**Mobile Intelligent Shopping System**

University logo

**Developed by:**

**Supervised by:**

**Dated:** June,

**Computer Science Department**

# **Acknowledgement**

First, I would like to show my gratitude to God for all his blessings. I would like to express my deepest concern and appreciation to all those who provided me the possibility to complete this project. A special gratitude to my project manager, Name, whose contribution in simulating suggestions and encouragement helped me throughout the project development.

Furthermore, I would like to highlight the efforts of my colleagues and friends. Without their support and encouragement I was unable to reach there. Last but not least, I have to appreciate the guidance given by other Supervisors as well as the panels especially in our project presentation that has improved our presentation skills. Thanks to their comment and advices.

# **Abstract**

A business is incomplete without having a website or application. It is one of the most powerful marketing tools available. The main reason that it is important to have an application is how people are likely to find you. Now a days, most people will go online and research products and companies before they make a purchase or any other help. For this reason of dependency on online, we have develop a web application that allow users to register themselves to access system features. MIS System is a web-based application that provides facility to the customers to add their shopping list and gives the price details of the products along with new deals and notification. In addition, the system can give the customer a recommended list from the trends found for a shopping round, such as a ‘BBQ party’ or ‘Home supplies’ they may be getting low on and coupons using data association techniques. Moreover, the system will predict the customer’s specific shopping list and give them promotional offers by using some data mining techniques to their mobile. When a customer wants to buy their shopping list without wasting any time, they get the price details of each product from anywhere. From a manager’s perspective this tool can be used to promote their store and their products. Moreover, they can get a better idea of the business and recover the weak areas of their business by getting intelligence reports from the data mined information. These reports will give help to the managers to run their businesses more effectively. There is an Admin, who is able to manage all the customers and managers as well as all the products. The system records all the data and information details to the database. The system is user friendly including all these features, which make this application convenient and easy to use. Since it is a web-based application therefore, it is accessible from anywhere in any device, with internet connection.

**List of Tables**

[Table 1: Register Use Case 11](#_Toc518178173)

[Table 2: Login Use Case 12](#_Toc518178174)

[Table 3: Remove Product Use Case 13](#_Toc518178175)

[Table 4: Enter Manage Details Use Case 13](#_Toc518178176)

[Table 5: Change Personal Details Use Case 14](#_Toc518178177)

[Table 6: Change Password Use Case 15](#_Toc518178178)

[Table 7: Add Item Use Case 16](#_Toc518178179)

[Table 8: Change Password Use Case 16](#_Toc518178180)

[Table 9: Add Coupon Use Case 17](#_Toc518178181)

[Table 10: Manage Order Use Case 18](#_Toc518178182)

[Table 11: Display Graph and Charts Use Case 18](#_Toc518178183)

[Table 12: Add Store Use Case 19](#_Toc518178184)

[Table 13: Map Product 19](#_Toc518178185)

[Table 14: Manage User Use Case 20](#_Toc518178186)

[Table 15: Delete User 21](#_Toc518178187)

[Table 16: Change Personal Details Use Case 21](#_Toc518178188)

[Table 17: Import Product Use Case 22](#_Toc518178189)

[Table 18: Manage Featured Coupon Use Case 23](#_Toc518178190)

[Table 19: Logout Use Case 23](#_Toc518178191)

[Table 20: Tools and Techniques 29](#_Toc518178192)

[Table 21: Login Test Case 42](#_Toc518178193)

[Table 22: Manage Details Test Case 43](#_Toc518178194)

[Table 23: Add Product Test Case 43](#_Toc518178195)

[Table 24: Add Item Test Case 44](#_Toc518178196)

[Table 25: Add Coupon Test Case 44](#_Toc518178197)

[Table 26: Manage Order Test Case 44](#_Toc518178198)

[Table 27: Map Product Test Case 45](#_Toc518178199)

[Table 28: Add Store Test Case 45](#_Toc518178200)

[Table 29: Manage User Test Case 46](#_Toc518178201)

[Table 30: Import Product Test Case 46](#_Toc518178202)

[Table 31: Change Password Test Case 47](#_Toc518178203)

[Table 32: Remove Product Test Case 47](#_Toc518178204)

[Table 33: Non-Functional Requirement Test Case 48](#_Toc518178205)

[Table 34: GUI Requirement Test Case 48](#_Toc518178206)

**List of Figures**

[Figure 1: Stages of Waterfall Model 7](#_Toc518184559)

[Figure 2: Use Case Diagram 11](#_Toc518184560)

[Figure 3: Architectural Design 26](#_Toc518184561)

[Figure 4: Sequence Diagram 27](#_Toc518184562)

[Figure 5: Class Diagram 28](#_Toc518184563)

[Figure 6: Register 30](#_Toc518184564)

[Figure 7: Login 30](#_Toc518184565)

[Figure 8: Product Details 31](#_Toc518184566)

[Figure 9: Coupon 32](#_Toc518184567)

[Figure 10: Order 32](#_Toc518184568)

[Figure 11: User 33](#_Toc518184569)

[Figure 12: Map Product 34](#_Toc518184570)

[Figure 13: Review 34](#_Toc518184571)

[Figure 14: Advertise Brand 35](#_Toc518184572)

[Figure 15: Map 36](#_Toc518184573)

**Table of Contents**

[**Acknowledgement** i](#_Toc518184574)

[**Abstract** ii](#_Toc518184575)

[**1.** **Introduction** 1](#_Toc518184576)

[**1.1.** **Problem Definition** 2](#_Toc518184577)

[**1.2.** **Aims and Objectives** 2](#_Toc518184578)

[**1.3.** **Project Scope** 3](#_Toc518184579)

[**2.** **Literature Review** 4](#_Toc518184580)

[**3.** **Issues related to the Project** 5](#_Toc518184581)

[**3.1.** **Existing System** 5](#_Toc518184582)

[**3.2.** **Current System** 6](#_Toc518184583)

[**3.2.1.** **Chosen Software Methodology** 7](#_Toc518184584)

[**4.** **Analysis and Requirements** 8](#_Toc518184585)

[**4.1.** **External Interface Requirements** 8](#_Toc518184586)

[**4.1.1.** **User Interface** 8](#_Toc518184587)

[**4.1.2.** **Software Interface** 8](#_Toc518184588)

[**4.1.3.** **Hardware Interface** 8](#_Toc518184589)

[**4.1.4.** **Communication Interface** 9](#_Toc518184590)

[**4.2.** **Functional Requirements** 9](#_Toc518184591)

[**4.2.1.** **Use Cases** 9](#_Toc518184592)

[**4.2.2.** **Use Case Diagram** 10](#_Toc518184593)

[**4.2.3.** **Use Case Description** 11](#_Toc518184594)

[**4.3.** **Non-Functional Requirements** 24](#_Toc518184595)

[**4.3.1.** **Reliability** 24](#_Toc518184596)

[**4.3.2.** **Availability** 24](#_Toc518184597)

[**4.3.3.** **Maintainability** 24](#_Toc518184598)

[**4.3.4.** **Security** 25](#_Toc518184599)

[**4.3.5.** **Portability** 25](#_Toc518184600)

[**4.3.6.** **Performance** 25](#_Toc518184601)

[**4.3.7.** **Database Requirement** 25](#_Toc518184602)

[**5.** **Design** 25](#_Toc518184603)

[**5.1.** **System Architecture Design** 25](#_Toc518184604)

[**5.1.1.** **Chosen System Architecture** 25](#_Toc518184605)

[**5.2.** **Sequence Diagram** 26](#_Toc518184606)

[**5.3.** **Class Diagram** 27](#_Toc518184607)

[**6.** **Implementation** 28](#_Toc518184608)

[**6.1.** **Tools and Techniques** 28](#_Toc518184609)

[**6.2.** **Coding Explanation** 29](#_Toc518184610)

[**6.2.1.** **Register:** 29](#_Toc518184611)

[**6.2.2.** **Login:** 30](#_Toc518184612)

[**6.2.3.** **Product Details:** 31](#_Toc518184613)

[**6.2.4.** **Coupon:** 31](#_Toc518184614)

[**6.2.5.** **Order:** 32](#_Toc518184615)

[**6.2.6.** **User:** 33](#_Toc518184616)

[**6.2.7.** **Map Product:** 33](#_Toc518184617)

[**6.2.8.** **Review:** 34](#_Toc518184618)

[**6.2.9.** **Advertise Brand:** 35](#_Toc518184619)

[**6.2.10.** **Map:** 35](#_Toc518184620)

[**6.3.** **Application Screenshots** 36](#_Toc518184621)

[**7.** **Testing** 37](#_Toc518184622)

[**7.1.** **Black Box Testing** 37](#_Toc518184623)

[**7.2.** **White Box Testing** 37](#_Toc518184624)

[**7.3.** **Test Approach** 38](#_Toc518184625)

[**7.4.** **Test Plan** 38](#_Toc518184626)

[**7.4.1.** **Features to be tested:** 38](#_Toc518184627)

[**7.4.2.** **Features not to be tested:** 39](#_Toc518184628)

[**7.5.** **Specific Test Cases** 39](#_Toc518184629)

[**8.** **Conclusions** 46](#_Toc518184630)

[**8.1.** **Discussion and Summary** 46](#_Toc518184631)

[**8.2.** **Possible Future work and Enhancements** 47](#_Toc518184632)

[**9.** **References and Bibliography** 47](#_Toc518184633)

# **Introduction**

This document provides a complete software requirement specification on Mobile Intelligent Shopping System. In this document, a complete description of product introduction, literature review, aims and scope, requirements, design, implementation, testing and conclusion are given for complete understanding of the project [1].

We all know that in the field of technology, smart phones with killer applications rules over the world. Ranging from online banking to online shopping or other home automation applications are useful and powerful. Usage of these kind of applications are cost efficient, attractive and convenient. So, our project is also based on intelligent shopping system [2].

Mobile Intelligent Shopping (MIS) System is a mobile based shopping application that brings a different experience to customers and managers. It focuses on providing a solution to complex and time wasting shopping behaviour of the people. The system is developed for the customers and managers/store owners to reduce their work load and provide ease in today’s life.

