**INSE 6260**

**Mischell Flor**

**5428483**

**Tanvin Hossaine**

**Software Requirements Specification**

**Document**

**Version: 0.3** **Date: (02/07/2013)**

**Table of Contents**

1. Introduction 4

1.1 Purpose 4

1.2 Scope 4

1.3 Definitions, Acronyms, and Abbreviations. 4

1.4 References 5

1.5 Overview 5

2. The Overall Description 5

2.1 Product Perspective 5

2.1.1 System Interfaces 6

2.1.2 Interfaces 6

2.1.3 Hardware Interfaces 6

2.1.4 Software Interfaces 6

2.1.5 Communications Interfaces 6

2.1.6 Memory Constraints 7

2.1.7 Operations 7

2.1.8 Site Adaptation Requirements 7

2.2 Product Functions 7

2.3 User Characteristics 8

2.4 Constraints 9

2.5 Assumptions and Dependencies 9

2.6 Apportioning of Requirements. 9

3. Specific Requirements 9

3.1 External Interfaces 9

3.1.1 User Interfaces 9

3.1.2 Hardware Interfaces 10

3.1.3 Software Interfaces 10

3.1.4 Communications Interfaces 10

3.2 Functions 10

3.2.1 Administrator Component 10

3.2.4 Common Component 13

3.3 Performance Requirements 14

3.4 Logical Database Requirements 14

3.5 Design Constraints 19

3.5.1 Standards Compliance 19

3.6 Software System Attributes 20

3.6.1 Reliability 20

3.6.2 Availability 20

3.6.3 Security 20

3.6.4 Maintainability 21

3.6.5 Portability 21

3.7 Organizing the Specific Requirements 22

3.7.1 System Mode 22

3.7.2 User Class 23

3.7.3 Objects 23

3.7.4 Feature 23

3.7.4.1 Problem Domain 23

3.7.4.1 Main Domain Concepts 25

3.7.4.2 System Overview 25

3.7.4.3 User Requirements 26

3.7.5 Stimulus 37

3. 7.6 Response 37

3.7.7 Functional Hierarchy 37

3.8 Additional Comments 37

4. Change Management Process 38

5. Document Approvals 38

6. Supporting Information 38

# 1. Introduction

## 1.1 Purpose

The purpose of this document is to elicit the functional and non-functional requirements of eWomen online business. Functional requirements are directly related to the functionality of the User and eWomen web application. Functional requirements include structural, behavioral, and creational requirements. Non-functional requirements provide constraints, which are indirectly related to the functionality of the system and it includes the quality and organizational requirements of User and eWomen web Application

The intended audience of this document is users and customers, system developers, testers, and writer of user manuals. This document serves as a common ground for communication among stakeholders involved aspect of the software development lifecycle.

|  |  |
| --- | --- |
| **Group of readers** | **Reasons for reading** |
| Users and customers | To give feedback about the requirements |
| Developers | To understand what functions and properties that the system must contain |
| Testers | To test the system against the requirements |
| Writers of user manuals | To follow up the status of the project against the requirements |

Table 1: SRS Audience

## 1.2 Scope

eWomen aims to serve as an online selling and shopping. The scope of this online business is to create a web application for the seller (eWomen) and buyers. This web application will enable women to select and buy different products that suit their needs and lifestyles.

## 1.3 Definitions, Acronyms, and Abbreviations.

|  |  |
| --- | --- |
| Availability | An attribute that indicates how easily eWomen is made available to the users. This attribute also indicates if the service provided is uninterrupted and is readily available as expected. |
| Performance | A quantitative measure that relates to the execution of a particular aspect or function in the system under a particular workload |
| Reliability | A set of attributes that bear on the capability of eWomen to be trustworthy and up-to-date resource of information |
| Security | It is the condition of being protected against unauthorized disclosure of information, modification, or destruction of information related to the application |
| Maintainability | This attribute indicates the degree of efficiency and ease with which eWomen could be maintained in the future |
| Portability | This attribute indicates the level of independence of the system as independent system can be made to work successfully in a new environment without making any changes. |

Table 2: Definitions

## 1.4 References

* [IEEE, 1998] IEEE Standard 830-1998. Recommended Practice for Software Requirements Specification. IEEE Computer Society. 1998
* [Donn Le Vie, 2007] Writing Software Requirements Specifications. By Donn Le Vie

## 1.5 Overview

This document are divided into four sections. The first section introduces the purpose and scope of eWomen, an online business system. The next section covers the product perspective, function, user characteristics, constraints, assumptions, dependencies and apportioning of requirements. It is for customers and potential users. The third section is for developers and testers. It covers all software requirements in details. The last section covers the revision history and document approvals.

