



## **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: 31<sup>st</sup> May 2017

First Supervisor: Ms. Theja Nadeeshani Perera Ilanperuma

Second Supervisor: Mr. Nishan Sembacuttiaratchy

## **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: 322/08/17

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

Extracts from University *Regulations* on Cheating, Plagiarism and Collusion

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

- (i) 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.
- (ii) Falsifying data in experimental results.
- (iii) 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.
- (iv) Bribery or attempted bribery of a person thought to have some influence on the candidate's assessment.
- (v) Collusion to present joint work as the work solely of one individual.
- (vi) Plagiarism, where the work or ideas of another are presented as the candidate's own.
- (vii) 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. " $e = mc^2$  (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**

As industries are fast growing and the busy life style, people are seeking for ways of purchasing products while remaining the cost effectiveness. The Small restaurant owners are always looking to increase food sales. Ordering food over the phone can be considered as a solution for the above matter. Phone Ordering of food also has many drawbacks like need for telephone operators, waiting till operator answer the phone, pronounce problems. Repeating credit card numbers at the phone, number many numbers. Mistakes when taking down order quantities, address etc.

As a solution for the above-mentioned problems author suggests a online shopping platform, which is free to register, so the small local restaurant owners can register and start selling by displaying the latest menu available. End customers can easily order food by selecting the food they prefer. Platform charges certain percentage as the service charge from order at the end of each order. Small restaurant owners can reach large amount of audience through the online platform.

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 an 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 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 .....	35
Abstract.....	36
Acknowledgement.....	37
Introduction.....	39
Background .....	40
Mission .....	42
Background of the problem.....	43
Solution .....	44
Existing web applications.....	45
Feasibility Study .....	46
Requirement Gathering and Analysis .....	47
System Design.....	48
Online Food ordering system process .....	56
Entity Relationship Diagram .....	57
Class Diagram .....	58
System Architecture.....	59
UI Design Scratches .....	60
Implementation .....	63
Data Dictionary Mart .....	71
Database .....	72
References .....	97

## **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 a server remote location your desktop it is an application hosted on Remote server. Software uses its own interfaces, designed to provide a special characteristic 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 provides 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'.

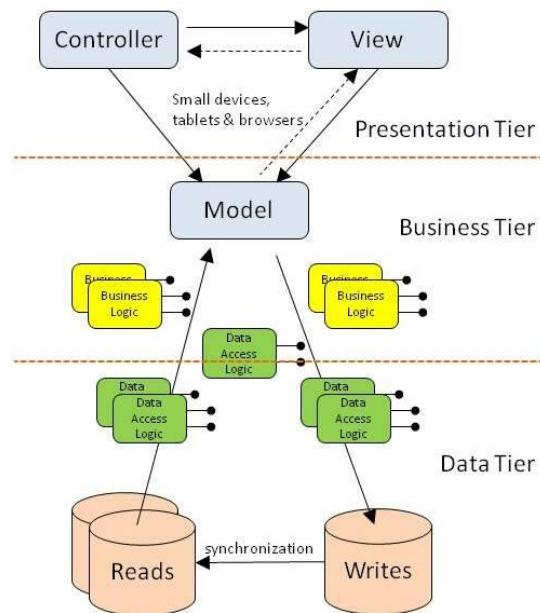
To overcome from these problems Author suggested an online platform for Mart and where gives the small restaurant and shop owners to maintain their profiles by partnering with ' Mart' and after registering with systems customers can order and their favourite food products and grocery items from their 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 2% 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 architectures. 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.

## **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

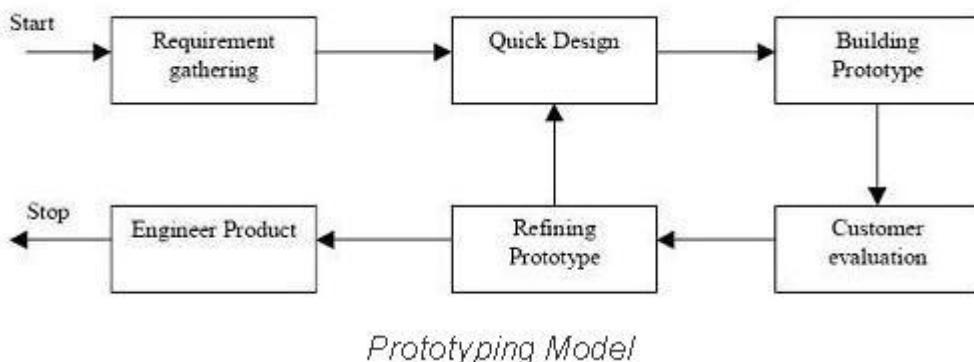
- Application database should be secured.
- All data should have a backup.
- User experience should be high.

## System Design

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 define 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.

Prototyping model consists of few steps



*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

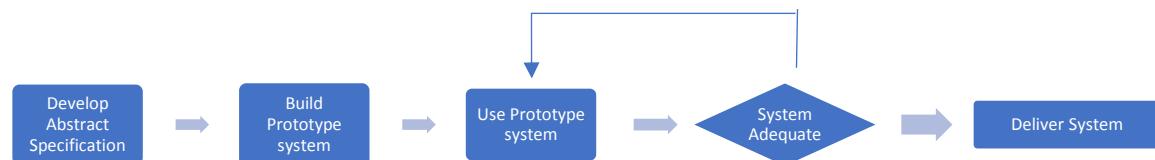
The final step of the process is to revise the prototype based on the feedback of the business analyst or the testers.

(study.com 2017)

There are several uses of prototype model, one of the most important purpose is to illustrate the inputs the data formats, messages, reports and the interactive dialogs to the customer. It allows to gain better understanding about the customer's needs.

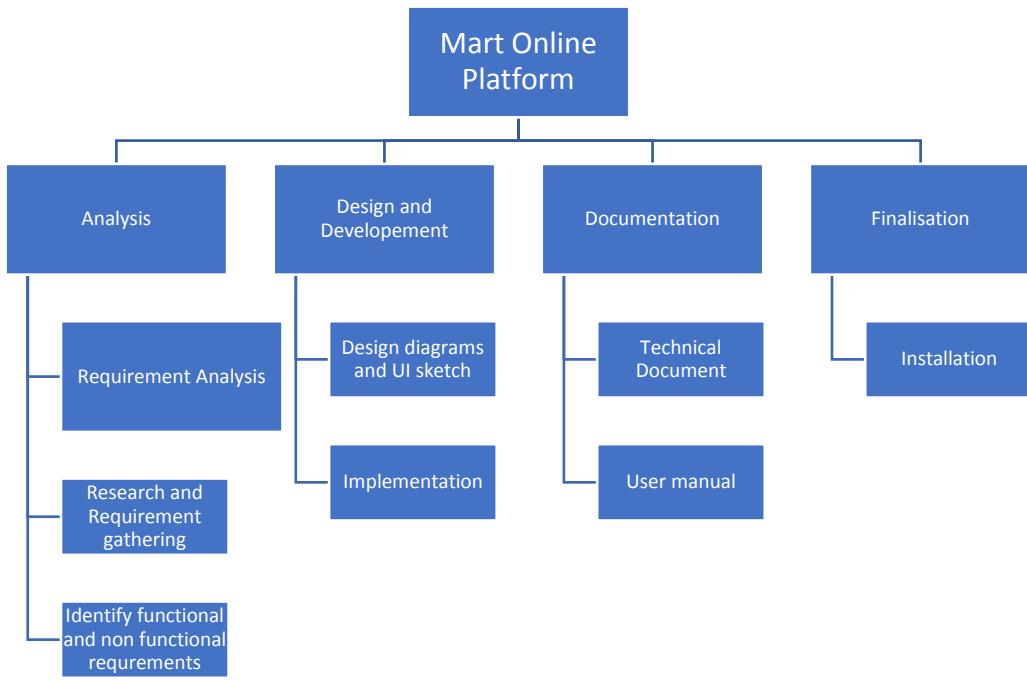
There are several types of prototyping also available. Developer used Evolutionary Prototyping method. First version of the Mart developed with minimal functionality. Only the well understood requirements are included in the first prototype.

(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*

## System Design

Below use case diagram provides the functional requirements identified during the Requirements Analysis Stage. Up to now System developer has identified 3 major user roles that in the system. They are,

1. Mart Admin User
2. Shop Owner(Merchant)
3. Customer
4. Courier Agent

Mart Admin user is the user who involve and operate the system with lot of privileges, Mart Admin user is an Employee of a Mart Organization, so that user has access to almost everything.

Shop Owner is user who partner with Mart system and publish their products on the Mart platform.

Customer is the user who sign up with the system and order products through the Mart platform by going through different products listed by the Shop owners and make payments for them

Courier user agent is the user who pick the delivery order to the customer when a customer place an order through the system and update the payment status, and order delivery status.

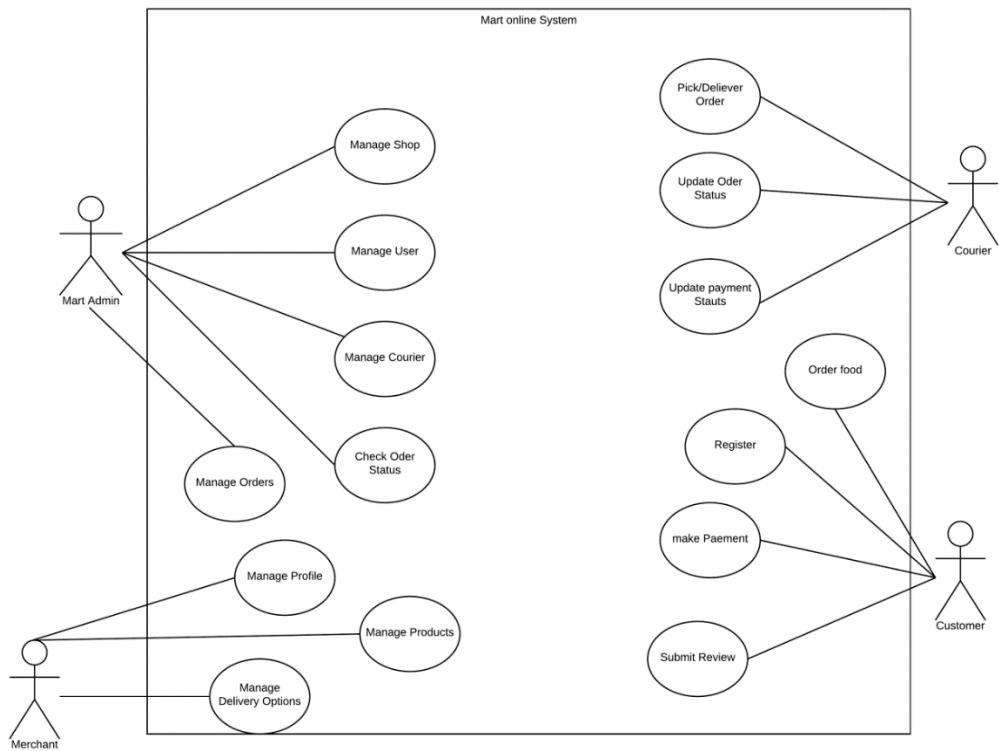


Figure 3 Use case Diagram

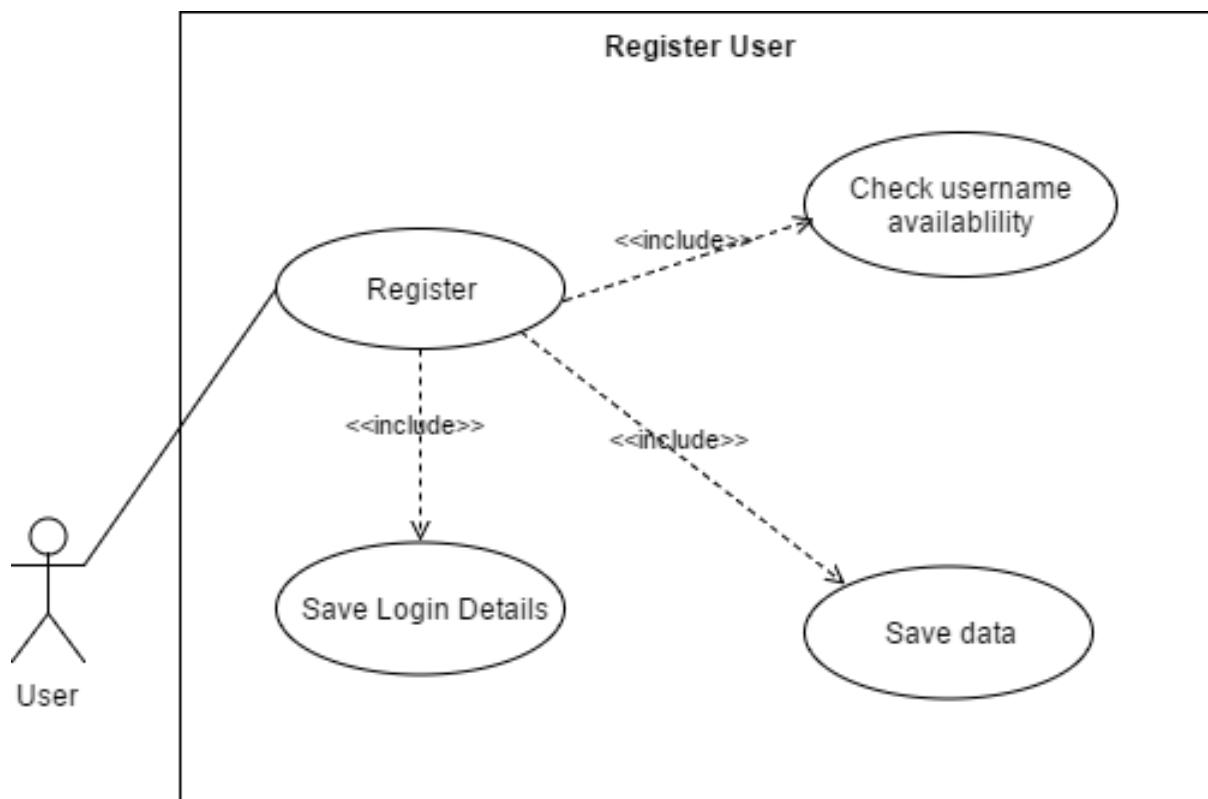


Figure 4 Register User Use Case

<b>Test Case</b>	Register
<b>Actors</b>	Admin, End Customer, Merchant, Delivery Person
<b>Overview</b>	
<ul style="list-style-type: none"> <li>• All users for the system are registering through this use case.</li> </ul>	
<b>Preconditions</b>	
User is registering for the first time.	
<b>Flow of events</b>	
<ul style="list-style-type: none"> <li>• Check database for email</li> <li>• Send invitation email</li> <li>• Save on detail on separate micro service account</li> <li>• Save data for role in database.</li> </ul>	
<b>Post of Condition</b>	The user is directed to the respective home page according to the user role.

