



Initial Design Report

Sinem Demirci

Gizem Baştürk

Önder Altan

Emre Işıklıgil

18.12.2009

Table of Contents

1. Introduction.....	3
1.1 Purpose.....	3
1.2 Problem Definition.....	3
1.3 Project Description and Features.....	3
2. Constraints.....	4
2.1 Time Constraints.....	4
2.2 Performance Constraints.....	5
2.3 Portability Constraint.....	6
2.4 Hardware Constraint.....	6
2.5 Software Constraint.....	7
3. Data Design.....	8
3.1 ER Diagrams.....	8
3.2 Database Schema.....	9
3.2.1 User Table.....	9
3.2.2 Product Table.....	9
3.2.3 Promotion Table.....	10
3.2.4 Questionnaire Table.....	10
3.2.5 MapElement Table.....	11
3.2.6 ShoppingHistory Table.....	11
3.2.7 ShoppingItems Table.....	12
3.2.8 Recipe Table.....	12
3.3 File Types.....	13
4. Architectural Design.....	17
4.1 Detailed Module Design.....	17
4.1.1 Component Diagram.....	17
4.1.2 CustomerIntermafe Component.....	18
4.1.3 AdministratorInterface Component.....	18
4.1.4 Customer Component.....	19
4.1.5 Administrator Component.....	19
4.1.6 Map Component.....	20

4.1.7 ShoppingList Component.....	21
4.1.8 Communication.....	22
4.1.9 Adver_&Prom_.....	23
4.1.10 Product&Department.....	23
4.1.11 DB_Access.....	24
4.1.12 DB.....	24
4.2 Detailed Class Design.....	24
4.2.1 Class Diagram.....	24
5. External Interfaces.....	25
5.1 Software Interfaces.....	25
5.1.1 Adobe Flash Player.....	25
5.2 Hardware Interfaces.....	25
5.2.1 Touch Screen.....	25
5.2.2 Card Reader.....	25
5.3 Graphical User Interfaces.....	26
5.3.1 Main Menu.....	26
5.3.2 User Profile.....	27
5.3.3 Virtual Keyboard.....	28
6. Behavioral Design	30
6.1 Activity Diagrams.....	30
6.2 Sequence Diagrams.....	34
7. Gantt Chart.....	36

1. Introduction

1.1 Purpose

This document describes how Supermarket Guide will be prepared to fulfill the requirements identified in the Software Requirements Specification prepared by Bounce. The details of the project will be explained in this document by using specific diagrams such as class, sequence, activity, ER diagrams.

1.2 Problem Definition

The size of supermarkets is getting bigger from day to day. Too many people waste their time in these huge supermarkets reluctantly. Sometimes people spend much time to find just one bottle or something else. Supermarket Guide Kiosk solves these problems with searching product, creating shopping list and giving their path. Besides, this kiosk gives these services with enjoyable ways.

1.3 Project Description and Features

The Supermarket Guide project is intended to provide software for supermarket kiosks to help the customers to find the items they are searching for and to give some necessary information about the items and places. Moreover, this kiosk gives many opportunities to customer. The supermarket guide provides easy way of finding products. It helps to find locations in a supermarket. In this project we planned to provide following features:

- * Getting promotion information: Customers deserve some promotions according to their shopping rate. Every customer has personal promotions. These personal promotions are determined by system according to their most preferring products or departments. Previous shopping information stores in database to reach products or department that more preferred than others.

- * Searching for a product: With this option customer search a product. The system returns properties of product and location information.

* Creating shopping list: The shortest and most advantageous path is created after this option. Primarily customers add products to list or remove products from list. Then finalize their list. According to this list the system determines the locations of the products and returns the shortest and most advantageous path of current list. With this option customer spend little time while shopping process.

* Recipes: This option is very useful to prevent forgetting some ingredients of a meal. Besides, it provides directions, ready time information. After customers choose recipe, the system takes ingredients information. According to this information it locates the products. Then it returns the shortest and most advantageous path.

*Online Access to System: Kiosk system can be accessed through web. This enables customers with limited time to create their shopping list before coming to the store.

*Usage of Administrators: Administrators of the kiosk system can construct a map of store including all departments and item information. They can publish advertisements on the interface of the users. They can put some questionnaires to the customer.

*Login of Costumers: Customers will be login with a card reader. Every loyalty customer will have their own card.

Briefly, this kiosk provides customers to spend time more efficiently. With this kiosk, customers can reach the products quickly. They can generate list of products. Well prepared GUI attracts customers and makes it easy to use kiosk systems. That is to say, this Supermarket Guide makes shopping convenient and then makes customers happy.

2. Constraints

2.1 Time Constraints

Total period intended for this project to be completed is two academic semesters. Design part will be completed in the first academic semester. During the second

semester, implementation of the project is going to be done. Testing of the product is going to be handled while implementing it.

Both software development and implementation periods have strict deadlines. We should distribute the workload between deadlines equally to be able to use our time efficiently.

At the end of the second semester, we are going to be completed implementation of the project and we are going to be using the product with all features included to it.

2.2 Performance Constraints

Supermarket Guide should respond the commands of the users in a reasonable time. Actually, there are two main tasks concerning performance of the product. One of them is retrieving data belonging to a user from database. The other one is executing the Flash application which does 3D visualization of the supermarket.

Some relations in the database occupy huge number of pages. In addition, size of them increase continuously. One example of great-size relations is the relation which holds the shopping history of the customers. Size of that relation increases every time when a customer with loyalty card buys something. It does not matter how big the size of the relation, Supermarket Guide should respond a user who wants to see his/her shopping history in a reasonable time.

3D visualization of the supermarket is the other issue which should be considered by the mean of performance. It can increase response time of Supermarket Guide to a user that increasing size of the supermarket. Because this kiosk solution is being developed for supermarkets which have lots of departments and great square, its response time should be reasonable while visualizing the inside of the supermarket.

2.3 Portability Constraints

Because Supermarket Guide is developed using PHP, it can be moved to any server that has PHP and MySQL support. There is no operating system dependency. To be able to access the server and play Flash applications, a browser and Flash Player should be installed on kiosks. If Supermarket Guide moved to a server with WAN connection, it can be accessed via WAN anywhere in the world. So, Supermarket Guide provides pretty enough portability.

2.4 Hardware Constraints

- The system should have minimum 1.3 Ghz Intel or AMD Processor.
- The system should have a main board with support of VGA LAN and sound.
- Minimum of 64 MB AGP Graphic Card should be available on the system.
- The system should have a DDR2 Memory with at least 512 MB capacity.
- An external storage of minimum 40 GB should be available on the system.
- The system should provide at least three USB2.0 connections.
- Minimum 15" LCD TFT Touch Screen should be connected to the system.
- A magnetic card reader should be available on the system.
- The system should be connected to a power supply that provide minimum 250 Watt power.
- The system should have a cooler bigger than 10x10 cm should be connected to the system.
- The system should be integrated into a metal kiosk frame.

2.5 Software Constraints

- Supermarket Guide could run on all distributions of Linux, MS Windows 95/98, MS Windows NT/2000/XP, 32-bit and 64-bit MS Windows Vista, 32-bit and 64-bit Windows Seven, 32-bit and 64-bit Windows Server 2003/2008.
- Since the product is developed using PHP web-based programming language, PHP server should be installed. That means the Apache Server should be installed on the system. Moreover, this server should support PHP.
- Since the product uses a database to keep track of products, the system should have MySQL Database System installed on it.
- Kiosks should have a web browser installed on it. Mozilla Firefox , MS Internet Explorer 6.0 or upper , Opera 7.0 or upper version should be used.
- Adobe Flash Player 7 or upper version should be installed on kiosks to see flash animations and applications.
- In order to use the system through web, internet connection should be established on the server and kiosks.