# 2. The Overall Description

## 2.1 Product Perspective

These days, many women are busy at work and with busy schedule, they no longer have time to go to shopping malls. In addition, the widespread use of web applications opens up an opportunity to do online business for sellers.eWomen System, a distributed and interactive software system, is a web application that sells various products for women who have different styles and fashions.

The perspective of eWomen with respect to how the software operates inside various constraints are dealt with in detail in section 3

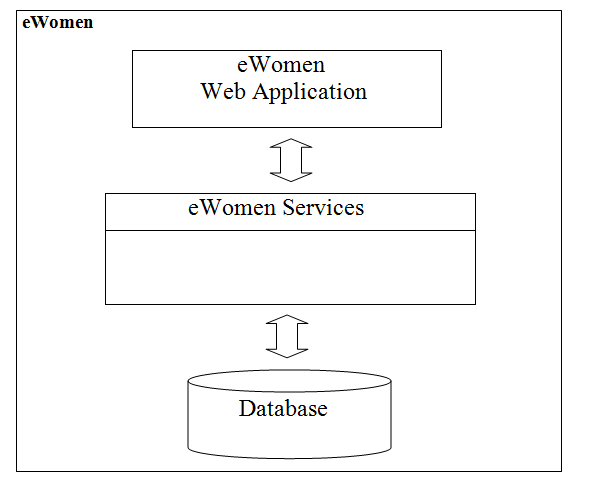


Figure 1: Conceptial Overview Diagram

As shown in figure 1, the core components eWomen are composed of three components such as Web application, Services and Database.

### 2.1.1 System Interfaces

*List each system interface and identify the functionality of the software to accomplish the system requirement and the interface description to match the system. These are external systems that you have to interact with. For instance, if you are building a business application that interfaces with the existing employee payroll system, what is the API to that system that designer’s will need to use?*

### 2.1.2 Interfaces

*Specify:*

1. *The logical characteristics of each interface between the software product and its users.*
2. *All the aspects of optimizing the interface with the person who must use the system*

*This is a description of how the system will interact with its users. Is there a GUI, a command line or some other type of interface? Are there special interface requirements? If you are designing for the general student population for instance, what is the impact of ADA (American with Disabilities Act) on your interface?*

The following shows a good overview of the user interface and how users will be able to interact with it. Sequences between dialogs and how to access them are clearly specified. Other things such as menus, icons, and all user messages are also detailed in this section

### 2.1.3 Hardware Interfaces

*Specify the logical characteristics of each interface between the software product and the hardware components of the system. This includes configuration characteristics. It also covers such matters as what devices are to be supported, how they are to be supported and protocols. This is not a description of hardware requirements in the sense that “This program must run on a Mac with 64M of RAM”. This section is for detailing the actual hardware devices your application will interact with and control. For instance, if you are controlling X10 type home devices, what is the interface to those devices? Designers should be able to look at this and know what hardware they need to worry about in the design. Many business type applications will have no hardware interfaces. If none, just state “The system has no hardware interface requirements” If you just delete sections that are not applicable, then readers do not know if: a. this does not apply or b. you forgot to include the section in the first place.*

### 2.1.4 Software Interfaces

### 2.1.5 Communications Interfaces

### 2.1.6 Memory Constraints

*Specify any applicable characteristics and limits on primary and secondary memory*. *Don’t just make up something here. If all the customer’s machines have only 128K of RAM, then your target design has got to come in under 128K so there is an actual requirement. You could also cite market research here for shrink-wrap type applications “Focus groups have determined that our target market has between 256-512M of RAM, therefore the design footprint should not exceed 256M.” If there are no memory constraints, so state.*

### 2.1.7 Operations

*Specify the normal and special operations required by the user such as:*

1. *The various modes of operations in the user organization*
2. *Periods of interactive operations and periods of unattended operations*
3. *Data processing support functions*
4. *Backup and recovery operations*

