

**Computer Science and Engineering Go Shopper**

# Analysis Specification - Version 1.2

### Document Number: Analysis Specification-001

* Team Number: C8
* Team Members:

|  |  |
| --- | --- |
| **Names** | **E-mails** |
| Datta Sainath D | dsd298@nyu.edu |
| Harsh Yadav | hy1217@nyu.edu |
| Shearyar Shamim Khan | ssk482@nyu.edu |
| Jayesh Punjaram Patil | jpp421@nyu.edu |

**Revision History**

|  |  |  |
| --- | --- | --- |
| **Date** | **Version** | **Description of Change** |
| 11/10/2016 | 1.0 | Initial Document |
| 10/11/2016 | 1.1 | Business and Technical Specification |
| 1/12/2016 | 1.2 | Analysis Specification |

# Table of Contents

## 1 Introduction

|  |  |  |
| --- | --- | --- |
|  | 1.1 | Purpose………………………………………………………1 |
| 2 | Scope |  |
|  | 2.1 | Identification………………………………………………....2 |
|  | 2.2 | Bounds………………………………………………….........3 |
|  | 2.3 | Objectives…………………………………………………....3 |
|  | 2.4 | System Overview…………………………………………….4 |

1. Business Requirements

### Technology…………………………………………………...5

* 1. Economics……………………………………………………5
  2. Regulatory and Legal………………………………………...5
  3. Market Considerations……………………………………….5
  4. Risk and Alternatives………………………………………...6
  5. Human Resources and Training……………………………...7

1. Context Diagram
   1. Diagram………………………………………………………8
2. Requirements

### Functional Descriptive Requirements………………………..8

* 1. Requirement Use Cases……………………………………....9
  2. Use Case Diagrams………………………...............................9
  3. Non-Functional Descriptive Requirements…………………..20

1. Analysis

### Component Architecture Diagram……………………………23

* 1. Component Description……………………………………....23
  2. Class Diagrams……………………….....................................25
  3. Class Relationship/Interaction Diagrams…………………….27
  4. Events

6.5.1 Motives……………………………………………….…28

6.5.2 Event Diagrams………………………………………….30

* 1. Activity/State Diagrams………………………………………..31
  2. State Logic……………………………………………………..33
  3. Behavior

6.8.1 Sequence Diagrams…………………………………..…..34

6.8.2 Collaboration Diagrams………………………………….35

1. Dictionaries

### Class………………………………………..…………………..36

* 1. Methods………………………………………………………...37
  2. Attributes……………………….................................................38
  3. Relationships……………………………………………………39
  4. Key Event……………………………………………………….40

1. INTRODUCTION
   1. **Purpose**

The purpose of this project is to save money and time for people who take lot of pain to buy groceries from different stores or end up paying an exorbitant amount by buying overpriced goods. Often, the consumers are not able to find all the groceries at the same place. Thus there is a need for an application that could inform customer (user) where the items in his list are available (Location of store) and also list its price. That way people could select which store they should be visiting for economical purchases. There are quite a few services and apps like Gasbuddy, Google Express, and other grocery store specific websites, which allow people to go through their list of items and buy them. Google express is even in the business of delivering these items as well. However, this application will be streamlined to use those services as well as develop an algorithm in this application for economical purchases.

1. **SCOPE**

|  |  |
| --- | --- |
| **The product can be used to** | **The product cannot be used for** |
| Provide interface for the user | No guarantee of Safety of Food Items |
| Provide interface for Store Owners | Items besides groceries |
| Update database by Store Owners | Stores that are not included in the application |
| Cataloging items with Prices and Stores |  |
| Personal User Profiles |  |
| Delivery Services |  |
| Discounts |  |
| List Prices from cheapest to most expensive and Stores in which it is available |  |

* 1. **Identification**

This is version 1.0 of Go Shopper Business Specification

Any major functional change in the documentation will be marked as X.0 (Example: 1.0, 2.0, 3.0)

Any minor functional change in the documentation will be marked as 1.X(Example: 1.1, 1.2, 1.3)

* 1. **Bounds**

Following are the bounds for the system:

It provides interface for users, store owners, and deals and discounts.

It provides database management for stores owners.

Delivery options for customers.

Directions for customers to the store.

* 1. **Objectives**

The main objective of this application is provide people with cheap groceries which are available around their local neighborhood and at the same time save the time that involved in buying groceries. This application is similar to various other applications like Gasbuddy, Amazon, and other web based shopping websites. In the other applications there is no option for searching a grocery store near the customer’s location which could satisfy the customer’s need. For example, Amazon does not provide groceries to people. Also, it does not provide a way through which people can compare prices of the same grocery at different stores. Also, the application Gasbuddy is based on gas. It tells people about nearby gas stations, but does not provide the prices of gas at that station. Therefore, the main objective of this project is to satisfy the grocery need of people and at the same time save their money and time.

**Deliverable Dates:**

|  |  |
| --- | --- |
| **Deliverables** | **Date** |
| Project Proposal | 10-6-2016 |
| Project Business Requirements | 10-20-2016 |
| Project Requirements | 11-10-2016 |
| Project Analysis | 12-01-2016 |
| Presentation | 12-08-2016 |

* 1. **System Overview**

The application helps the customers buy economic groceries from the stores near their location. The customers have to make a list of all the groceries that they want and based on that list, a list of all the nearby stores will be generated with the product name and its cost. The user will then be able to select a particular grocery store from the list of stores shown on the screen in the order of increasing distances from the user’s location. The selected grocery store will display the user the items sold by that particular grocery store. If the user is interested to buy any item, he can add the item directly into the cart. After adding all of his interested items into the cart, he will be given an option to checkout (for premium users) or to navigate him to that desired store to pick up his items.

The premium customers will be asked to choose to which location the items are to be delivered. Then, he will be directed to a secured payment gateway to enter his/her credit card number. The delivery of goods is done by company-employed deliverymen. This application will also have a separate tab, which contains any special discounts on products that are listed by the nearby grocery stores through this application.

Another major feature of this application is to intelligently list the desired items of the user from different grocery stores nearby to minimize the overall cost of the all the products put together. The list of items wanted by the user is written in a specific page of the application by the user. When the user is interested to buy those products, this application runs an algorithm which compares the rates of all the items provided by the user and shows the items that are present in different stores with the minimum cost.