3 Data Design

3.1 ER Diagram

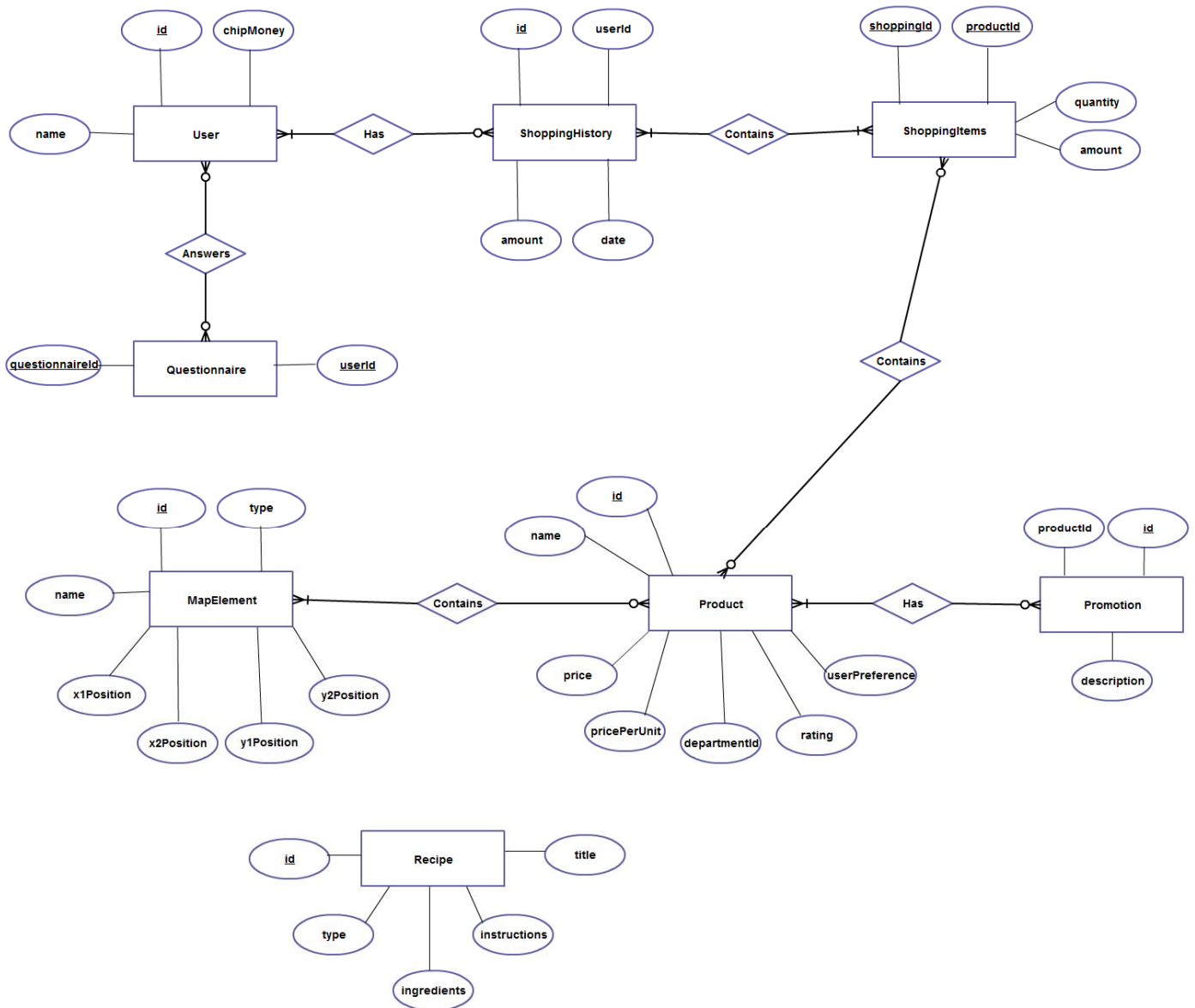


Figure.1- ER Diagram

3.2 Database Schema

3.2.1 User Table

This table holds the necessary information about the customers. Since we are intended to use the card system in logging this software, it is not needed to hold a password field.

```
CREATE TABLE User
(
    id integer NOT NULL AUTO_INCREMENT,
    name varchar(25) NOT NULL,
    chipMoney int,
    PRIMARY KEY(id)
)
```

3.2.2 Product Table

Product table holds the information about the product itself. In order to hold statistical data, “rating” and “userPreference” fields are included.

```
CREATE TABLE Product
(
    id integer NOT NULL AUTO_INCREMENT,
    name varchar(25) NOT NULL,
    price integer NOT NULL,
    pricePerUnit integer NOT NULL,
    department varchar(50) NOT NULL,
    rating integer,
    userPreference integer,
    PRIMARY KEY(id) )
```

3.2.3 Promotion Table

Promotion table holds the promotion information of the related product. Since most of the products do not have a promotion, it is not formed as a separate column in the Product table. “productId” should match with the “id” column of the Product table.

```
CREATE TABLE Promotion
(
    id integer NOT NULL AUTO_INCREMENT
    productId integer NOT NULL,
    description varchar(60) NOT NULL,
    PRIMARY KEY(id),
    FOREIGN KEY(productId) REFERENCES Product(id)
)
```

3.2.4 Questionnaire Table

This table is formed to hold the information about which user answered which questionnaire or complaint in the system. Since we do not want users to answer the same questionnaire more than one time, both “questionnaireId” and “userId” are decided to be the primary key of the table.

```
CREATE TABLE Questionnaire
(
    questionnaireId integer NOT NULL,
    userId integer NOT NULL,
    PRIMARY KEY(questionnaireId, userId),
    FOREIGN KEY(userId) REFERENCES User(id)
)
```

3.2.5 MapElement Table

MapElement table holds the information about the overall map of the supermarket. All construction elements are indicated with positions and names. “type” column shows the type of construction element which are supermarket departments, cash points and other type of areas.

```
CREATE TABLE MapElement
(
    id integer NOT NULL AUTO INCREMENT,
    name varchar(50) NOT NULL,
    x1Position integer NOT NULL,
    x2Position integer NOT NULL,
    y1Position integer NOT NULL,
    y2Position integer NOT NULL,
    type integer NOT NULL,
    PRIMARY KEY(id)
)
```

3.2.6 ShoppingHistory Table

This table holds the information about each shopping which is done by user. This table is used to provide statistical information to the customer afterward.

```
CREATE TABLE ShoppingHistory
(
    id integer NOT NULL AUTO_INCREMENT,
    userId integer NOT NULL,
    date date NOT NULL,
    amount double NOT NULL,
    PRIMARY KEY(id),
```

```
FOREIGN KEY(userId) REFERENCES User(id)
)
```

3.2.7 ShoppingItems Table

ShoppingItems table holds the information about the items which are bought in a specific shopping.

For one shopping item, both “shoppingId” and “productId” cannot be same at the same time. Therefore it is assigned as the primary key of the table. “shoppingId” should match with the “id” column of the ShoppingHistory table and “productId” should match with the “id” column of the Product table.

```
CREATE TABLE ShoppingItems
(
    shoppingId integer NOT NULL,
    productId integer NOT NULL,
    quantity int NOT NULL,
    amount double NOT NULL,
    PRIMARY KEY(shoppingId, productId),
    FOREIGN KEY(shoppingId) REFERENCES ShoppingHistory(id),
    FOREIGN KEY (productId) REFERENCES Product(id)
)
```

3.2.8 Recipe Table

This table holds the necessary information about recipes. Afterward, this data are used to make offerings to the customer.

```
CREATE TABLE Recipe
```

```
(  
    id integer NOT NULL AUTO_INCREMENT,  
    type varchar(40) NOT NULL,  
    title varchar(40) NOT NULL,  
    ingredients text,  
    instructions text,  
    PRIMARY KEY(id)  
)
```

3.3 File Types

The supermarket kiosk software needs to store questionnaires and complaint forms in the system. They are considered to be stored in the XML file format since XML is a very useful format. It is easily read, parsed and written by many applications. The below is the format of an example questionnaire file:

```
<?xml version="1.0" encoding="UTF-8" ?>  
<questionnaires>  
    <questionnaire id = "1">  
        <question>  
            <question_text>Question-1 text is here</question_text>  
            <choice>Choice-1 text is here</choice>  
            <choice>Choice-2 text is here</choice>  
            <choice>Choice-3 text is here</choice>  
            <choice>Choice-4 text is here</choice>  
        </question>  
        <question>  
            <question_text>Question-2 text is here</question_text>  
            <choice>Choice-1 text is here</choice>  
            <choice>Choice-2 text is here</choice>
```

```
<choice>Choice-3 text is here</choice>
<choice>Choice-4 text is here</choice>
</question>
<question>
  <question_text>Question-3 text is here</question_text>
  <choice>Choice-1 text is here</choice>
  <choice>Choice-2 text is here</choice>
  <choice>Choice-3 text is here</choice>
  <choice>Choice-4 text is here</choice>
</question>
</questionnaire>
<questionnaire id = "2">
  <question>
    <question_text>Question-1 text is here</question_text>
    <choice>Choice-1 text is here</choice>
    <choice>Choice-2 text is here</choice>
    <choice>Choice-3 text is here</choice>
  </question>
  <question>
    <question_text>Question-2 text is here</question_text>
    <choice>Choice-1 text is here</choice>
    <choice>Choice-2 text is here</choice>
    <choice>Choice-3 text is here</choice>
  </question>
  <question>
    <question_text>Question-3 text is here</question_text>
    <choice>Choice-1 text is here</choice>
    <choice>Choice-2 text is here</choice>
    <choice>Choice-3 text is here</choice>
  </question>
  <question>
```

```

        <question_text>Question-4 text is here</question_text>
        <choice>Choice-1 text is here</choice>
        <choice>Choice-2 text is here</choice>
        <choice>Choice-3 text is here</choice>
    </question>
</questionnaire>
</questionnaires>

```

Complaint form format is similar to the questionnaire format. An example is shown below:

```

<?xml version="1.0" encoding="UTF-8" ?>
<complaints>
    <complaint id = "1">
        <complaint_question>
            <question_text>Question-1 text is here</question_text>
            <choice>Choice-1 text is here</choice>
            <choice>Choice-2 text is here</choice>
        </complaint_question>
        <complaint_question>
            <question_text>Question-2 text is here</question_text>
            <choice>Choice-1 text is here</choice>
            <choice>Choice-2 text is here</choice>
        </complaint_question>
        <complaint_question>
            <question_text>Question-3 text is here</question_text>
            <choice>Choice-1 text is here</choice>
            <choice>Choice-2 text is here</choice>
        </complaint_question>
        <complaint_question>
            <question_text>Question-4 text is here</question_text>

```



```
<choice>Choice-1 text is here</choice>
<choice>Choice-2 text is here</choice>
</complaint_question>
</complaint>
<complaint id = "2">
  <complaint_question>
    <question_text>Question-1 text is here</question_text>
    <choice>Choice-1 text is here</choice>
    <choice>Choice-2 text is here</choice>
    <choice>Choice-3 text is here</choice>
    <choice>Choice-4 text is here</choice>
  </complaint_question>
  <complaint_question>
    <question_text>Question-2 text is here</question_text>
    <choice>Choice-1 text is here</choice>
    <choice>Choice-2 text is here</choice>
    <choice>Choice-3 text is here</choice>
    <choice>Choice-4 text is here</choice>
  </complaint_question>
  <complaint_question>
    <question_text>Question-3 text is here</question_text>
    <choice>Choice-1 text is here</choice>
    <choice>Choice-2 text is here</choice>
    <choice>Choice-3 text is here</choice>
    <choice>Choice-4 text is here</choice>
  </complaint_question>
</complaint>
</complaints>
```