*(Note: This is sometimes specified as part of the User Interfaces section.) If you separate this from the UI stuff earlier, then cover business process type stuff that would impact the design. For instance, if the company brings all their systems down at midnight for data backup that might impact the design. These are all the work tasks that impact the design of an application, but which might not be located in software.*

### 2.1.8 Site Adaptation Requirements

*In this section:*

1. *Define the requirements for any data or initialization sequences that are specific to a given site, mission, or operational mode*
2. *Specify the site or mission-related features that should be modified to adapt the software to a particular installation*

*If any modifications to the customer’s work area would be required by your system, then document that here. For instance, “A 100Kw backup generator and 10000 BTU air conditioning system must be installed at the user site prior to software installation”.*

*This could also be software-specific like, “New data tables created for this system must be installed on the company’s existing DB server and populated prior to system activation.” Any equipment the customer would need to buy or any software setup that needs to be done so that your system will install and operate correctly should be documented here.*

## 2.2 Product Functions

The major functions of eWomen are categorizes as Administrator Component, Registered User Component, and Visitor Component. The functions under each category are as follows:

1. Administrator Component

1.1 Manage User

1.2 Manage Category

1.3 Manage Products

1.4 Monitor Feedback

2. Registered User Component

2.1 Authentication

2.2 Update User Account

2.3 Manage Shopping Cart

2.4 Online Payment

2.5 Track Product

2.6 Feedback

3. Visitor Component

3.1 Create User Account

4.Common Component

4.1 Browse

4.2 Search

4.3 Help

Each of the above mentioned functions are describe in detail in Section 3.2

## 2.3 User Characteristics

This section describes the intended users of the system

|  |  |  |
| --- | --- | --- |
| **User Groups** | **Description** | **Number of users** |
| Administrators | * User who is in charge in managing the users as well as maintaining the online business application. * Expert in Information Technology system | 1-2 person |
| Members | * Users who are registered in the system * Literate in Computer * Have an experience in e-payment system | N person |
| Visitors | * First-time users of the system | N person |

## 2.4 Constraints

* The use of SSL protocol to enhance security could be a possible constraint during implementation as some browsers may not support SSL.
* eWomen shall be made available 24x7. Hence the design and implementation of the system shall not be susceptible to failure or error frequently
* eWomen does not support cookies that automate user authentication
* Design shall be consistent and simple in order to ease maintenance

## 2.5 Assumptions and Dependencies

* The rendering of the web application depends on the availability and the version of the required software products such as browser configurations, Photoshot, etc.
* eWomen is available in the World Wide Web
* The product collection in eWomen depends on the changes/updates made by the administrator periodically

## 2.6 Apportioning of Requirements.

The potential features that could be built are numerous but the first version will focus on basic web features with a strong focus on selling and buying process. Once the product foundation is in place, the functionality footprint could evolve with future versions and include more advance possibilities in other stages of the product life cycle process.

# 3. Specific Requirements

## 3.1 External Interfaces

### 3.1.1 User Interfaces

* The web application shall be designed for 1027x768 screen resolutions as it is the most widely used screen size even among the target audience. Hence, the design shall be optimized and aim at perfection for its screen size although it shall also be rendered in 800x600 screen resolutions
* Liquid layouts shall be used so that the layout stretches to the current user's window size. Frozen layout shall be avoided
* The web application shall be rendered in the following Web Browsers: MS- Internet Explorer, Firefox, and Netscape
* The organization of contents, the location of links and the layout shall be consistent throughout the application
* The product logo shall be displayed throughout the application
* The choice of background and foreground colors shall be in contrast and in tune with the logo
* The content shall be evenly spaced with white spaces in between so that it increases readability
* The main page shall contain only the most essential details and does not appear cluttered
* The user interface format shall be user-friendly and easy to use
* All the navigational links shall be placed together at the top, bottom, right, or left consistently through the entire web application
* The system shall display error messages precisely and briefly within the current window as a dialog box

### 3.1.2 Hardware Interfaces

* There are no special hardware interface requirements

### 3.1.3 Software Interfaces

* The system shall use MySQL server as database management system. Communication with the database is through ODBC connections.
* Client Operating System: Microsoft Windows NT, 2000, XP, Vista, Linux, Mac OS x
* Web Server: Glassfish server