MIS System is a web-based application that provides facility to the customers to add their shopping list and gives the price details of the products along with new deals and notification. In addition, the system can give the customer a recommended list from the trends found for a shopping round, such as a ‘BBQ party’ or ‘Home supplies’ they may be getting low on and coupons using data association techniques. Moreover, the system will predict the customer’s specific shopping list and give them promotional offers by using some data mining techniques to their mobile. When a customer wants to buy their shopping list without wasting any time, they get the price details of each product from anywhere.

From a manager’s perspective this tool can be used to promote their store and their products. Moreover, they can get a better idea of the business and recover the weak areas of their business by getting intelligence reports from the data mined information. These reports will give help to the managers to run their businesses more effectively, gain the maximum use from existing data for the future, inform about cutting edge trends and legislation information to keep them aware of the changing business climate and help them market their promotions and new products.

Both customer and manager have to register themselves in order to use the system. There is an Admin, who is able to manage all the customers and managers as well as all the products. The system records all the data and information details to the database. The system is user friendly including all these features, which make this application convenient and easy to use. Since it is a web-based application therefore, it is accessible from anywhere in any device, with internet connection.

In this report, Problem Definition, Aims and Objective and Scope of the project will be discussed in next sections. Under Literature Review section, a good review and research on related projects will be provided, for understanding of existing. After that, a comparison of Existing System and Current System will be discussed next, which are helpful in project development. Moving further, Analysis and Requirement are gathered and will be explained in next section. As it is a very important part on which project depends. The conclusion of analysis will lead through the Design phase and then Implementation phase, which are based on these requirements. Each step will explain to show the complete flow of the system. Results of these Design and Implementation phase will discuss in the next section, to show the outcome of previous phase. Later on, Testing approach will justify and testing methods will explain under next section. Last but not least, Conclusion part will explain the overall project understanding and development that was faced during each phase. Appendix and Bibliography will be mentioned at the end for the reference.

## **Problem Definition**

Currently, there is a lot of work load and burden on store managers because they are unable to maintain records automatically. A lot of time was wasted for the fast going people, who don’t have time to rush to shopping centres and buy products. There should be a solution to these obstacles.

There is need of developing a web-based application which supports many different data mining techniques to read customer orders and produce trends for promotional offers, as well as sending reports to the managers to help them in running their business with all this big data that has been provided to them.

Finding a solution to current complex and time-wasting shopping behaviour of the customers as well as making it accessible, easy to use and understandable will be beneficial to the customer in helping them with their shopping.

Store owners can also gain the maximum use of sales data by utilising the algorithms techniques of data mining. Also, it will make it easier for the owners to track their regular customer base.

This system is for customers and store managers. The system will help to reduce their work load, make targeted deals easier to see and make everyday life easier. This will also be beneficial to small store owners as they may not have the resources such as money and technologies to employ such a system e.g. Smart basket system and have their store marketed in this fashion. However, it does assume the store owners to already have a database or create one using the system.

## **Aims and Objectives**

The main goal of this system is to design and develop a web-based mobile application that will help the customer to add their shopping list and to speed up the existing system. The major aims and objective list is mention below:

* To design and implement a web-based system that will use customer shopping information to data mine trends and output specific promotional offers to the customer.
* To output the trends to the managers who will use the application in a report style, to manage, improve and help their business further.
* To make the application user friendly that will be easy to use, navigate and understand so the customers can easily input their information as well as retrieve information from the website.
* To cut down the unwanted hassle of spending time grocery shopping. This would entice them to use the application more and more.
* To make this app convenient to use, so making it a web application means it is accessible on any device so long as they have an internet connection.
* To produce a software that manages the sales activity done in a supermarket, maintaining the stock details, maintaining the records of the sales done for a month/year.
* To make it efficient so that the users will consume less time in calculation
* To complete the sales activity within a fraction of seconds, whereas manual system makes the user to write it down which is a long procedure and so paper work will be reduced and the user can spend more time on the monitoring the supermarket.
* To register the user and let them manage their profiles

## **Project Scope**

The major part of the project planning that involves determining and documenting a list of specific project goals, deliverables and tasks is called Scope of the project. It will be a web based application that any user can easily use it from anywhere. The scope of this project is divided as follows:

* It will be a very useful tool for adding a shopping list and products that the customer’s desire and getting back the deals from the one provided from the system.
* It will recommend a list to the customers from the trends found for a shopping round, such as a ‘BBQ party’ or ‘Home supplies’
* It will be able to give coupons using data association techniques.
* Moreover, it will predict the customer’s specific shopping list and give them promotional offers by using some data mining techniques to their mobile.
* It will be used to promote store and its products.
* It will generate an intelligence report that helps manager to get a better idea of the business and recover the weak areas of their business. These reports will give help to the managers to run their businesses more effectively, gain the maximum use from existing data for the future.
* It will notify customers about new offers and products.

Besides Scope, there are boundaries and limitation for this project. These are as follows:

* User needs to be registered first in order to use the system.
* There will be only grocery products for shopping
* Notifications will be sent to only registered users.

Stakeholders for this project are Users, who will user this application, Supervisor, who supervises this projects and Teachers, who helps in project development. User and Manager will input through and view output results on display screen.

# **Literature Review**

Literature review is an important part of software development life cycle as it helps in collecting and acquiring useful information to develop and maintain a project. it is a description of relevant research on a specific topic and gives an overview of what has been said and discovered by the researchers and developers.

For this project, history shows a large number of researches and experiments performed in this field area. Many of them were failed due to various reasons. So the purpose of developing this system is to overcome those problems.

Dr. Suryaprasad J, founder of “A Novel Low Cost Intelligent Shopping Cart” [3], proposed a system that assist the customer to search the product from the given list with their respective price list and able to select them. It was a low cost intelligence based shopping card that all the selected items were moved to the cart automatically.

After that, Amine Karmouche suggested a solution that is able to scan both dynamic and static products, included in shopping area. It was used with the help of RFID Reader antennas and performed aisle-level scanning on the cart, rather than conducting the RFID observation on each individual cart [4].

Mr. P Chandrasekar was another researcher who provides the information in “Smart Shopping Cart with Automatic Billing System through RFID and ZigBee” [5] to develop a shopping cart having PID (Product Identification Device) that includes microcontroller, LCD, RFID and ZigBee. RFID reader was able to read the purchasing information in shopping cart. All the product information was stored in EEPROM, which sends all the data to Central Billing System through ZigBee module. This CBS gets the information and access the database of the products to calculate the total amount of purchased items.

Another researcher Satish Kamble described in his book, “Developing a Multitasking Shopping Trolley System” [6] to develop a system that assist the user in everyday life in shopping. It was a great help in purchasing items online. The main purpose was to develop a proposed system to provide ease and convenience for assisting shopping.

In today’s world, online shopping is one of the most famous shopping types. But still having some problems that customer faced. For example, Fujitsu (2005) proposed a system of revolutionary shopping with the emergence of their U-Scan Shopper. It was wireless, trolley-mounted computer that gave customers a useful information and provide convenience as they move through a store. It also notifies them about special deals and reminds them about their history purchase. There were built-in scanners, digital store directories and wireless communication that has been around a while but hasn't really caught on in a big way. These carts tend to be expensive, nearly $1,200 per-cart.

Later, IBM proposed "Wireless web page" that permit users to add their shopping lists from anywhere and view them at the store using the mobile devices [7]. The cart was a great help in assisting the consumers to locate the items with the help of a map embedded in the system. Despite their high costs, these system were rendered the simple but useful service of displaying product related information and their location. This information helped intelligent shopping systems to understand how they were integrated as physical hardware. There was also a limitation to this that a hardware system can retain, such as price to the store owner and convenience to the customer. As they cannot have the same cart experience at every store they go to shop.

There was another application, named iGrocer. It was a mobile application that runs on a Smartphone having bar-code reader [8]. Customers were allowed to view product information using bar code reader to scan item barcodes. More importantly, it has a list of useful features such as nutrition profiling, personalized product categories etc. “iGrocer” architecture is a classic example of the client-proxy server architecture with the proxy server handling all the intermediate communication. Customers were able to create a profile from either the phone client or the through the iGrocer website. The new user information is then sent to the proxy, that later stores it to the database. The information also included customer nutrition profiles, wish lists, recipes, purchase history and the shopping list.

All these researches gave new ideas for this project. For example, save the products as a shopping list and make it cheapest available to them. It also gave an idea of improving data mining techniques which will be used in this project. However, iGrocer was used as more of a personal nutritionist rather than a smart shopping system and so the focus wasn’t on convenience or the shopping experience as these were just a bi product from the actual focus.

# **Issues related to the Project**

Given below is the comparison between issues in existing system and solution in current system.

## **Existing System**

We have seen that shopping at big malls is becoming a daily activity in big cities. A lot of hassle has been seen at the malls, especially on holidays and weekends. Same at the billing counter. People now a days are in hurry and have no time to spend a lot of time at shopping malls. We have surveyed some of the existing systems in previous section.

According to a survey, some of the systems were used as more of a personal nutritionist rather than a smart shopping system and so the focus wasn’t on convenience or the shopping experience as these were just a bi product from the actual focus.

Also, some of them permit users to add their shopping lists from anywhere and view them at the store using the mobile devices. The cart was a great help in assisting the consumers to locate the items with the help of a map embedded in the system. Despite their high costs, these system were rendered the simple but useful service of displaying product related information and their location. This information helped intelligent shopping systems to understand how they were integrated as physical hardware. There was also a limitation to this that a hardware system can retain, such as price to the store owner and convenience to the customer. As they cannot have the same cart experience at every store they go to shop.

Beside these systems, there is still need a system that contains all these features in a single application that will efficient and convenient for users. There are some reasons that proves the need of a new system:

* To develop a smart intelligence based shopping system
* To reduced cost
* To increase efficiency
* To exchange paper work with computer work/storage
* To save time management
* To generate report on a daily basis
* To facilitate online payment
* To use data mining techniques for shopping
* To create a personalised shopping space with login system

## **Current System**

We have study the existing systems and their problems because it is always necessary to study and recognize the problems of existing system, which will help in finding out the requirements for the new system. System study helps in finding different alternatives for better solution.

