline and budget (Sommerville, 2010). However, the final outcome of the feasibility study is deciding on whether or not the system development is continued. By doing this in the correct way the project a lot of time, money and resources can be saved.

There are many different types of feasibility studies. Prior to starting this project the following feasibility studies were done:

* Technical feasibility
* Schedule Feasibility
* Economic feasibility
* Operational feasibility

**Technical feasibility**

Technical feasibility is compatibility of technology we are using resources at hand in implementing the project. Technical resources considered in this project are the technical knowledge, capacity of hardware devices and capability of software. Further assessment on different types of frontend and backend platforms were done before choosing a specific platform.

**Schedule Feasibility**

Schedule feasibility is analysing whether the system can be completed within the timeline given. According to the schedule feasibility of this system, a decision was made to work part-time on the project implementation process.

**Economic Feasibility**

Economic feasibility is analysing the cost and the revenue of the project. In this project one of the main concerns was the affordability of additional software and hardware needed to implement the project. Keeping the project within the given budget was challenging as well. However it was decided that the cost of hardware and software needed to implement the system.

**Operational Feasibility**

Operational Feasibility involves testing the operational scope of the system. When the operational scope of the system is high, the usability of the system will also be high. Without doubt the system is highly user friendly with fully GUI. But it was decided it would be better to conduct a training program to make users more comfortable and to get the maximum efficiency of the system.

**Requirement Gathering and Analysis**

Functional requirements.

Functional requirements are need identify the most important task, functions services. Functions that are required to deliver the final product, these requirements are used to assess and evaluate the software.

These requirements are identified

Users of the system,

Restaurant Owners, End Customers, Delivery Agents and Admin Users should be able to Register with the system and respective views should be appear for them.

Users who are not registered with the system only able to explore the shops and food items listed.

All Users should be able to update their profile pages.

Admin Users should be able to add new restaurants and generate reports or add food items to the restaurant page.

End Customers of the system should be able to browse restaurants and their products.

End Customers should be able to add items to cart.

End Customers should be able to Modify the ordered items and remove the unnecessary items from the cart before placing the order.

Customer should be able to select a delivery address and specific delivery time.

Customer should be able to search for Restaurant or Food item

Respective restaurant owners and delivery agents should receive a notification when customer place the order.

**Non-Functional Requirements**

Application database should be secured.

All data should have a backup.

User experience should be high.