4. Architectural Design

4.1 Detailed Module Design

4.1.1 Component Diagram

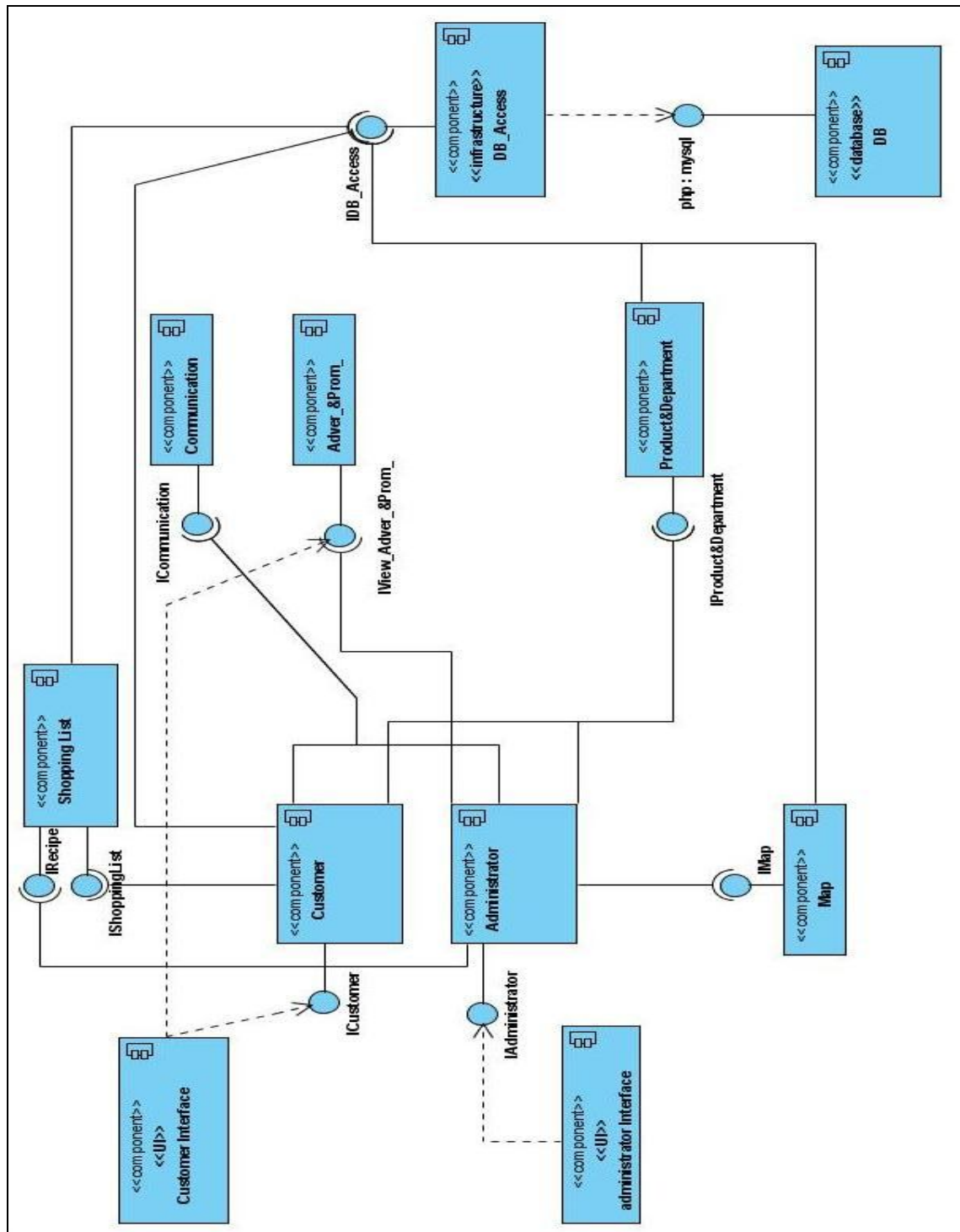


Figure.2-Component Diagram

Our design is divided into components. There exist eleven components namely:

- CustomerInterface
- AdministratorInterface
- Customer
- Administrator
- Map
- ShoppingList
- Communication
- Adver_&Prom_
- Product&Department
- DB_Access
- Database

4.1.2 CustomerIntermafe Component

Customer will enter the system by using provided user interface. This interface will be prepared by system developers and it will provide easy accessibility to customers. Also UI will interact with the Adver_&Prom_ component. The reason is that advertisements and promotion will be displayed on the kiosks independent from customers. CustomerInterface component enables UI requirements.

4.1.3 AdministratorInterface Component

Administrators of the system will have different UI than the customers. AdministratorInterface component keeps the subsystem components of UI.

4.1.4 Customer Component

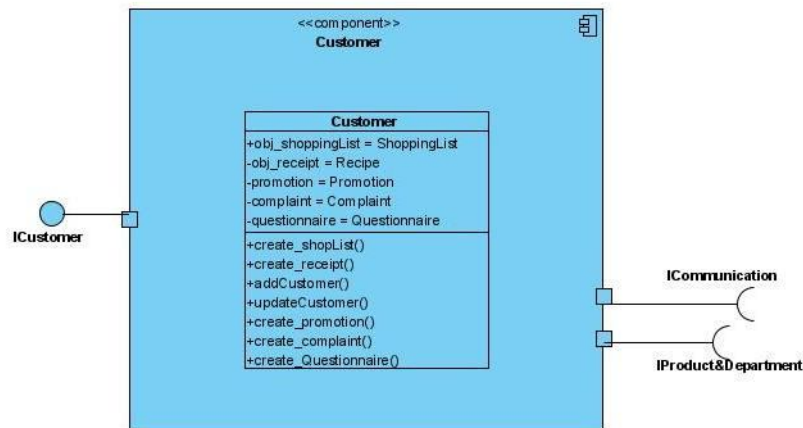


Figure.3- Customer Component

Customer component is composed of Customer class. For each user there will be a Customer object. That object will keep the information of customers. The provided interface of Customer component which is ICustomer, will be used by CustomerInterface component. By UI, a customer can update his/her information using updateCustomer() method. In addition, Customer component will interact with the Communication, Product&Department and ShoppingList component using the ICommunication, IProduct&Department and IShoppingList interfaces respectively. Using Customer class, customers can create their shopping list, find recipe, fill the questionnaire, fill the form of complaint.

4.1.5 Administrator Component

Administrator component is composed of Administrator class. The provided interface of IAdministrator communicates with the Administrator UI. Administrator component uses the ICommunication, IAdver_&Prom_, IProduct&Department and IMap interfaces. By these interfaces administrator can create questionnaire and complaint form then get the results, can determine the advertisements and promotions, can create

the map by putting departments and can fill the departments with the corresponding products.

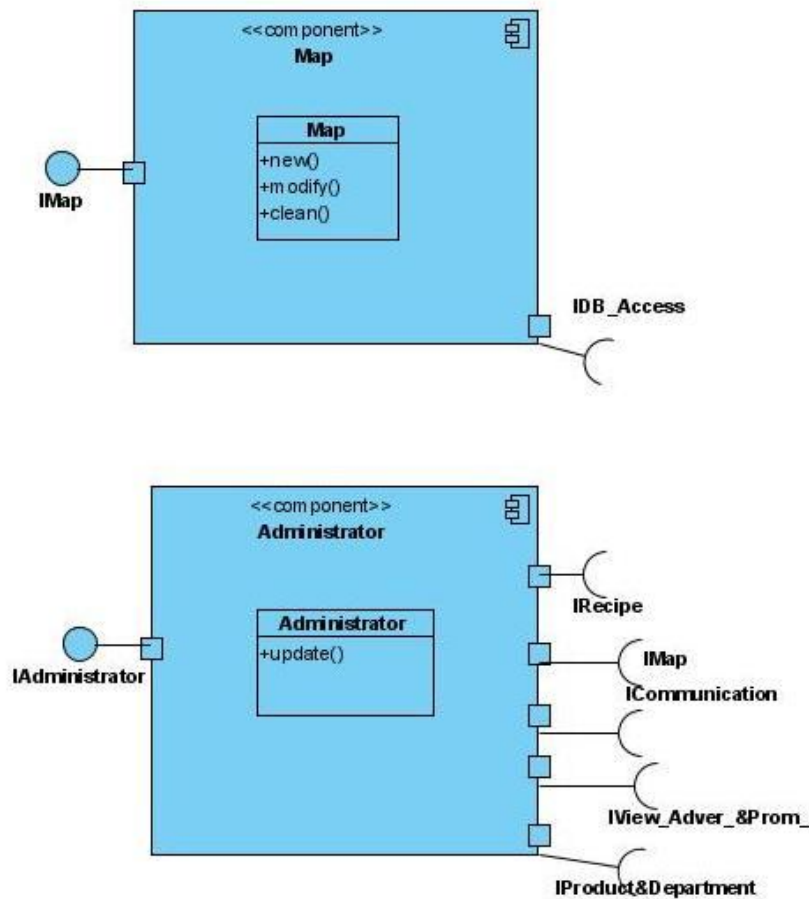


Figure.4- Map and Administrator Component

4.1.6 Map Component

Map component contains Map class. That class creates, modifies and cleans a map using create(), modify() and clean() methods respectively. Map component provides an IMap interface to make use of the map object by system. Also Map component uses database by accessing DB_Access interface that is IDB_Access.

4.1.7 ShoppingList Component

This component is used for creating shopping list, creating path according to the shopping list created, finding recipe and path of list which is used in recipe. This component provides IShoppingList interface and it uses IDB_Access interface to reach DB.

ShoppingList component is composed of ShoppingList, Online_shoppingList, Algorithm and Recipe classes. ShoppingList class creates shopping list of customer by add/remove methods. At the end customer finalizes the list. If want, a customer can get the path of list using createPath() method and get the printed copy of path using printPath(). When createPath() is called, using pre-determined algorithm, the path is created. Algorithm is executed by Algorithm class.

Using Recipe class Customer can get the recipe and get the list of products for a recipe. An administrator can put a recipe or delete it.

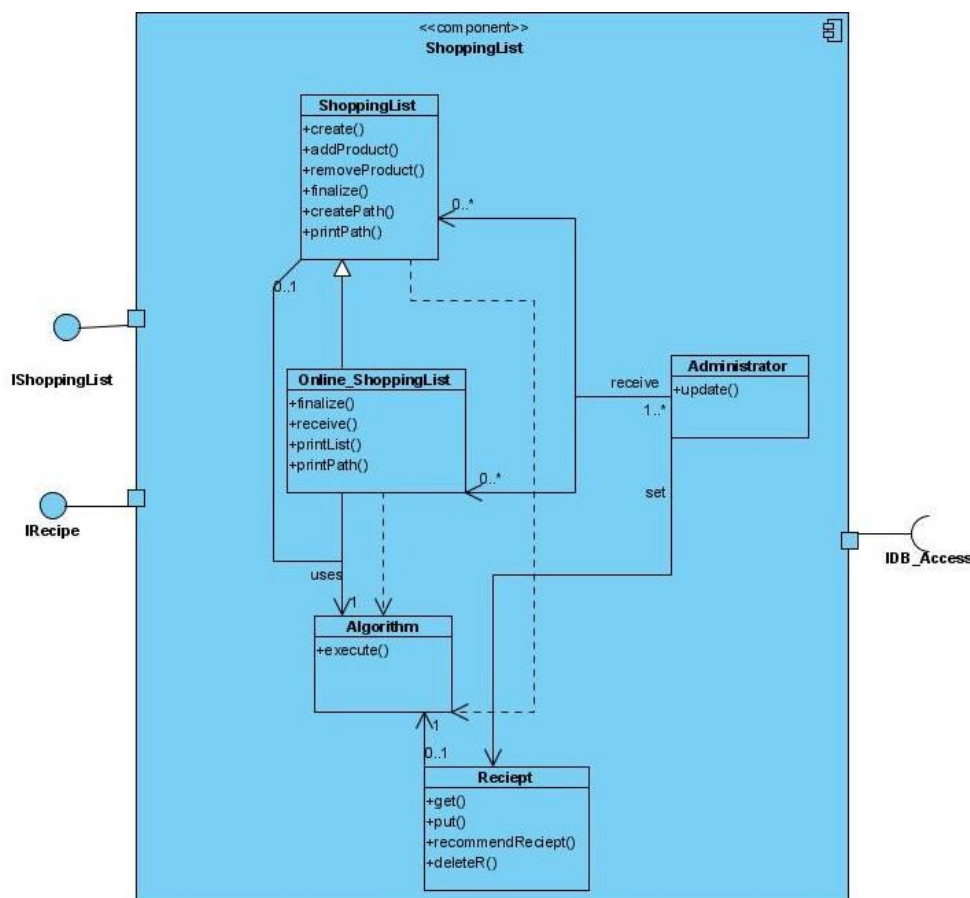


Figure.5- Shopping List Component

4.1.8 Communication

Communication component has Complaint and Questionnaire classes. Using the ICommunication interface of Communication component required components interacts with this component.