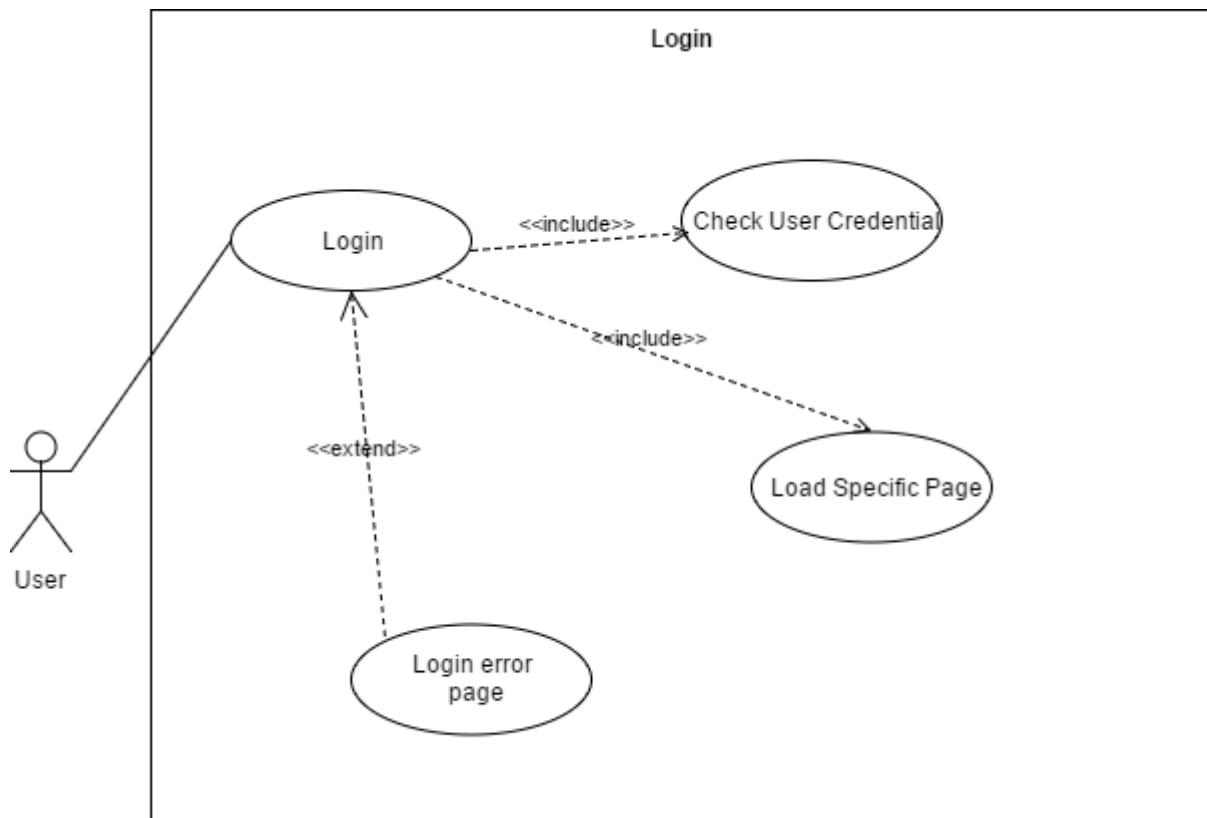


Figure 5 Login Usecase

<b>Test Case</b>	Login
<b>Actors</b>	Admin, End Customer, Merchant, Delivery Person
<b>Overview</b>	
<ul style="list-style-type: none"> <li>• All users for the system are login through this use module.</li> </ul>	
<b>Preconditions</b>	
User is already registered	
<b>Flow of events</b>	
<ul style="list-style-type: none"> <li>• Check database for credentials according to user input</li> <li>• Show error message if not matching.</li> <li>• Save on detail on separate micro service account</li> <li>• Save data for role in database.</li> </ul>	
<b>Post of Condition</b>	The user is directed to the respective home page according to the user role.

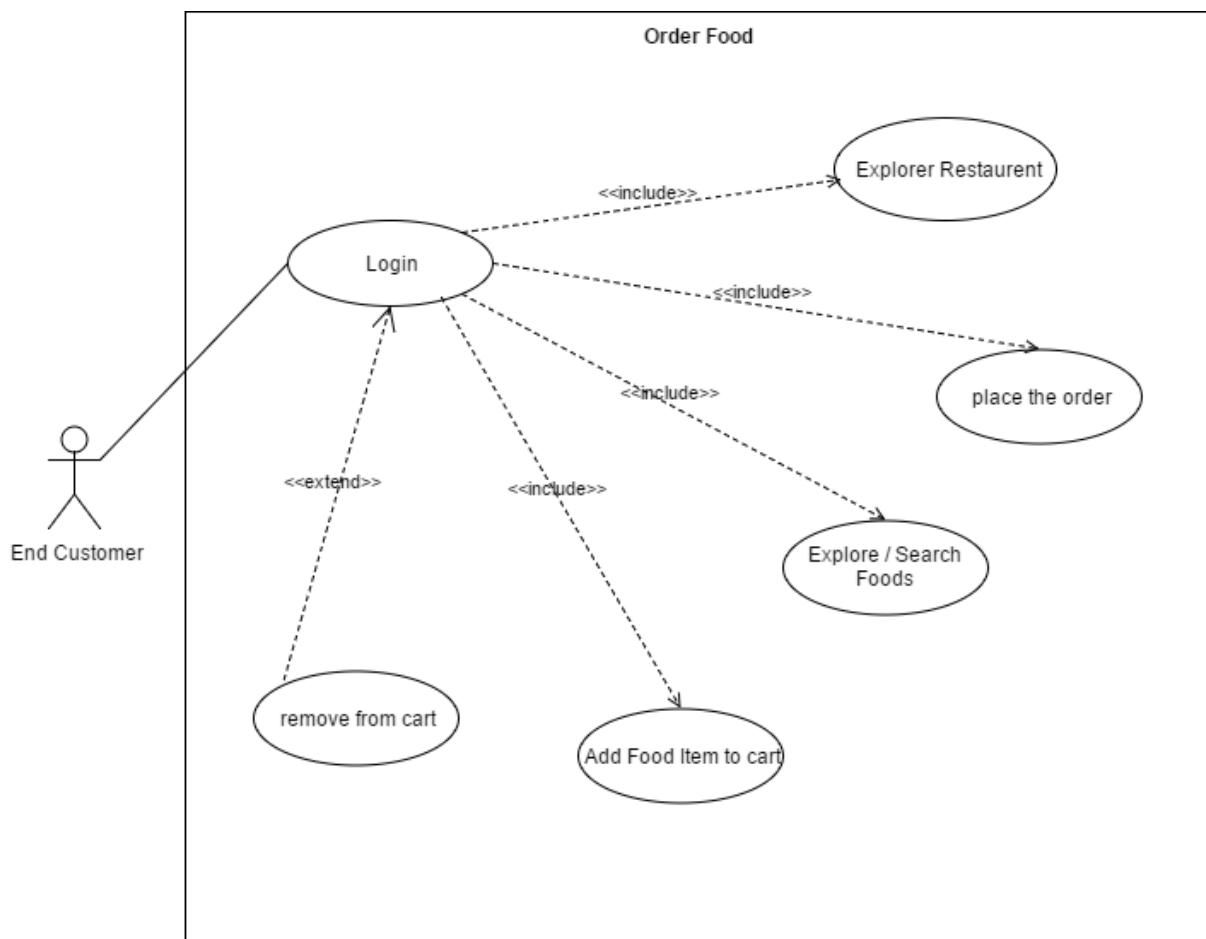
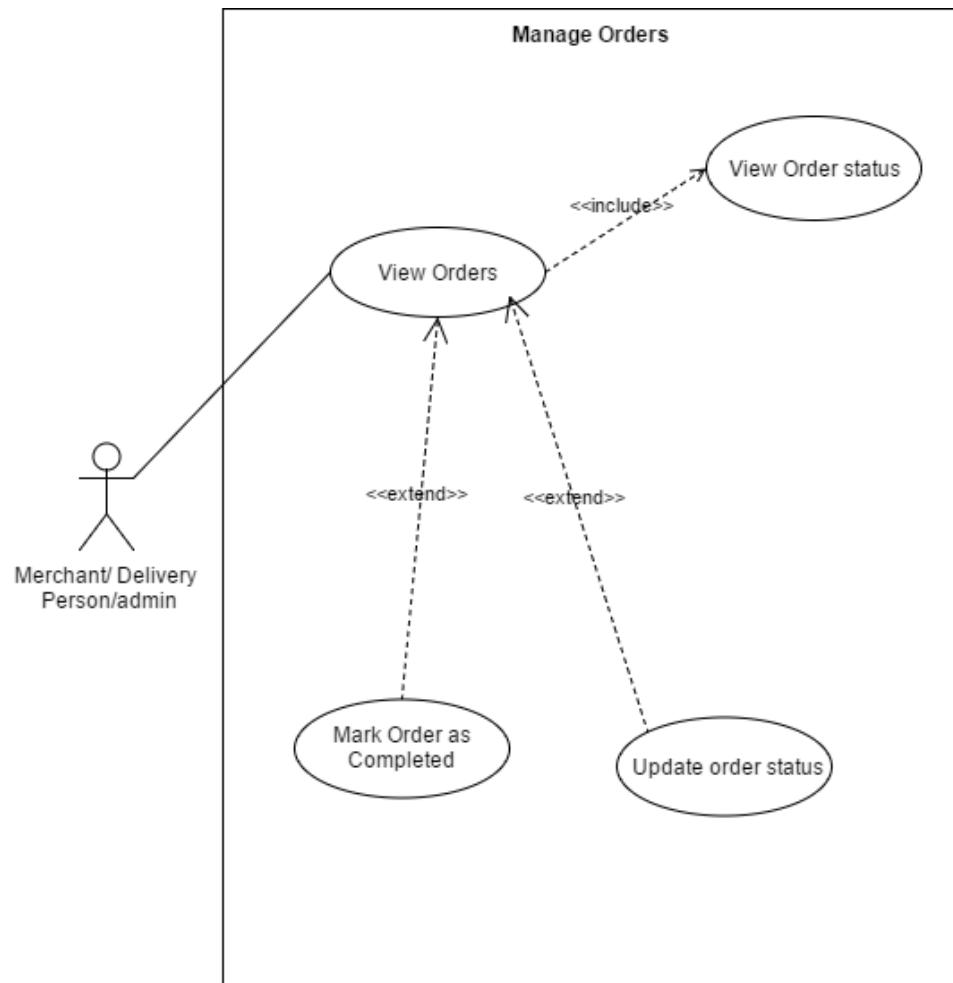


Figure 6 Order Food use case

<b>Test Case</b>	Order food
<b>Actors</b>	Admin, End Customer
<b>Overview</b>	
<ul style="list-style-type: none"> <li>• Admin users and end customers of the system can order food through this module.</li> </ul>	
<b>Preconditions</b>	
User is logged in	
<b>Flow of events</b>	
<ul style="list-style-type: none"> <li>• Search for a restaurant.</li> <li>• Select and go inside the restaurant page / alternatively user can directly search for the food</li> <li>• Select food and add to cart button</li> <li>• After adding the food click on the checkout button on the cart summary view.</li> <li>• User direct to Checkout page.</li> </ul>	
<b>Post of Condition</b>	The user is directed to the respective home page according to the user role.



*Figure 7 Manage Orders Use case*

<b>Test Case</b>	Manage Order
<b>Actors</b>	Admin, Delivery Person, Merchant
<b>Overview</b>	
<ul style="list-style-type: none"> <li>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.</li> </ul>	
<b>Preconditions</b>	
User is logged in	
<b>Flow of events</b>	
<ul style="list-style-type: none"> <li>Orders page should show.</li> <li>Check the active orders tab</li> <li>It shows the active orders currently.</li> <li>In the completed orders tab, it shows the completed orders.</li> </ul>	
<b>Post of Condition</b>	After click on the Order it shows complete details about the order.

## Online Food ordering system process

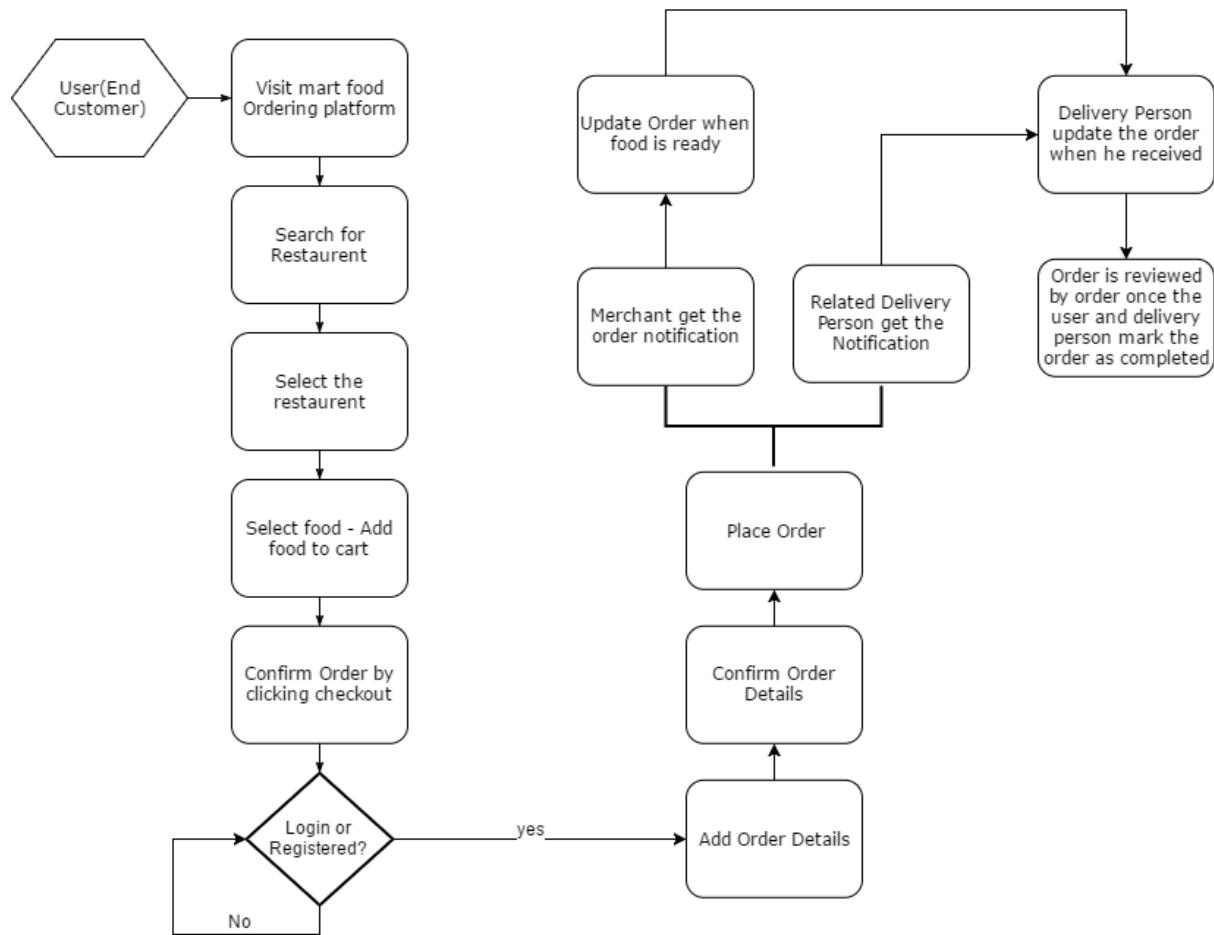


Figure 8 flowchart

Above flow chart represents the business process of mart shopping platform including the off the system process related to the business of mart shopping platform.