**System Design**

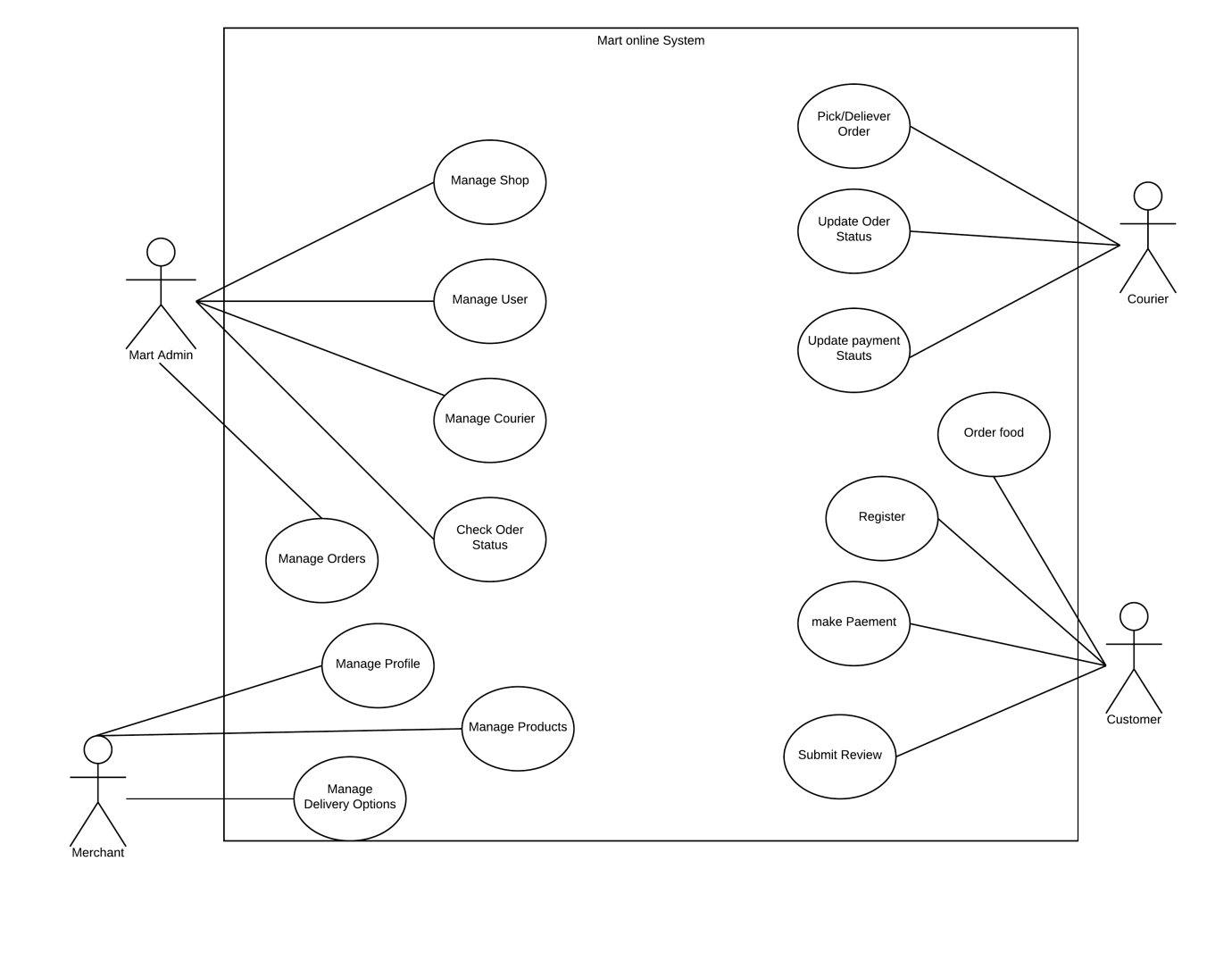


Figure 3 Use case Diagram

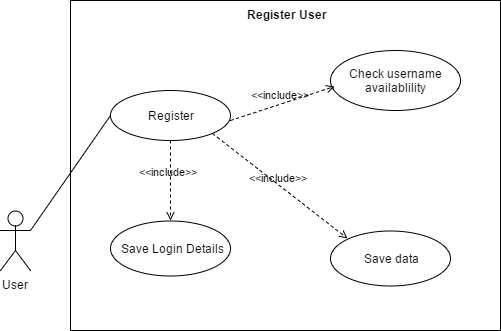


Figure 4 Register User Use Case

|  |  |
| --- | --- |
| **Test Case** | Register |
| **Actors** | Admin, End Customer, Merchant, Delivery Person |
| **Overview** | |
| * All users for the system are registering through this use case. | |
|  | |
| **Preconditions** | |
| User is registering for the first time. | |
|  | |
| **Flow of events** | |
| * Check database for email * Send invitation email * Save on detail on separate micro service account * Save data for role in database. | |
| **Post of Condition** | The user is directed to the respective home page according to the user role. |

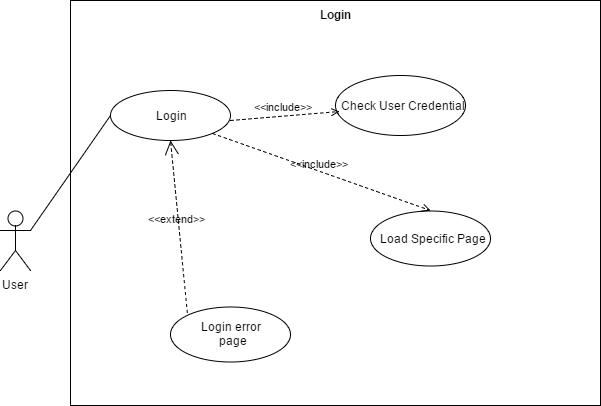


Figure 5 Login Usecase

|  |  |
| --- | --- |
| **Test Case** | Login |
| **Actors** | Admin, End Customer, Merchant, Delivery Person |
| **Overview** | |
| * All users for the system are login through this use module. | |
|  | |
| **Preconditions** | |
| User is already registered | |
|  | |
| **Flow of events** | |
| * Check database for credentials according to user input * Show error message if not matching. * Save on detail on separate micro service account * Save data for role in database. | |
| **Post of Condition** | The user is directed to the respective home page according to the user role. |

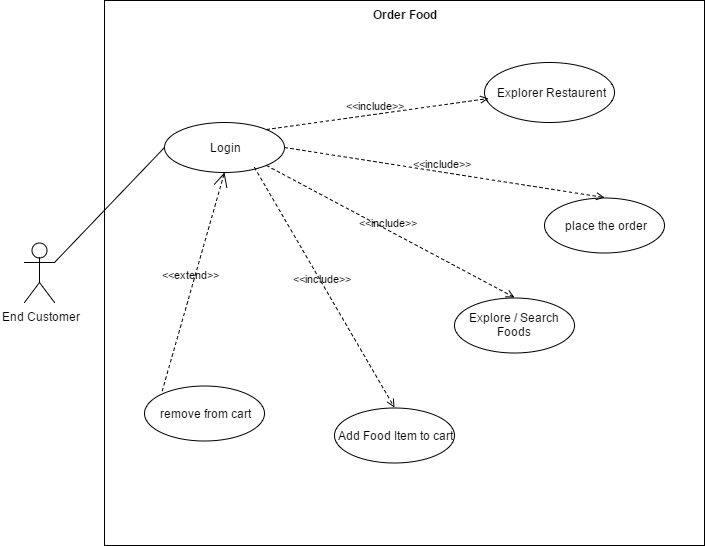


Figure 6 Order Food use case

|  |  |
| --- | --- |
| **Test Case** | Order food |
| **Actors** | Admin, End Customer |
| **Overview** | |
| * Admin users and end customers of the system can order food through this module. | |
|  | |
| **Preconditions** | |
| User is logged in | |
|  | |
| **Flow of events** | |
| * Search for a restaurant. * Select and go inside the restaurant page / alternatively user can directly search for the food * Select food and add to cart button * After adding the food click on the checkout button on the cart summery view. * User direct to Checkout page. | |
| **Post of Condition** | The user is directed to the respective home page according to the user role. |