1. BUSINESS REQUIREMENTS
   1. **Technology**

The application will enable the customers to buy the groceries at economical prices and also it will guide them to buy those items. In addition, the customers will also be able to get the products delivered to their home. This will also help the stores to sell more items as the people living near those stores will be guided to those stores to buy the items.

* 1. **Economics**

Minimum features of the app are free and advanced features like delivery services will be available for premium users who pay additional amount to get a premium membership for one year.

* 1. **Regulatory and Legal**

As per the Regulatory and Legal, the application shall have contracts with stores and retail chains to have access to their database, along with complete listing of all the items present in their store with their prices and also, if there are any discounts and offers on products, then they should also be shown by the application. Also, the permission to show the store and it GPS location on the application is required.

* 1. **Market Considerations**

The application helps and improves the overall quality of middle income people by providing economic food items to them and thus saving their money on the same items that they used to purchase before.

People can buy quality products from nearby stores at economical prices and also, by the help of the application they can get to know the offers and discounts available in the nearby stores on the items that they require.

People can get access to particular products from nearby stores.

* 1. **Risks and Alternatives**

Business Risk: When users show less interest to sign up as a premium customer.

Probability: It is less probable as the customers would have a free delivery option if they are premium customers and also they can frequently order items as per their need.

How discovered: Product Analysis before launch.

Responsible Party Status: Yet to be discovered.

Mitigation Plan: Deals will be first available to premium customers and free delivery option.

Operational Risk: When the stores are not ready to share their database.

Probability: It is less probable as the stores would want to draw more attention through this application.

How discovered: Discovered when the store owners hesitated to share database, but later on they were convinced to share it.

Responsible Party Status: They are ready to share their data.

Mitigation Plan: We will also display sponsored deals on the homepage related to a particular store.

Technology Risk: Threat to user’s personal information such as saved credit card and other details.

Probability: Less probable.

How discovered: General problem related to payment systems in technology.

Responsible Party Status: Software modelling.

Mitigation Plan: We shall provide secure gateways.

Economic Risk: The stores that are not part of the application provide better quality product more economically.

Probability: Less probable.

How discovered: Analysis.

Responsible Party Status: Neutral.

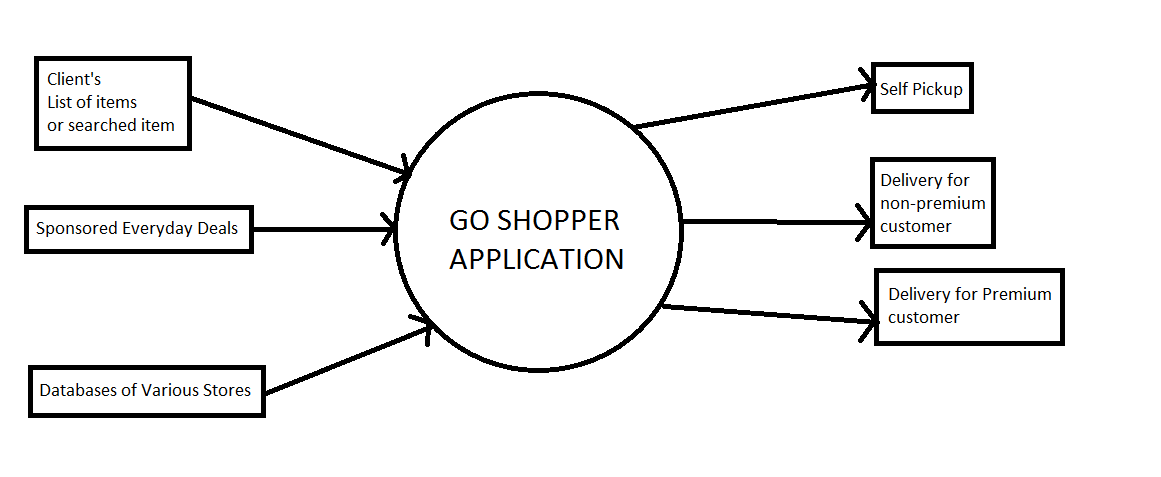
Mitigation Plan: We shall get as many stores online as possible

* 1. Human Resources and Training

We need not train people in the initial phase of the product. Later, depending on the market demand we can set up our own delivery service or outsource it to third party service. We will need to train people if we set up own delivery service.

1. **Context Diagram**

High-level context diagram identifying system boundaries



1. **Requirements**
   1. **Functional Descriptive Detailed Requirements**
2. The application helps the customers buy economic groceries from the stores near their location (around 2km radius)
3. The customers have to make a list of all the groceries (generally around 5-10 items) that they want and based on that list, a list of all the nearby stores will be generated with the product name and its cost.
4. The user will then be able to select a particular grocery store from the list of stores shown on the screen in the order of increasing distances from the user’s location.
5. The selected grocery store will display the user the items sold by that particular grocery store. If the user is interested to buy any item, he can add the item directly into the cart.
6. The premium customers will be asked to choose to which location the items are to be delivered. Then, he will be directed to a secured payment gateway to enter his/her credit card number.
7. The delivery of goods is done by company-employed deliverymen.
8. Another major feature of this application is to intelligently list the desired items of the user from different grocery stores nearby to minimize the overall cost of the all the products put together.
9. The list of items wanted by the user is written in a specific page of the application by the user. When the user is interested to buy those products, this application runs an algorithm which compares the rates of all the items provided by the user and shows the items that are present in different stores with the minimum cost.
   1. **Requirement Use Cases**

The use cases consist of all operations that are performed by the user, store owners, system and other third parties (delivery and payment authentication. The user can have access to login, catalog information, cost management which involve offers and discounts on products across different stores, payments and refunds related to products, customer support which involve resolving user queries and delivery options for the users which involve free delivery for premium customers, paid delivery for non-premium customers and self-pickup options for non-premium customers. Also, the third party is involved with deliveries and payment authentication through banks. The store owners also have access to adding, updating and deleting the list of items, discounts and offers that are provided by their store. All these details are checked by the system administrator and then added to the system. Also, the count of all the items is maintained by the system, so that no wrong information is displayed to the user and no problems are faced.