MIS system is a user friendly application. It saves user’s time and helps to maintain its products and features. It includes all the individual functionalities of existing systems with the combination of new ones. This app is very simple in design and also to implement. The main objective of building this application is to upgrade the existing one and enhance its features.

In this application, both customer and manager have to register themselves in order to use the system. There is an Admin, who is able to manage all the customers and managers as well as all the products. The system records all the data and information details to the database. The system is user friendly including all these features, which make this application convenient and easy to use. Since it is a web-based application therefore, it is accessible from anywhere in any device, with internet connection.

The system will allow the manager to add and maintain the products and allow the customer to add these items in their shopping list. Managers will also get a better idea of the business and recover the weak areas of their business by getting intelligence reports from the data mined information. These reports will give help to the managers to run their businesses more effectively, gain the maximum use from existing data for the future, inform about cutting edge trends and legislation information to keep them aware of the changing business climate and help them market their promotions and new products.

This application will provide a facility to the customers to add their shopping list and gives the price details of the products along with new deals and notification. In addition, the system can give the customer a recommended list from the trends found for a shopping round, such as a ‘BBQ party’ or ‘Home supplies’ they may be getting low on and coupons using data association techniques. Moreover, the system will predict the customer’s specific shopping list and give them promotional offers by using some data mining techniques to their mobile. When a customer wants to buy their shopping list without wasting any time, they get the price details of each product from anywhere.

### **Chosen Software Methodology**

Waterfall process model is chosen and will be used for the development of this project. This model is preferred because it is simple, easy and reasonable approach, especially when the requirements are well understood. It is a straight forward model which means there is no risk and no change in requirements for this project. Each phase of this model has specific deliverables and review process. It also allows easy testing and analysis and saves significant amount of time.

Waterfall process model is the traditional methodology which is popular version of software development life cycle. This is a classic model which works in a linear sequential flow. Every time new phase begins only if the previous one is completed. There is no need to go back and change the requirements.

The waterfall model is the first process model which is very simple to understand and use. It is used when the requirements are very clear and there is no need of any change. It is mostly used in small system or projects which requires no change in requirements. It is most understandable, easy and simple model for the beginners. The testing can also be done very easier.

Figure : Stages of Waterfall Model

# **Analysis and Requirements**

Analysing the requirements of the Mobile Intelligent Shopping (MIS) System, the list of functional and non-functional requirements have been identified as necessary for the initial version of the application. These are presented below:

## **External Interface Requirements**

External interface requirements includes User interface, Software interface and Hardware interface requirements.

### **User Interface**

User interface is everything that is designed to the information system, which a person may interact. We all know that presentation and immersion is everything. An application becomes more attractive due to its presentation. So, for this system,

* A visual GUI will be used to display menus and sub menus.
* Buttons, check boxes and drop down menus will be displayed for selecting options.
* Backgrounds will be used for making UI more attractive.
* Screens will be updated to display the output results of the input
* Map will be embedded with attractive icons and options.

### **Software Interface**

The application is a web based so it will be run on any operating system, which includes proper working browser. It will be implemented using PHP and HTML/CSS languages. An internet connection is most important as this system works with internet connection. The system will use MySQL as its database.

### **Hardware Interface**

The hardware requirement is very simple for this kind of system. As the system is a web application, so it runs on any hardware that can runs a simple browser. Although the hardware should be good enough to handle the peak times of web servers. Some of the requirements are as follows:

* At least 1GB ram and 100MB memory for any operating system
* Dual core processor
* Fast Internet connection

### **Communication Interface**

The main communication protocol is HTTP. This is used to transfer data and information back and forth, from client to server. HTTP GET and POST is used to send the information over the web browser securely.

## **Functional Requirements**

Following are the functional requirements for this system:

* The system will be able to allow customers and managers to register themselves along with their personal details. An unregistered person will not be able to access system features, both frontend and backend.
* The system will be able to let users add item into their shopping list.
* Users will also able to view and manage their shopping lists under their account
* The system will be able to display deals, coupons and featured coupons. Deals will be from other stores, coupons will be made manually by the store manager and featured coupons will be made by the system using a data association technique.
* The system will allow managers to add product with its details into the system, which will be viewed by the customers.
* The system will be able to store each product details to the database automatically.
* The manager will be able to manage orders from the customers. He will be able to see the shopping list that customer has made.
* The system will be able to generate reports and graphs and display it to the manager.
* The system will allow admin to view the seller invoices, along with the customer’s shopping list.
* The system will also able to let admin manage the multiple stores and managers, including their details.
* Admin will be able to manage customers as well in order to update, modify and delete their details.
* Admin will be able to manage coupons.
* The system will be able to allow admin to map each product on the website for manager and customers.
* Admin will be able to manage products and their categories.
* The system will be able to populate external party database information

### **Use Cases**

In software engineering, a use case is a series of actions and events that defines the interactions between actor and a system to achieve a specific goal. It is a methodology to identify, clarify and organize system requirements. Actor can be a human or other external system. The following use cases are chosen which are best fit for the system,

* Register
* Login
* Manage details
* Add product
* Add item
* Add coupon
* Manage order
* Map product
* Add store
* Manage user
* Import product
* Change password
* Manage featured coupon
* Remove product
* Logout

### **Use Case Diagram**

A Use Case Diagram is the graphical representation of the interaction of different elements of a system between them. These elements can be an Actor, Use Cases and others. A Use Case is used to analyse and identify the specific tasks used in the system, by the user. The purpose of this diagram is to depict the way of interaction of the user with the system.

Here, the Use Case Diagram of our system is shown below which shows the task or services (use cases), provided by the system. These services are done by specific user (actor).

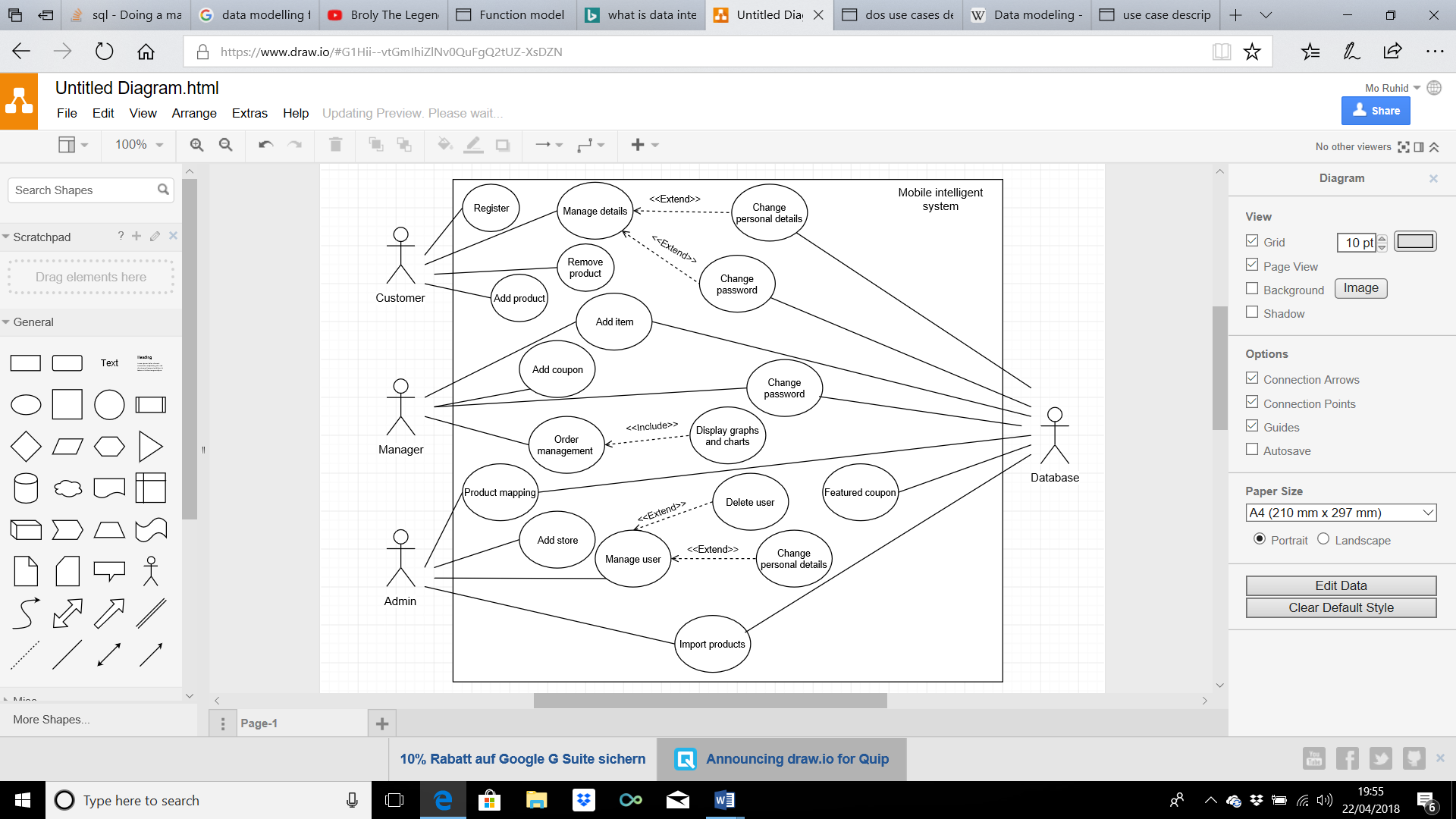


Figure : Use Case Diagram

### **Use Case Description**

Following are the description of each use case:

Table : Register Use Case

|  |  |
| --- | --- |
| **Name** | Register |
| **ID** | UC\_001 |
| **Description** | When wanting to complete their shopping list, a customer must register to the system |
| **Primary Actor** | Customer |
| **Secondary Actor** | N/A |
| **Pre-Condition** | The user is on a screen where there is a sign-up option |
| **Post Condition** | The user is signed up to the system and the previous page they were on will be displayed again |
| **Trigger** | Customer selects to sign up |
| **Main Success Scenario** | 1. Customer is navigating the website where there is an option to sign up 2. The customer is then prompted to sign up using their details 3. Verification successful 4. The customer has made an account and is directed to their previous page |
| **Variation** | 2a – Customer tries to sign up with existing details  2b – Error message pops up informing the customer they already have these details used  2c – Return to Main success scenario step 2 |

Table : Login Use Case

|  |  |
| --- | --- |
| **Name** | Login |
| **ID** | UC\_002 |
| **Description** | Customer will be able to login to the application |
| **Primary Actor** | Customer |
| **Secondary Actor** | N/A |
| **Pre-Condition** | Customer must be registered |
| **Post Condition** | Customer has been logged in successfully |
| **Trigger** | Customer opens login menu |
| **Main Success Scenario** | 1. Customer enters username 2. Customer enters password 3. Customer selects Login option 4. System receives credentials 5. System verifies credentials from database 6. Login successful |
| **Variation** | 1a – Customer enters invalid username  2a – Customer enters invalid password |

Table : Remove Product Use Case

|  |  |
| --- | --- |
| **Name** | Remove Product |
| **ID** | UC\_003 |
| **Description** | When an order is in the shopping list the user can remove products in their shopping list |
| **Primary Actor** | Customer |
| **Secondary Actor** | N/A |
| **Pre-Condition** | The customer is in their shopping list |
| **Post Condition** | The customers shopping list is updated and their product is removed from their shopping list |
| **Trigger** | Customer selects a product to remove from their shopping list |
| **Main Success Scenario** | 1. A customer has made a shopping list which contains at least one product 2. They choose to remove that product 3. The customer is shown their shopping list and the shopping list is updated with the product removed |
| **Variation** | N/A |

Table : Enter Manage Details Use Case

|  |  |
| --- | --- |
| **Name** | Manage Details |
| **ID** | UC\_004 |
| **Description** | Assuming a customer is already signed up to the system they may want to change their personal details as well as their login details |
| **Primary Actor** | Customer |
| **Secondary Actor** | N/A |
| **Pre-Condition** | The customer is on a page where they can see they can see the ‘my account’ button |
| **Post Condition** | The customer is taken back to my account page and the update is successful |
| **Trigger** | Customer selects the ‘my account’ button |
| **Main Success Scenario** | 1. The user is on a page where they can see the ‘my account’ button on the top right of the screen 2. They click on the button 3. They are successfully taken to the ‘my account’ page where they can change their personal details and their login details |
| **Variation** | V1 – Customer tries to check the ‘my account’ page when they are not signed up to the system   1. The user is navigating the website, but their details are not signed up with the system 2. They click the ‘my account’ button 3. They are taken to the page where they will be required to register to the system with their details instead |

Table : Change Personal Details Use Case

|  |  |
| --- | --- |
| **Name** | Change Personal Details |
| **ID** | UC\_005 |
| **Description** | The customer can update their personal details when in the ‘my account’ page |
| **Primary Actor** | Customer |
| **Secondary Actor** | Database |
| **Pre-Condition** | The customer is on the ‘my account’ page and can see the button where it allows the customer to update their personal details |
| **Post Condition** | The customer is taken back to ‘my account page’ and the update is successful |
| **Trigger** | Customer selects the ‘Update personal details’ button |
| **Main Success Scenario** | 1. The user already has an existing account and is in the ‘my account’ page of the system 2. They then click on the ‘update personal details’ button 3. They are then directed to the page where they can update their personal details 4. They click save 5. Verification successful 6. The personal details are successfully updated to the system |
| **Variation** | V1 – Customer tries to check the ‘my account’ page when they are not signed up to the system   1. The user is navigating the website, but their details are not signed up with the system 2. They click the ‘my account’ button 3. They are taken to the page where they will be required to register to the system with their details instead |

Table : Change Password Use Case

|  |  |
| --- | --- |
| **Name** | Change Password |
| **ID** | UC\_006 |
| **Description** | The customer can update their password when in the ‘my account’ page |
| **Primary Actor** | Customer |
| **Secondary Actor** | Database |
| **Pre-Condition** | The customer is on the ‘my account’ page and can see the button where it allows the customer to update their password |
| **Post Condition** | The customer is taken back to ‘my account page’ and the update is successful |
| **Trigger** | Customer selects the ‘password’ button |
| **Main Success Scenario** | 1. The user already has an existing account and is in the ‘my account’ page of the system 2. They then click on the ‘change password’ button 3. They are then directed to the page where they can change their password 4. They click save 5. Verification successful 6. The password is successfully updated to the system |
| **Variation** | V1 – Customer tries to check the ‘my account’ page when they are not signed up to the system   1. The user is navigating the website, but their details are not signed up with the system 2. They click the ‘my account’ button 3. They are taken to the page where they will be required to register to the system with their details instead   V2 – Customer doesn’t verify the password correctly   1. The customer wants to change their password 2. The passwords do not match 3. Return to Main success scenario step 3 |

Table : Add Item Use Case

|  |  |
| --- | --- |
| **Name** | Add Item |
| **ID** | UC\_007 |
| **Description** | The manager can add a product to be displayed on their store on the website |
| **Primary Actor** | Manager |
| **Secondary Actor** | Database |
| **Pre-Condition** | The manager is on the dashboard and can see the ‘add product’ button |
| **Post Condition** | The system is updated, and the manager is taken back to the dashboard |
| **Trigger** | Manager clicks the ‘add product’ button on the dashboard |
| **Main Success Scenario** | 1. The manager has signed in using his login details 2. They have successfully logged in and can view their dashboard 3. They click on the ‘add product’ button 4. They fill in the details of the product and click save 5. The product is successfully added to the database and the website 6. They are taken back to the dashboard |
| **Variation** | N/A |

Table : Change Password Use Case

|  |  |
| --- | --- |
| **Name** | Change Password |
| **ID** | UC\_008 |
| **Description** | The manager can change their password for their login details |
| **Primary Actor** | Manager |
| **Secondary Actor** | Database |
| **Pre-Condition** | The manager is on the main dashboard page and can see the change password button |
| **Post Condition** | The manager is taken back to dashboard and the update is successful |
| **Trigger** | Manager selects the ‘change password’ button |
| **Main Success Scenario** | 1. The manager is on the dashboard where they can see the change password button 2. They click the button 3. They are then directed to the page where they can change their password 4. They click save 5. Verification successful 6. The password is successfully updated to the system |
| **Variation** | V1 – Manager doesn’t verify the password correctly   1. The customer wants to change their password 2. The passwords do not match 3. Return to Main success scenario step 3 |

Table : Add Coupon Use Case

|  |  |
| --- | --- |
| **Name** | Add Coupon |
| **ID** | UC\_009 |
| **Description** | The manager can add their own coupon for their products |
| **Primary Actor** | Manager |
| **Secondary Actor** | N/A |
| **Pre-Condition** | The manager is on the main dashboard page and can see the add coupon button |
| **Post Condition** | The manager is taken back to dashboard and the coupon is successfully added to the website |
| **Trigger** | Manager selects the ‘Add coupon’ button |
| **Main Success Scenario** | 1. The manager is logged into the system and can see various options on his dashboard 2. He clicks the add coupon button 3. He is prompted to add a coupon to be saved under the coupon section of the website 4. He enters the details about the coupon 5. The coupon is successfully added |
| **Variation** | N/A |

Table : Manage Order Use Case

|  |  |
| --- | --- |
| **Name** | Manage Order |
| **ID** | UC\_010 |
| **Description** | The manager can see orders made to their store |
| **Primary Actor** | Manager |
| **Secondary Actor** | N/A |
| **Pre-Condition** | The manager is on the dashboard |
| **Post Condition** | The manager is taken back to dashboard |
| **Trigger** | Manager selects the ‘orders’ button |
| **Main Success Scenario** | 1. The manager is logged in and can view their dashboard 2. The manager selects the order button 3. He is taken to the information about the orders made to his store |
| **Variation** | N/A |

Table : Display Graph and Charts Use Case

|  |  |
| --- | --- |
| **Name** | Display Graphs and Charts |
| **ID** | UC\_011 |
| **Description** | The manager can see the graphs and charts related to his customers order history |
| **Primary Actor** | Manager |
| **Secondary Actor** | N/A |
| **Pre-Condition** | There is some historical data about orders |
| **Post Condition** | N/A |
| **Trigger** | The manager is on the main dashboard page |
| **Main Success Scenario** | 1. The manager is successfully logged in 2. The manager is on the main dashboard page 3. They can then see graphs and charts about their order history |
| **Variation** | V1 – Manager doesn’t see any information   1. The manager is logged on 2. He clicks on the dashboard 3. He can’t see any graphs or charts because there is no historical data 4. Go to Main success scenario step 2 |

Table : Add Store Use Case

|  |  |
| --- | --- |
| **Name** | Add Store |
| **ID** | UC\_012 |
| **Description** | The admin can add stores to the system |
| **Primary Actor** | Manager |
| **Secondary Actor** | N/A |
| **Pre-Condition** | They are in the user management page |
| **Post Condition** | They have added the store successfully and will appear in the manage section |
| **Trigger** | The admin clicks on the add store button |
| **Main Success Scenario** | 1. The admin is logged in 2. The admin can see the user management button on the left-hand side 3. The admin clicks on the user management button and is directed to the page 4. The admin clicks on the button add store 5. He then adds information about the store and clicks save 6. The store is successfully added to the system |
| **Variation** | N/A |