A customer can fill the questionnaire or complaint forms. These forms are prepared and put into the system by administrator and results also be reached by administrator.

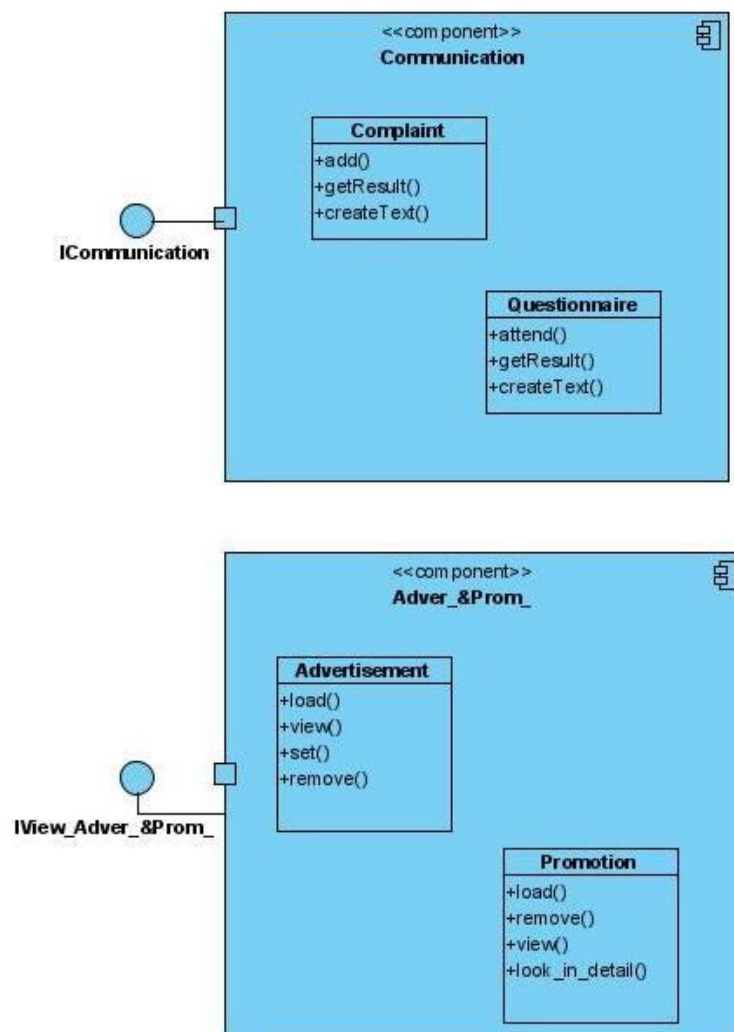


Figure.6- Communication and Adver_&Prom Components

4.1.9 Adver_&Prom_

Adver_&Prom_ component contains Advertisement and Promotion classes. Using the IAdver_&Prom_ interface, other components interact with this component.

The administrator determines the advertisement and promotions. He/she loads or removes them to the system. Using view(), advertisements and promotions can be viewed on the screen.

4.1.10 Product&Department

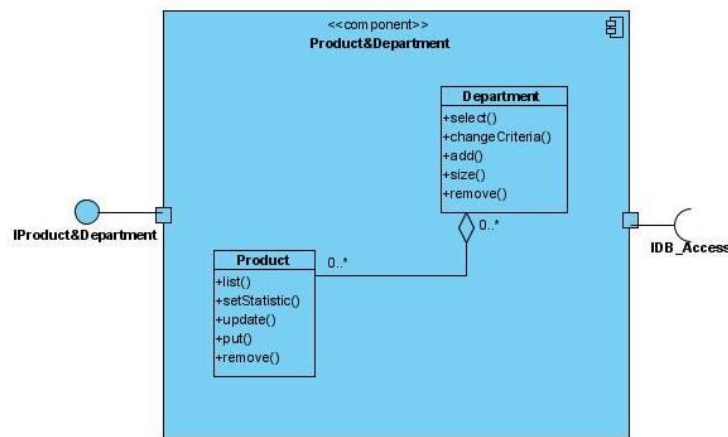


Figure.7- Product&Department Component

Product&Department component is composed of Product and Department classes. Administrators and customers can reach that component by using provided interface of IProduct&Department. Using IDB_Access interface, products and departments interact with DB. Using the product class a customer can get the product list and get statistic of the product. An administrator can add/remove or modify the product. A customer also, selects department. Departments are determined by administrators by using add/size/remove methods.

4.1.11 DB_Access

DC_Access is an infrastructure. It makes available of using DB.

4.1.12 DB

DB is a database. It contains every product, department, user information. When an information is required, system get data from database by using provided interface.

4.2 Detailed Class Design

4.2.1 Class Diagram

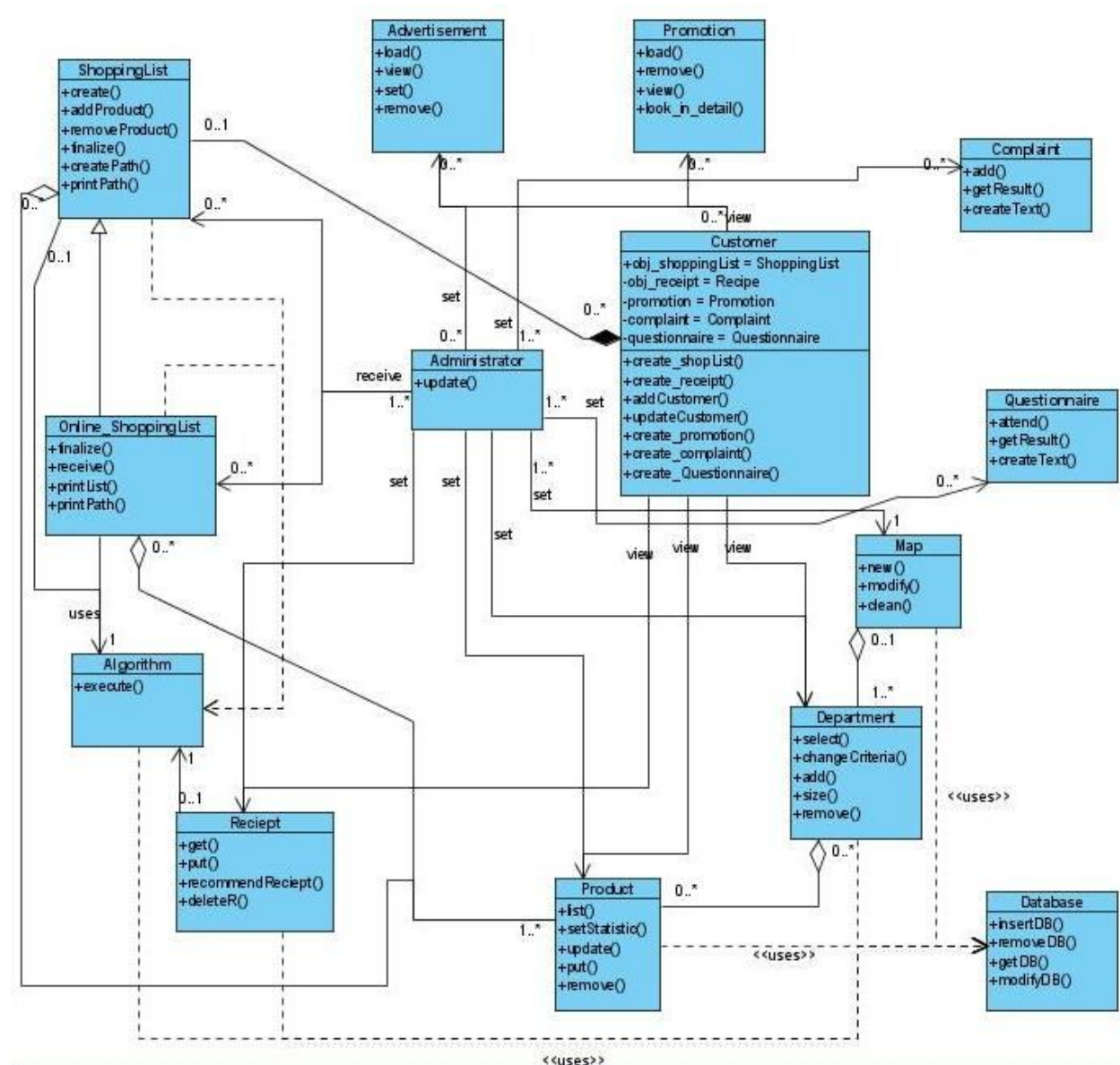


Figure.8- Class Diagram

5. External Interfaces

5.1 Software Interfaces

Because Supermarket Guide is a web-based application which uses local network it does not deploy any external communication software to establish a communication between server and stand-alone kiosks.

Supermarket Guide does not use any external library and software other than Adobe Flash Player. All features and facilities can be implemented using PHP core except viewing 3D visualization.