* 1. **Use Case Diagrams**

C:\Users\ddatt\AppData\Local\Microsoft\Windows\INetCacheContent.Word\Use Case 1.png

C:\Users\ddatt\AppData\Local\Microsoft\Windows\INetCacheContent.Word\Untitled Diagram (1).png

**Use Case Descriptions**

|  |  |  |
| --- | --- | --- |
| **Secure Login/Logout** | | |
| **Description** | Secure Login and Logout into user account. | |
| **Pre- Conditions** | **Pre- Conditions: The user shall be using the latest version of the application/ web browser.** | |
| **Flows** | **Basic or Normal Flows** | 1. When the user opens the application/ web browser, he must be directed to the login page.  2. The user must enter his username and password details |
|  | **Alternative Flows** | 1. If the username or password is incorrect then the user must be redirected to the login page again.  2. If the wrong password is entered more than 5 times then an email notification to change the password should be sent to the user and his account must be blocked until then. |
| **Post Conditions** | If the user enters correct login details, then the login is successful and the user can enter the list of diagrams that he wants. | |
| **Special Requirements** |  | |
| **Extension Points** |  | |

|  |  |  |
| --- | --- | --- |
| **Cost Manager** | | |
| **Description** | Tells about the offers and discounts available to the users. | |
| **Pre- Conditions** | **The user shall successfully logged in.** | |
| **Flows** | **Basic or Normal Flows** | 1. When the user enters the list of items that he wants, the system goes through all the information and depending on the list the system shows all the offers and discounts available in the nearby stores. 2. The user can select the discounts and offers based on his requirements. |
|  | **Alternative Flows** |  |
| **Post Conditions** | The discounts and offers should be applied to the items selected by the user and base do on that the pricing should be altered. | |
| **Special Requirements** |  | |
| **Extension Points** |  | |

|  |  |  |
| --- | --- | --- |
| **Catalog Information** | | |
| **Description** | Represents the details of all the items that the user wants in the nearby stores along with their prices. | |
| **Pre- Conditions** | **The user must have logged into his account** | |
| **Flows** | **Basic or Normal Flows** | After successful login to the application, the user must select the items that he wants from the available list of items. |
|  | **Alternative Flows** | 1. If the username or password is incorrect then the user must be redirected to the login page again.  2. If the wrong password is entered more than 5 times then an email notification to change the password should be sent to the user and his account must be blocked until then. |
| **Post Conditions** | If the user enters correct login details, then the login is successful and the user can enter the list of diagrams that he wants. | |
| **Special Requirements** |  | |
| **Extension Points** |  | |

|  |  |  |
| --- | --- | --- |
| **Payments and Refunds** | | |
| **Description** | The system should process the payments and refunds as per user requirements. | |
| **Pre- Conditions** | **The user should be logged in and the items to be bought by him should be in a list with their prices and discounts applied with a final amount. Also, the delivery details must be added in case of premium users.** | |
| **Flows** | **Basic or Normal Flows** | 1. When the user has completed selecting the items, the offers and discounts related to those products are applied.  2. The final payment is then produced by the system.  3. In case of the refund, the user must select the items to be returned and refunded.  4. To initiate the payment the user can make payment either through credit/ debit card.  5. The authentication of payment can be done by the third party. |
|  | **Alternative Flows** | 1. If the items are not eligible for refunds, then the user must be prompted about it. |
| **Post Conditions** | The user is directed to the payment options. | |
| **Special Requirements** | **The system must check the status of all the items for offers, discounts and refunds.** | |
| **Extension Points** |  | |

|  |  |  |
| --- | --- | --- |
| **Customer Service** | | |
| **Description** | To solve the user queries related to products. | |
| **Pre- Conditions** | **The user must be logged in to the system.** | |
| **Flows** | **Basic or Normal Flows** | 1. Incase the user has any query related to the product, then he can contact the customer service for seeking help. 2. The customer service and be either through the email or phone call. |
|  | **Alternative Flows** |  |
| **Post Conditions** | User feedback for the help provided shall be asked. | |
| **Special Requirements** |  | |
| **Extension Points** |  | |

|  |  |  |
| --- | --- | --- |
| **Delivery Details** | | |
| **Description** | Secure Login and Logout into user account. | |
| **Pre- Conditions** | **The user must be logged in to the system and should have the final payment produced by the system and also the user must be a premium user for free delivery options.** | |
| **Flows** | **Basic or Normal Flows** | 1. When the user is ready with the final list of items with the final payment. He is then asked for a delivery option.  2. If the user is a premium customer then he can get the products delivered for free at his home.  3. If the user is not a premium customer then he may choose to pick the items directly from the stores or he may get them delivered at home for an added cost.  4. If the user has the home delivery item selected then he is prompted to enter his address and contact details.  5. Based on the address provided by the user, an approximate date range is provided to the user, about when his product will get delivered. |
|  | **Alternative Flows** |  |
| **Post Conditions** | An approximate delivery date must be provided to the user in case of home delivery. | |
| **Special Requirements** | **For free delivery, the user must be a premium customer.** | |
| **Extension Points** |  | |
| **Payment Authorization** | | |
| **Description** | The payment authorization is done by the third party which is the bank in case of the credit/ debit cards. | |
| **Pre- Conditions** | **The user must enter his card details for payment of items.** | |
| **Flows** | **Basic or Normal Flows** | 1. When the user enters his card details for the payment of items, the payment authorization must be done by the bank whose credit card is used to determine the correctness of the user.  2. If the details entered are correct, the order is successful. |
|  | **Alternative Flows** | 1. In case of wrong details, the user must be prompted to enter the correct details next time and the payment is not done. |
| **Post Conditions** |  | |
| **Special Requirements** |  | |
| **Extension Points** |  | |

|  |  |  |
| --- | --- | --- |
| **Customer Support** | | |
| **Description** | Provide the customer support to all the customer queries. | |
| **Pre- Conditions** | **The user must be logged in to his account** | |
| **Flows** | **Basic or Normal Flows** | 1. If the user has any query related to the product, then he can contact the system to resolve the queries and this can be done through email or phone call. |
|  | **Alternative Flows** |  |
| **Post Conditions** |  | |
| **Special Requirements** | **The user queries must be resolved within 24 hours.** | |
| **Extension Points** |  | |