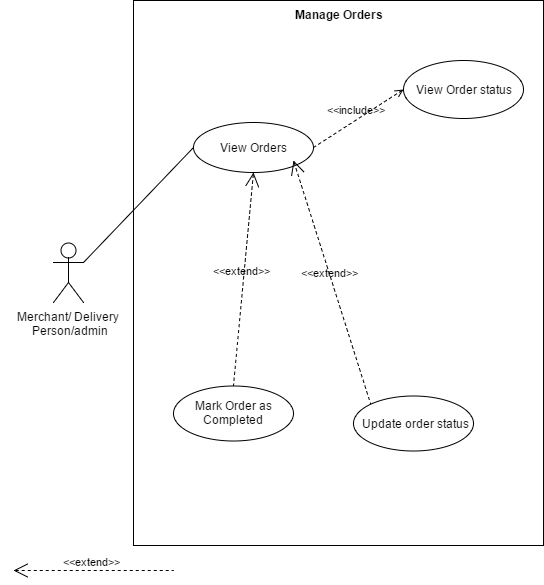


Figure 7 Manage Orders Use case

|  |  |
| --- | --- |
| **Test Case** | Manage Order |
| **Actors** | Admin, Delivery Person, Merchant |
| **Overview** | |
| * All the above users have the manage order page, but only the admin users have access to all orders, other user only have shown their own orders. | |
|  | |
| **Preconditions** | |
| User is logged in | |
|  | |
| **Flow of events** | |
| * Orders page should show. * Check the active orders tab * It shows the active orders currently. * In the completed orders tab, it shows the completed orders. | |
| **Post of Condition** | After click on the Order it shows complete details about the order. |