## Entity Relationship Diagram

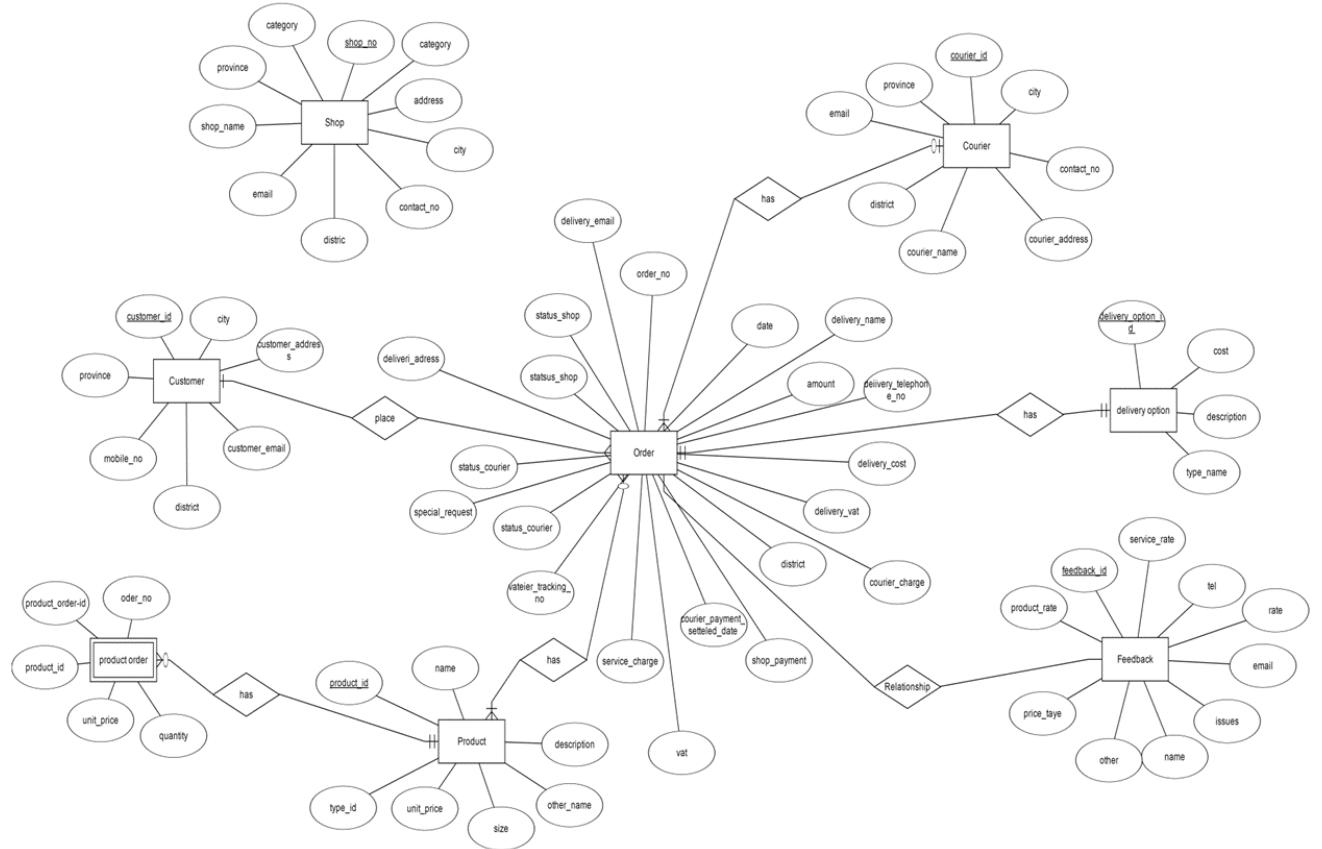
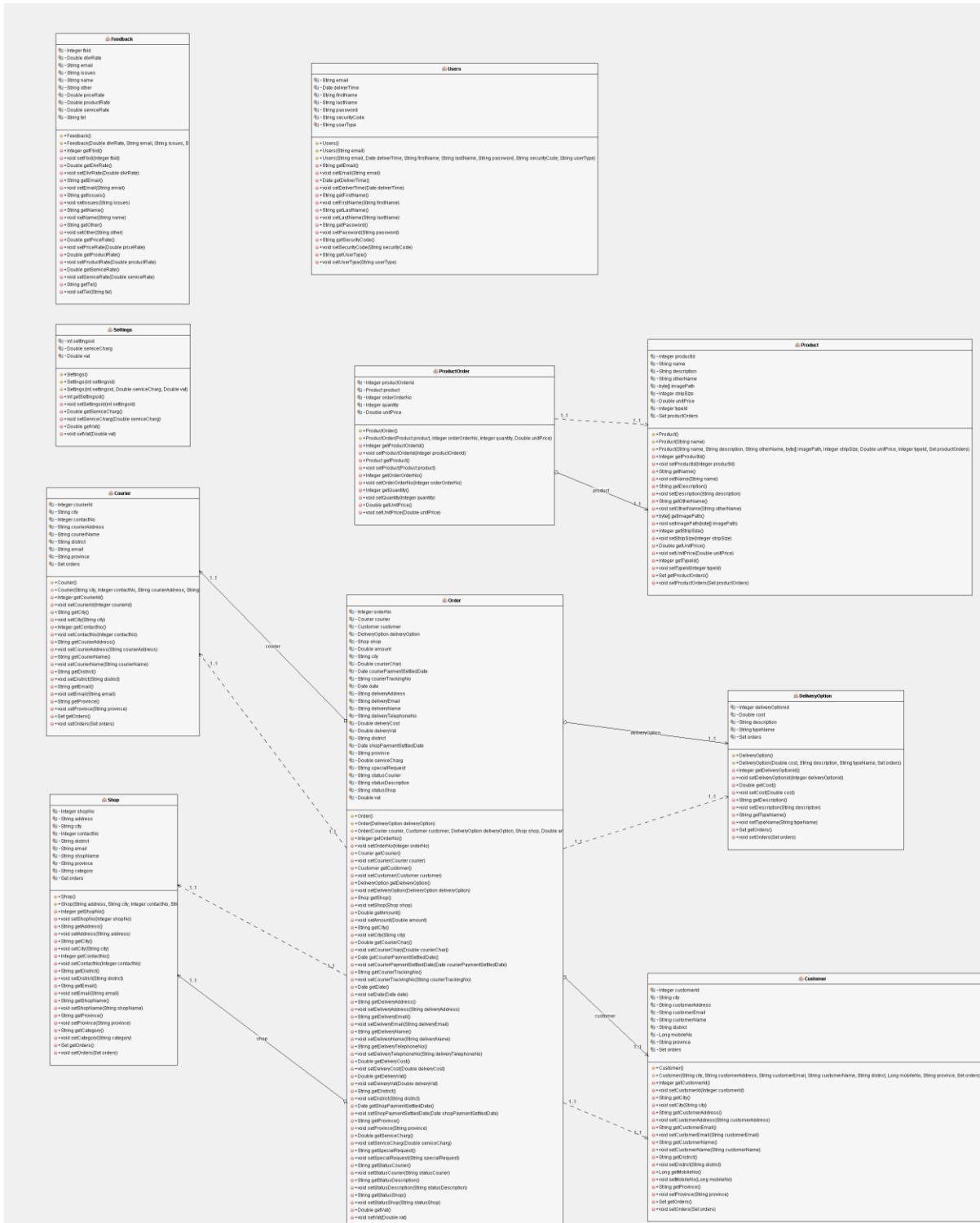


Figure 9 Entity Relationship Diagram

Entity Relationship diagram of Mart platform represents the relationship between restaurant owners, end customers, delivery persons(couriers), products, orders and feedbacks.

## 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 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.

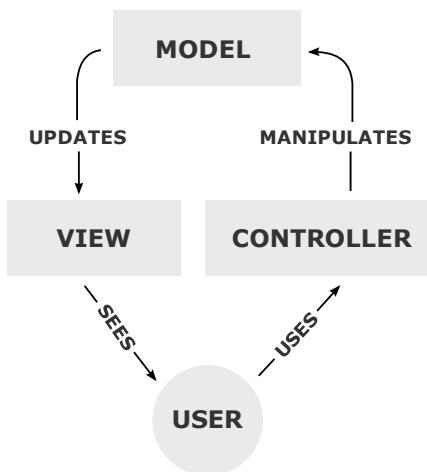


Figure 11 Model View Controller- From Wikipedia (MVC Architecture, 2017)

### 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 Scratches

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.

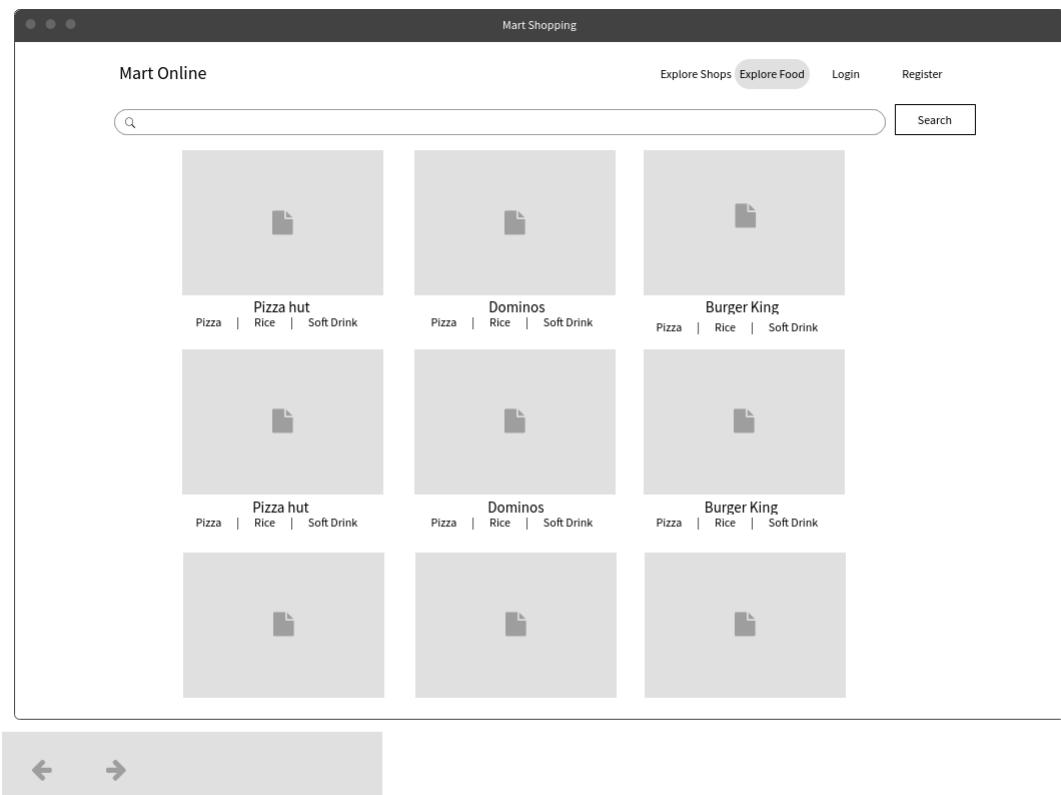
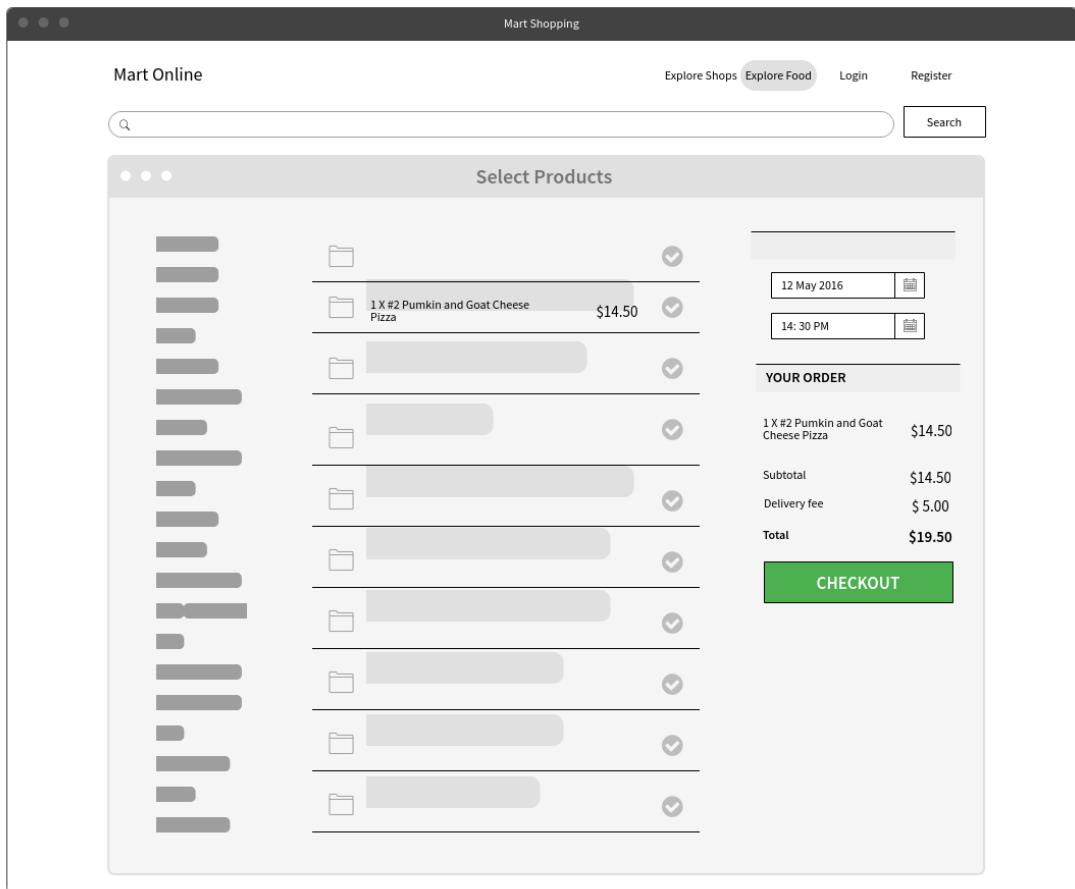


Figure 12 Scratch for Shops UI



*Figure 13 Product Add to cart UI Scratch*

+

The screenshot shows a web-based admin interface titled "Admin Page". At the top, there is a navigation bar with five tabs: "Completed Orders", "Active Orders" (which is highlighted in grey), "Revenue Report", "Best Selling Products", and "best Shops". Below the tabs is a search bar containing a magnifying glass icon and the placeholder text "# 45798754".

The main content area displays two active delivery orders:

Order ID	Remaining Time	Deliver date	Deliver time	Status
# 45798754	02 Hr: 52 Min :45 Sec	05/02/2017	05:02 AM	✓
# 45798754	02 Hr: 52 Min :45 Sec	05/02/2017	05:02 AM	✓

Each order row includes a small folder icon next to the order ID, the remaining delivery time, the delivery date, the delivery time, and a checked circular status indicator.

Figure 14 Ui for Admin - Active Orders

## Implementation

Implementation Architecture of the system

Layers are logical separation of code, Tiers are the physical separation of the program according to the deployment of the application. (Oellermann, 2001) Mart application code is implemented with logically separated 3 layers that made easier to maintain the code. These 3 logical layers are implemented with MVC (Model View Controller) Pattern. In the 3 layers code is separated according to the implementation of code. Model is the object that holding data, which are represented from the view, Controller is the layer which handles the request calls from the view layer and send a new view according to the model layer of the application.

Application deployed on a single machine as a server, since the MySQL database also installed in the same server instance. Since that application has no physical separation. Future enhancements of the application will be installed in separate physical machines, data access layer, business logic layer and the presentation layer.

When it comes to development stage, it is necessary to select the most suitable technologies and the architecture, system architecture's main responsibility to select the most suitable technology considering about the capacity of the team, when it comes to a team development and should consider about the affordability of the technology. Recently what author experience was acquiring of products that are closed source and freely available now common thing, so when selecting the appropriate technology, author recommend selecting open source products.

Developer selected Java as the programming language and the MySQL used as the database language. The main reason for choosing the above technologies are they are free. MySQL is also open source and the Open JDK is also open source software. Open Source Spring framework is used as the framework for developing the application. Spring boot framework version 1.5.4

Hibernate 5.2.10 used as the object relational mapping tool

Using Object relational mapping tool author released from lot of data access problems and reduced the SQL code must write multiple times. It also generates boilerplate code for basic CRUD operations.

```
Session session = sessionFactory.openSession();
session.save(emp); // emp is the model object passed to the method
```

It is necessary to select proper IDE for the development task of the application. With experience of the developer, developer selected IntelliJ Idea 2017.04 community edition for the development of the spring boot application, and for the front end designing developer used Jet Brains Web Strom 2017.04 which really helped the developer suggesting the Java Script suggestions at the code development stage. Developer also used Heidi SQL 9.4 for physical implementation of the database.

Developer used Key Cloak for user authentication and role management with spring security. Key Cloak Microservice is an Open Source Identity and access management solution for modern applications. This application is developed by Red Hat. Key Cloak adds authentication to applications and secure services with minimum effort.

There are two types of servers available, they are

1. Application Server
2. Web Server

Apache Tomcat is an Open Source web server that is developed by apache software foundation. There is a built-in web container called Catalina in tomcat directory. It loads all http related request and has privilege to initiate the GET and POST method's object.

## **Client Side Technology**

### **Materialize CSS**

Materialize CSS is the client side technology used to develop the user interface of the Mart Online Shopping. Materialize is a UI Component library created with CSS, JavaScript and HTML. Materialize UI components helps in building attractive, consistent and functional web pages and web apps.

Material Design, is designed and created by Google, Goal of developing the material design is to create unified user experience across any platform.

Materialize CSS has inbuilt responsive designing. Materialize CSS has own components like buttons, checkboxes, and text areas which are designed according to material design concepts.

Materialize CSS also contains special user interface components, that are unique to material design.

Screen toasts, cards, navigation bars etc. Material CSS is free to use.

### **jQuery**

jQuery is fast and concise Java Script library created by John Resig in 2006. jQuery simplifies HTML document traversing, event handling, animating and ajax interactions for client side applications.

jQuery simplifies various tasks by writing less code.

DOM manipulation, Ajax Support, Event handling, Animations are some of the features provided by jQuery.

## **Object Oriented Principles**

For the development of mart Object Oriented principles heavily used by the developer since the application was developed in Java language and using the Spring boot framework.