|  |  |  |
| --- | --- | --- |
| **Add/ Update/ Delete Item** | | |
| **Description** | The store owner can add, update or delete the items, offers and discounts. | |
| **Pre- Conditions** | **The store owner must be logged in to the system.** | |
| **Flows** | **Basic or Normal Flows** | 1. The store owner can add items to te list of items available in his store along with it price, offers and discounts, if any.  2. The store owner can delete any particular item, offer or discount that is available on items on his store.  3. The store owner can update the price, offer, name or discount available on the items present in his store. |
|  | **Alternative Flows** | 1. For updation and deletion of item/ offer/ discount, it should be already present in the system, else it shall cause an error. |
| **Post Conditions** | The additions/ updates/ deletions must be implemented immediately to the system. | |
| **Special Requirements** |  | |
| **Extension Points** |  | |

|  |  |  |
| --- | --- | --- |
| **Store Delivery** | | |
| **Description** | Tells the items that are to be picked up from the store for delivery | |
| **Pre- Conditions** | **The store owner shall have a list of items that are to be picked up from his store.** | |
| **Flows** | **Basic or Normal Flows** | 1. The store owner shall get information about the products that are to picked from his store for delivery. 2. The items can be either picked up by a delivery service or customers who have not selected the home delivery option. |
|  | **Alternative Flows** | 1. If the items are not picked up, then the store owner must update the information on the system, so that it can be picked soon |
| **Post Conditions** | The count of the picked up items is decremented from the total count of those items. | |
| **Special Requirements** |  | |
| **Extension Points** |  | |

**5.4 Non-Functional Descriptive Detailed Requirement**

**System Capabilities, conditions, and constraints:**

The application shall handle thousands of users at once. It shall secure transactions and refunds. The application shall be able to quickly update customers on what items are available in the catalog. It shall only handle transactions and delivery fulfillments. Finally the application shall be constrained to the geographical area of New York City. Geographical area is going to be the main constraint. We shall start with the Manhattan area and then spread our market progressively.

**Physical Resource Requirements:**

In the initial testing phases we shall use a single server and then increase the number of servers as per the user capacity and demand.

It shall use the services of GPS and shall be available on smartphones, laptops, and **desktops.**

**Computer Hardware Requirements (User Computer Hardware Requirements):** 2GB RAM, Dual Core Processors, 80GB Memory, 1Ghz Processor

**User Mobile Device Hardware Requirements**: smart phone with 512MB RAM which can handle applications.

**Computer Hardware Resource Requirements:** Graphical display, input mechanism-keyboard/mouse, touch screen (in case of smart phone)

**Computer Software Requirements:** Computer should have an updated operating systems and a web browser to visit the website

**Computer Communications Requirements:** The application shall fully support Wi-Fi capabilities. It shall provide secure communication between the user and website by using a reliable and secure browser.

**Environmental Conditions:** None

**System Performance Characteristics:** The system shall hold all orders for 30 minutes, which will give customers the opportunity to cancel their order. The system shall after the hold period process orders in less than a minute and contact the delivery service and/or storeowners to ready the order.

**Safety Requirements:** No goods shall be sold after the expiration date.

**Security and Privacy Requirements:** SSL/HTTPS 2.0

**System Human Interface:** To be clear and concise, graphical, breadcrumb navigation. We just create separate pages for each functionality displayed in the user case diagrams. Our main aim is to keep the interface clear and easily understandable.

**System Maintainability:** Whenever an update is available to technologies, we shall immediately update with new technologies and better functionalities.  
Example: Upgrading our product from Android 6 compatibility to Android 7.

**System Quality Factors:** Usability, Scalability, Security, Correctness & Consistent.

**Design and Construction Constraints:** High Cohesion and low coupling.

**Life Cycle Model:** Spiral Model

**Policies and Standards-Methods, tools and techniques.**

**Personnel-Related Requirements:** Personnel for this product is considered as customer/user. The customer/user must be able to know how to use a web browser or an application on a smartphone

**Training-Related Requirements:** Training should be given to store owner on usage of application.

**Logistics-Related Requirements:** Logistics is important to maintain all the items from different stores.

**Packaging Requirements:** Third party which is responsible for packing of perishable items and non-perishable items should be well trained in logistics as perishable items should be taken of. The third party responsible for packaging the items for delivery should also be trained.

**Precedence and Critically Requirements:** None

**Other Non-Functional Requirements:** None

1. **Analysis**

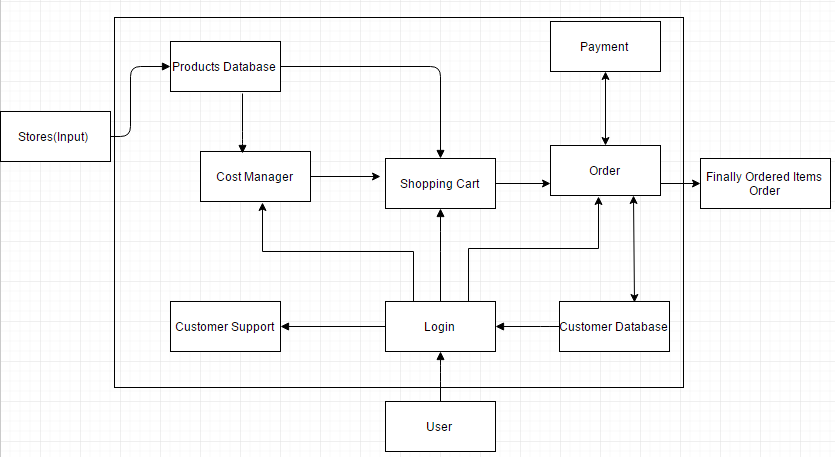
Component (Component/Package/Subsystem) Architecture

* 1. **Component Architecture Diagram**

Architecturally significant design elements (components, subsystems, interfaces) and their relationships

Logical system architectures derived from functional requirements and the domain (context, scope, and boundary) descriptions

Component architecture diagram for go shopper:



* 1. **Component Description**

https://lh5.googleusercontent.com/w6viKNVnH3Qa-fRsofC3oJHP4EDOGThvBJulHUpXIdL0OAK-8kbfgfDTKQ92gsJ8sSFJpwbRQ9LV6nd0I0C-LpgV4TwPVgr4x24OjnBllb9kbJUsGME_QoCBIDGZdzWPYmjYs1o6qG8mHonUxA