# Online Food ordering system process

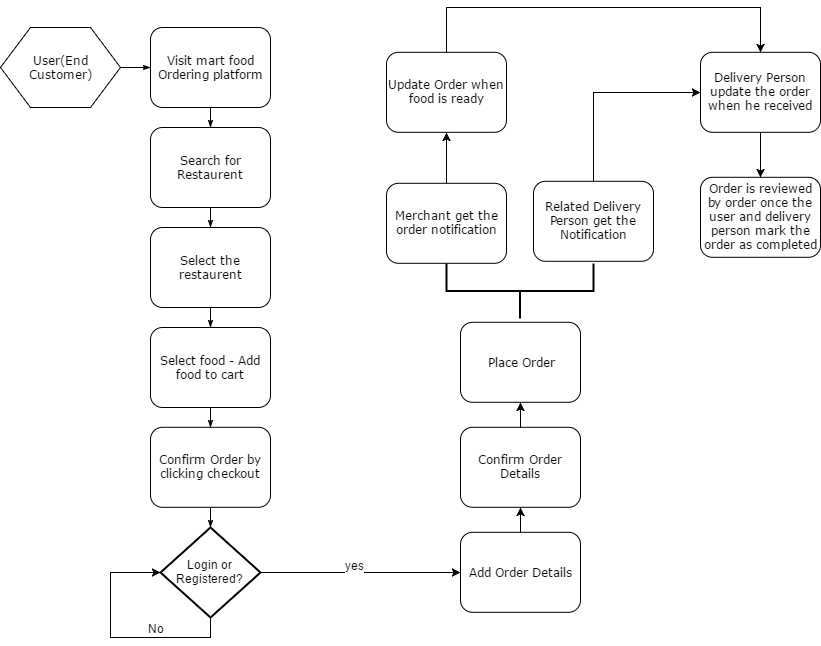


Figure 8 flowchart

# Entity Relationship Diagram

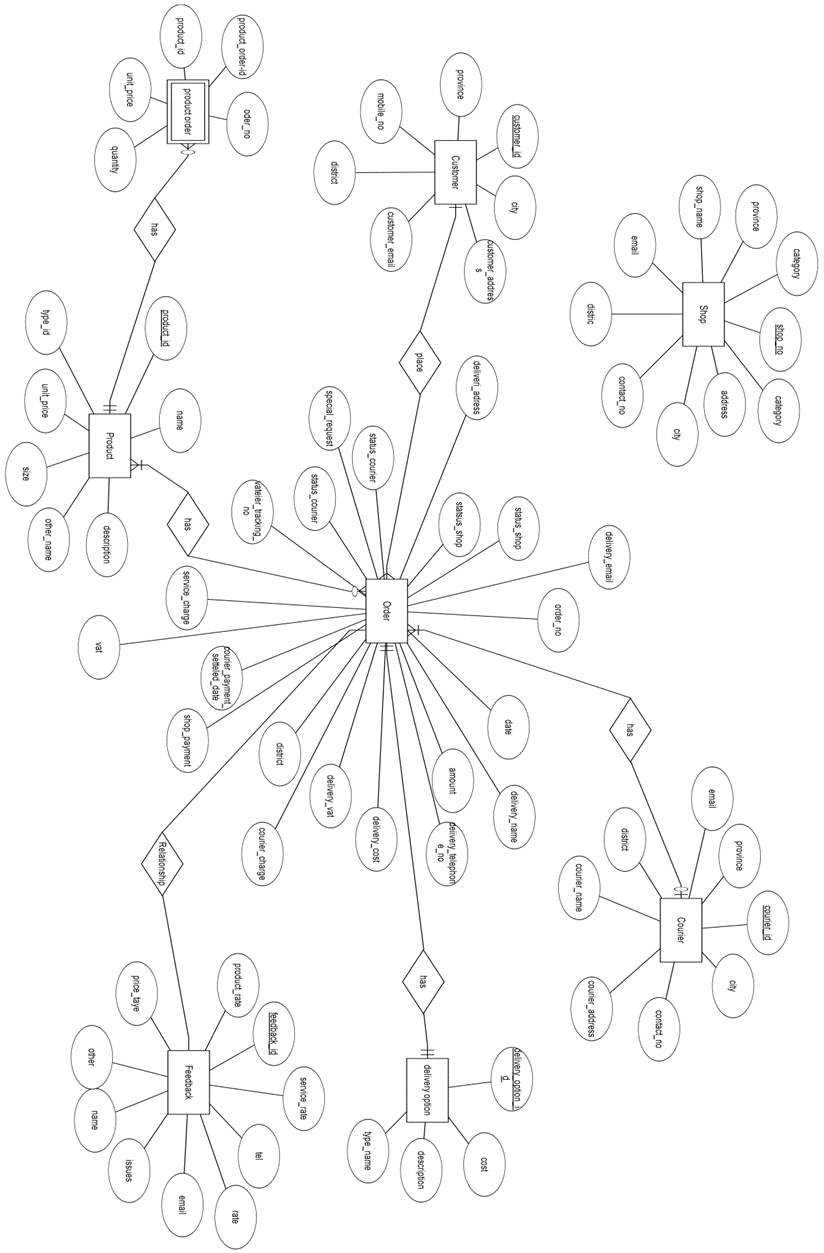


Figure 9 Entity Relationship Diagram

# Class Diagram

Due to the complexity of the class diagram to make it clear view of the classes below the classes are shown separately and finally the complete class diagram is placed.