5.1.1 Adobe Flash Player

Supermarket Guide communicates with Adobe Flash Player to be able to retrieve data from an embedded Flash Application which takes care of 3D visualization part.

Adobe Flash Player takes input from users via Flash Application and sends output to Supermarket Guide. Actually, it is an interface between Flash Application and Supermarket Guide but, it provides interaction between users and Supermarket Guide.

5.2 Hardware Interfaces

Supermarket Guide uses some electronic devices to make the users be able to interact with the system. These electronic devices provide interfaces between the users and the system. They enable interaction of the users with the system.

There are two main hardware which provide an interface between the users and the system. One of them is touch screen and the other one is card reader.

5.2.1 Touch Screen

Touch screen is the main device which establishes communication between the users and the system. It is the main input and the main output device. It provides an interface between the users and the system which enables user to send command to Supermarket Guide.

Touch screen takes input from the users and sends output to Supermarket Guide. It also takes input from Supermarket Guide and sends output to the users. Inputs of touch screen are clicks of the users on the screen. It sends coordinates of the clicked point to the system. So, it makes analog to digital conversion.

5.2.2 Card Reader

Card reader provides an interface between the users and Supermarket Guide. It enables the users to be able to log into the system.

Card reader takes input from card of the users by reading magnetic tapes of them and sends outputs to Supermarket Guide. Inputs of card reader are signs on the magnetic tapes. Output type of a card reader is a simple double number which is account number of a user. Card reader also makes analog to digital conversion like touch screen.

5.3 Graphical User Interfaces

Because Supermarket Guide System does not include keyboard or any other input device other than touch screen and card reader, Graphical User Interface of the system should be designed such a way that all possible commands which a user may give to the system should be able to be given through it. That, Graphical User Interface should be designed so that the users could use all features of the system by touching to the screen.

Supermarket Guide provides a Graphical User Interface consisting of buttons and textboxes to enable the users give commands. Using that buttons, the users could change between menu pages, write to textboxes, select among choices.

Some examples pages which include main features of Graphical User Interface are illustrated in subsections of this section.

5.3.1 Main Menu

The main menu is the menu opened when a user logs into the system using loyalty card. All submenus and features could be accessed by following some access paths starting from the main menu. Access paths are followed by clicking on the buttons on the touch screen. The main menu window is illustrated below.

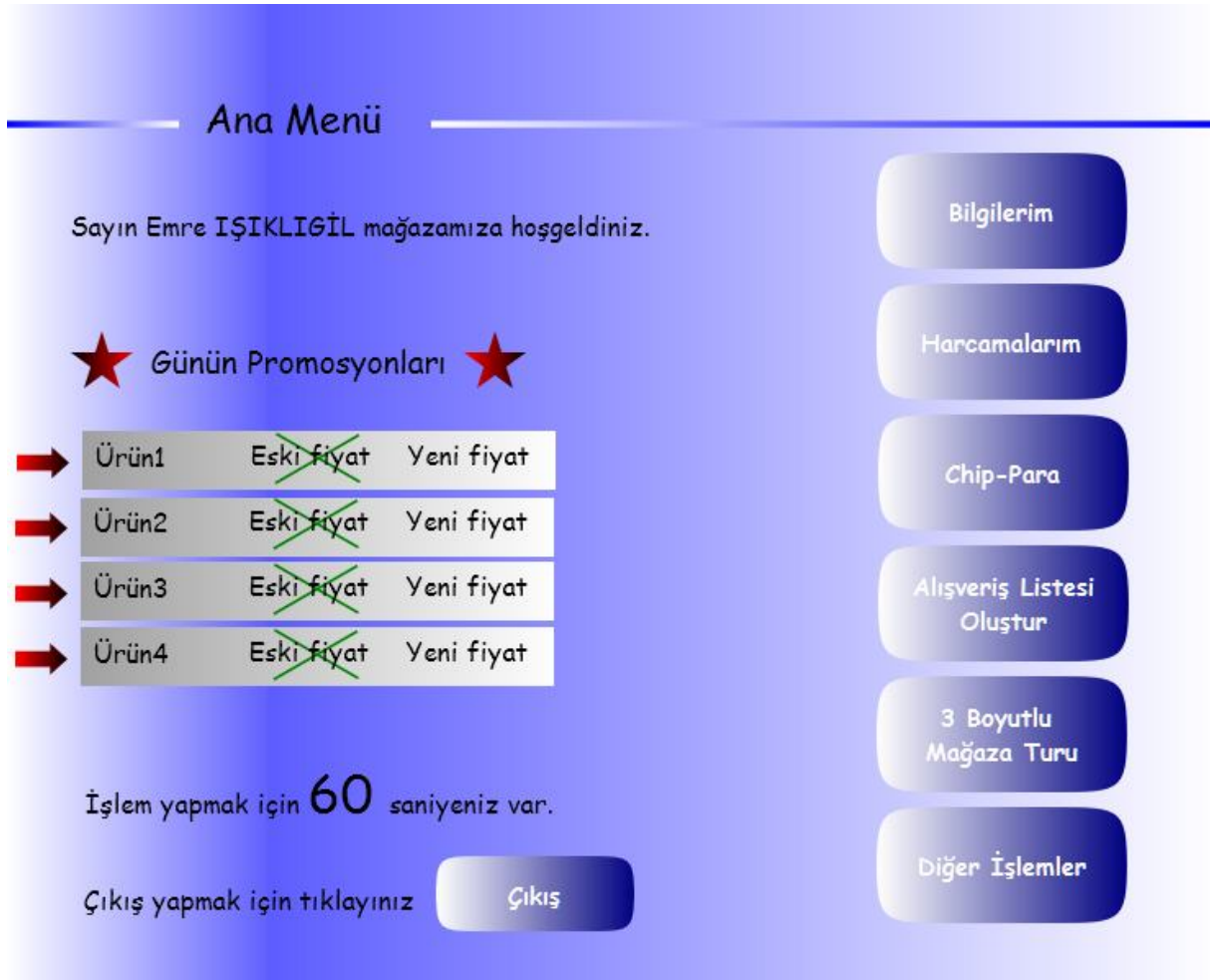


Figure.9- Main Menu

Graphical User Interface of Supermarket Guide give maximum 60 seconds idle time to the users. If no input is entered during 60 seconds, session of the current user is automatically ended.

5.3.2 User Profile

Users of the system can update their individual information using User Profile menu. This menu illustrates use of textboxes. Below a screenshot of User Profile Menu is figured out.

Bilgilerim

Ana menüye dönmek için tıklayınız. **Ana Menü**

Ad: EMRE Soyad: IŞIKLIGİL Ev Telefonu: 0312 123 45 67

E-Mail: emre_isikligil@yahoo.com Cep Telefonu: 0589 123 45 67

Adres: 100. YIL İŞÇİ BLOKLARI MAHALLESİ
432. SOKAK H BLOK A-6
BALGAT/ANKARA

İşlem yapmak için **60** saniyeniz var. Çıkış yapmak için tıklayınız. **Çıkış**

Figure.10- User Profile

There is a button linked to main menu in all submenus. Using this, all access paths accessed from main menu are also accessed from submenus. This property makes user interface easy-use and understandable.

5.3.3 Virtual Keyboard

Virtual Keyboard is one of the most important features of the Graphical User Interface. It makes everything easy while updating contents of textboxes. When a user wants to change content of a textbox, the virtual keyboard pops up immediately and enables the user change the content easily. A screenshot while changing the content of a textbox is figured out below. While changing content of a textbox all other textboxes and buttons projected out from the screen for a simple look.

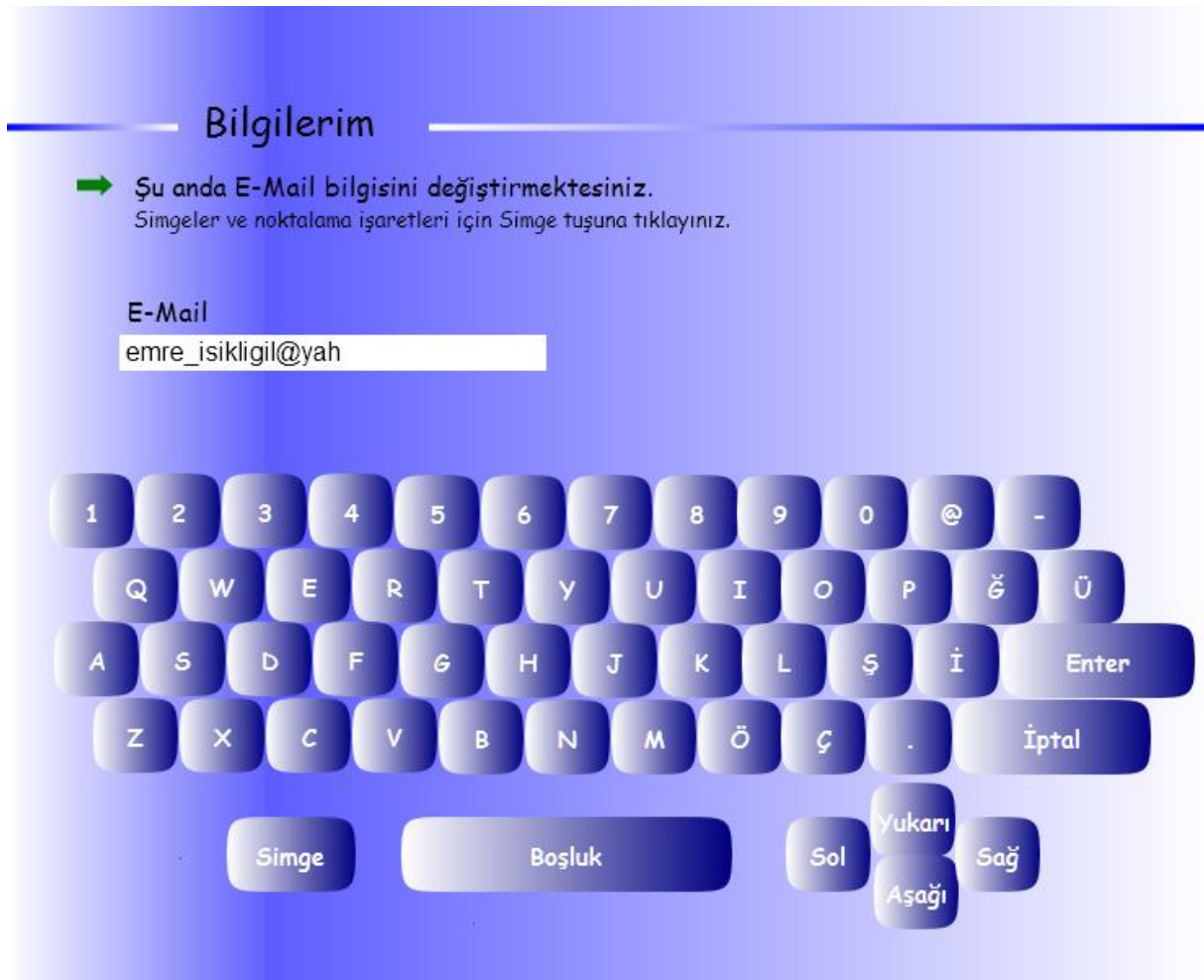


Figure.11- Virtual Keyboard

Virtual Keyboard makes it easy to use the GUI and provides high functionality to it.