This system/block will have all the stores in a selected geographical area initially and then spread all over the country as the product is used by several users.

https://lh5.googleusercontent.com/C0UhMSIoRLU8ss9zpGrs257mtsn2J8eS8MZolQlSFkVQ2a3XkMks7RxYMamAko4erwIR8P1YZbqkt8hzpM1D9JUv4PH-x3J15H1S86L0si-A015VVJ4vkbl9LZqPdrykaLmIwGcFIh5gqper-Q

This system/block consists of all the stores’ product details which include product name, product id, quantity and other technical details of products.  Shopping by a user or searching of different products from the cost manager done by the algorithm.

https://lh4.googleusercontent.com/07pEzlFTiny3I8R2sCjX_u0StUW3ClG1HTUkoeOcGeqHnlnJGtcO2s2adj_YD8ijnUA2MNN_QVcvpiCjebfnWeyXyw1PUtmlaVGS7xmZszmgGFHMCFLc4HdVVAk716UfMgar4A_HQjSwLosPAw

This system/block takes a list of products from the user as input.  The list of items will be chosen economically from the locality by user the user’s location taken during the login process.  The items will be searched from the Products Database and then added into the cart

https://lh4.googleusercontent.com/OWGsiXrF_QXWq2hMDfhnkFD94ZeQwY29X6i0CbUNcZt68zi_FizVsAIox08Crce0CVtex57wdyAH63hgOmYINnyjQWvSKd1DdaSAL3oXf3S_Ow4N6eC_LTCOUCNAax_HUPzF24BA26505e2UpQ

Customer Support is a system which can be used once the user logins to his account he can interact with the customer service agents.

https://lh6.googleusercontent.com/7onvZzEIKA7uBfTAC8XriYCRKpvtsZs9N5kERS9pshQYSIZ73K1SasrGeHqZ2ezB8s90GFw7WF-tOwY-jGCho0r9E0yuI7s9J7QZuJh9W53IGRV_PQ_ZSbSQA2spWsJ6m3hRCOwj3RVxc90QSQ

Shopping Cart is a block which can is used to input all the products and their quantities.  This block will also carry all the necessary data of the products which is carted to be bought.  This block will let us update the quantities of the item carted or to add or delete a product from the cart.  This block will help is checkout process too.

https://lh6.googleusercontent.com/Gv8-gf-FlBOlVBLuDKTD71oYGthVZNqItJtMWAJm9pi7-6WzLISbqGy_rLXIarTFhXWyeN9I2ljladb942vpEHO9-VGeycnRAByEpFi0_DzOvgfSRe9wXixwcIlrVuHNg_hUdoqf_CjvcoCdhA

The Login block will serve as a normal login interface which asks the user to enter his user name and password.  This block will also provide the user to create an account, if he is using this application for the first time.  This block will also detect the current location of the user.

https://lh6.googleusercontent.com/SGhoH04c77IWXCx6gZ_1t_R-jOJUAEAypq_sHfANkra3OlnIFemXhgS-9sv9CYDEKR1wJ5wgBFDKH_CdfF_uw23JqAyM8T5ydh5TGP37o0Z4niRXwyD2vFUcw7PV8Vd_PnQdqyOT2fFWhQmIVw

This block represent the user, who is using the application.

https://lh6.googleusercontent.com/Ff0lcuOPUdx2SAgflc-xxmonokFo_S4Md1I3IcI89S6YF4488MIryBDNzm58vDWl0H-UeglwUMAC4Hq3qHSP4hRHIsjpEElTsijIGTEKjmMw9l63s3qWM2g8ODZ2eTCFiesOv-q6HjH7p7ZzHAv

Payment block is represents the payment gateway through which a user is allowed to make payments using various modes like credit card, debit card, Paypal etc.

https://lh6.googleusercontent.com/SaQjLucOq7D-Wg5CwJYhsPUx89QXpuQT2G52TKIaIBFIBuqKb_Ht07CdVYYJVZDeWCvmzi6VpZ6BX2PL5RvimLoRaZvJTNJs6nld62YRj_XNmzMwCoxc14GPmqvLVFTaRGimLI_FaGP20mbjHg

This is the most important block which takes the shipping information from the customer database (if there is no shipping address in the customer database, then it asks the user to input here), has all the details regarding the products about their prices, quantities, and delivery dates.  The user can login into his account and check all of his current and past orders in this system.

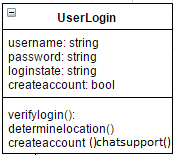
https://lh3.googleusercontent.com/m4xuLH__u1fECKyqBiwSR1DWpOeQJMtQpC76fiG6r_MqRYY6sYInnQqgPMgkK32HXstU55OZ42RQRtA6kbL-Cbogj_5yx1hPi9vSJHzV_IcpXozOh8G2mSIHh1sp7y4S0wNJfMn_qpgZ4T22Jg

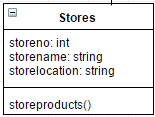
This system/block consists of user information like customer name, customer address, shipping addresses, phone number, emailed, credit card information. This block will also have his wish list item too.

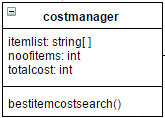
https://lh6.googleusercontent.com/ZmCgj2icS1Mg0LAwk6pJ0HXvBb2eqc3_6QeotDn8naNf8zuJgVN_17nvUVXa0vCHxQ4yBLH7rgya7wo1_oGmOtYGxN9rXt67-BBd6Hs-tHs4nl_Ztbh2htoAdjZejvoeWfDJoNynXso9jK2MkQ

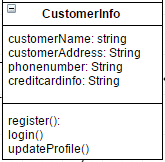
Finally Ordered Items (order) is the output of the system which contains the order item details and their quantities and shipping details which is emailed to the user for his reference.