### 3.1.4 Communications Interfaces

* The system shall use an internet connection to the User and eWomen System
* The communication is effected by means of Hyper Text Transfer Protocol 1.1 (HTTP 1.1) to complete transactions. The web browser in which the web application is rendered shall comply with the standards of HTTP 1.1 or the latest versions
* eWomen shall also make use of protocols such as SSL for secure communication on the internet

## 3.2 Functions

This section defines the functionality that the system must provide in order to meet the business goals

### 3.2.1 Administrator Component

**3.2.1.1 Manage User**

**Purpose:** This feature shall allow an administrator to edit or delete a user from user lists

**Functional Requirements:**

* Edit User: This function shall enable an administrator to edit user account information
* Delete User: This function shall enable an administrator to delete user account

**3.2.1.2 Manage Category**

**Purpose:** This feature enables the administrator to maintain the product categories and keep them updated by adding new categories or by re-organizing the products into different categories

**Functional Requirements:**

* This feature shall provide add new category function. The new category name shall be not more than 40 characters and the feature shall accept only alphabetic characters as input
* The system shall also provide modify category option. This option enables the administrator to modify the category name and to move the products from/to the specific category
* The system shall provide an option to delete a category. Before deleting the category, the system should confirm if the constituent products shall be moved to another category or delete the entire category along with sub-categories and products.

**3.2.1.3 Manage Products**

**Purpose:** This feature shall allow an administrator to add, edit, or delete products

**Functional Requirements:**

* Add product: This function shall enable an administrator to add new products in the system
* Edit product: The system shall provide to an administrator the ability to modify an existing product
* Delete product: An administrator shall have the right to delete a product.
* View product inventory: An administrator shall have the right to view all products inventory such as new arrivals, existing and returned products and products that are ready for shipping and delivery.

**3.2.1.4 Monitor Feedback**

**Purpose:** This feature shall allow an administrator to monitor the feedback from customers

**Functional Requirements:**

* The system shall provide this function in the administrator page.
* The administrator shall be able to view the feedbacks associated with the product along with the control buttons to delete, modify or move them.
* The administrator shall be able to perform operations such as delete irrelevant feedback and delete old feedbacks if the feedback list is too long and cumbersome
* The administrator shall be able to post a feedback in response to an existing feedback

**3.2.2 Registered User Component**

**3.2.2.1 Authentication**

**Purpose:** This feature shall authenticate the identity of the user before providing access to the User Management or online payment

**Functional Requirements:**

* The users with an account shall be authenticated before accessing their accounts
* The system shall provide a form with Username, password fields and logon button
* The authentication feature of the system shall provide a hyperlink for Forgotten password
* The system shall provide an error message if either the username or password is incorrect and prompt the user to try again, displaying the login form once again.

**3.2.2.2 Update User Account**

**Purpose:** This feature shall allow a user to manage its own account

**Functional Requirements:**

* Edit Account: This function shall enable an existing member to edit its account
* Delete Account: This function shall enable an existing member to delete its account

**3.2.2.3 Manage Shopping Cart**

**Purpose:** This feature shall allow a user to add, modify and edit its e-cart

**Functional Requirements:**

* The system shall enable a user to add one or multiple items to its e-cart
* User shall be allowed to modify the quantity of e-cart items
* The system shall provide a user the ability to delete items in e-cart
* The system shall provide a link to pay the items in e-cart

**3.2.2.4 Online Payment**

**Purpose:** This feature shall allow a user to perform an online payment

**Functional Requirements:**

* The system shall provide to user an option to pay by credit or debit card
* The system shall provide the total price of the product before and after taxes
* The system shall provide an insurance for shipping and delivery
* The system shall provide to user an option to select a delivery carrier
* The system shall verify the payment
* The system shall provide a tracking code and a receipt after the payment transaction
* The system shall provide an error message for invalid entries in e-payment form
* The system shall provide an agreement form to a user before the payment transaction
* The system shall ensure that e-payment is secured
* User shall be a member in order to do e-payment

**3.2.2.5 Track Product**

**Purpose:** This feature shall allow a user to track his\her products shipping and delivery status

**Functional Requirements:**

* The system shall provide a form with tracking code field and submit button
* The current product's status shall be visible to user