Figure 10 Courier Class

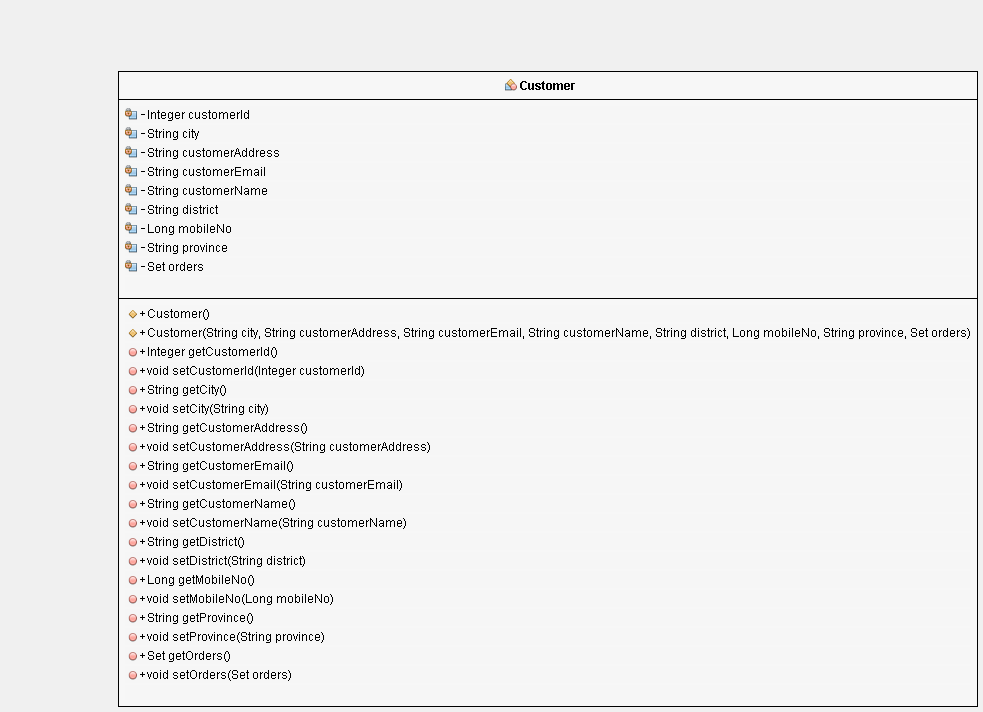


Figure 11Customer Class

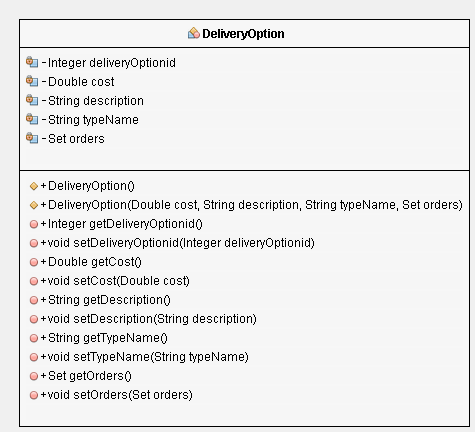


Figure 12 Delivery Option

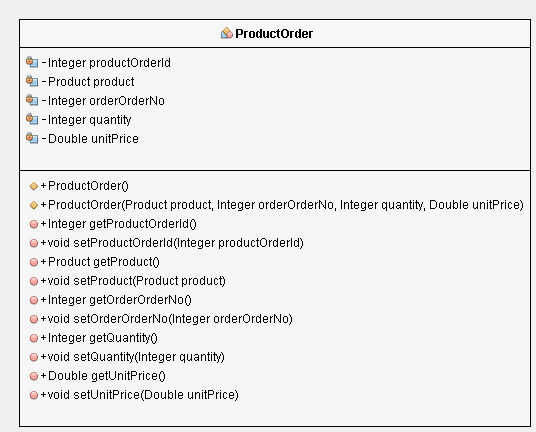


Figure 13 Place Order

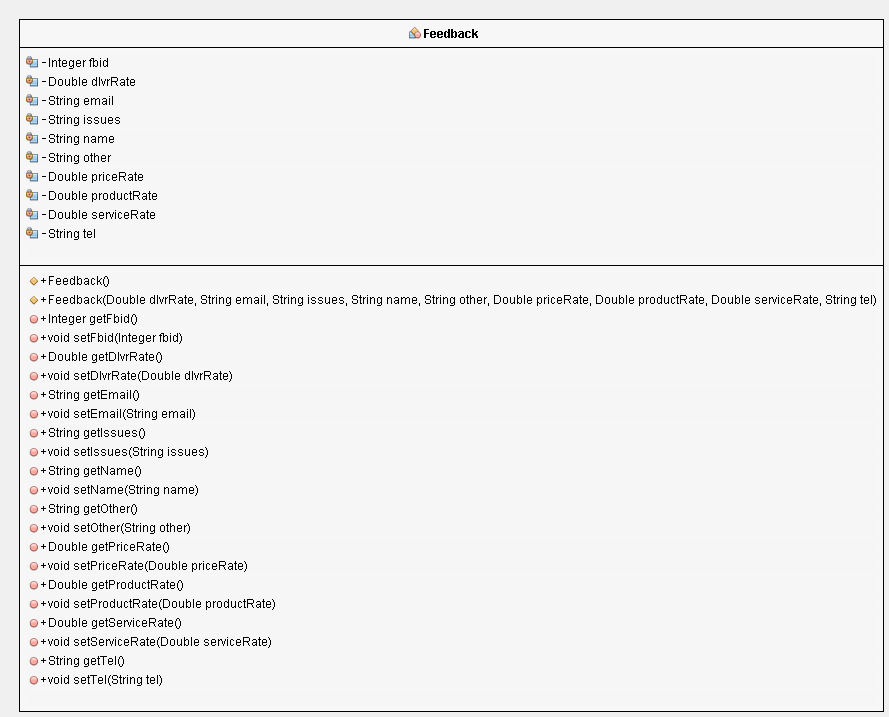


Figure 14 Feedback

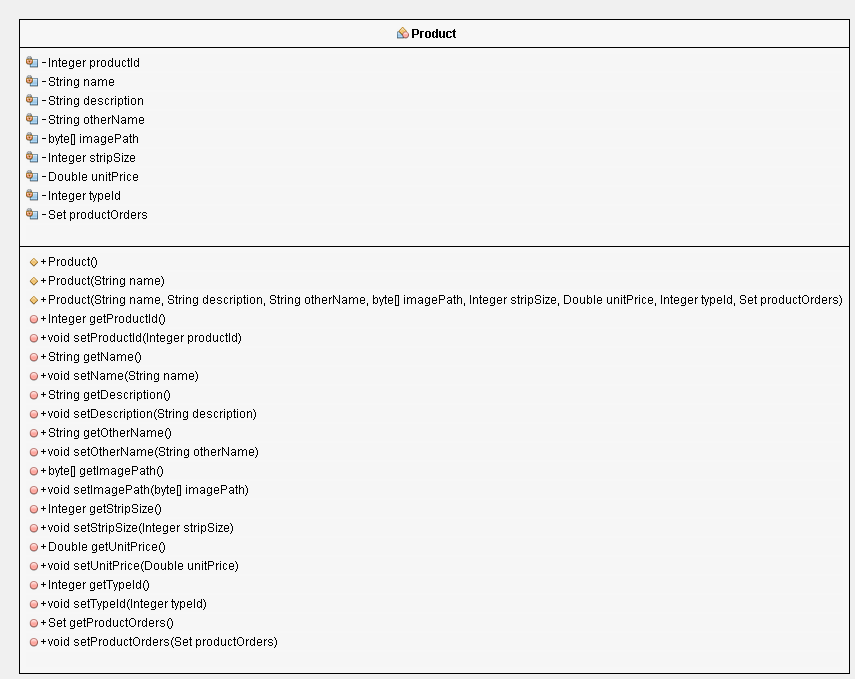


Figure 15 Product Class

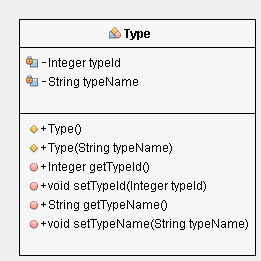


Figure 16 Type Class

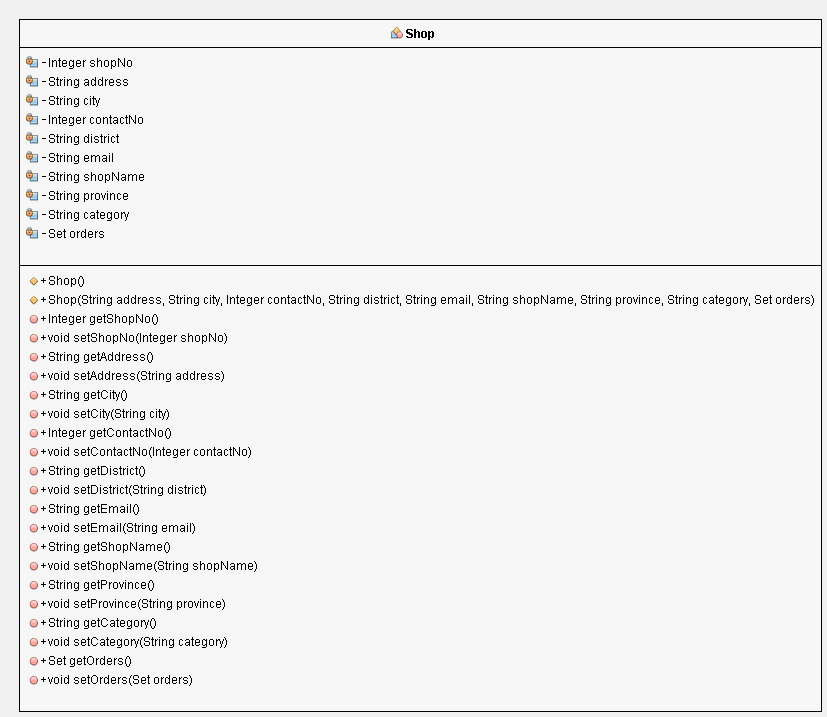


Figure 17 Shop Class

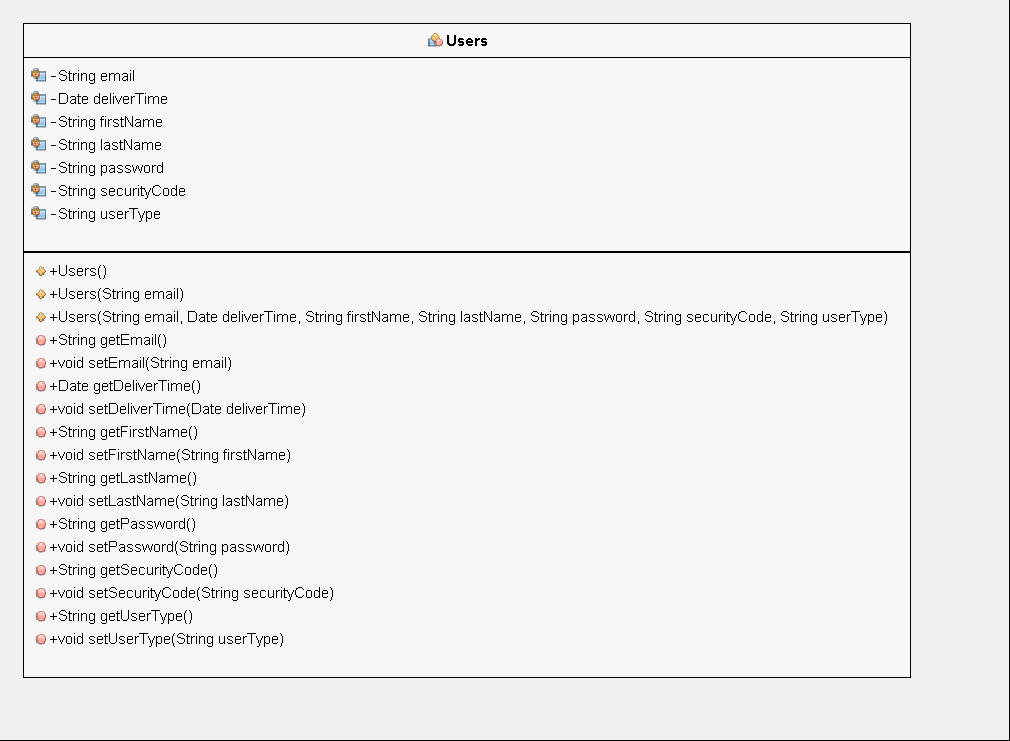


Figure 18 Users Class



Figure 19 Complete Class Diagram

# System Architecture

Mart Shopping platform has been implemented with MVC architecture.

MVC – Model – View- Controller is a design pattern for architecture of web applications. Many language frameworks are supporting this architecture it helps users to lose the dependency between each layer. It helped developer to develop the system parallelly. It helped developer to maintain the code clean and more organized manner. Since the software process model is prototyping, changes for the prototype was easily managed because of the MVC architecture since it reduced dependencies of the code.