## Abstraction

Abstraction is simply the art of focusing on the essential and ignoring the rest. With abstraction, it allowed developer to lose coupling inside the code. Java interfaces are used to implement the abstraction in Object oriented programming. Below example code represents how the application was used in the Mart Online System.

```
package com.zambrone.service;
import com.zambrone.entity.Shop;
import org.springframework.dao.DataAccessException;
import java.util.List;

/**
 * Created by Chamith on 05/05/2017.
 */
public interface ShopService {

    /**
     *
     * @param shop
     * @throws DataAccessException
     */
    void registerNewShop(Shop shop) throws DataAccessException;

    /**
     *
     * @return @throws DataAccessException
     */
    List<Shop> getAllShop() throws DataAccessException;

    /**
     *
     * @param name
     * @return
     * @throws DataAccessException
     */
    Shop getShopByName(String name) throws DataAccessException;

    /**
     *
     * @param id
     * @return
     * @throws DataAccessException
     */
    Shop getShopByID(Integer id) throws DataAccessException;

    /**
     *
     * @param phone
     * @return
     * @throws DataAccessException
     */
    Shop getShopByPhone(String phone) throws DataAccessException;
```

```

    /**
     *
     * @param shop
     * @throws DataAccessException
     */
    void updateShop(Shop shop) throws DataAccessException;

    /**
     *
     * @param id
     * @throws DataAccessException
     */
    void removeShop(Integer id) throws DataAccessException;

    /**
     *@return allshops
     * @param category
     * @throws DataAccessException
     */
    List<Shop> getShopByCategory(String category) throws DataAccessException;
}

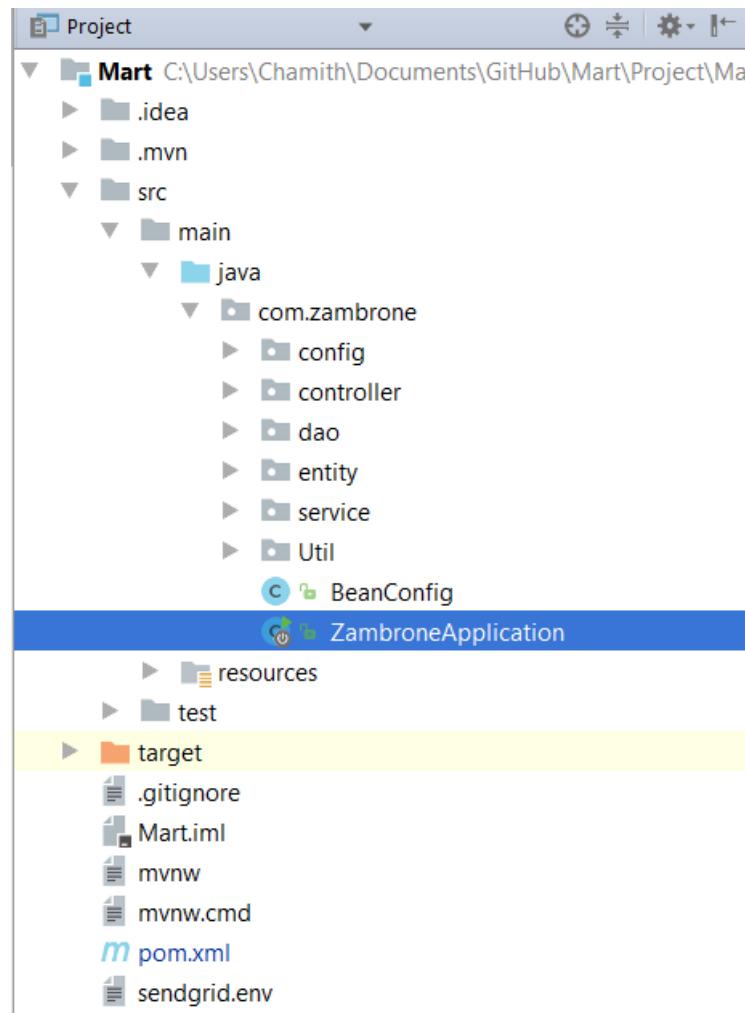
```

Since java interfaces are already public there is no need of adding the access modifier “public”. The above code sample shows how the shop what are the most important methods that should implemented in the Shop Service, interface methods do not have a body to implement, interfaces provide full abstraction in java.

Using interfaces, it provides loose coupling between the classes, it also uses encapsulation, since the inner changes of an implemented class of and interface method changes are not visible to the outside classes. Spring frameworks dependency injection also provides the encapsulation to the developer.

## Design Patterns

Development of the Mart platform done using the Model View Controller Architectural design pattern was used for development, increased code reusability, less coupling between the layers, it also leads to better and easier code maintenance since it provides code separation for model, view and controllers.



Sample code for controller class used for Customer request and respons controlling in Mart platform is mentioned below.

```
package com.zambrone.controller;

import amazon.s3.AmazonS3Template;
import com.amazonaws.services.s3.model.CannedAccessControlList;
import com.amazonaws.services.s3.model.ObjectMetadata;
import com.amazonaws.services.s3.model.PutObjectRequest;
import com.zambrone.config.JsonResponse;
import com.zambrone.entity.Customer;
import com.zambrone.service.CustomerService;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.beans.factory.annotation.Value;
import org.springframework.http.HttpHeaders;
import org.springframework.http.HttpStatus;
import org.springframework.http.ResponseEntity;
import org.springframework.validation.BindingResult;
import org.springframework.web.bind.annotation.*;
import org.springframework.web.multipart.MultipartFile;
import org.springframework.web.util.UriComponentsBuilder;

import javax.validation.Valid;
import java.util.List;
import java.util.UUID;
```

```

/**
 * Created by Chamith on 04/07/2017.
 */
@RestController
@RequestMapping("/customer")
public class CustomerController {

    public static final Logger logger =
LoggerFactory.getLogger(CustomerController.class);

    @Autowired
    CustomerService customerService;

    private AmazonS3Template amazonS3Template;
    private String bucketName;

    //setup/initialise the amazon
    @Autowired
    public CustomerController(AmazonS3Template amazonS3Template,
    @Value("${amazon.s3.default-bucket}") String bucketName) {
        System.out.println("template name " + amazonS3Template);
        System.out.println("bucket name " + bucketName);
        this.amazonS3Template = amazonS3Template;
        this.bucketName = bucketName;
    }

    //save new customer
    @RequestMapping(value = "add", method = RequestMethod.POST)
    public @ResponseBody
    JsonResponse addCustomer(@Valid @ModelAttribute(value = "customer") Customer
    customer, @RequestParam(value = "proimage") MultipartFile file, BindingResult
    result) {

        System.out.println("Content file part " + file.getName());
        JsonResponse res = new JsonResponse();

        String fileid = UUID.randomUUID().toString();
        if (!result.hasErrors()) {
            if (!file.isEmpty()) {
                try {
                    ObjectMetadata objectMetadata = new ObjectMetadata();
                    objectMetadata.setContentType(file.getContentType());

                    //save image to profile image s3 bucket
                    amazonS3Template.getAmazonS3Client().putObject(new
                    PutObjectRequest(bucketName, fileid, file.getInputStream(), objectMetadata)
                    .withCannedAcl(CannedAccessControlList.PublicRead));
                    customer.setImagePath("https://s3-ap-southeast-
1.amazonaws.com/martonline/" + fileid);
                    customerService.registerNewCustomer(customer);
                    res.setStatus("SUCCESS");
                    res.setResult("successfully uploaded");
                    System.out.println("Customer " + customer);
                }
            }
        } catch (Exception e) {

            res.setStatus("FAIL");
            res.setResult("Data Not Submitted");
            System.out.println("Submission errors" +
result.getFieldError());
        }
    } else {

        customerService.registerNewCustomer(customer);
        System.out.println("Customer " + customer);
    }
}

```

```

        res.setStatus("SUCCESS");
    }
} else {
    res.setStatus("FAIL");
    res.setResult("Data Not Submitted");
    System.out.println("Submission errors" + result.getFieldError());
}

return res;
}

// -----Retrieve All Customers-----
-----

@RequestMapping(value = "/customer/", method = RequestMethod.GET)
public ResponseEntity<List<Customer>> listAllCustomers() {
    List<Customer> customers = customerService.getAllCustomer();
    if (customers.isEmpty()) {
        return new ResponseEntity(HttpStatus.NO_CONTENT);
        // You many decide to return HttpStatus.NOT_FOUND
    }
    return new ResponseEntity<List<Customer>>(customers, HttpStatus.OK);
}

// -----Retrieve Single Customer-----
-----

@RequestMapping(value = "/customer/{id}", method = RequestMethod.GET)
public ResponseEntity<?> getCustomer(@PathVariable("id") int id) {
    logger.info("Fetching Customer with id {}", id);
    Customer customer = customerService.getCustomerByID(id);
    if (customer == null) {
        logger.error("Customer with id {} not found.", id);
        return new ResponseEntity("Customer with id " + id
            + " not found", HttpStatus.NOT_FOUND);
    }
    return new ResponseEntity<Customer>(customer, HttpStatus.OK);
}

// -----Create a Customer-----
-----

@RequestMapping(value = "/customer/", method = RequestMethod.POST)
public ResponseEntity<?> createCustomer(@RequestBody Customer customer,
UriComponentsBuilder ucBuilder) {
    logger.info("Creating Customer : {}", customer);

    customerService.registerNewCustomer(customer);

    HttpHeaders headers = new HttpHeaders();

headers.setLocation(ucBuilder.path("/api/customer/{id}").buildAndExpand(customer.getCustomerId()).toUri());
    return new ResponseEntity<String>(headers, HttpStatus.CREATED);
}

// ----- Update a Customer -----
-----

@RequestMapping(value = "/customer/{id}", method = RequestMethod.PUT)
public ResponseEntity<?> updateCustomer(@PathVariable("id") int id,
@RequestBody Customer customer) {
    logger.info("Updating Customer with id {}", id);

    Customer currentCustomer = customerService.getCustomerByID(id);
}

```

```

    if (currentCustomer == null) {
        logger.error("Unable to update. Customer with id {} not found.", id);
        return new ResponseEntity<Customer>(currentCustomer,
HttpStatus.BAD_REQUEST);
    }

    customerService.updateCustomer(currentCustomer);
    return new ResponseEntity<Customer>(currentCustomer, HttpStatus.OK);
}

// ----- Delete a Customer-----
-- 

@RequestMapping(value = "/customer/{id}", method = RequestMethod.DELETE)
public ResponseEntity<?> deleteCustomer(@PathVariable("id") int id) {
    logger.info("Fetching & Deleting Customer with id {}", id);

    Customer customer = customerService.getCustomerByID(id);
    if (customer == null) {
        logger.error("Unable to delete. Customer with id {} not found.", id);
        return new ResponseEntity("Unable to delete. Customer with id " + id +
" not found.",
                           HttpStatus.NOT_FOUND);
    }
    customerService.removeCustomer(id);
    return new ResponseEntity<Customer>(HttpStatus.NO_CONTENT);
}
}

```

Singleton method is used to implement the email sending class. Singleton design pattern helped developer to instantiate only one object at once to send email sending object which multiple objects creating might cause problems.

```

private static SendEmail sendEmail = null;

private SendEmail() {
}

public static SendEmail getSendEmail() {

    if (sendEmail == null) {
        sendEmail = new SendEmail();
    }
    return sendEmail;
}

```

Code comments are used for internal references.

# Data Dictionary Mart

## Legend

- 🔑 Primary key Primary
- 🔑 key disabled
- 🔑 User-defined primary
- 🔑 key Unique key
- 🔑 Unique key disabled
- 🔑 User-defined unique key
- ⚡ Active trigger
- ⚡ Disabled trigger
- Primary key relation
- User-defined primary key
- ◀ relation Foreign key relation
- ◀ User-defined foreign key relation
- ➔ Input
- ➔ Output
- ➔ Input/Output
- N Nullable

## 1. Mart Shopping Online Platform

### 1.1. Tables

#### 1.1.1. Table: category (Category)

This table contains data about the categories of different food items sold by Restaurants.

Ex - Pizza, Pasta, Burger

#### Columns

Name		Data type	N	Description / Attributes
	category id	int(10, 0)	N	Primary key of the category
	category name	varchar(100)	N	Category name
	tag	varchar(100)	N	tag name 1
	tag1	varchar(100)	N	tag name 1
	tag3	varchar(100)	N	tag name 3

## Database

### 1.1.2. Table: courier (Courier)

This table contains data about the Courier who deliver the order to the door step of the customer. All Courier related master data contain on this table.

#### Columns

	Name	Data type	N	Description / Attributes
🔑	courierId	int(10, 0)		Primary Key of the Courier table Identity / Auto increment column
	city	varchar(255)	✗	City where the courier is available
	nic	varchar(50)	✗	National Identity Card Number of the Courier Service Provider Owner
	address	varchar(50)	✗	Permanent Address of the courier provider
	image_path	varchar(5000)	✗	image of the courier provider
	contact_no	int(10, 0)	✗	contact number of the courier provider
	vehicle_no	varchar(255)	✗	vehicle number of the courier provider
	courier_name	varchar(255)	✗	name of the courier provider
	district	varchar(255)	✗	district of the courier provider
	email	varchar(255)	✗	email of the courier provider
	province	varchar(255)	✗	province of the courier provider
	imagepath	varchar(10000)	✗	

### 1.1.3. Table: customer (Customer)

All end user customer related data recorded in this table.

#### Columns

	Name	Data type	N	Description / Attributes
🔑	customerId	int(10, 0)		Primary key of the customer table is customer unique id Identity / Auto increment column

	Name	Data type	N	Description / Attributes
	city	varchar(255)		City of the Customer
	geolocation	varchar(255)		Geo Cordinations of the customer address
	customerAddress	varchar(255)		Address of the Customer
	customerEmail	varchar(255)		Email of the Customer
	customerName	varchar(255)		Name of the Customer
	district	varchar(255)		District of the Customer
	mobileNo	bigint(19, 0)		mobile number of the customer

### 1.1.5. Table: feedback (Customer Feedback)

This table contains the data about the feedback from the end customer. This table contains different data about the feedback

#### Columns

	Name	Data type	N	Description / Attributes
	fbid	int(10, 0)		Primary key of the feedback Identity / Auto increment column
	dlvrRate	double(22)		delivery rating of the feedback
	email	varchar(255)		email of the feedback
	issues	varchar(255)		issue of the feedback
	name	varchar(255)		name of the feedback
	other	varchar(255)		other of the feedback
	priceRate	double(22)		price rate of the feedback
	productRate	double(22)		product rate of the feedback
	serviceRate	double(22)		service rate of the feedback
	tel	varchar(255)		phone number of the feedback user

## 1.1.6. Table: order (Order)

This table contains data about the order

### Columns

	Name	Data type	N		Description / Attributes
key	orderNo	int(10, 0)			Primary key of the order table Identity / Auto increment column
	amount	double(22)	N		amount of the order
	city	varchar(255)	N		city of the order
	courierCharj	double(22)	N		courier of the order
	courierPaymentSettledDate	datetime	N		Courier payment settled date
	courierTrackingNo	varchar(255)	N		Tracking Number of the Courier
	date	datetime	N		Date of the courier
	deliveryAddress	varchar(255)	N		Delivery address of the Courier
	deliveryEmail	varchar(255)	N		Email address of the email
	deliveryName	varchar(255)	N		delivery person name of the customer
	deliveryTelephonenNo	varchar(255)	N		phone number of the delivery person
	delveryCost	double(22)	N		delivery cost for the order
	delveryVat	double(22)	N		vat for the order
	district	varchar(255)	N		delivery district for the order
	shopPaymentSettledDate	datetime	N		payment shop settled date for the order
	province	varchar(255)	N		province for the order
	serviceCharg	double(22)	N		service charge for the order
	specialRequest	varchar(255)	N		special request for the order
	statusCourier	varchar(255)	N		status of the order
	statusDescription	varchar(255)	N		description of the order
	statusShop	varchar(255)	N		status of the shop
	vat	double(22)	N		vat of the order
	courierId	int(10, 0)	N		id of the courier
	customerEmail	varchar(255)	N		email of the customer of the order
	delivery_optionid	int(10, 0)			delivery option of the order
	shopNo	int(10, 0)	N		shop of the order

### 1.1.7. Table: product (Products)

This table contains data about different product items.