**3.2.2.5 Feedback**

**Purpose:** This feature enables the user to post feedbacks

**Functional Requirements:**

* This feature shall display a form that records the personal details of users such as name, email and the feedback
* The feedback is entered in text
* The feedback entered by the user shall be posted at the end of the related product along with the date, name, email id and the comment

**3.2.3 Visitor Component**

**3.2.3.1 Create User Account**

**Purpose:** This feature shall allow a user to manage its own account

**Functional Requirements:**

* Create an Account: This function shall enable a non-member to create an account
* Edit Account: This function shall enable an existing member to edit its account
* Delete Account: This function shall enable an existing member to delete its account

### 3.2.4 Common Component

**3.2.4.1 Browse**

**Purpose:** This feature provides the ability to navigate through product items in eWomen grouped by category

**Functional Requirements:**

* The system shall provide the options to view the products grouped by category, by price, by brand, and an overall product list. This enable the user to navigate through the product catalog easily
* The system shall provide a list of products grouped by category when the option "Categories" is selected by a user
* The system shall display an error message when any of the options do not work properly (the page name changed and not updated or due to a problem with the user's internet browser) and the system shall redirect to the main page

**3.2.4.2 Search**

**Purpose:** This feature allows user to search a product using keywords or query strings within the product collection

**Functional Requirements:**

* The system shall provide a form with the search field for a basic search. The system shall also provide an advanced search function
* The advanced search function shall also provide options to search a specific product by entering the category name and price range, size type and product name. This is an additional fields to the keyword or query string. The search engine shall narrow the scope to find a product.
* The system shall accept a keyword or query string as an input on the search field. The keyword must be up to 20 characters. The query string must be up to 100 characters. The keyword and query string must be written in English.
* The results shall be displayed as hyper links in decreasing order of importance. The hyperlink shall connect to the particular page
* The system shall display a message "no match found" when the product does not exist in the product collection.

**3.2.4.3 Help**

**Purpose:** This feature allows user to view help topics and obtain help if required. Help topics guide users on how to use a feature available in the system

**Functional Requirements:**

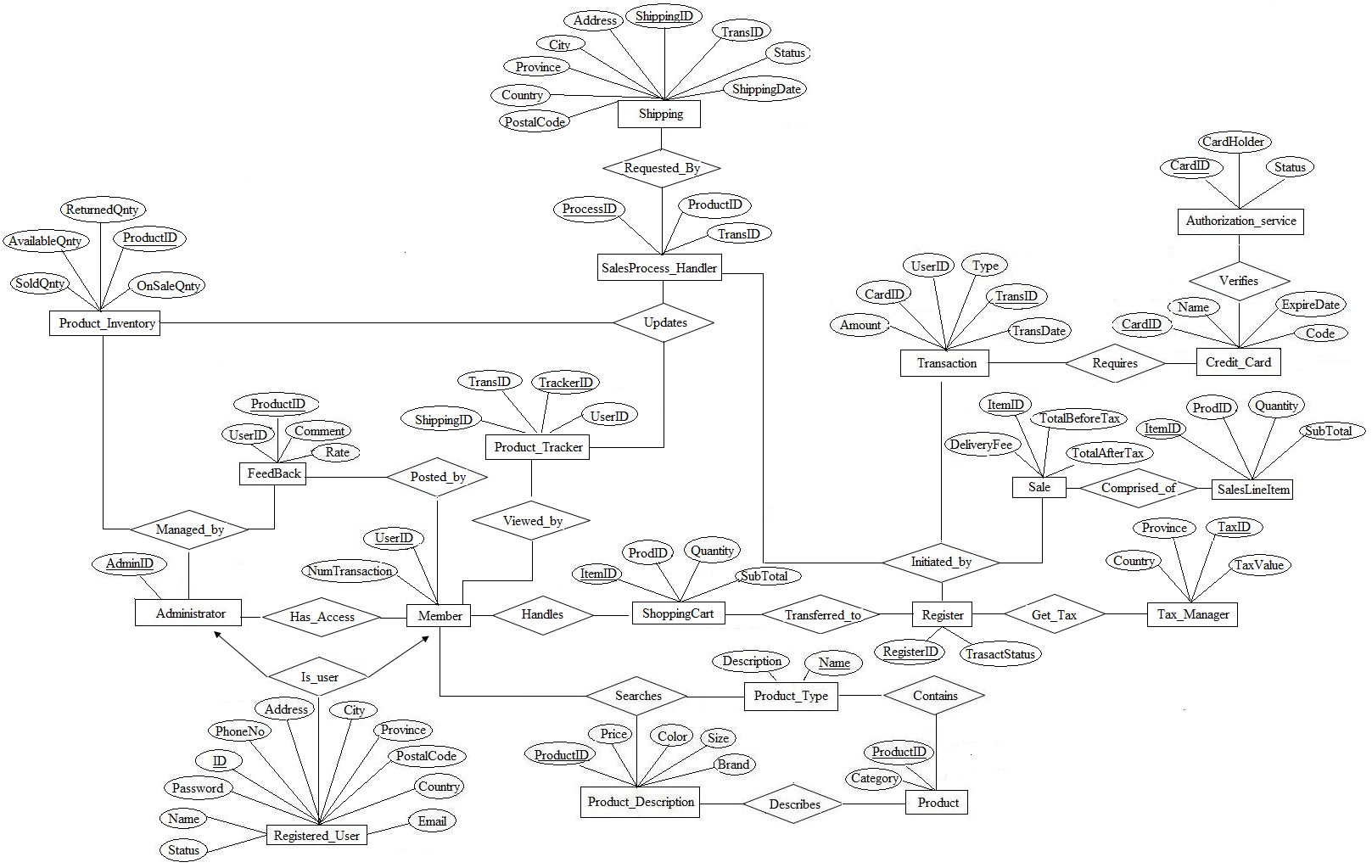
* The system shall provide a help button in every page in eWomen. This help button shall open a new window with a help list grouped by topic
* The system shall also display the content of a topic selected by a user
* The system shall provide a search tab to find the required help topic faster
* The system shall accept a keyword or query string as an input in the search help field. The keyword must be up to 10 characters and the query string must be up to 20 characters.
* The system shall provide context-sensitive help for each feature available in eWomen