6. Behavioral Design

6.1 Activity Diagrams

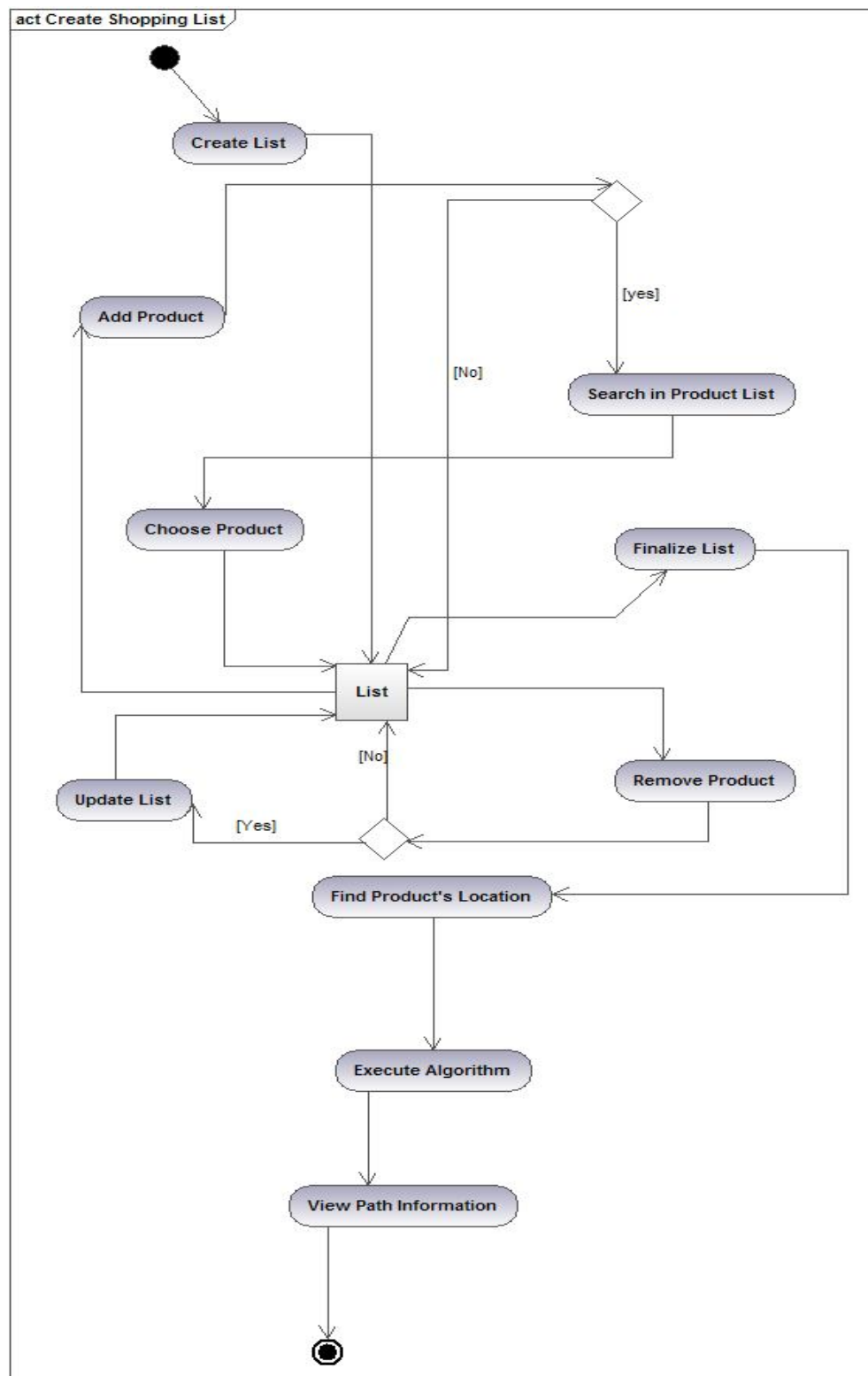


Figure.12- Activity Diagram I: Create Shopping List

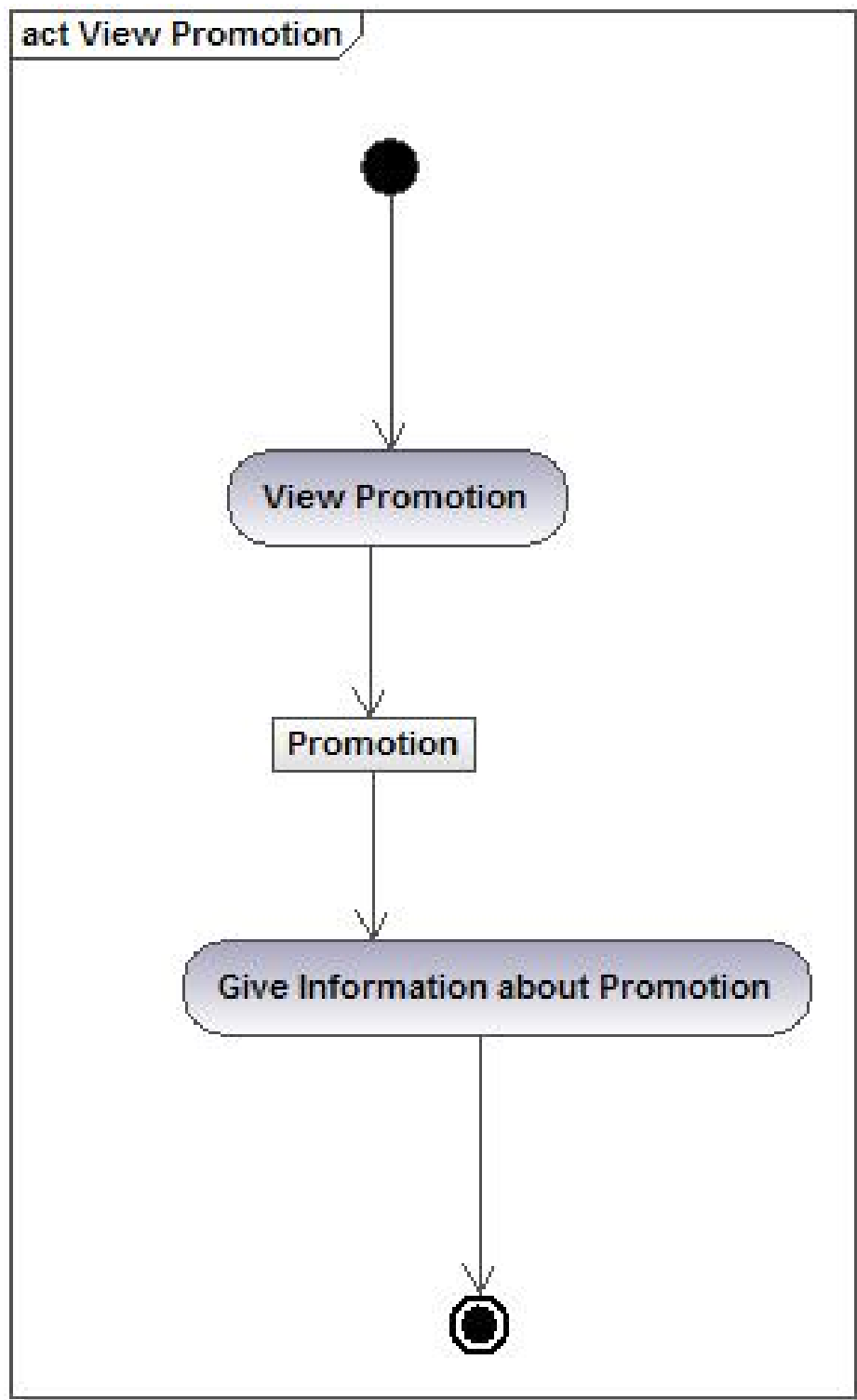


Figure.13- Activity Diagram II: View Promotion

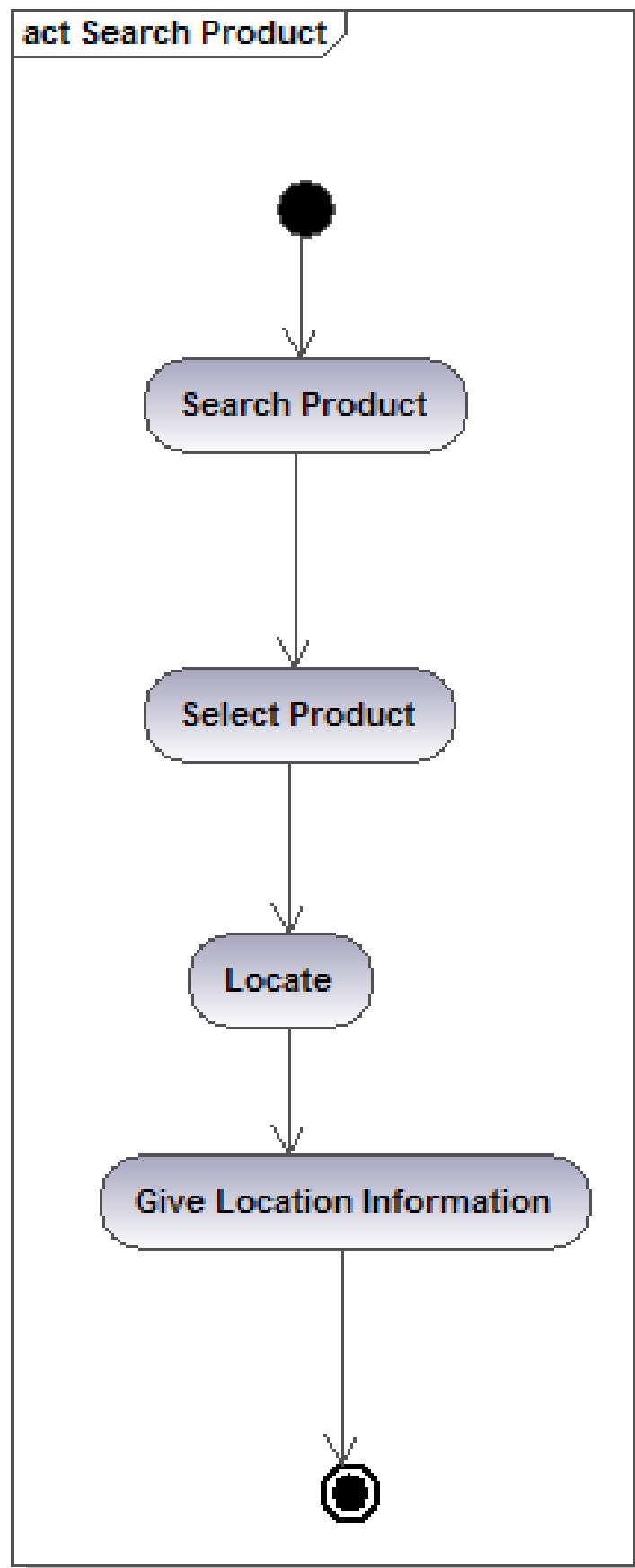