Table : Map Product

|  |  |
| --- | --- |
| **Name** | Product mapping |
| **ID** | UC\_013 |
| **Description** | The admin can change the way the website displays certain products as well as change their categories and description |
| **Primary Actor** | Admin |
| **Secondary Actor** | Database |
| **Pre-Condition** | There is some data of products in the database |
| **Post Condition** | The admin is taken back to the dashboard |
| **Trigger** | The admin clicks on the product mapping button in the user management page |
| **Main Success Scenario** | 1. The admin is logged in and clicks on the user management page 2. The admin then selects the store he wants to edit 3. The admin then selects the product he wants to edit 4. A new page is open were he can change the details of the selected product 5. The product details are successfully updated |
| **Variation** | V1 – Admin doesn’t see any products   1. The admin is in the user management section 2. The admin cannot see any stores or products in that store 3. The pre-condition must be met 4. Then admin is then taken back to the dashboard |

Table : Manage User Use Case

|  |  |
| --- | --- |
| **Name** | Manage User |
| **ID** | UC\_014 |
| **Description** | The admin can change the personal details as well as delete users from the system |
| **Primary Actor** | Admin |
| **Secondary Actor** | N/A |
| **Pre-Condition** | There is some users or stores in the system |
| **Post Condition** | N/A |
| **Trigger** | The admin clicks on the user management page |
| **Main Success Scenario** | 1. The admin is logged in 2. The admin clicks on the user management page 3. The admin is then taken to the page successfully |
| **Variation** | N/A |

Table : Delete User

|  |  |
| --- | --- |
| **Name** | Delete user |
| **ID** | UC\_015 |
| **Description** | The admin can delete customers, managers and stores from the system |
| **Primary Actor** | Admin |
| **Secondary Actor** | Database |
| **Pre-Condition** | The admin must be in the user management page |
| **Post Condition** | The admin is taken back to the dashboard |
| **Trigger** | The admin clicks on the customer, manager or store that is in the system |
| **Main Success Scenario** | 1. The admin is logged in 2. The user can identify the user management screen on the left-hand side and clicks on it 3. They are taken to the new page and can see details about the customer, managers and stores in the system 4. The admin clicks on one of these options 5. The admin is taken to a new page were the information about which category he has clicked in 6. There is an option to delete the user form the system after selecting them 7. The admin chooses to delete the chosen user 8. The record is then deleted from the system |
| **Variation** | N/A |

Table : Change Personal Details Use Case

|  |  |
| --- | --- |
| **Name** | Change personal details |
| **ID** | UC\_016 |
| **Description** | The admin can change personal details of customers, managers and stores from the system |
| **Primary Actor** | Admin |
| **Secondary Actor** | N/A |
| **Pre-Condition** | The admin must be in the user management page |
| **Post Condition** | The admin is taken back to the dashboard |
| **Trigger** | The admin clicks on the customer, manager or store that is in the system |
| **Main Success Scenario** | 1. The admin is logged in 2. The user can identify the user management screen on the left-hand side and clicks on it 3. They are taken to the new page and can see details about the customer, managers and stores in the system 4. The admin clicks on one of these options 5. The admin is taken to a new page were the information about which category he has clicked in 6. There is an option to change personal details of that user 7. The admin chooses the user 8. The admin changes certain details of the user 9. Verification successful 10. The record is updated |
| **Variation** | N/A |

Table : Import Product Use Case

|  |  |
| --- | --- |
| **Name** | Import Product |
| **ID** | UC\_017 |
| **Description** | The admin can import a database of products to a store |
| **Primary Actor** | Admin |
| **Secondary Actor** | Database |
| **Pre-Condition** | The admin must be in a store’s page |
| **Post Condition** | The admin is taken back to the stores page |
| **Trigger** | The admin clicks on the import button |
| **Main Success Scenario** | 1. The admin is logged in 2. The admin clicks on the user management button on the left 3. The admin then chooses a store from the stores already existed 4. The admin clicks on the import products button on the page 5. They are prompted to identify which database it is to be taken from 6. The import is successful |
| **Variation** | N/A |

Table : Manage Featured Coupon Use Case

|  |  |
| --- | --- |
| **Name** | Manage Featured Coupon |
| **ID** | UC\_018 |
| **Description** | The database uses data association technique to display featured coupons |
| **Primary Actor** | N/A |
| **Secondary Actor** | Database |
| **Pre-Condition** | There must be information of products and product categories |
| **Post Condition** | The coupon is displayed in the featured coupon list |
| **Trigger** | The customer has made a shopping list and confirmed it |
| **Main Success Scenario** | 1. The customer has created a shopping list and confirmed it 2. The database uses a data association technique 3. Coupons are then on the certain users featured coupons page |
| **Variation** | N/A |

Table : Logout Use Case

|  |  |
| --- | --- |
| **Name** | Logout |
| **ID** | UC\_019 |
| **Description** | The customer and managers will be able to logout |
| **Primary Actor** | Customer, Manager |
| **Secondary Actor** | N/A |
| **Pre-Condition** | User must be logged in |
| **Post Condition** | User has been logged outs successfully |
| **Trigger** | The customer and manager opens up a main page |
| **Main Success Scenario** | 1. Customer and Manager selects logout option 2. System quits user account 3. System displays login menu |
| **Variation** | N/A |

## **Non-Functional Requirements**

Non-Functional requirements are also called System Quality Attributes. They depicts the overall performance and other feathers of the system that effects run time behaviour. Non-Functional requirements for the system are as follows:

### **Reliability**

The system will perform its intended functions and operations without expecting any error, except Operating system error. All errors will handled in a graceful manner. It will be reliable and achieves at least 95% reliability.

### **Availability**

The system will available all the time with internet connection.

### **Maintainability**

The system will be modified and updated in case of any defects. It will have the ability to change the system components, in order to meet new functionalities. It will easily adapt new features and customization. All upgrades will be simple and safely performed.

### **Security**

The system will provide security to each user. All accounts and user’s data will be safe and secured. The system will provide data integrity as well.

### **Portability**

The system will be portable to any operating system.

### **Performance**

All the system components will perform well and correct the perceived response will immediate and it will not waste valuable resources. Initial interface load time will take less than 2 seconds. 95% of other system functionalities will not take more than 5 seconds.

### **Database Requirement**

MySQL database will be used for registration and data storage for each user.

# **Design**

A Software Design is a written description of a software product that a software designer writes in order to give a software development team overall guidance to the architecture of the software project. It specifies detailed features specifications of smaller pieces of the design. Usually, it contains Architecture design, Design Class diagram and Sequence diagram. The goal is to provide users better understanding of the project at design level.

## **System Architecture Design**

A typical object-oriented information system is designed in terms of several Architectural layers or subsystems. Architectural Diagram is used to represent the components or entities of a system and the interaction between them.

It consists of three layers; Presentation, Domain and Technical services.

### **Chosen System Architecture**

Our system is 3-tier Architectural design. It consists of three tiers, Presentation, Business Logic and Data Access. The components and their interaction is shown below in diagram. Double arrow line represents two way interaction between components.

Chosen architecture of the system is shown below:

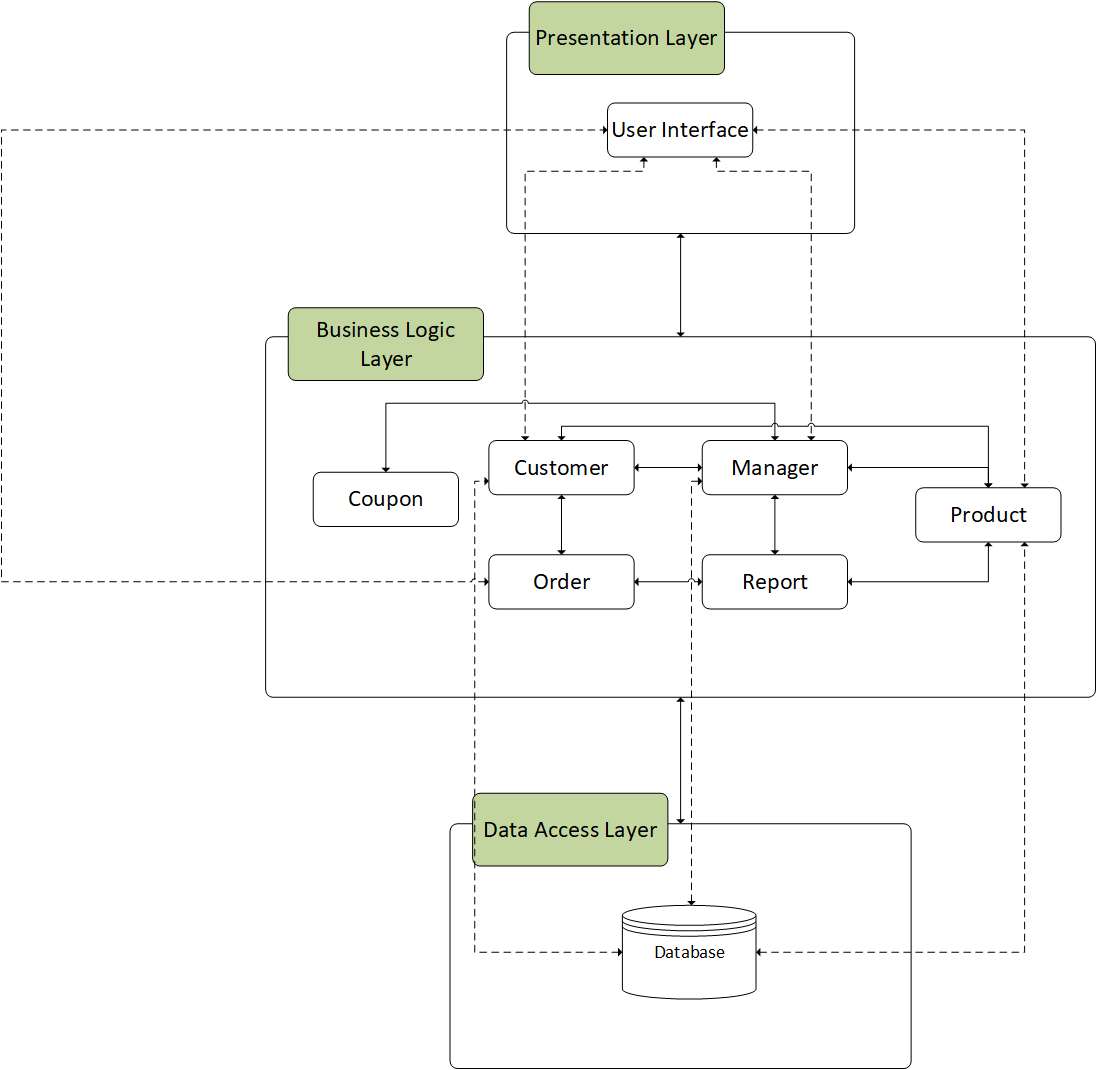


Figure : Architectural Design

## **Sequence Diagram**

A sequence diagram shows the interactions of objects that are arranged in a time sequence. It describes the objects and classes involved in the specific scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Given sequence diagram shows the generic flow of our system. The interaction of objects and exchange of messages shows the overall working in a visual form, as shown in the diagram below:

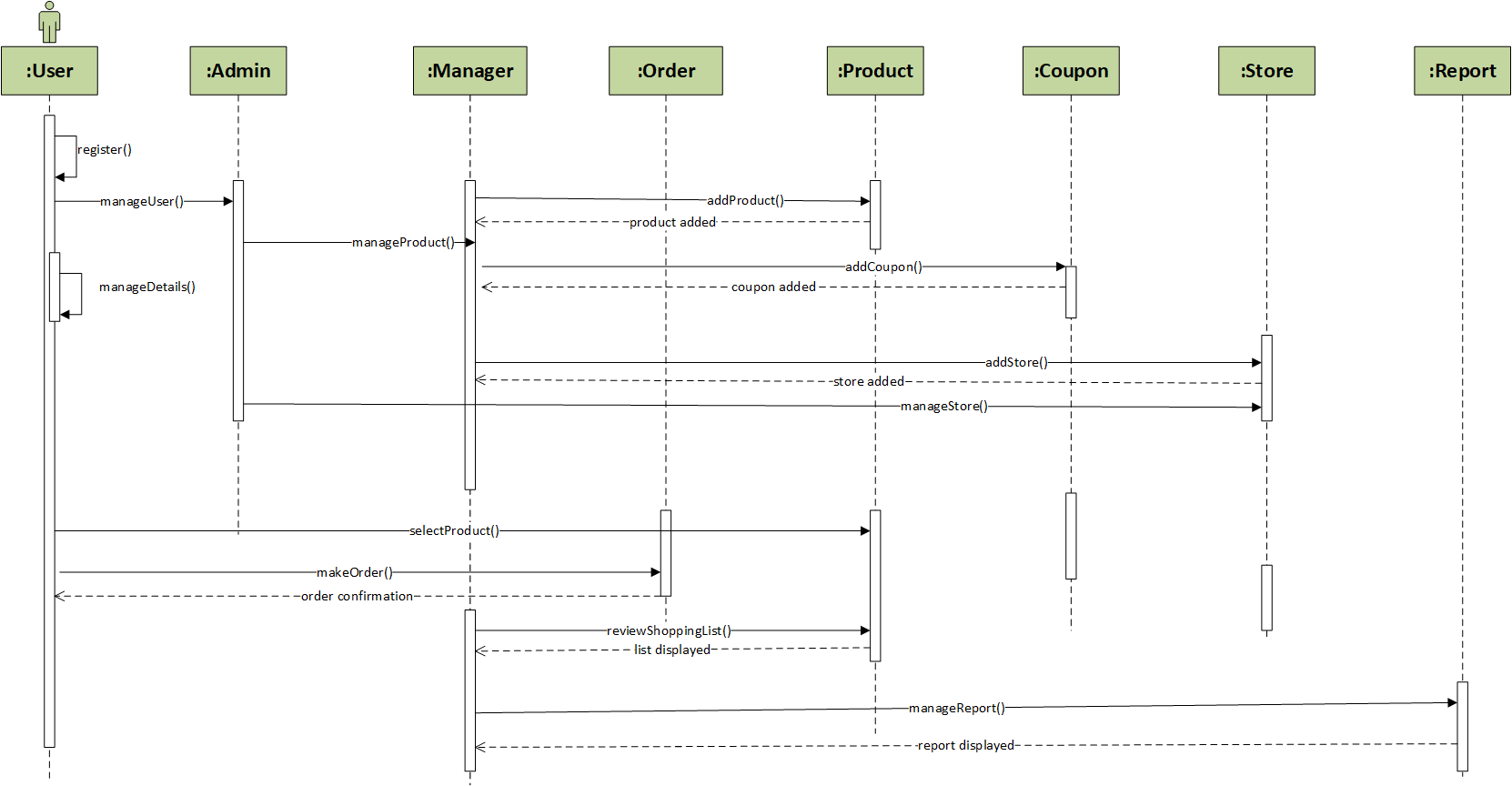


Figure : Sequence Diagram

## **Class Diagram**

For further implementation, class diagram plays an important role. It helps to identify and design the classes that are necessary for our system including fields and methods. Each class has its own purpose and working. It is a static structure that describes the basic structure of overall system’s classes with their attributes, operations and relationship between them. Interface classes are also shown to define the classes related to interface. There are also sub classes which inherited from the parent class for more understanding. This can be shown in diagram below:

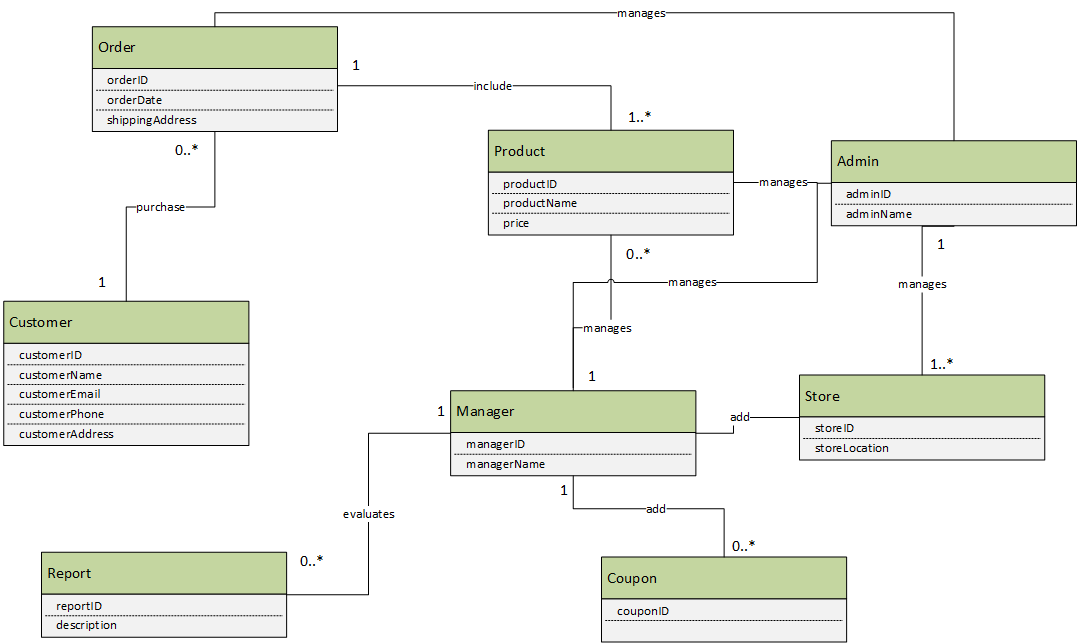


Figure : Class Diagram

# **Implementation**

After designing, implementation process begins which is done by coding our system specifications. The whole work is divided into modules. This is the longest phase of the SDLC. Here, we used some technologies which helps in the development, like for UI design, main coding environment, server side implementation and many more. An implementation is a realization of a technical specification or algorithm as a program, software component, or other computer system through computer programming and deployment.

## **Tools and Techniques**

To satisfy the technical requirements of a product, we need some tools for this purpose. There are some phases in a software development process and every phase needs some tools and techniques for its completion.

Currently, we have UML diagrams and now there is a need of designing interfaces and implementation. For this purpose, following are the possible tools and techniques, used for this project

Table : Tools and Techniques

|  |  |
| --- | --- |
| **Tools & Technologies** | **Description & Purpose** |
| MS Word | A tool which is used for prepared project report for better understanding of requirements of the application. Ms Word is chosen because it is a standard word-processing program, which provides best features and easy to use. |
| Adobe Photoshop | A tool which is used to design and edit the project’s graphics. It is easy for use that helps us in making creative graphics like text, icons and images to our application. We chose this tool to create attractive graphics, icons and backgrounds for our system |
| MS Visio | A UML tool which is used to create system’s UML diagrams. It a best tool that offers a wide variety of built-in shapes, objects and stencils to work with. The purpose of using this tool is that it let user create diagram as easy as possible. |
| PHP | A server side scripting language that helps in designing web-based application. It is also known as general purpose language. This project is developed in PHP. It is embedded with HTML for project development and design |
| HTML/CSS | A standard mark-up language that is used for designing web pages and web applications, with the integration of CSS or JavaScript. The reason of using it because it is easy to understand, use and write to implement web application. |
| SQL Server | A language which is used for storing, manipulating and retrieving data in databases |

## **Coding Explanation**

The implementation phase of MIS system includes coding part, from requirements and design phase. Each part play their own role, described below:

### **Register:**

Implementation part for registration is given below:



Figure : Register

Here, we can see that customer personal details are received from customer input and store it to the system. Verification process is done in order to validate customer data. System displays a successful message in case of valid registration, otherwise displays some error message.

### **Login:**

Implementation part for login is given below:

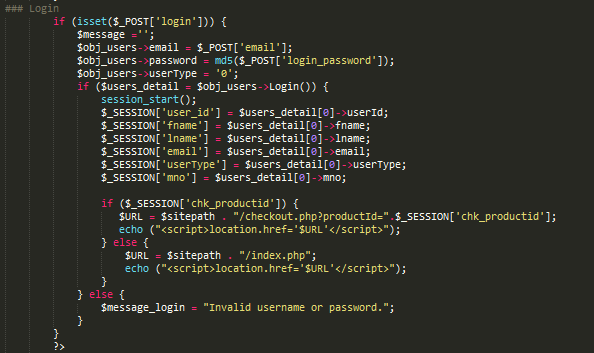


Figure : Login

Here, we can see that customer enters his username and password and user type, as there are two users (customer and manager). System receives credentials and verifies them from database. After successful validation, a message is displayed and customer is able to access the system. While, an error displayed in case of login fails.