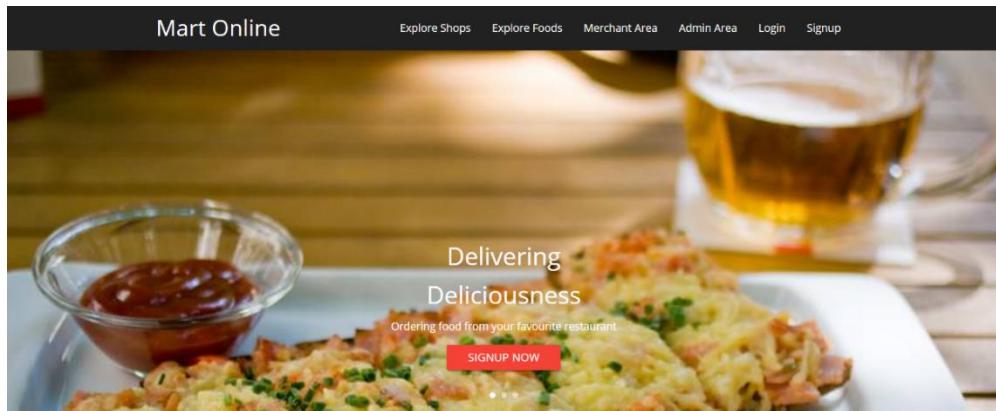
#### Columns

	Name	Data type	N	Description / Attributes
key	productId	int(10, 0)		Primary Key of the Product Identity / Auto increment column
	name	varchar(255)		name of the product
	description	varchar(10000)	N	description of the product
	other_name	varchar(255)	N	other name of the product
	imagePath	tinyblob	N	Image path for the product

## User interfaces

### Home Screen

This is the Home Screen of the Mart Online Food Ordering system. This view is gives the end customer clear idea about how the Mart online platform food ordering process is working. This page is common for all users, users can explore food, explore shops by navigating on the navigation bar.



### How It Works



## Explore Shops

This interface displays the end customer about the restaurants registered with the system. End customer able to search their preferred restaurant by restaurant name, area, and by the category.

Mart Online

Login   Signup

Explore > Shops

≡

### Explore Shops

Search Ex - Area, Price, Food Name, Description, Category

 <b>Cafe Fort</b> Fort, Colombo. Casual Dining  <a href="#">MORE INFO</a>	 <b>Aasife</b> Rajagiriya, Colombo. Casual Dining  <a href="#">MORE INFO</a>
 <b>Shandong</b> Rajagiriya, Colombo. Casual Dining  <a href="#">MORE INFO</a>	 <b>Chinese Dragon</b> Bambalapitiya, Colombo 04. Casual Dining  <a href="#">MORE INFO</a>
 <b>Hotel de Pillawoos</b> Bambalapitiya, Colombo 04. Casual Dining  <a href="#">MORE INFO</a>	

**Mart Online Shopping**  
©2017

**Links**

[Explore Foods](#)  
[Explore Shops](#)  
[Merchant Area](#)  
[Admin Area](#)

[More Links](#)

## Explore Food

Below interface allows end customer to add their preferred food to cart in required quantities, finally to checkout them.

Mart Online

Explore Shops Explore Foods Merchant Area Admin Area

Explore > Shops > Products

≡ PRODUCTS ABOUT

Small Cart 6 Items in cart  
Grand Total : Rs .79.01  
Checkout

Dry Bun	X
4 X 12.50	50
Submarine	X
2 X 14.50	29

Rice & Curry  
Simple lunch non veg  
+  
Submarine  
Soft and chewy fresh Sub Bun.with polish sausage  
+  
Dry Bun  
12.50  
+  
MORE INFO 14.50 price  
MORE INFO 14.50 price  
MORE INFO 14.50 price

Eggplant Curry  
Curry made with Egg plant.  
+  
Sushi  
I am a very simple card. I am good at containing small bits of information.  
+  
MORE INFO 14.50 price  
MORE INFO 14.50 price

Mart Online Shopping

@2017

Links

Explore Foods  
Explore Shops  
Merchant Area  
Admin Area

More Links

## Order page

This page allows end customer to place the order for the food items in the cart. End Customer will be able to select a delivery recipient name address, phone number, delivery date and delivery time.

The screenshot shows the 'Place My Order' section of the Mart Online website. The form fields include:

- Order ID (input field)
- Recipient Name (input field)
- Address (input field)
- City (input field)
- Email (input field)
- Phone (input field)
- Delivery Date (date input field) - dd/mm/yyyy
- Delivery Time (time input field) - --:--

A green 'SUBMIT >' button is located at the bottom of the form. To the right, a sidebar displays the 'Small Cart' with 2 items and a total of Rs. 27.00. The items listed are Dry Bun (1 X 12.50) and Submarine (1 X 14.50).

Mart Online Shopping  
©2017

Links

- Explore Foods
- Explore Shops
- Merchant Area
- Admin Area

More Links

Other interfaces are attached in the appendix with description.

## **Tools Used to develop the system**

JetBrains IntelliJ IDEA used as Integrated Development Environment(IDE) for the system development.

Github desktop used as Git Client for the development of the system as it provides easy graphical user interface.

Developer has obtained Ultimate version of the IntelliJ IDEA for the London metropolitan university student ID and the keeping the project as private repository in GitHub.

HeidiSQL used as administration tool for MySQL which is free and opensource tool.

Application uses Amazon S3 bucket for storage needs, to store the user profile pictures, shop logo, shop cover image saving, product image hosting. Developer chose Asia Pacific (Singapore) as the region of the bucket to minimise the latency in delivering the content.

## **System Requirements**

The software operational environment is the environment (hardware and software resources) needed for the proper functioning of the system after its deployment. Different tools and frameworks that need to be present in the client's computer for the system to work properly are discussed.

The system requirements are of 2 types: Client-side requirements and server-side requirements. Client-side is called the operations done by the client in a client-server relationship in a computer network. Likewise, the server-side is called the operations done by the server in a client-server computer network.

<b>Client Side</b>	<b>Server Side</b>
Device with Windows, Linux or MacOS	Ubuntu 14.04 Server Edition
Google Chrome 4.0,Microsoft Edge 12.0, Firefox 3.5, Safari 4.0 or above version	Apache2 Server, Apache Tomcat 7 and MySQL Server 5.5
512 MB RAM	512MB RAM
5GB free Storage	20GB free Storage
Internet Connection	High speed Internet Connection with Static IP AWS Account for S3 bucket functionality Domain name(optional)

## **Implementation Process**

Implementation process of the system done by using the parallel implementation process chose to develop the system because management requested to implement the assignment submission part and resource sharing part as soon as possible. It was part of Institute's management decision to reduce paper usage as it reduces 60% of expenses of the organization.

Implementation process of the system done for the first increment of delivery user registration, Assignment submission and the Resource sharing parts were implemented.

## **Hardware and Software Installation**

For the implementation process of the system, developer suggested the institute to host the application in a virtual private server than hosting in a dedicated server or by hosting the application or in-house hosting because of several reasons. Developer has done a market research on pricing of each solution. In house dedicated server is the most expensive method of hosting, as the servers need more energy for air conditioning and always on high speed internet connection, another main expense is company has to pay dedicated IP address. Also organization might hire a network engineer to maintain the network. Biggest disadvantage is company have to invest big amount of money to purchase the server nearly costs more than 300,000 LKR, monthly maintenance cost will be around 50,000 LKR. Advantage of the in-house hosting is since all data retained in the organization, Manawa higher education institute do not have to worry about the data security.

Dedicated hosting server is less expensive in house hosting solution. Dedicated server on cloud means that server is dedicated specifically for our application hosting purpose, all resources of the server is completely dedicated for our application development. Because this server is located on data centre in an organization, organization can cut down the costs of air conditioning and internet connection fees but the dedicated server costs around 15500 LKR per month, that is much cheaper than the in house hosting solution, but still it's a considerable cost for the company.

DEDICATED HOSTING PROVIDER				
FEATURES	ZNETLIVE® THE HOSTING AUTHORITY	BIGROCK Domains, Websites and more.	HostGator	GoDaddy®
Price	₹ 6110/mo	₹ 7299/mo.	₹ 9795/mo.	₹ 8199/mo.
RAM	8 GB	4 GB	4 GB	2 GB
RAM type	DDR3 ECC RAM	DDR3 RAM	DDR2 ECC	No Information
Server type	Intel Xeon E3-1230	Intel E3-1220LV2	Intel E3-1220LV2	Intel Xeon processor

Figure 15 – Dedicated server price comparison -

Virtual Private server hosting, also known as VPS hosting is much less expensive than dedicated server hosting. The VPS acts like a server but it is in actuality apart of one physical server. VPS uses resource sharing system. It has its own operating system, hard drive space, and bandwidth. Developer has done market research vps service providers and found reliable hosting solutions.

Rank	Company	Price	Exclusive Offer	Setup Fee	Memory	Space	Traffic	OS
1	 MOCHAHOST	\$7.98	50% OFF only through <a href="#">Top11Hosting</a> <a href="#">click for details</a>	FREE	0.5 GB	40 GB	500 GB	 
2		\$29.95	–	FREE	512 MB	40 GB	750 GB	
3	 GreenGeeks®	\$39.95	–	FREE	512 MB	25 GB	500 GB	
4.		\$16.99	–	FREE	1 GB	40 GB	1 T	 

Figure 16- VPS server comparison (Hosting, 2017)

Prices are starting from \$5 per month from reliable sources like Digital Ocean ([digitalocean.com](http://digitalocean.com)) Amazon also provides EC2 server instances but the developer selected digital ocean as he already used digital ocean service previously and a satisfied customer. You have the same technical support as with shared hosting for most issues. You can choose from semi managed hosting where the web host handles some maintenance and fully managed services.

## Testing

Purpose of the software testing is to identify whether the software meets the its requirements mentioned in the software requirements specification. Software testing phase also assures the quality of the final product.

Software testing process has two main goals.

1. Demonstrate the developer and the customer the software meets its requirements.
2. Discover the defects, find system crashes.

First goal leads to the validation testing, there tester able to identify whether the system functioning as expected.

Second Goal leads tester for defect testing, these tests are designed to expose defects, so those teste cases are not needed to be designed to test the system in expected manner (Sommerville, 2011)

Validation and Verification also another two processes that are combined with software development process, goal of these two processes also to establish the confidence to the system is to fit for purpose.

In functional testing, the structure of the program not considered, Test cases are based on the values input and the produced output from the specific inputs. There is no attempt to analyse the code, internal structure of the program, In the mart test plan also it contains mostly these type of testing, this also known as black box testing.

Backbox testing allows to identify the following errors, incorrect or missing functions in the system, missing components of the interface, errors presented in the data model.

Another type of testing is white box testing which is also known as structural testing, glass box testing, clear box testing logic-driven testing. Performing white box test needs special knowledge on code inspection.

White box test helps to understand whether all paths in a process are correctly operational, logical decisions are true and false condition. Advantages of white box testing is it forces the developer to carefully think about the implementation.

(Agarwal, Gupta and Tayal, 2010)

## Software Test Plan

Software test plan is the outline of the test strategy and overall test approach for the Mart Shopping platform

Mart online Shopping Platform Test Plan	
Test plant ID	1
Brief Introduction	<p>This document describes the test plan for the Mart Shopping platform.</p> <p>This document supports the following objectives.</p> <p>Identify the functionalities needs to be tested and the features that not need to be tested.</p> <p>Identify required resources and provide an estimate of the teste efforts.</p> <p>List the deliverable elements of the test activities.</p>
Features to be tested	<ul style="list-style-type: none"><li>• View Shops</li><li>• View products listed by shops</li><li>• Add products to cart</li><li>• Checkout the cart</li><li>• Register as shop owner (merchant)</li><li>• register as customer</li><li>• Register as delivery person</li><li>• Login, add products to system</li><li>• add shops to system</li><li>• Check the order status</li><li>• Update order status as merchant</li><li>• update the order as delivery person</li><li>• Review the order</li><li>• View Shop description</li></ul>
Features need not to be tested	Password forgot, Pay on delivery, About page
Approach	Black box testing
Environment	Ordinary desktop or laptop PC with internet connection and updated google chrome browser
Test Deliverables	Test cases, test data, Test strategy
Roles	Tester
Schedule	

## Test case # 1

<b>Test Case #: 1.01</b>	<b>Test Case Name:</b> Login
<b>System:</b> Mart Online Shopping platform	<b>Subsystem:</b> Login
<b>Designed by:</b> Chamith Wickramarathna	<b>Design Date:</b> 05/05/2017
<b>Executed by:</b> Chamith Wickramarathna	<b>Execution Date:</b> 06/06/2017
<b>Short Description:</b>	This test case includes the testing for the user login function.

**Pre-condition:** User is already registered with the system.

User navigated to the home page

Step	Action	Expected System Response	Pass/ Fail	Scenario
1	Click on the login in the navigation bar, upper right corner of the screen.	User should navigate to the login page	Pass	
2	Enter the user name or email in the text box		Pass	
3	Type the password in the password field		Pass	
4	Click on the Login button	User should navigate back to the Home page where user was.	Pass	Username should appear in the upper right corner of the navigation bar and the login item should be disappeared.

<b>Test Case #: 1.02</b>	<b>Test Case Name:</b> Signup
<b>System:</b> Mart Online Shopping platform	<b>Subsystem:</b> Signup
<b>Designed by:</b> Chamith Wickramarathna	<b>Design Date:</b> 05/05/2017
<b>Executed by:</b> Chamith Wickramarathna	<b>Execution Date:</b> 06/06/2017
<b>Short Description:</b>	This test case includes the testing for the user login function.

**Pre-condition:**

User navigated to the home page

<b>Step</b>	<b>Action</b>	<b>Expected System Response</b>	<b>Pass/ Fail</b>	<b>Scenario</b>
1	Click on the signup on navigation	User should navigate to the signup page	Pass	
2	Enter the user name or email in the text box		Pass	
3	Type the password in the password field		Pass	
4	Click on the Login button	User should navigate back to the Home page where user was.	Pass	Username should appear in the upper right corner of the navigation bar and the login item should be disappeared.

<b>Test Case #:</b> 1.03	<b>Test Case Name:</b> Explore Shops
<b>System:</b> Mart Online Shopping platform	<b>Subsystem:</b> Explore Shops
<b>Designed by:</b> Chamith Wickramarathna	<b>Design Date:</b> 05/05/2017
<b>Executed by:</b> Chamith Wickramarathna	<b>Execution Date:</b> 06/06/2017
<b>Short Description:</b>	This test case includes the testing for the user login function.

**Pre-condition:**

User navigated to the home page

<b>Step</b>	<b>Action</b>	<b>Expected System Response</b>	<b>Pass/ Fail</b>	<b>Scenario</b>
1	Click on the Explore Shops on navigation bar	User should navigate to the Explore Shops page	Pass	

<b>Test Case #:</b> 1.04	<b>Test Case Name:</b> Explore Shops
<b>System:</b> Mart Online Shopping platform	<b>Subsystem:</b> Search Shop
<b>Designed by:</b> Chamith Wickramarathna	<b>Design Date:</b> 05/05/2017
<b>Executed by:</b> Chamith Wickramarathna	<b>Execution Date:</b> 06/06/2017
<b>Short Description:</b>	This test case includes the testing for the user login function.

**Pre-condition:**

User navigated to the home page

<b>Step</b>	<b>Action</b>	<b>Expected System Response</b>	<b>Pass/ Fail</b>	<b>Scenario</b>
1	Click on the Explore Shops on navigation bar	User should navigate to the Explore Shops page	Pass	
2	Type the Shop name in the search bar, click on the search bar and type the name of restaurant	It should filter out the restaurant name from the list.	Pass	

<b>Test Case #:</b> 1.05	<b>Test Case Name:</b> Explore Shops
<b>System:</b> Mart Online Shopping platform	<b>Subsystem:</b> Search Shop
<b>Designed by:</b> Chamith Wickramarathna	<b>Design Date:</b> 05/05/2017
<b>Executed by:</b> Chamith Wickramarathna	<b>Execution Date:</b> 06/06/2017
<b>Short Description:</b>	Test shops are listed correctly

**Pre-condition:**

User navigated to the home page

<b>Step</b>	<b>Action</b>	<b>Expected System Response</b>	<b>Pass/ Fail</b>	<b>Scenario</b>
1	Click on the Explore Shops	User should navigate to the	Pass	

	on navigation bar	Explore Shops page		
2	Type the Shop name in the search bar, click on the search bar and type the name of restaurant	It should filter out the restaurant name from the list.	Pass	

<b>Test Case #:</b> 1.06	<b>Test Case Name:</b> Explore Shops
<b>System:</b> Mart Online Shopping platform	<b>Subsystem:</b> Explore products in Shop
<b>Designed by:</b> Chamith Wickramarathna	<b>Design Date:</b> 05/05/2017
<b>Executed by:</b> Chamith Wickramarathna	<b>Execution Date:</b> 06/06/2017
<b>Short Description:</b>	Products are listed in the shop.