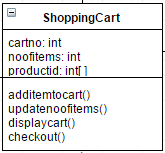
* 1. **Class Diagrams (Individual Class Diagrams)**

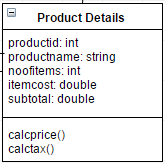


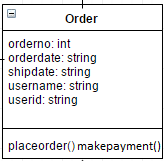






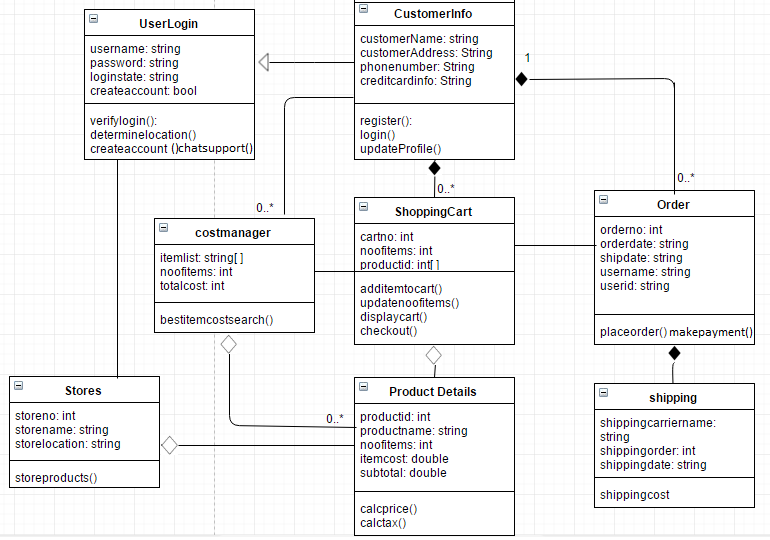








* 1. **Class Relationship/Interaction Diagrams**



* 1. **Events**

Events to which the system responds:

1. UserLogin: The system asks the user to login into his/her account by providing the correct username and password. If the user enters correct username and password then login is successful, otherwise the user is again asked for correct username and password. For the first time users, there is an option for creating a new account.
2. CustomerInfo: For the first time users, there is an option for creating a new account which asks for all user details like name, address, contact information and payment method.
3. User Verification: For the users entering usernme and password, if the details entered are correct, the verification is successful, otherwise it fails and user is asked to enter the details again.
4. CostManager/ List Creation: It involves the user to enter the list of items that he/she needs and the quantity of each item. Based on the list of items entered by the user a total cost (minimum based on the system algorithm) is generated and offers and discounts of each item is also displayed to the user.
5. ShoppingCart: After selecting all the items to purchase and availing all the discounts and offers applicable to items purchased, the user must add all the items to the shopping cart. Each customer is assigned a unique cart number and all products have a unique product identification number. The customer can view his/her account and also remove the items that are not required and after confirming all the items the customer is directed to checkout.
6. Order: After payment of the product is completed, each order is assigned a unique order number, order date, an estimated shipping/ delivery date, customer’s name and user id.
7. Stores: When the user enter’s the list of items to be purchased by him/ her, the system checks for all the stores and the products in the database and based on the prices of that product, it chooses the least amount that is available for that product and at the same time it also checks the availability of the product and location of the store and based on that, it selects that product for the user. Each store has a unique store number, store name and store location.
8. ProductDetails: Each product selected by the system for the user has a unique product id, product name, quantity present in different stores and cost at different stores.
9. Shipping: For the customers who select shipping or are premium customers are given a shipping carrier name, shipping order number and an estimated shipping/ delivery date. For non-premium customers who opt for delivery are charged an extra amount for shipping depending on the shipping address.
   * 1. **Motives**

1. UserLogin: The system asks the user to login into his/her account by providing the correct username and password. If the user enters correct username and password then login is successful, otherwise the user is again asked for correct username and password. For the first time users, there is an option for creating a new account.

2. CustomerInfo: For the first time users, there is an option for creating a new account which asks for all user details like name, address, contact information and payment method.

3. User Verification: For the users entering usernme and password, if the details entered are correct, the verification is successful, otherwise it fails and user is asked to enter the details again.

4.CostManager/ List Creation: It involves the user to enter the list of items that he/she needs and the quantity of each item. Based on the list of items entered by the user a total cost (minimum based on the system algorithm) is generated and offers and discounts of each item is also displayed to the user.

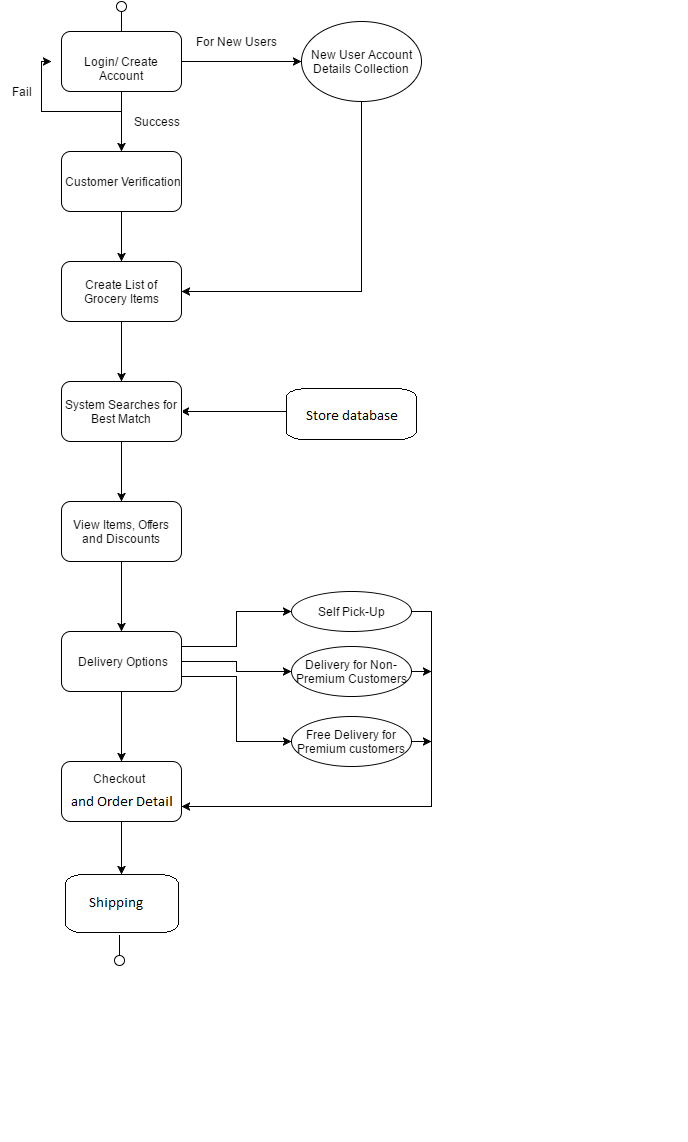
5. ShoppingCart: After selecting all the items to purchase and availing all the discounts and offers applicable to items purchased, the user must add all the items to the shopping cart. Each customer is assigned a unique cart number and all products have a unique product identification number. The customer can view his/her account and also remove the items that are not required and after confirming all the items the customer is directed to checkout.