### **Product Details:**

Product details are implemented in a following way:



Figure : Product Details

### **Coupon:**

Coupon details are implemented in a following way:

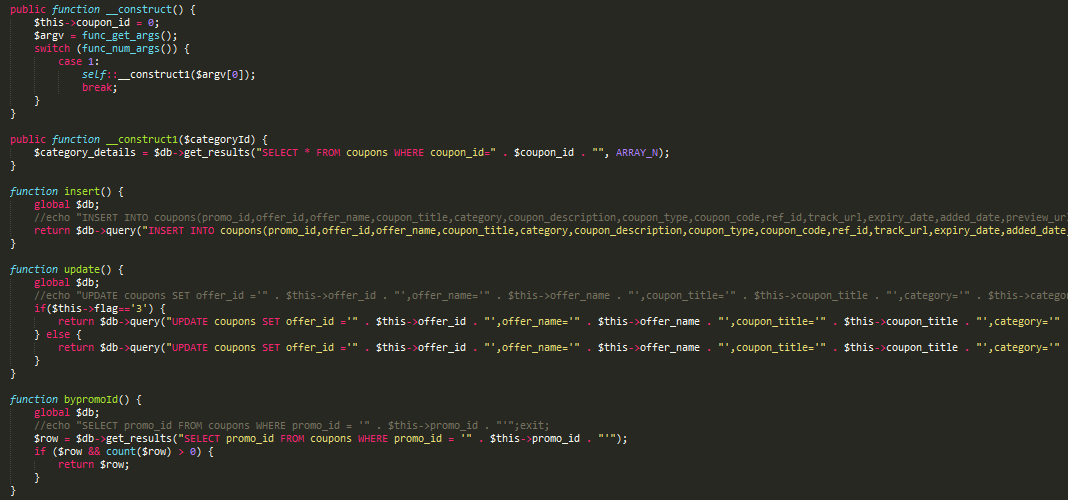


Figure : Coupon

Here, we can see that all the details of coupon are stored in database, using insert() function. Whereas, update() is used for updating all the information regarding coupon. Similarly, some other functions are also used here for their own purpose.

### **Order:**

Order details are implemented in a following way:

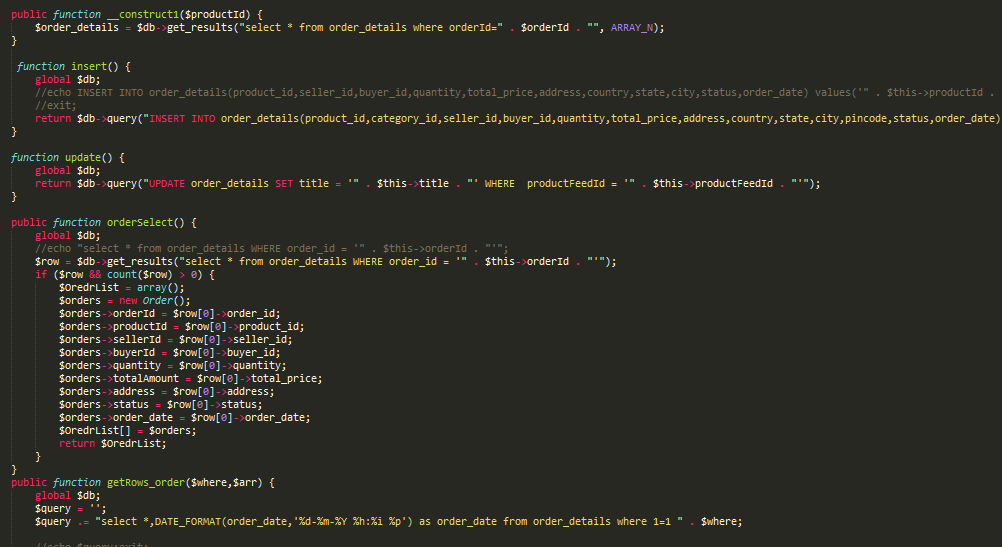


Figure : Order

Here, we can see that all the details of order are stored in database, using insert() function. This order is made by customers. Whereas, update() is used for updating all the information regarding order. Similarly, some other functions are also used here for their own purpose. The details of the order can be selected from selectOrder().

### **User:**

User details are implemented in a following way:



Figure : User

Here, we can see that all the details of each user are stored in database, using insert() function. This order is made by customers. Whereas, update() is used for updating all the information regarding user. Similarly, some other functions are also used here for their own purpose. The details of the order can be selected from byUserId().

### **Map Product:**

Implementation of mapping a product is done in a following way:



Figure : Map Product

Here also, we can see that all the details of each product are stored in database, using insert() function. This order is made by customers. Whereas, update() is used for updating all the information regarding product. Similarly, some other functions are also used here for their own purpose. The details of the order can be selected from bymapProductId() and deleted from deletebymapProductId(), from the database.

### **Review:**

Implementation of review is done in a following way:

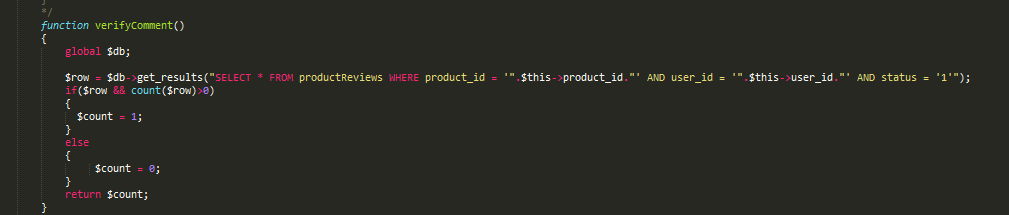


Figure : Review

Reviews are managed by managers. It consists of comments, written by the customers. Each review is verified by using product\_id and user\_id, from the database.

### **Advertise Brand:**

Implementation of advertise brand by uploading image is done in a following way:



Figure : Advertise Brand

Any advertising can be done by promoting its image. So, in this function, the details of the product brand is collected and stored in a database. Image(s) for each product brand is inserted in jpg, jpeg, png and gif format. System displays a successful message in case of uploading image, otherwise an error message is displayed from the system.

### **Map:**

Implementation of map is done in a following way:



Figure : Map

Here, we can see that the information of store is inserted under insert(), to set the category map of the store. Any map element is accessed through byCategoryMapId(), in which all the details of the map are given.

## **Application Screenshots**

Given below are the screenshots of the application:

Screenshots

# **Testing**

Testing is executing a system in order to identify any gaps, errors, or missing  
requirements in contrary to the actual requirements. After the implementation and quality assurance, our system needs to be tested. For this purpose, each module is tested against the requirements to make sure that the product is actually performing well without any issues. Testing is also an important phase of SDLC. During this phase all types of testing like unit testing, functional testing, GUI testing as well as non-functional testing are done. The main intension is to find bugs and defects in the process of verification and validation that meets all of the requirements.

## **Black Box Testing**

Black box testing is also known as Functional Testing. It is a testing technique that focuses on requirements and verifies the functionality of our system and ignores the internal mechanism or coding part of the system. Under Black box testing, we only concentrates on input and output of our system. Many applications are tested by Black Box testing methods. Majority of the test cases are covered to discovered bugs. It is often used for validation.

**Advantages:**

1. Testing is done on the basis of initial requirements and specifications.
2. Testing is done in user perception instead of designer’s perception
3. Test cases are generated immediately after completion of specification of the system.
4. Defects are fixed and re-tested
5. No need to have implementation or coding knowledge

**Types:** The major types of Black Box testing are Functional, Non-Functional and Regression testing.

**Tools:** Testing tools for Black Box testing are mainly Record and Playback tools. They record test in the form of some scripts like JavaScript, VB Script, and Perl etc. QTP, Selenium, Load runner and JMeter are used as well.

## **White Box Testing**

White Box testing is also known as Structural Testing, Code-Based Testing or Glass Box Testing. It is based on internal functionality, which includes application’s code and internal logic of the system application. Internal software and code working is tested by this technique. Under this, we are able to test the coverage of code statements, branches, broken or poorly structured paths, flow of specific input and output, conditional loop functionality and internal security holes. White Box testing is often used for verification. It can be done at system, integration and unit level.

**Advantages:**

1. Testing is done thoroughly as all paths covered
2. Test case are easily automated
3. Testing is done even if GUI is not present
4. Defects are fixed immediately
5. Eliminates the communication cost between QA and developers

**Types:** Unit testing, Statement testing, Path testing and branch testing etc. are main types of White Box Testing.

**Tools:** Testing tools for White Box testing are CppUnit, JUnit, EMMA, Veracode, NUnit, csUnit, Zeta Test, dot cover and many more.

## **Test Approach**

At this stage, test approach would be Black Box Testing and White Box Testing. It consist of a process of verifying that a solution works for the user. It is not system testing (ensuring software does no crash and meets documented requirements), but it is a test that user acceptance the solution.

## **Test Plan**

A test plan outlines the strategy that will be used to test an application, the resources that will be used, the test environment in which testing will be performed, and the limitations of the testing and the schedule of testing activities. Typically, the Quality Assurance Team Lead will be responsible for writing a Test Plan.