Figure.14- Activity Diagram III: Search Product

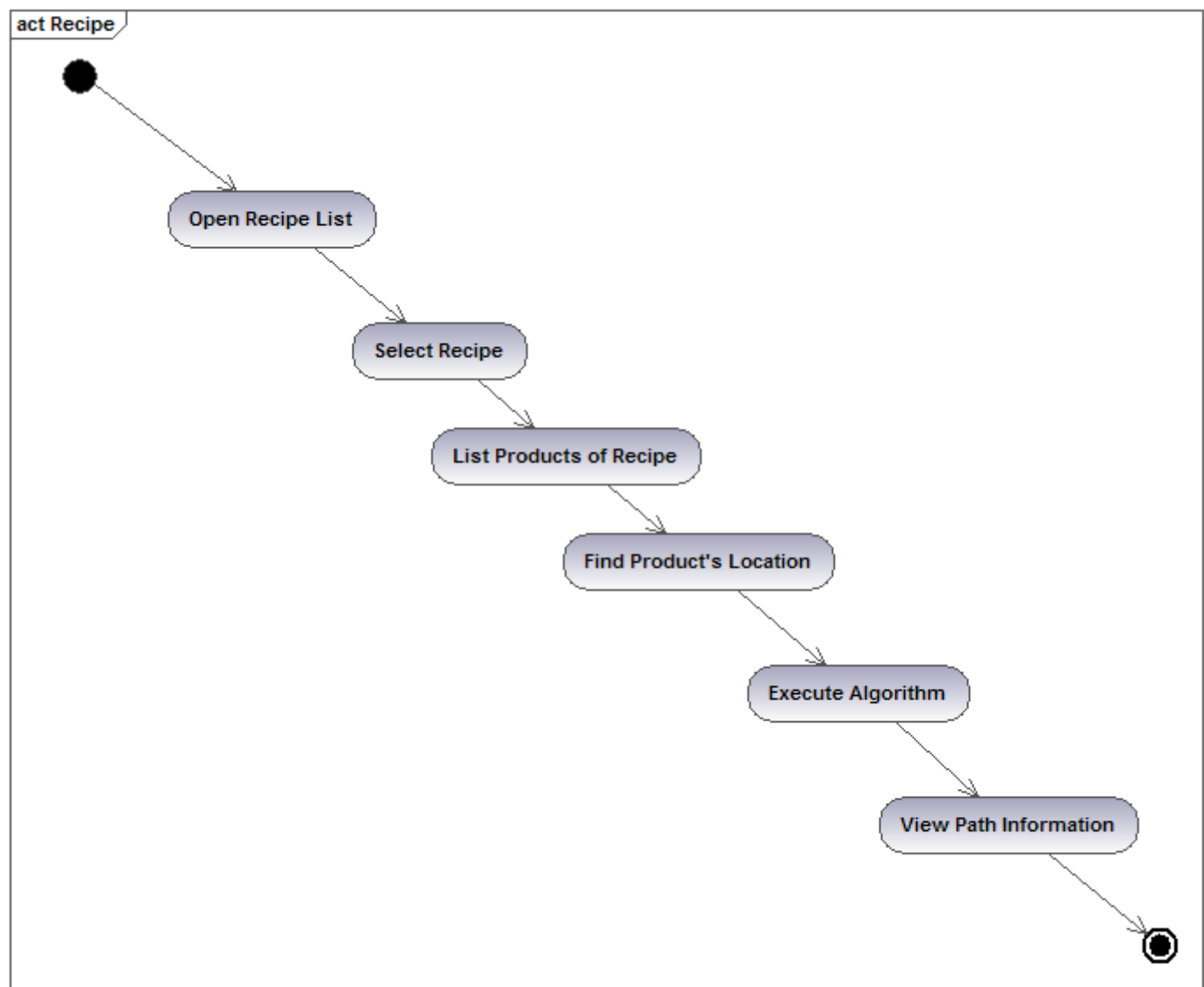


Figure.15- Activity Diagram IV: Recipes

6.2 Sequence Diagrams

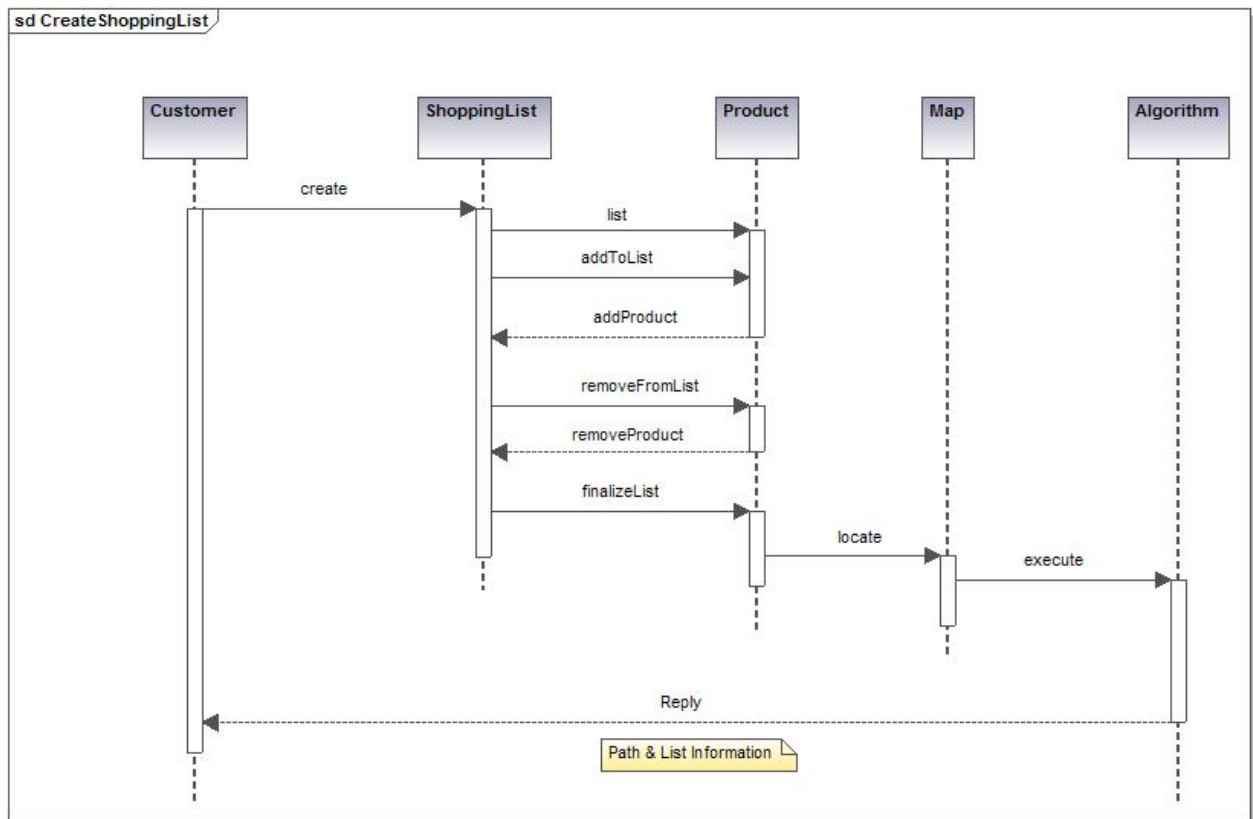


Figure.16- Sequence Diagram I: Create Shopping List

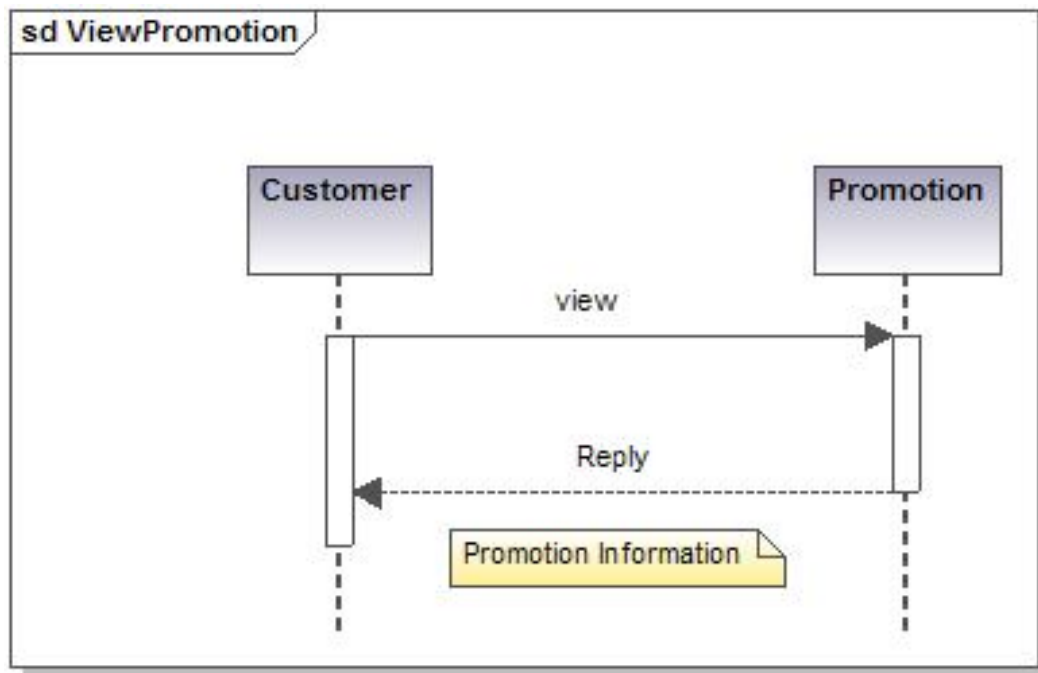