6. Order: After payment of the product is completed, each order is assigned a unique order number, order date, an estimated shipping/ delivery date, customer’s name and user id.

7. Stores: When the user enter’s the list of items to be purchased by him/ her, the system checks for all the stores and the products in the database and based on the prices of that product, it chooses the least amount that is available for that product and at the same time it also checks the availability of the product and location of the store and based on that, it selects that product for the user. Each store has a unique store number, store name and store location.

8. ProductDetails: Each product selected by the system for the user has a unique product id, product name, quantity present in different stores and cost at different stores.

9. Shipping: For the customers who select shipping or are premium customers are given a shipping carrier name, shipping order number and an estimated shipping/ delivery date. For non-premium customers who opt for delivery are charged an extra amount for shipping depending on the shipping address.

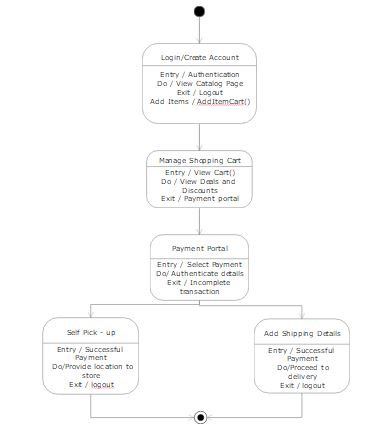
**6.5.2 Event Diagrams**

* 1. **Activity/State (Scenario) Section**

C:\Users\Harsh Yadav\AppData\Local\Microsoft\Windows\INetCacheContent.Word\Untitled Diagram (1).png

C:\Users\Harsh Yadav\AppData\Local\Microsoft\Windows\INetCacheContent.Word\Untitled Diagram (2).png

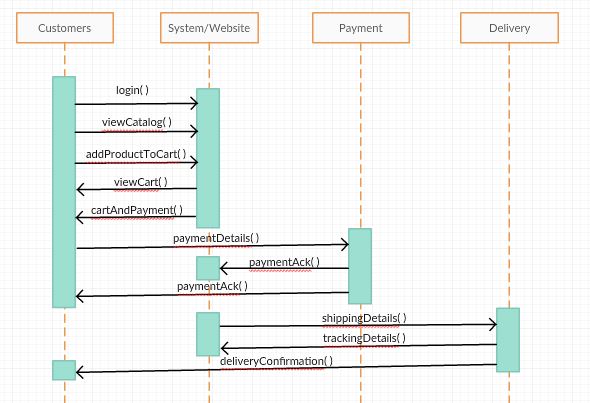
* 1. **State Logic**



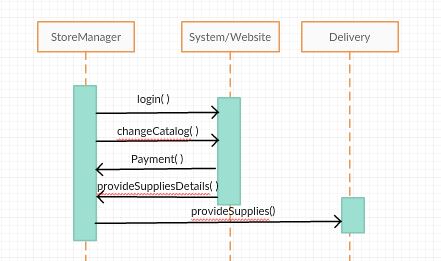
* 1. **Behavior**

**6.8.1 Sequence Diagrams**

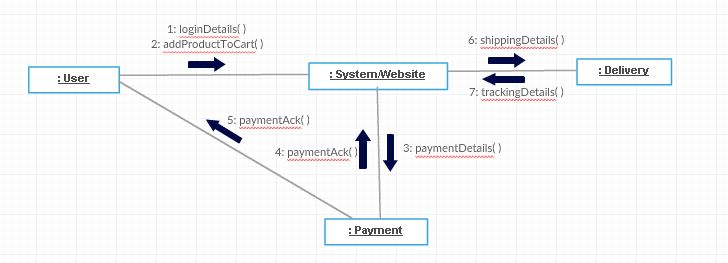
*(Generic Sequence)*



Between Store Manager and System



**6.8.2 Collaboration Diagrams**



1. **Dictionaries**
   1. **Class**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Description** | **Methods** | **Attributes** |
| UserLogin | Lets the user login to the system | verifyLogin(), determineLocation(), createAccount(), chatSupport() | username, password, loginState, createAccount |
| Stores | This class shall store the number of products that a stores has, the store's name, and its location. | storeproducts() | storeNo, storeName, storeLocation |
| Costmanager | This class shall calculate the cost of all the products that the customer wants and shall give the custmer the cheapest option that a store is offering. | bestItemCostSearch() | itemList, noOfItems, totalCost |
| CustomerInfo | This class shall store the customer's information such as their name, address, phone number, and credit card. The customer can choose to login, register, or update their profiles. | register(), login(), updateProfile() | customerName, customerAddress, phoneNumber, creditCardInfo |
| ShoppingCart | Customers can add the items they want to purchase to the shopping cart. This class can update the number of items, display the current, and proceed to check out. | addItemToCart(), updateNoOfItems(), displayCart(), checkOut() | cartNo, noOfItems, productID |
| ProductDetails | This class shall calculate the price and tax of a product and display it along with its name and the number of it a customer wants | calcPrice(), calcTax() | productID, productName, noOfItems, itemCost, subtotal |
| Order | This class shall allow a customer to place an order or make a payment | placeOrder(), makePayment() | orderNo, orderDate, shipDate, username, userID |
| Shipping | This class shall calculate the shipping cost and record the carrier's name | shippingCost() | shippingCarrierName, shippingOrder, shippingDate |