### **Features to be tested:**

Some of the features to be tested are as follows:

* Register
* Login
* Manage details
* Add product
* Add item
* Add coupon
* Manage order
* Map product
* Add store
* Manage user
* Import product
* Change password
* Manage featured coupon
* Remove product
* Logout

### **Features not to be tested:**

Following are the Features not be tested, which are from developer’s point of view:

* Power used by processor
* Memory consumed by the application
* Risk factor
* Operating system errors

## **Specific Test Cases**

Following are some test cases for our application:

Table : Login Test Case

|  |  |
| --- | --- |
| **Test Case 1** | |
| **Purpose** | Register |
| **Setup** | Not defined |
| **Instructions** | 1. Open registration page 2. Enter personal details 3. Registration successful |
| **Expected Results** | Registration successful. |
| **Actual Results** | Registration successful. |

Table : Manage Details Test Case

|  |  |
| --- | --- |
| **Test Case 2** | |
| **Purpose** | Manage Details |
| **Setup** | Not defined |
| **Instructions** | 1. Login 2. Opens my account page 3. Change personal details 4. Details updated |
| **Expected Results** | Details are managed successfully. |
| **Actual Results** | Details are managed successfully. |

Table : Add Product Test Case

|  |  |
| --- | --- |
| **Test Case 3** | |
| **Purpose** | Add Product |
| **Setup** | Not defined |
| **Instructions** | 1. Navigate product list 2. Select products to add them 3. Products are added to the shopping cart 4. Display a successful message |
| **Expected Results** | Products added successfully  . |
| **Actual Results** | Products added successfully |

Table : Add Item Test Case

|  |  |
| --- | --- |
| **Test Case 4** | |
| **Purpose** | Add Item |
| **Setup** | Not defined |
| **Instructions** | 1. Login 2. Select Add Product button 3. Enter item details 4. Item added |
| **Expected Results** | Item added by manager successfully. |
| **Actual Results** | Item added by manager successfully. |

Table : Add Coupon Test Case

|  |  |
| --- | --- |
| **Test Case 5** | |
| **Purpose** | Add Coupon |
| **Setup** | Not defined |
| **Instructions** | 1. Login 2. Select Add Coupon button 3. Add coupon 4. Coupon added in application |
| **Expected Results** | Coupon added successfully |
| **Actual Results** | Coupon added successfully |

Table : Manage Order Test Case

|  |  |
| --- | --- |
| **Test Case 6** | |
| **Purpose** | Manage Order |
| **Setup** | Not defined |
| **Instructions** | 1. Login 2. Select Order button 3. View the order information |
| **Expected Results** | Order is managed successfully |
| **Actual Results** | Order is managed successfully |

Table : Map Product Test Case

|  |  |
| --- | --- |
| **Test Case 7** | |
| **Purpose** | Map Product |
| **Setup** | Not defined |
| **Instructions** | 1. Selects specific store 2. Selects the product 3. Modify the product details 4. Product updated |
| **Expected Results** | Product mapping is done successfully |
| **Actual Results** | Product mapping is done successfully |

Table : Add Store Test Case

|  |  |
| --- | --- |
| **Test Case 8** | |
| **Purpose** | Add Store |
| **Setup** | Not defined |
| **Instructions** | 1. Select User Management button 2. Select Add Store button 3. Enter store information 4. Store added |
| **Expected Results** | Store added successfully |
| **Actual Results** | Store added successfully |

Table : Manage User Test Case

|  |  |
| --- | --- |
| **Test Case 9** | |
| **Purpose** | Manage User |
| **Setup** | Not defined |
| **Instructions** | 1. Select User Management button 2. Select User 3. Modify details |
| **Expected Results** | User is managed successfully |
| **Actual Results** | User is managed successfully |

Table : Import Product Test Case

|  |  |
| --- | --- |
| **Test Case 10** | |
| **Purpose** | Import Product |
| **Setup** | Not defined |
| **Instructions** | 1. Select User Management button 2. Select store 3. Select Import Product button 4. Prompt to identify the database 5. Product imported |
| **Expected Results** | Product imported successfully |
| **Actual Results** | Product imported successfully |

Table : Change Password Test Case

|  |  |
| --- | --- |
| **Test Case 11** | |
| **Purpose** | Change Password |
| **Setup** | Not defined |
| **Instructions** | 1. Opens my account 2. Select change password button 3. Enter new password 4. Verification successful 5. Password changed |
| **Expected Results** | Password changed successfully |
| **Actual Results** | Password changed successfully |

Table : Remove Product Test Case

|  |  |
| --- | --- |
| **Test Case 12** | |
| **Purpose** | Remove Product |
| **Setup** | Not defined |
| **Instructions** | 1. Create shopping list 2. Select product to remove 3. Product removed |
| **Expected Results** | Product is removed successfully |
| **Actual Results** | Product is removed successfully |

Table : Non-Functional Requirement Test Case

|  |  |
| --- | --- |
| **Non-Functional Test Requirements** | **Verdict** |
| System is performing well | Pass |
| System can hold large amount of data | Pass |
| System has appropriate response time | Pass |
| System is reliable | Pass |
| System uses reliable server | Pass |
| Transfer of data | Pass |
| System is able to secure user data | Pass |
| System is able to maintain its components and modification of components. | Pass |
| System is easy to understand and well organised | Pass |
| System is able to keep data integrity | Pass |

Table : GUI Requirement Test Case

|  |  |
| --- | --- |
| **GUI Test Requirements** | **Verdict** |
| Clear, meaningful and easily navigable | Pass |
| Concise information i-e optimal information in minimum space | Pass |
| Understandable icon that are familiar | Pass |
| Appropriate color scheme, text and images | Pass |
| Content focused and visual simplicity | Pass |
| Embedded map | Pass |
| Map zoom in/ zoom out feature | Pass |

# **Conclusions**

Mobile Intelligent Shopping (MIS) System is a mobile based shopping application that brings a different experience to customers and managers. It focuses on providing a solution to complex and time wasting shopping behaviour of the people. The system is developed for the customers and managers/store owners to reduce their work load and provide ease in today’s life. This system is successfully designed and developed to fulfil the necessary requirements, as identified in the analysis and requirement phase. The system is user friendly, easy to use, low cost and does not required any kind of training. The old manual system was suffering from a series of drawbacks. The present project has been developed to meet the new requirements in the modern age. All the previous related work has been explained in literature review section. Based on analysis and requirements, system design has been achieved, which includes system architecture design, sequence diagram and class diagram. Moving towards Implementation phase, which contains used technology, coding and algorithm of the system. Each step has been explained to show the complete flow of the system. Results of these Design and Implementation phase will discuss in the next section, to show the outcome of previous phase. Later on, testing approach justified and testing methods has been explained under next section

Waterfall methodology has been chosen and used for the development of this project. This model is preferred because it is simple, easy and reasonable approach, especially when the requirements are well understood. It is a straight forward model which means there is no risk and no change in requirements for this project. Each phase of this model has specific deliverables and review process. It also allows easy testing and analysis and saves significant amount of time.

## **Discussion and Summary**

Mobile Intelligent Shopping (MIS) System is a mobile based shopping application that brings a different experience to customers and managers. It focuses on providing a solution to complex and time wasting shopping behaviour of the people. The system is developed for the customers and managers/store owners to reduce their work load and provide ease in today’s life.

While the implementation of this system, developing team has gone through various research and existing systems. Most of those systems were unsuccessful due to following reasons, given below:

* Lack of high technical knowledge.
* Systems was unable to modify its features.
* Most systems were only based on front end.
* High cost due to many hardware recourses.
* Lack of security

So, we are unable to use the existing system and developed our new smart shopping system with new technology. It brings a different level experience to customers and managers. It focuses on providing a solution to complex and time wasting shopping behaviour of the people. The system is developed for the customers and managers/store owners to reduce their work load and provide ease in today’s life.

Moreover, system maintains security of the customer’s useful information in its database to remove the confidentiality issues. Our main focus is to assist the web based application on any mobile device so that the technology becomes cheaper. Here, we have used web services for the communication of client and server. It is obtained thorough major understanding of new technologies with compare to the background researches. All these activities led to a completed system.

## **Possible Future work and Enhancements**

The application is developed in a very short period of time and all efforts have been taken so that this project is very efficient in its execution there still exists some scope of improvement in our project. In future, more features can be added so that the app will become more reliable. The following lists some of the enhancement that can be added into the project.

* They system can make use of intricate routing system architecture in order to prevent loosing data of smart shopping cart.
* The system can send alerts and messages to the user about new deals and offers
* Voice technology can be used
* Customers can share their shopping list to their friends

# **References and Bibliography**

1. Standards Association, 2016. IEEE Recommended Practice for Software Requirements Specifications. IEEE Std 830 1998.
2. D.N.S.Kuruwitaarachchi, R.A.P.Buddhika - [Mobile Intelligent Shopping Guide](http://www.ijsrp.org/research-paper-0416.php?rp=P525239) - published at: "International Journal of Scientific and Research Publications (IJSRP), Volume 6, Issue 4, April 2016 Edition".
3. Suryaprasad, J., Kumar, B.P., Roopa, D. and Arjun, A.K., 2011. A novel low-cost intelligent shopping cart.
4. Karmouche, A. and Salih-Alj, Y., 2012, December. Aisle-level scanning for pervasive RFID-based shopping applications. In *Computer Systems and Industrial Informatics (ICCSII), 2012 International Conference on* (pp. 1-4). IEEE.
5. Chandrasekar, P. and Sangeetha, T., 2014, February. Smart shopping cart with automatic billing system through RFID and ZigBee. In *Information Communication and Embedded Systems (ICICES), 2014 International Conference on* (pp. 1-4). IEEE.
6. Kamble, S., Meshram, S., Thokal, R. and Gakre, R., 2014. Developing a Multitasking Shopping Trolley Based On RFID Technology. *International Journal of Soft Computing and Engineering (IJSCE) ISSN*, pp.2231-2307.
7. IBM aims to satisfy data-hungry shoppers [Online] Available: http://news.cnet.com/IBM-aims-to-satisfy-data-hungry-shoppers/2100- 345\_3- 5823169. html [Accessed: January 05, 2016]
8. Shekar, S., Nair, P. and Helal, A.S., 2003, March. iGrocer: a ubiquitous and pervasive smart grocery shopping system. In *Proceedings of the 2003 ACM symposium on Applied computing* (pp. 645-652). ACM.
9. Han, J., Pei, J. and Kamber, M., 2011. *Data mining: concepts and techniques*. Elsevier.
10. Hanwate, A. and Thakare, P., 2015. SMART TROLLEY USING RFID. *International Journal of Research in Science and Engineering e-ISSN*, pp.2394-8299.