**Pre-condition:**

User navigated to the home page

Step	Action	Expected System Response	Pass/ Fail	Scenario
1	Click on the Explore Shops on navigation bar	User should navigate to the Explore Shops page	Pass	
2	Click on a Shop Name listed in shop list.	User should navigate to products listed by the shop	Pass	
4	Click on the Mart Online branding on the upper left corner.	Navigate back to home page	Pass	
5	Repeat step 1		Pass	
6	Search a Shop Name or by Area	Relevant results should be filtered	Pass	
7	Repeat Step 2		Pass	

<b>Test Case #:</b> 1.07	<b>Test Case Name:</b> Explore Shops
<b>System:</b> Mart Online Shopping platform	<b>Subsystem:</b> Explore products in Shop
<b>Designed by:</b> Chamith Wickramarathna	<b>Design Date:</b> 05/05/2017
<b>Executed by:</b> Chamith Wickramarathna	<b>Execution Date:</b> 06/06/2017
<b>Short Description:</b>	Test Products are listed in the shop.

**Pre-condition:**

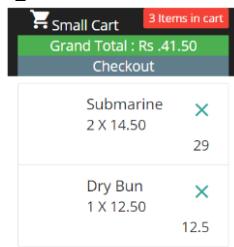
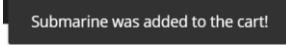
User navigated to the home page

<b>Step</b>	<b>Action</b>	<b>Expected System Response</b>	<b>Pass/ Fail</b>	<b>Scenario</b>
1	Click on the Explore Shops on navigation bar 	User should navigate to the Explore Shops page	Pass	
2	Click on a Shop Name listed in shop list.	User should navigate to products listed by the shop	Pass	
4	Click on the Mart Online branding on the upper left corner. 	Navigate back to home page	Pass	
5	Repeat step 1		Pass	
6	Search a Shop Name or by Area	Relevant results should be filtered	Pass	
7	Repeat Step 2	User should navigate to products listed by the shop	Pass	

<b>Test Case #:</b> 1.08	<b>Test Case Name:</b> Add/Remove Products to Cart
<b>System:</b> Mart Online Shopping platform	<b>Subsystem:</b> Mini Cart
<b>Designed by:</b> Chamith Wickramarathna	<b>Design Date:</b> 05/05/2017
<b>Executed by:</b> Chamith Wickramarathna	<b>Execution Date:</b> 06/06/2017
<b>Short Description:</b>	Mini cart handled in client side.

**Pre-condition:**

Successfully completed Script 1.08 and navigated last step of the script

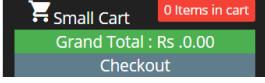
Step	Action	Expected System Response	Pass/ Fail	Scenario
1	Click on the  icon on the products listed in the page listed.	<p>Product should be added to the mini cart placed in the right side of the page</p>  <p>And screen toast should show at the upper right corner of the page saying the PRODUCT_NAME was added to the cart.</p>  <p>Items should be added, and grand total should be updated, quantity should be increased upon adding.</p> <p>Total number of items should be updated in the red</p>	Pass	

		badge of the Mini cart box. 		
2	Repeat step 1	Total product count should be increased by 1 and grand total item count should be increased.	Pass	
4	Click on the  on the item at the Mini cart.	Total number of items in the cart should be decreased.  Grand total should be deducted by the value of the item removed,  Total quantity of the item reduced.	Pass	
5	Repeat step 4 until all items get removed	Grand total should be like below   Small Cart  Grand Total : Rs .0.00 Checkout		

<b>Test Case #:</b> 1.09	<b>Test Case Name:</b> Checkout Order
<b>System:</b> Mart Online Shopping platform	<b>Subsystem:</b> Checkout
<b>Designed by:</b> Chamith Wickramarathna	<b>Design Date:</b> 05/05/2017
<b>Executed by:</b> Chamith Wickramarathna	<b>Execution Date:</b> 06/06/2017
<b>Short Description:</b>	Mini cart handled in client side.

**Pre-condition:**

Successfully logged in as a client user

Step	Action	Expected System Response	Pass/ Fail	Scenario
1	Go to Shop page	Products should be listed	Pass	
2	Add products to cart by clicking on + sign.	Products should be added to the cart.	Pass	
4	Add click on the checkout button on the mini cart  	User should redirect to place order page.	Pass	
5	Fill the relevant details and click on Continue and confirm	User will have directed to Payment option selection page.	Pass	
6	Click on "Pay on delivery"	User will get the message saying order confirmed order will be delivered on "date selected" between "time duration provided"	Pass	

<b>Test Case #:</b> 1.10	<b>Test Case Name:</b> Checkout Order
<b>System:</b> Mart Online Shopping platform	<b>Subsystem:</b> Checkout
<b>Designed by:</b> Chamith Wickramarathna	<b>Design Date:</b> 05/05/2017
<b>Executed by:</b> Chamith Wickramarathna	<b>Execution Date:</b> 06/06/2017
<b>Short Description:</b>	Mini cart handled in client side.

**Pre-condition:**

Successfully logged in as a client user

Step	Action	Expected System Response	Pass/ Fail	Scenario
1	Go to Shop page	Products should be listed	Pass	
2	Add products to cart by clicking on + sign.	Products should be added to the cart.	Pass	
4	Add click on the checkout button on the mini cart  	User should redirect to place order page.	Pass	
5	Fill the relevant details and click on Continue and confirm	User will have directed to Payment option selection page.	Pass	<p>Try click on the continue field without filling the relevant details.</p> <p>If address Phone number Delivery date or delivery time is missing</p> <p>User will get a message in red colour stating the please fill this field</p>
6	Click on "Pay on delivery"	User will get the message saying order confirmed order will be delivered on "date"	Pass	

		selected" between "time duration provided"		
--	--	--	--	--

<b>Test Case #: 1.11</b>	<b>Test Case Name:</b> Add Courier
<b>System:</b> Mart Online Shopping platform	<b>Subsystem:</b> Admin Courier
<b>Designed by:</b> Chamith Wickramarathna	<b>Design Date:</b> 05/05/2017
<b>Executed by:</b> Chamith Wickramarathna	<b>Execution Date:</b> 06/06/2017
<b>Short Description:</b>	

**Pre-condition:**

Successfully logged in as an admin user

Step	Action	Expected System Response	Pass/ Fail	Scenario
1	Navigate to Admin panel		PASS	
2	Click on the Courier Tile/Button	Should open courier page	PASS	
3	Click on Add Courier	Should open add courier page	PASS	
4	Type National Identity card number, address, Vehicle number, vehicle type, email, vehicle category, Working area(City),		PASS	
5	Click on the Submit button	It should receive Success Message Toast at right top screen	PASS	<p>Try submitting without filling required fields.</p> <p>User should get an error message below each text input</p> <p style="color: red;"><b>This field is required.</b></p>

<b>Test Case #:</b> 1.12	<b>Test Case Name:</b> Add Shop
<b>System:</b> Mart Online Shopping platform	<b>Subsystem:</b> Admin> Shop
<b>Designed by:</b> Chamith Wickramarathna	<b>Design Date:</b> 05/05/2017
<b>Executed by:</b> Chamith Wickramarathna	<b>Execution Date:</b> 06/06/2017
<b>Short Description:</b>	

**Pre-condition:**

Successfully logged in as an admin user

Step	Action	Expected System Response	Pass/ Fail	Scenario
1	Navigate to Admin panel		PASS	
2	Click on the Courier Tile/Button	Should open courier page	PASS	
3	Click on Add Manage Shop	Should open add courier page	PASS	
4	Fill the required fields.		PASS	
5	Click on the Submit button	It should receive Success Message Toast at right top screen	PASS	User should get an error message below each text input <b>This field is required.</b>

## **6. Conclusions**

With the rapid development of new technology and busy life style of the people, people always try to save their time and money. Small business owners are always trying to increase their sales to grow their business. People who are not busy with their work always and do not have satisfied income to cover their living cost intension is to improve their income. Mart online shopping platform is Developed focusing on the above three categories of people, so it is very much clear the target audience is high and the opportunity for becoming the success in Sri Lankan market is high for the Mart Shopping platform.

## **Recommendation for future enhancements**

Several online surveys and offline surveys done with the help of Survey Monkey site([surveymonkey.com](https://www.surveymonkey.com)) and with printed papers. Survey was available with the below link at [surveymonkey.com](https://www.surveymonkey.com/r/8W63P9W).

<https://www.surveymonkey.com/r/8W63P9W>

Printed copy of the survey attached at the appendix section the at the appendix section. After the analysis done within the age limit of 18 – 65.

84% of positive results were able to receive for the author. Several additional ideas were added to the main business idea of the system for future enhancements.

Main suggestions for the system were without limiting for the restaurant food, expand the original idea also for the grocery food items which is a daily need for everyone.

As per security concern for the delivery person suggestion was presented to manually verify them at physical registration place. Two factor authentications to be presented as a future enhancement for the system.

Another suggestion was since this site target market was Sri Lanka it should be available in All the major languages should be integrated Sinhala and Tamil along with English.

Integration of the online payment gateway to accept credit cards was another major concern as it is easier than the pay on delivery method and it will also resolve many problems that will arise in future. Developer decided to after completing successful Beta test with public and then implement the credit or debit card payment facility.

## **Self-reflection / Critical Appraisal**

Development of the program was bit challenging at the beginning with for the developer, and selecting requirements according to the priority also a challenge.

Features available at IntelliJ Idea.

Smart Completion, Chain completion, Static member completion, Data flow analysis, Detecting Code duplications, inspections and quick fixes, Easy search everywhere.

Smart code support for Java Script and support for Spring bean detection were some of the admirable features offered by IntelliJ Idea, Developer previously used net beans and Eclipse developer felt the difference between those two IDE's and the powerful features of IntelliJ Idea.

## References

- Anon, (2017). [image] Available at: <http://1.bp.blogspot.com/-6cUGzkUsb3Q/TyQlarxNo4I/AAAAAAAUAU8/81PNgALailg/s1600/mvc.jpg> [Accessed 19 Apr. 2017].
- Meier, A. and Stormer, H. (2009). EBusiness & eCommerce. Berlin: Springer.
- Rawsthorne, P., Rawsthorne, P. and profile, V. (2017). MVC in a three-tier architecture. [online] Criticaltechnology.blogspot.com. Available at: <http://criticaltechnology.blogspot.com/2011/09/mvc-in-three-tier-architecture.html> [Accessed 19 Apr. 2017].
- Roebuck, K. (2012). SaaS - Software as a Service. Emereo Publishing.
- Miller, D. (2009). Building a Project Work Breakdown Structure. Boca Raton: Auerbach Publications.
- MVC Architecture. (2017). [image] Available at: <https://upload.wikimedia.org/wikipedia/commons/a/a0/MVC-Process.svg> [Accessed 24 Apr. 2017].
- Oellermann, W. (2001). Architecting Web services. Berkeley, CA: Apress.
- Study.com. (2017). What is Software Prototyping? - Definition, Models & Tools - Video & Lesson Transcript | Study.com. [online] Available at: <http://study.com/academy/lesson/what-is-software-prototyping-definition-models-tools.html> [Accessed 24 Apr. 2017].
- www.tutorialspoint.com. (2017). SDLC Software Prototype Model. [online] Available at: [https://www.tutorialspoint.com/sdlc/sdlc\\_software\\_prototyping.htm](https://www.tutorialspoint.com/sdlc/sdlc_software_prototyping.htm) [Accessed 24 Apr. 2017].
- Agarwal, B., Gupta, M. and Tayal, S. (2010). Software engineering & testing. Sudbury, Mass.: Jones and Bartlett.
- Gloriafood.com. (2017). What are the advantages of online food ordering. [online] Available at: <https://www.gloriafood.com/restaurant-ideas/advantages-ofonlinefood-ordering> [Accessed 1 Apr. 2017].
- Networksolutions.com. (2017). What is Ecommerce?. [online] Available at: <http://www.networksolutions.com/education/what-is-e-commerce/> [Accessed 1 Apr. 2017].
- Wohl, A. (2008). Succeeding at SaaS. 1st ed. Bohemia, NY: Wohl Associates
- Sommerville, I. (2011). Software Engineering. 9th ed.