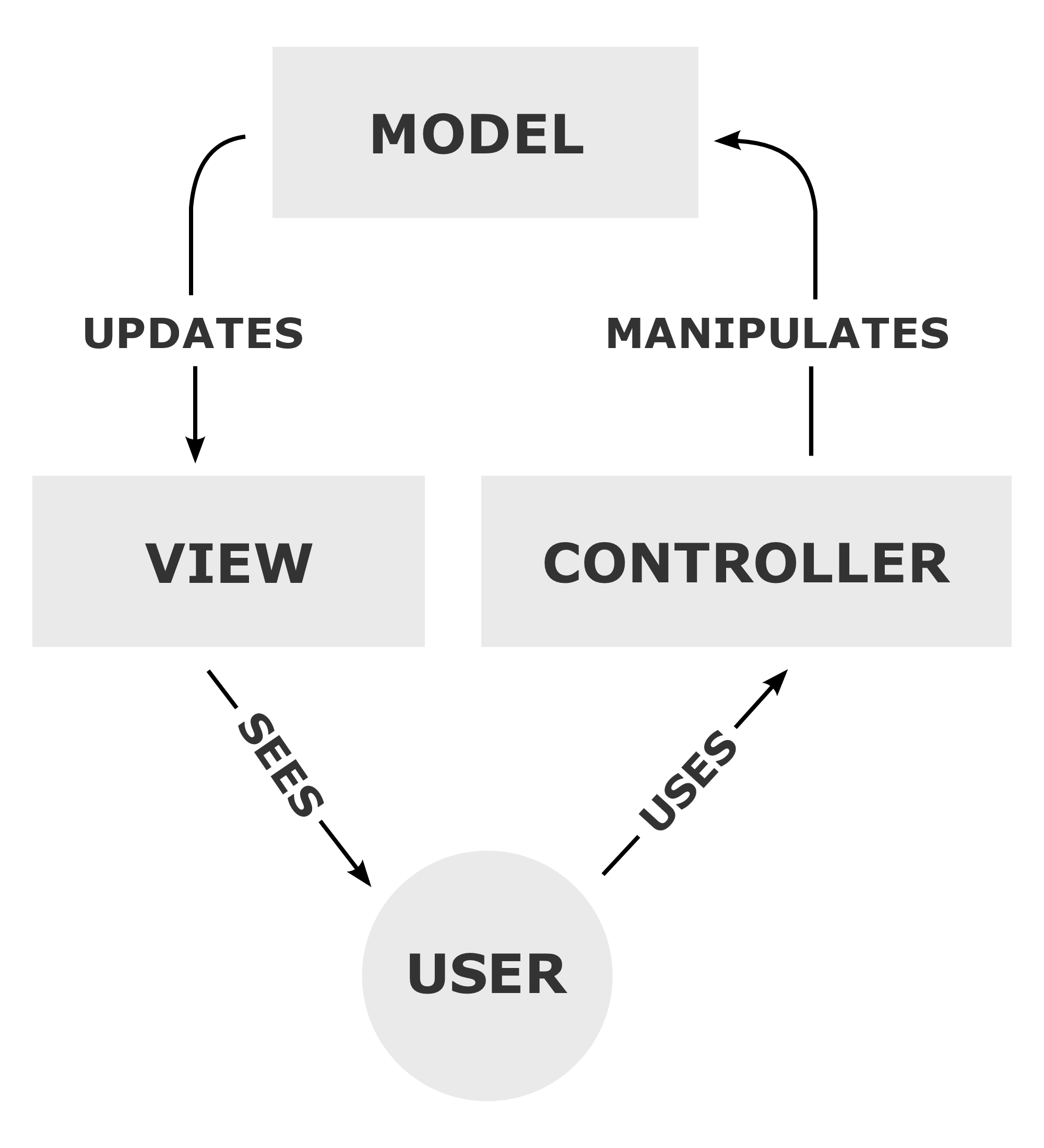


Figure 20 Model View Controller- From Wikipedia (MVC Architecture, 2017)

**Model**

Models are the parts of the application that contains the logic. Model objects retrieve and store data from database.

**View**

View is simply the User interface(UI) of the application. It shows the data loaded from the model.

**Controller**

Controllers are the components stay between the user and the model. It handles the user requests and responses to the user interaction through the data passed from model through the model.

# UI Design

User interface design is one of the most important process element in developing software, user interfaces should be attractive and user friendly. Otherwise there will be no use of the total system if the users are not interested with the user interfaces.

Mart system is developed using the prototyping methodology, UI design scratches also used in the requirement gathering phase also when the requirements are not clear.

Below are few of the digitally drawn User interface stretches.