Figure.17- Sequence Diagram II: View Promotion

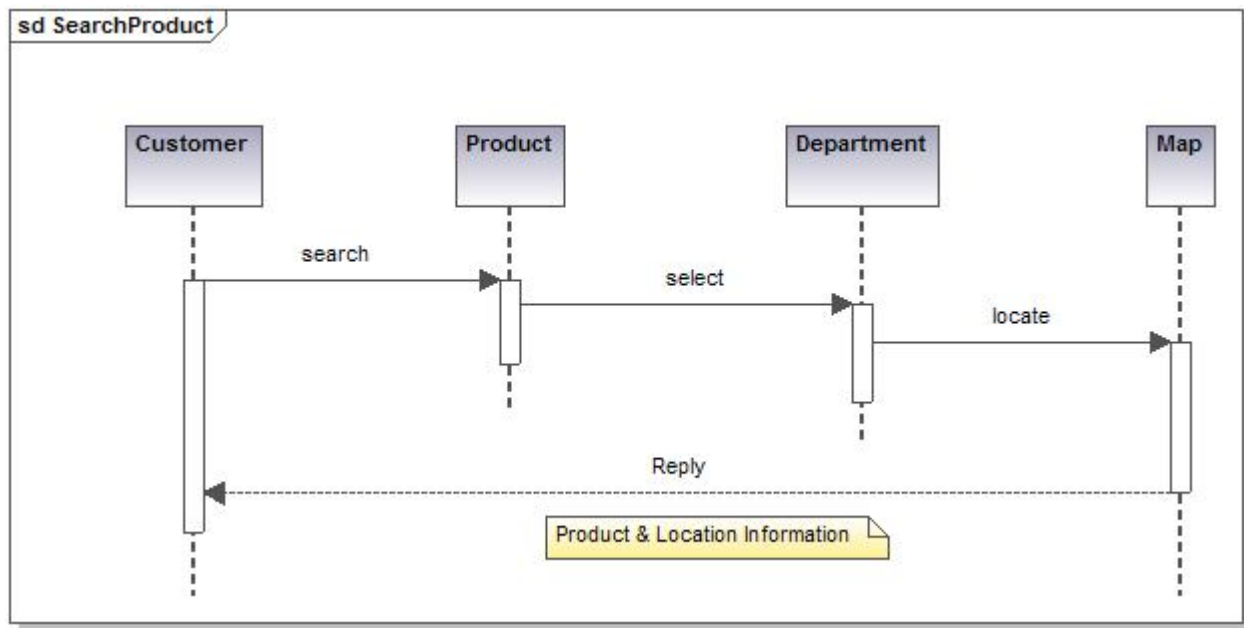


Figure.18- Sequence Diagram III: Search Product

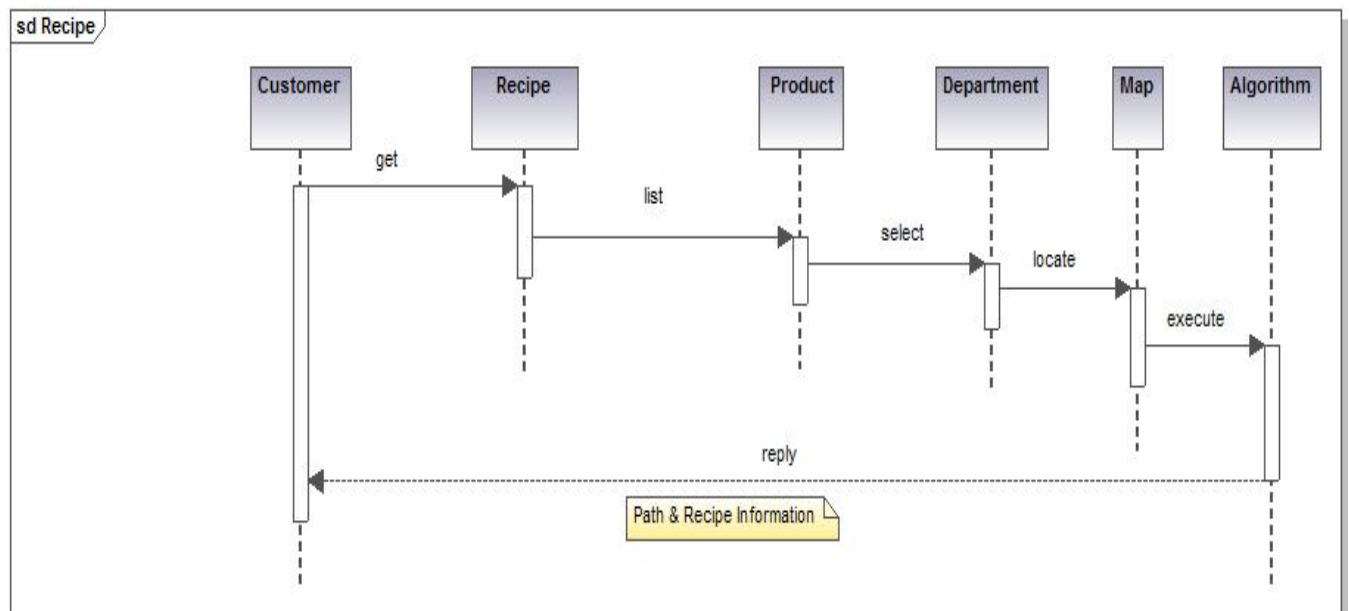


Figure.19- Sequence Diagram IV: Recipes

7. Gantt Chart

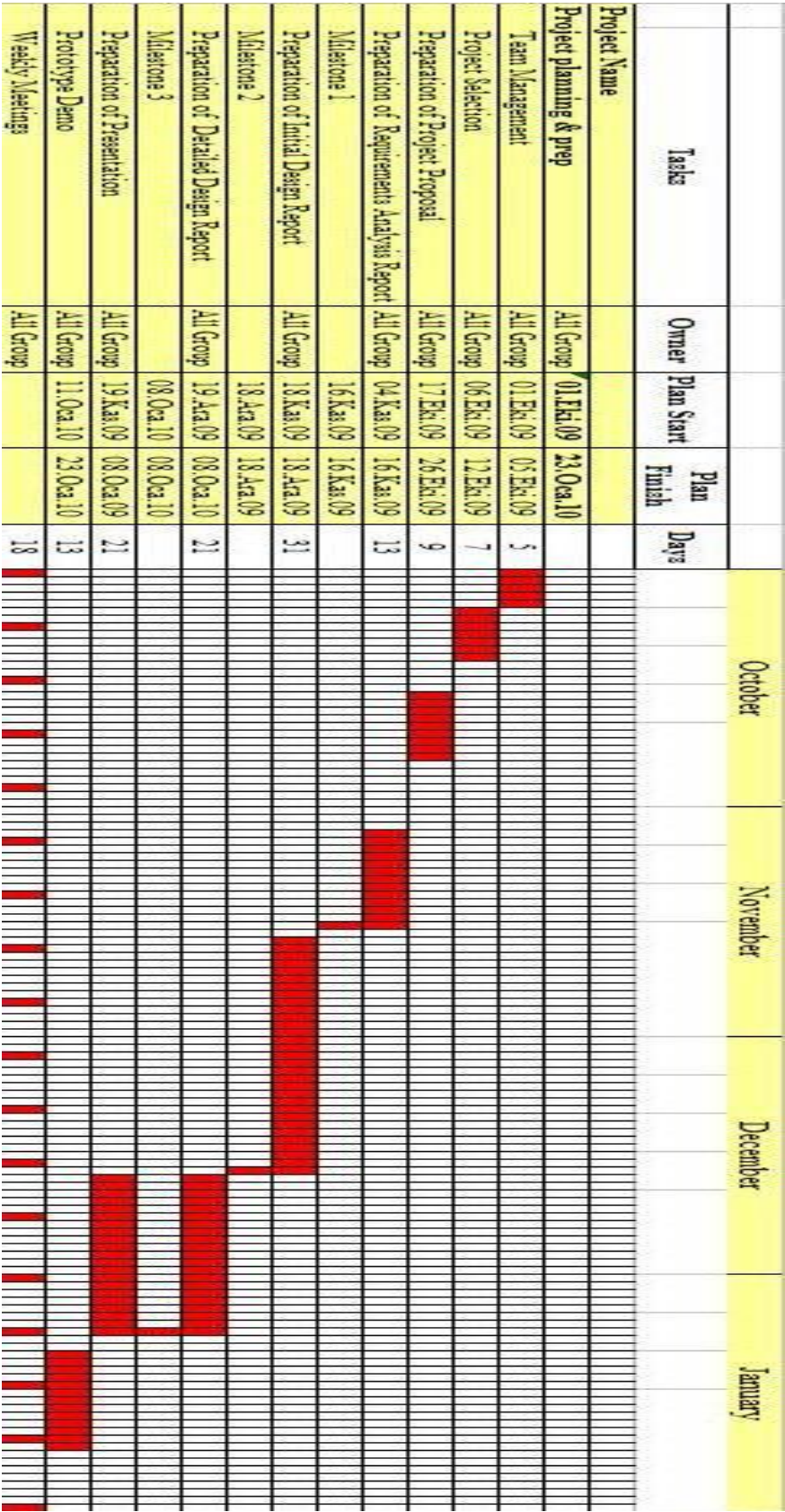


Figure.20-Gantt Chart