* 1. **Method**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Description** | **Class** | **Arguments** |
| verifyLogin() | Here the system shall verify the login credentials of the customer. | UserLogin | Username  Password |
| determineLocation() | This method shall determine the location of the customer. | UserLogin | GPS Location |
| createAccount() | This method shall create an account for a new customer. | UserLogin | CustomerName  CustomerAddress  PhoneNumber  Creditcardinfo  Password |
| chatSupport() | It shall allow allow a customer to talk with customer support in case he or she needs help. | UserLogin | Username |
| storeProducts() | This method lists the products in stores that are nearby the customer's location | Stores | Storeno  Storename  Storelocation |
| bestItemCostSearch() | This method shall search through all the stores nearby the customer for the product the customer wants that is the cheapest. | CostManager | Itemlist  Noofitems  Totalcost |
| register() | This method shall register a new user to be a customer of the app. | CustomerInfo | CustomerName  CustomerAddress  PhoneNumber  Creditcardinfo |
| login() | This mehod shall allow a customer to login to the system. | CutomerInfo | Username  Password |
| updateProfile() | This method shall allow a customer to update their profile | CustomerInfo | CustomerName  Username  Password  Creditcardinfo |
| addItemToCart() | This method allows a customer to add items to a shopping cart. | ShoppingCart | Productid  Productname  Itemcost  Subtotal |
| updateNoOfItems() | This method shall update the number of items in the cart. | ShoppingCart | noofitems |
| displayCart() | This method shall display the shopping cart to customers. | ShoppingCart | Cartno  Noofitems  Productid |
| checkOut() | This method shall allow a customer to check out the items in their cart | ShoppingCart | Cartno  Noofitems  Array of Productid |
| calcPrice() | This method shall calculate the price of the product | ProductDetails | Itemcost |
| calcTax() | This method shall calculate the tax on the product | ProductDetails | Itemcost  Shipdate  Orderno |
| placeOrder() | This method shall allow the customer to place an order. | Order | Orderno  Shipdate  Orderdate |
| makePayment() | This method shall allow the customer to place a payment | Order | Creditcardinfo  Orderno  Orderdate  Shipdate |
| shippingCost() | This method will calculate the shipping cost | Shipping | Orderno  Orderdate  Shipdate |

* 1. **Attribute**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Description** | **Class** | **Simple/Complex** | **Types** | **Size** |
| username | The customer's username | UserLogin | Simple | string | small |
| password | The customer's password | UserLogin | Simple | string | small |
| loginState | State of the customer's login. Is he still logged in or not for example | UserLogin | Simple | string | small |
| createAccount | This is a bool as to whether an account has been created or not | UserLogin | Simple | bool | small |
| storeNo | This is the store number of the store that is available to sell product. | Stores | Simple | int | small |
| storeName | This variable holds the name of the store | Stores | Simple | string | small |
| storeLocation | This string contains the location of a nearby store. | Stores | Simple | string | small |
| itemList | This variable holds the list of items the customer wants | Costmanager | Complex | array of strings | large |
| NoOfItems | This is the number of items | Costmanager | Simple | int | small |
| totalCost | This is the total cost of the items. | Costmanager | Simple | int | small |
| customerName | Name of the customer | CustomerInfo | Simple | string | small |
| customer Address | Address of the customer | CustomerInfo | Simple | string | small |
| phoneNumber | This is the phone number | CustomerInfo | Simple | string | small |
| creditCardInfo | This is the credit card information of the customer | CustomerInfo | Simple | string | small |
| cartNo | This is the cart number | ShoppingCart | Simple | int | small |
| NoOfItems | This is the number of items in the cart | ShoppingCart | Simple | int | small |
| productID | This is the ID number of the product | ShoppingCart | Simple | int | small |

* 1. **Relationship**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Description | From class | To class | Optional/Mandatory | Cardinality |
| Directed Association | Carries all the necessary information of the customers and verifies about Login credentials | CustomerInfo | UserLogin | Mandatory | 0…\* |
| Association | Location of the user is taken for determining the stores | UserLogin | Stores | Mandatory | 1 |
| Association | Takes the necessary information of the customer for computing economical products | CustomerInfo | CostManager | Optional | 0…\* |
| Aggregation | Gathers all the product details | ProductDetails | CostManager | Optional | 0…\* |
| Aggregation | Gathers all the product details for customer to add directly | Product Details | CostManager | Mandatory | 0…\* |
| Composition | Takes the necessary information of the customer for computing economical products | ShoppingCart | CustomerInfo | Mandatory | 0…\* |
| Association | Takes the information of all the products and uses it in order to manage orders and calculate shipping elements | ShoppingCart | Order | Mandatory | 0…\* |
| Composition | Takes the necessary information of the customer for computing delivery dates | Order | CustomerInfo | Mandatory | 0…\* |
| Composition | Takes the information from order to calculating the shipping cost and shipping dates | Shopping | Order | Mandatory | 1 |

* 1. **Key Events**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Name** | **Description** | **Motive** | **Actions** | **Pre-Conditions** | **Post-Conditions** | **State Change** |
| UserLogin | Enables the user to login to the system by entering username and password | System Login | Entering Login details by user | None | Login verification | User verification state |
| CustomerInfo | Enables new user to create account | Addition of new user's details | Account Creation | None | Account creation and Login | User enteres the system |
| User Verification | Verifies correctness of username and password entered by the user | User Verification | Successful or Un-successful Login | Username and password required by user | Login to system | User enteres the system |
| CostManager/ List Creation | User view's catalog and enter's list of items to purchase | Creation of list ot items to buy | user input for items to be bought | Login required | Display list of items at stores | Items list prepared |
| ShoppingCart | Adding items from different stores to cart and viewing offers and discounts | Item selection | Adding products to cart | User's list of items to be purchased is required | Availing offers and discounts | Adding items and their quantity |
| Order | Handles checkout and assigning of order number | Unique order number assignment | Purchase of products and gets order number | Checkout | Getting the delivery details | Checkout and order number |
| Stores | Checks for product availability and price at a store | Price and quantity check | Display item to user based on price | List creation by user | If item is purchased, then decremet the quantity of that product | Item quantity updated |
| ProductDetails | Stores product details | Used during purchase of item | Uniquely identifies a product | Availability of product at a store | None | None |
| Shipping | Provides shipping details to user or pick up locations | Gives estimated delivery time | User is informed of product delivery | Checkout | None | Delivery or pick up of item |