## Appendix 1: Class diagram - detailed

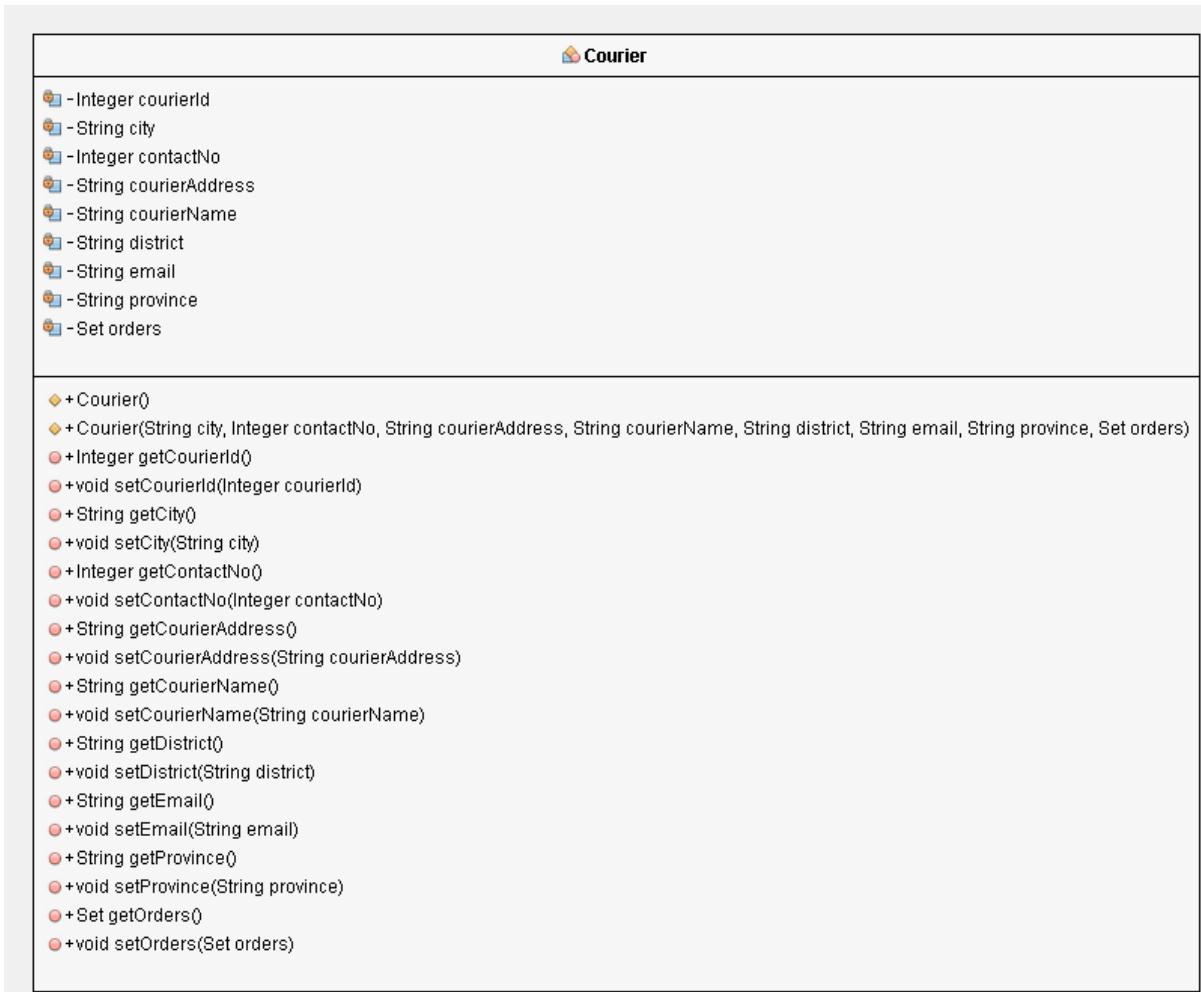


Figure 17 Courier Class

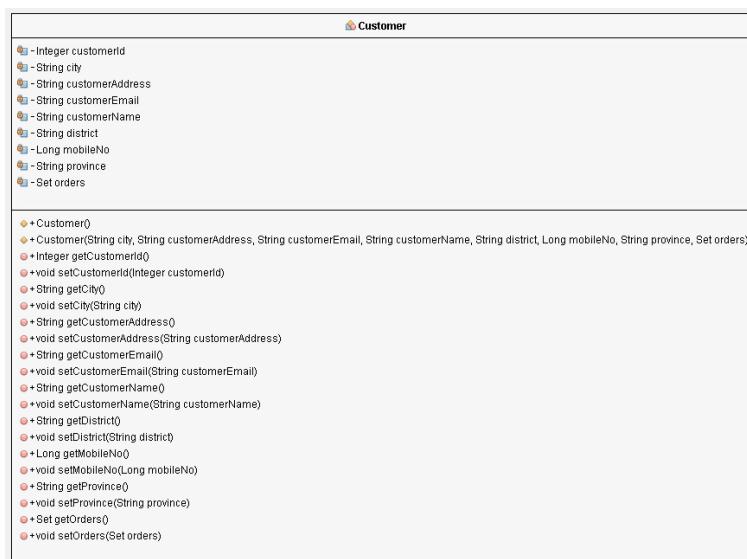


Figure 18 Customer Class

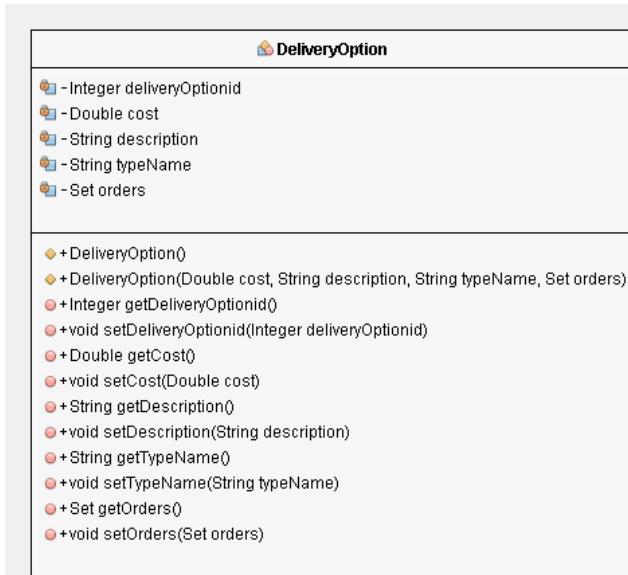


Figure 19 Delivery Option

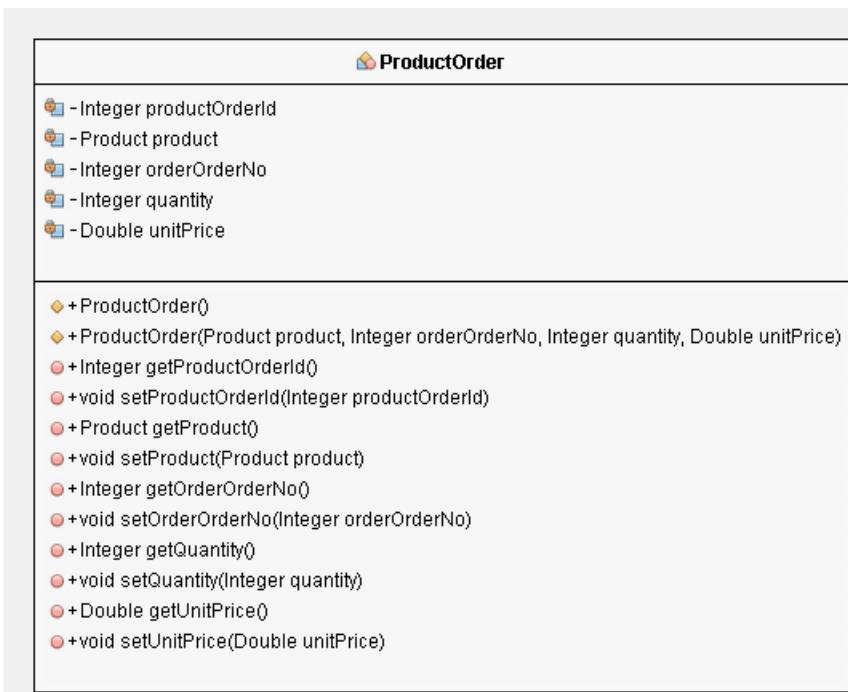


Figure 20 Place Order

Feedback	
<pre> ✉ - Integer fbid ✉ - Double dlvrRate ✉ - String email ✉ - String issues ✉ - String name ✉ - String other ✉ - Double priceRate ✉ - Double productRate ✉ - Double serviceRate ✉ - String tel </pre>	
<pre> ◆ +Feedback() ◆ +Feedback(Double dlvrRate, String email, String issues, String name, String other, Double priceRate, Double productRate, Double serviceRate, String tel) ● +Integer getFbid() ● +void setFbid(Integer fbid) ● +Double getDlvrRate() ● +void setDlvrRate(Double dlvrRate) ● +String getEmail() ● +void setEmail(String email) ● +String getIssues() ● +void setIssues(String issues) ● +String getName() ● +void setName(String name) ● +String getOther() ● +void setOther(String other) ● +Double getPriceRate() ● +void setPriceRate(Double priceRate) ● +Double getProductRate() ● +void setProductRate(Double productRate) ● +Double getServiceRate() ● +void setServiceRate(Double serviceRate) ● +String getTel() ● +void setTel(String tel) </pre>	

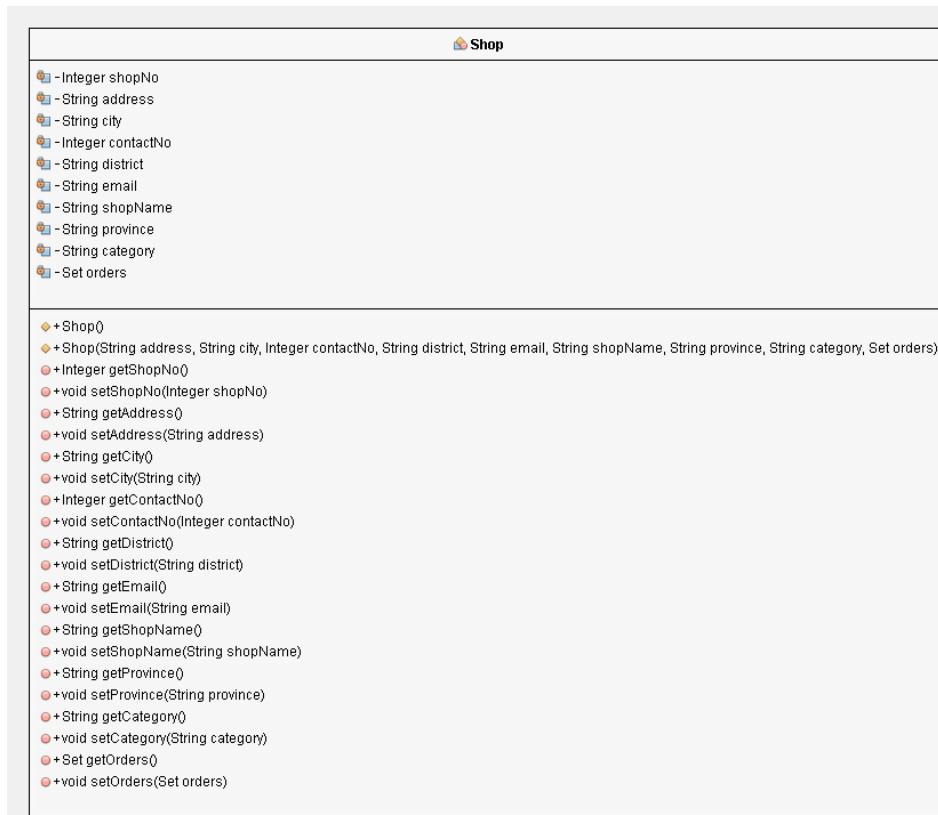
Figure 21 Feedback

 Product	
 - Integer productId  - String name  - String description  - String otherName  - byte[] imagePath  - Integer stripSize  - Double unitPrice  - Integer typeId  - Set productOrders	
 + Product()  + Product(String name)  + Product(String name, String description, String otherName, byte[] imagePath, Integer stripSize, Double unitPrice, Integer typeId, Set productOrders)  + Integer getProductId()  + void setProductId(Integer productId)  + String getName()  + void setName(String name)  + String getDescription()  + void setDescription(String description)  + String getOtherName()  + void setOtherName(String otherName)  + byte[] getImagePath()  + void setImagePath(byte[] imagePath)  + Integer getStripSize()  + void setStripSize(Integer stripSize)  + Double getUnitPrice()  + void setUnitPrice(Double unitPrice)  + Integer getTypeId()  + void setTypeid(Integer typeId)  + Set getproductOrders()  + void setproductOrders(Set productOrders)	

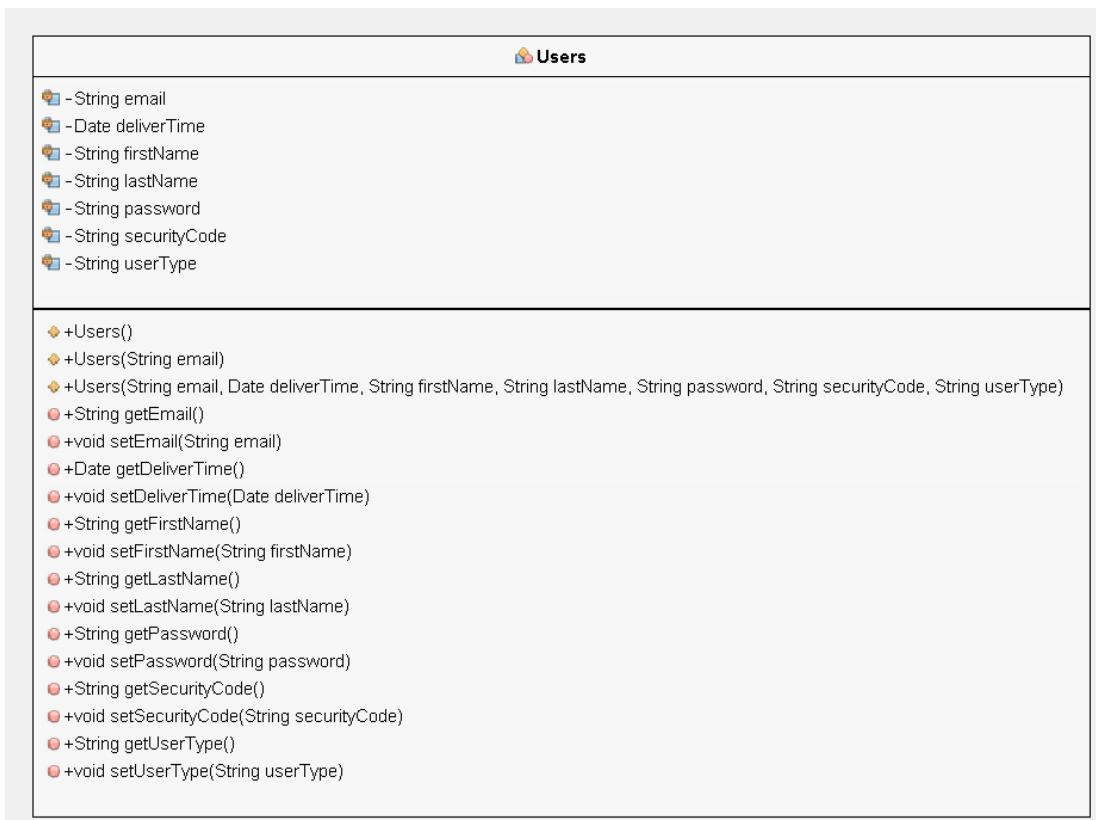
Figure 22 Product Class

 Type	
 - Integer typeId  - String typeName	
 + Type()  + Type(String typeName)  + Integer getTypeid()  + void setTypeid(Integer typeId)  + String getTypeName()  + void setTypeName(String typeName)	

Figure 23 Type Class



*Figure 24 Shop Class*



*Figure 25 Users Class*

## Appendix 2: User interfaces

### Mart Shopping

Please, login into your account

Enter your email

---

Enter your password

---

[Forgot Password?](#)

**LOGIN**

[Create account](#)

### Add Customer Details

**Mart Online**

Explore Shops   Explore Foods   Merchant Area   Admin Area

Customer > Customer Data

**Add Customer**

<input type="text"/> Customer ID	<input type="text"/> Customer Name
<input type="text"/> Address	
<input type="text"/> Email	<input type="text"/> Category
<input type="text"/> City	<input type="text"/> Province
<input type="text"/> Mobile Number	<input type="file"/> Picture

**SUBMIT ►**

**Mart Online Shopping**  
©2017

**Links**

Explore Foods  
Explore Shops  
Merchant Area  
Admin Area

[More Links](#)

## Add products

The screenshot shows the 'Add Products' form. At the top, there's a navigation bar with 'Mart Online' and links to 'Explore Shops', 'Explore Foods', 'Merchant Area', and 'Admin Area'. Below this, the breadcrumb path 'Admin > Shops > Add Item' is shown. The main form has fields for Product ID, Product Name, Description, Other name, Size, Unit Price, Category (with a dropdown menu 'Choose Category'), and Autocomplete. A green 'SUBMIT >' button is at the bottom.

Mart Online Shopping  
@2017

Links

Explore Foods  
Explore Shops  
Merchant Area  
Admin Area

More Links

## Add Shops

The screenshot shows the 'Add Shop' form. At the top, there's a navigation bar with 'Mart Online' and links to 'Explore Shops', 'Explore Foods', 'Merchant Area', and 'Admin Area'. Below this, the breadcrumb path 'Admin > Shops > Add Shop' is shown. The main form has fields for Shop ID, Shop Name, Address, City, Province, Email, and Category. A green 'SUBMIT >' button is at the bottom.

Mart Online Shopping  
@2017

Links

Explore Foods  
Explore Shops  
Merchant Area  
Admin Area

More Links

## Search restaurant by food category

Mart Online

Login   Signup

Explore > Shops

pizz

Cafe Fort  
Fort, Colombo. Pizza  
★★★★★  
[MORE INFO](#)

Mart Online Shopping

@2017

Links

Explore Foods  
Explore Shops  
Merchant Area  
Admin Area

[More Links](#)

## Search Restaurant by Area

Mart Online

Login   Signup

Explore > Shops

Rajagiri

Aasife  
Rajagiriya, Colombo. Casual Dining  
★★★★★  
[MORE INFO](#)

Shandong  
Rajagiriya, Colombo. Casual Dining  
★★★★★  
[MORE INFO](#)

Mart Online Shopping

@2017

Links

Explore Foods  
Explore Shops  
Merchant Area  
Admin Area

[More Links](#)

## Merchant Area

### Completed Orders

Mart Online

Explore Shops Explore Foods Merchant Area Admin Area

Login > Merchant Area

≡

Merchant Area

Orders to be delivered: 3

Total Number of Orders: 24

Total Sales: \$12.00

COMPLETED ORDERS ACTIVE ORDERS REVENUE REPORT TOP SELLING

Show entries

Order ID	Ordered Date	Delivered Date	Delivered Time	Order Status	Order Total
#74645454	05/03/2017	05/05/2017	17:50	Completed	\$10.87
#74645454	05/03/2017	05/05/2017	17:50	Completed	\$25
#74645454	05/03/2017	05/05/2017	17:50	Completed	\$18.00

Showing 1 to 3 of 3 entries

Previous 1 Next

⚙️

Store Settings

Mart Online Shopping

©2017

Links

Explore Foods  
Explore Shops  
Merchant Area  
Admin Area

More Links

## Active Orders - Detailed

Mart Online

Explore Shops Explore Foods Merchant Area Admin Area

Login > Merchant Area

Merchant Area

Orders to be delivered: 3

Total Number of Orders: 24

Total Sales: \$12.00

COMPLETED ORDERS ACTIVE ORDERS REVENUE REPORT TOP SELLING

Product ID	Product Name	Status
#756454544	Dry Bun	<input checked="" type="checkbox"/> Completed
Select all and Mark as Order Completed		
Product ID	Product Name	Status
#756454544	Dry Bun	<input type="checkbox"/> Completed
#756454547	Submarine	<input type="checkbox"/> Completed
#756454544	Donut	<input checked="" type="checkbox"/> Completed
Select all and Mark as Order Completed		

Store Settings

Mart Online Shopping

©2017

Links

Explore Foods  
Explore Shops  
Merchant Area  
Admin Area

More Links

## Merchant Area - Top Selling Products

Mart Online

Explore Shops Explore Foods Merchant Area Admin Area

Login > Merchant Area

≡

Merchant Area

Orders to be delivered: 3

Total Number of Orders: 24

Total Sales: \$12.00

COMPLETED ORDERS	ACTIVE ORDERS	REVENUE REPORT	TOP SELLING
#1 Cheese Pizza \$10.00 X 10			\$100.00
#2 Dry Bun \$5.00 X 3			\$15.00
#3 Submarine \$5.00 X 2			\$10.00