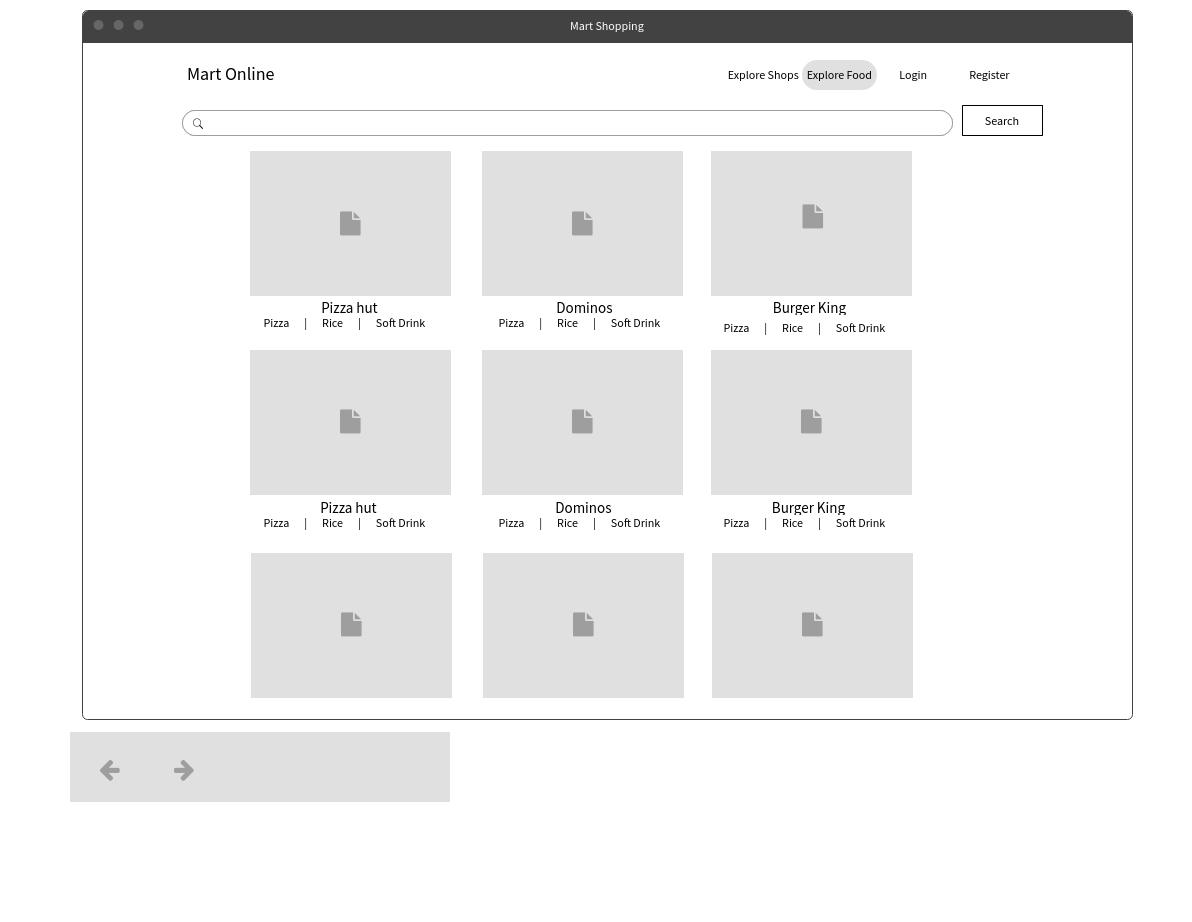


Figure 21 Scratch for Shops UI

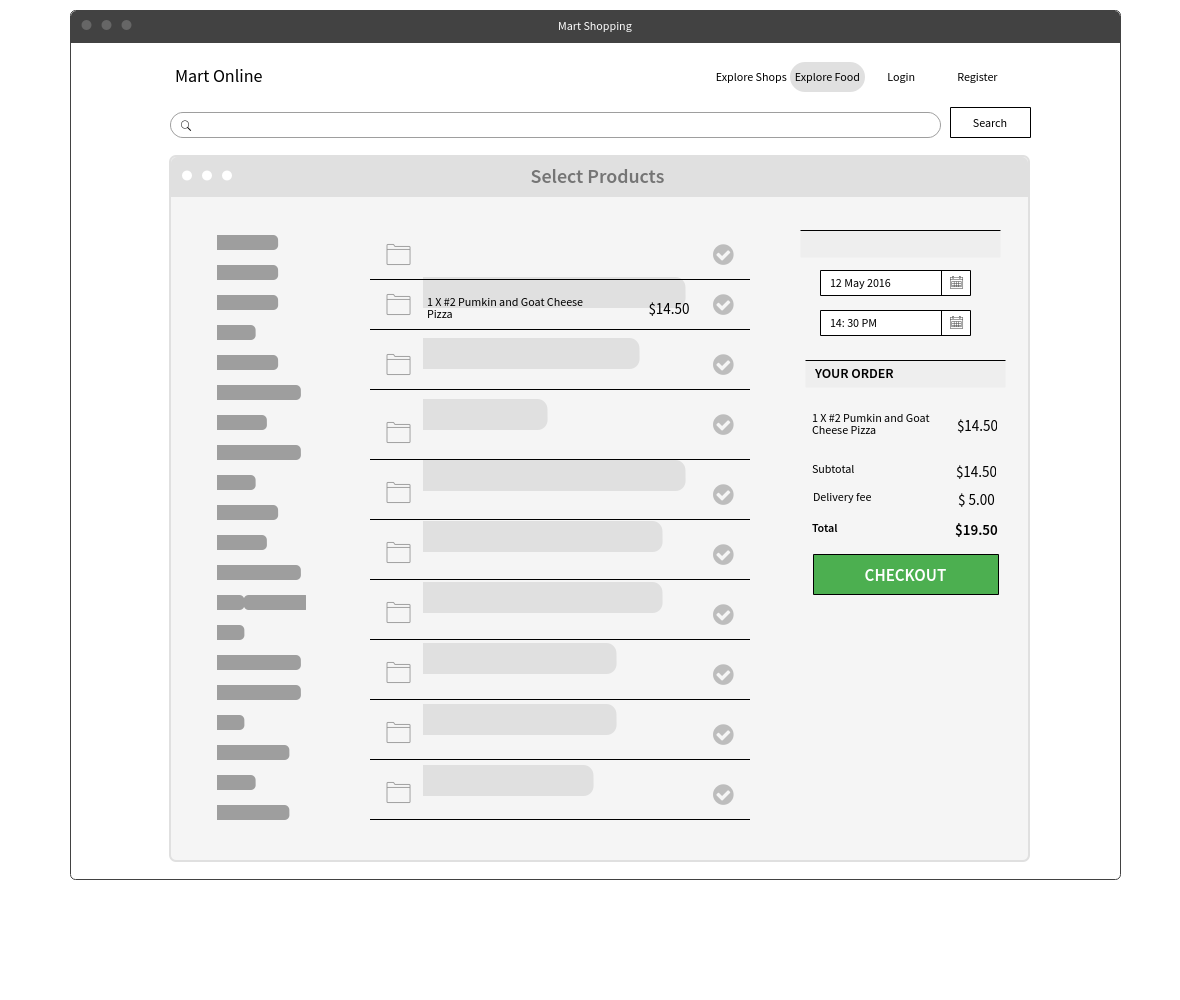


Figure 22 Product Add to cart UI Scratch

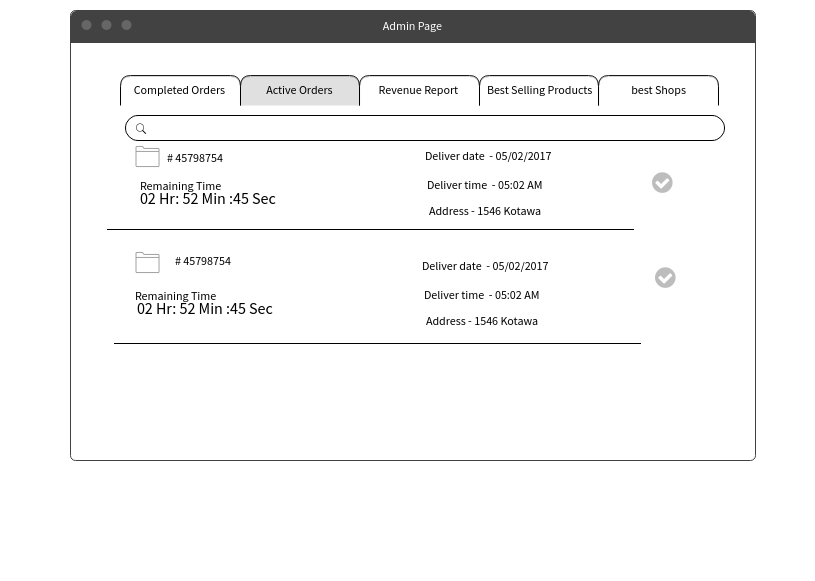
+

Figure 23 Ui for Admin - Active Orders

# Implementation