⚙ Store Settings

Mart Online Shopping  
©2017

Links

Explore Foods  
Explore Shops  
Merchant Area  
Admin Area

More Links

## Admin

Mart Online

Explore Shops Explore Foods Merchant Area Admin Area

Login > Admin Area

≡

### Admin Area

Users  
3  
(This month)  
**12**  
(Total)

Merchants  
3  
(This month)  
**12**  
(Total)

Orders  
3  
(This month)  
**12**  
(Total)

Sales  
**\$300**  
(This month)  
**\$1200**  
(Total)

COMPLETED ORDERS      ACTIVE ORDERS      REVENUE REPORT      TOP PRODUCTS SELLING

Search Anything on Orders

Order #7456845  Order Date : 23-06-2017 Delivery Time : 05:00 PM Order Total : \$14  08 days 12:15:12
Order #7656846  Order Date : 11-05-2017 Delivery Time : 08:00 PM Order Total : \$54
Order #7756845  Order Date : 21-05-2017 Delivery Time : 07:00 PM Order Total : \$24

Manage Store

Store Settings

**Mart Online Shopping**  
©2017

**Links**

Explore Foods  
Explore Shops  
Merchant Area  
Admin Area

More Links

## Admin revenue report

Mart Online

Explore Shops Explore Foods Merchant Area Admin Area

Login > Admin Area

≡

### Admin Area

Users  
3 (This month)  
**12** (Total)

Merchants  
3 (This month)  
**12** (Total)

Orders  
3 (This month)  
**12** (Total)

Sales  
\$300 (This month)  
**\$1200** (Total)

---

COMPLETED ORDERS ACTIVE ORDERS REVENUE REPORT TOP PRODUCTS SELLING

Show entries Search:

Shop ID	Shop Name	City	Contact	Total No Orders	Total Revenue
S0564784	Shandong	Rajagiriya, Colombo	0117845985	45	\$1500.87
S0564784	Chinease Dragon	Bambalapitiya, Colombo	0112547896	25	\$1009.00
S0564785	Pizza Hut	Malabe, Colombo	0112729729	41	\$1800.00

Showing 1 to 3 of 3 entries Previous **1** Next

Manage Store

Store Settings

Mart Online Shopping  
©2017

Links

Explore Foods  
Explore Shops  
Merchant Area  
Admin Area

More Links

# Admin Revenue Report

Mart Online

Explore Shops Explore Foods Merchant Area Admin Area

Login > Admin Area

≡

## Admin Area

 Users  
3  
(This month)  
12  
(Total)

 Merchants  
3  
(This month)  
12  
(Total)

 Orders  
3  
(This month)  
12  
(Total)

 Sales  
\$300  
(This month)  
\$1200  
(Total)

---

COMPLETED ORDERS ACTIVE ORDERS REVENUE REPORT TOP PRODUCTS SELLING

Show entries Search:

Shop ID	Shop Name	City	Contact	Total No Orders	Total Revenue
S0564784	Shandong	Rajagiriya, Colombo	0117845985	45	\$1500.87
S0564784	Chinease Dragon	Bambalapitiya, Colombo	0112547896	25	\$1009.00
S0564785	Pizza Hut	Malabe, Colombo	0112729729	41	\$1800.00

Showing 1 to 3 of 3 entries Previous 1 Next

 Manage Store

 Store Settings

Mart Online Shopping

@2017

Links

Explore Foods  
Explore Shops  
Merchant Area  
Admin Area

More Links

## Delivery Agent Active Orders

Mart Online

Explore Shops Explore Foods Merchant Area Admin Area

Login > Delivery Agent Area

### Delivery Agent Area

Orders to be delivered: 3

Total Number of Delivered Orders: 24

Total Sales: \$12.00

COMPLETED ORDERS

ACTIVE ORDERS

Order Details	Order Date : 23 - 06 - 2017 Delivery Time : 05:00 PM
 <p>08 days 12:06:02 Order #784754 Order of Chinese Dragon</p> <p>Contact Person : Chamith Chathuka Delivery Address : 69/1 Horahena Road Hokandara East Phone Number : 0702178300</p>	<p>Delivery Charges : \$2 Order Total : \$14 Paid Status : Paid</p>
 <p>08 days 12:06:03 Order #784754 Order of Shandong</p> <p>Contact Person : Chamith Chathuka Delivery Address : 69/1 Horahena Road Hokandara East Phone Number : 0702178300</p>	<p>Order Date : 23 - 06 - 2017 Delivery Time : 05:00 PM</p> <p>Delivery Charges : \$2 Order Total : \$16 Paid Status : Paid</p>

Mart Online Shopping

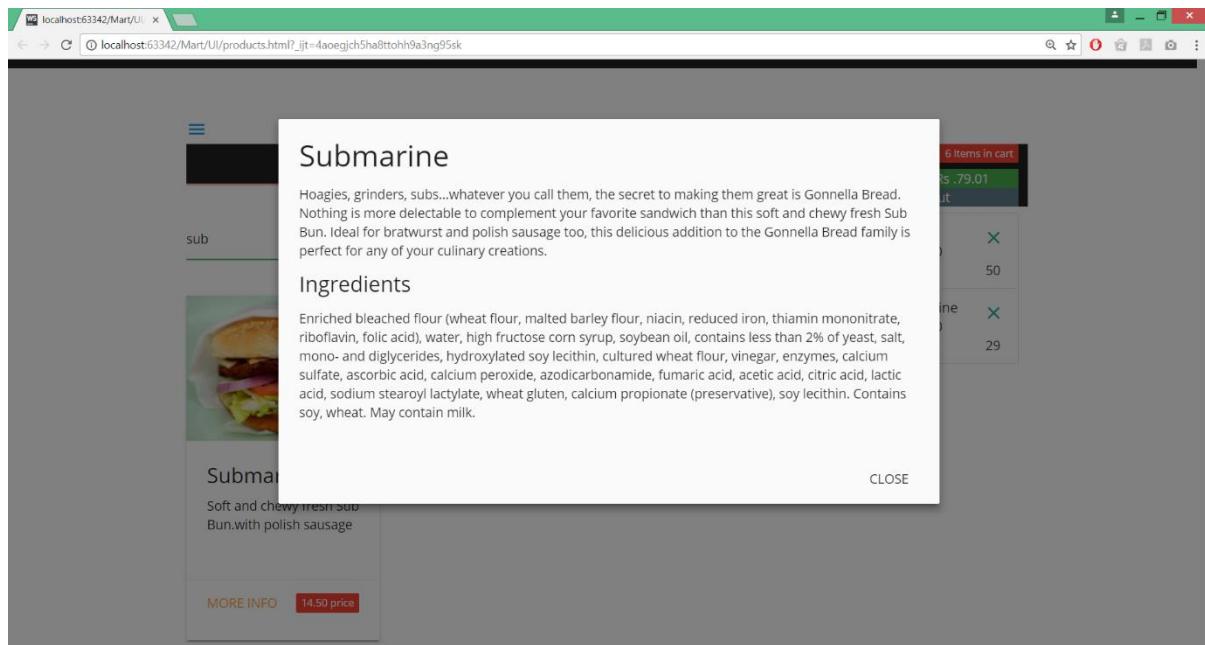
@2017

Links

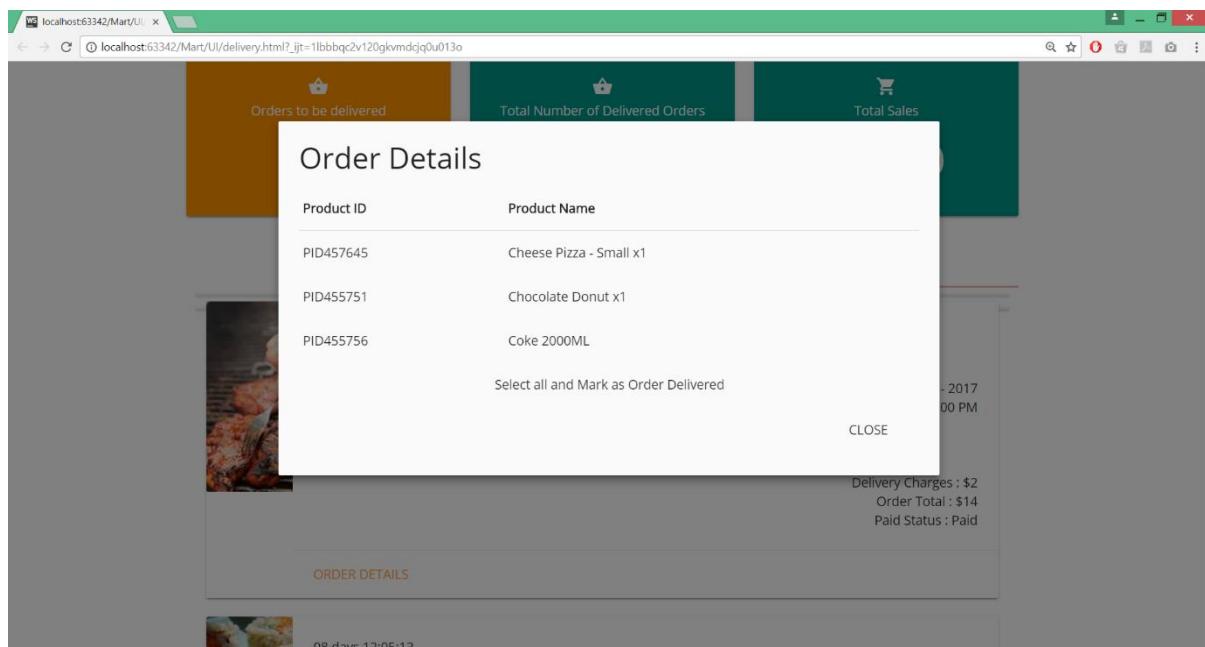
Explore Foods  
Explore Shops  
Merchant Area  
Admin Area

More Links

## Explore Food – More details



## Detailed Order information – For Admin/ Merchant (Shop Owner)/Delivery Agent/ Product ordered customer.



## **Appendix Questioner Form**

### **Online Ordering Survey**

\* Required

Which category below includes your age?

Below 18

18-25

25-35

35-50

50-60

Above 60

Have you ever ordered food online?

Yes

No

What are the things you like to order online? \*

Electronic Items

Cuisine (Restaurant Food)

Grocery Items

Clothes

Supplements

Other:

Which of the following according to you is the most preferred method of ordering food?

Through an online web portal

Phone call to the restaurant

Pick up

Dine in

Other:

What is your preferred way of ordering online? \*

Through Mobile Phone App

Through Web Site

Do you like shopping at Super Markets? \*

Yes

No

Other:

How you like to pay for your Ordered items? \*

Credit or Debit Card - Online

Pay Cash - On Delivery

Pay by Cred/ Debit card on delivery

Other:

Why do you like to order food from an online web portal?

Less risk of mistakes made with order when placed online

It is easier than talking to a live person

You can see the entire menu

Process of ordering is faster

Online orders let you skip the line and let you avoid being put on hold

Other (please specify)

How many times a month do you order a delivery?

Less than once a month

4-5 times a month

Almost every other day

Other:

What kind of uncertainties do you face while ordering food online?

Confirmation that the order has been placed

What the status of the order is

Other:

## Summary of responses received for the survey.

50 responses



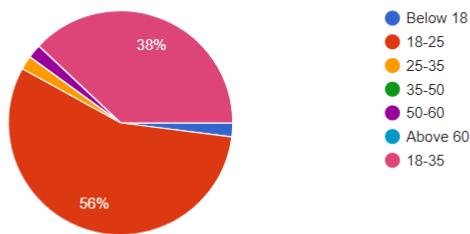
SUMMARY INDIVIDUAL

Accepting responses



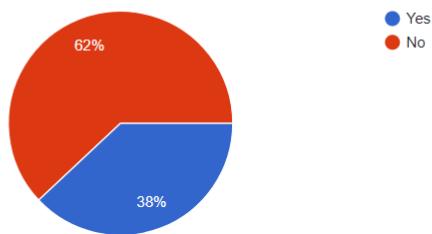
### Which category below includes your age?

50 responses



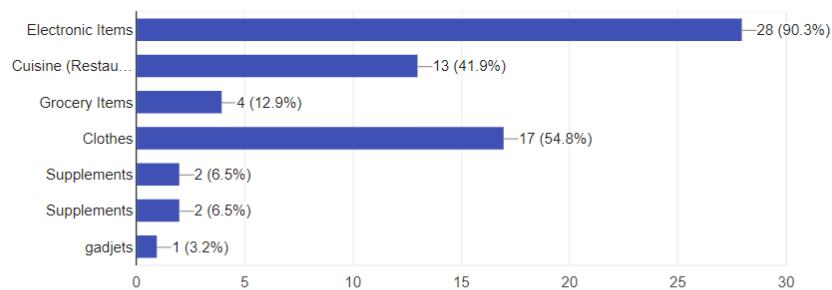
### Have you ever ordered food online?

50 responses



### What are the things you like to order online?

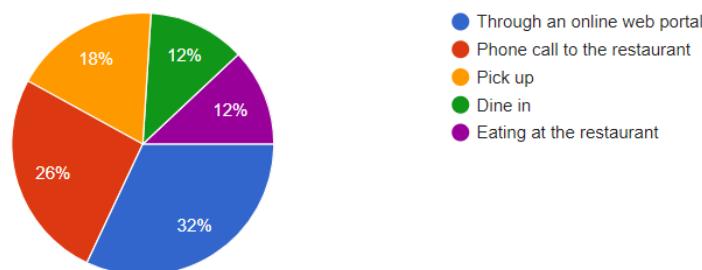
31 responses





Which of the following according to you is the most preferred method of ordering food?

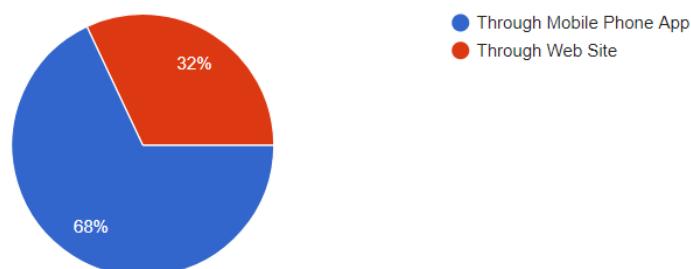
50 responses



---

What is your preferred way of ordering online?

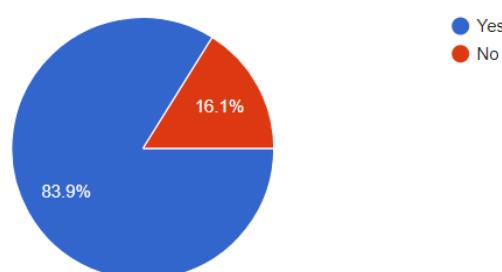
50 responses



---

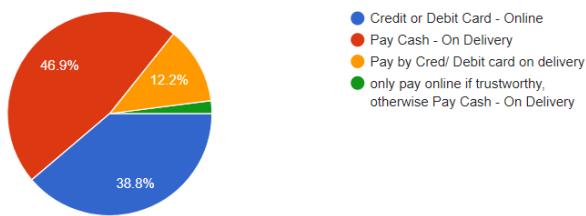
Do you like shopping at Super Markets?

31 responses



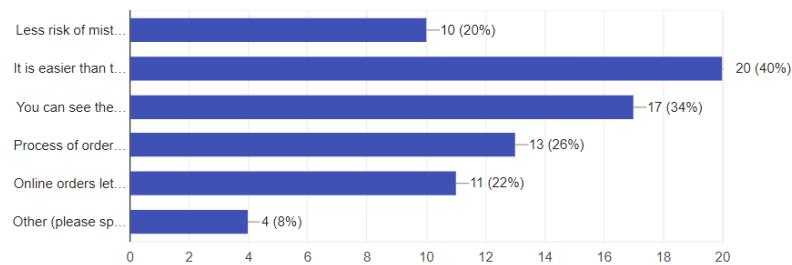
### How you like to pay for your Ordered items?

49 responses



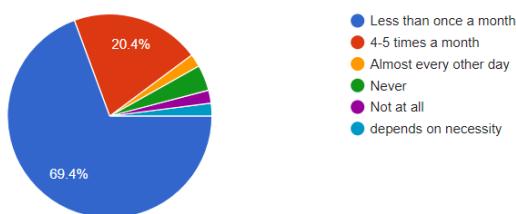
### Why do you like to order food from an online web portal?

50 responses



### How many times a month do you order a delivery?

49 responses



### What kind of uncertainties do you face while ordering food online?

49 responses