## 3.3 Performance Requirements

* Depending on the server capacity, the database and the complexity of the web application, the speed of the internet connectivity, eWomen shall support a minimum of 50 users simultaneously.
* The type of information handled in eWomen web application shall predominantly be textual and images within the size of 1MB (each) shall be handled
* Although subject to variation depending on the speed of the internet connection, 95% of the transactions of a client shall be processed within 5s.

## 3.4 Logical Database Requirements

The backend database of eWomen is designed on relational model where data is organized in relations (tables). The E-R diagram for the system is shown below



The schema of eWomen is given below in alphabetical order.

|  |  |
| --- | --- |
| Schema | Description |
| *Administrator(AdminID)* | *Administrator* has full access to eWomen system. The primary key is *AdminID* |
| *Authorization\_service( CardID ,CardHolder, Status)* | *Authorization\_Service* authorizes the credit payment. The primary key is *CardID* |
| *ComprisedOf(Sale.ItemID,SalesLineItem.ItemID)* | *ComprisedOf* defines many to many relationship between Sales and SalesLineItem. *ItemID* is a foreign key from *Sale* and *SalesLineItem* |
| *Contains(ProductID, Name)* | *Contains* defines many to many relationship between *Product\_Type* and *Product*. *ProductID* is a foreign key from of *Product* and *Name* is a foreign key from *Product\_Type* |
| *Credit\_Card(CardID, Name, ExpireDate, Code)* | *Credit\_Card* stores the information for credit card. The attribute *code* is the three digits number located at the back of the card. The The primary key is *CardID* |
| *Feedback(UserID, ProductID, Comment, Rate)* | The *Feedback* stores the information for user's feedback that purchased the products. The attribute *Rate* indicates user's satisfactory level from the purchased product. *Comment* attribute is a user's statement on the product. The primary key is *ProductID* |
| *Get\_Tax(RegisterID, TaxID)* | *Get\_Tax* defines many to many relationship between *Register* and *Tax\_Manager*. |
| *Handles(UserID, ItemID)* | *Handles* defines many to many relationship between Member and *Shopping Cart*. |
| *HasAccess(AdminID, UserID)* | *HasAccess* defines many to many relationship between *Administrator* and *Member*. *AdminID* is a foreign key from *Administrator* and *UserID* is a foreign key from *Member*. |
| *Is\_User(AdminID,UserID)* | *Is\_user* defines many to one relationship. A registered user is either an Administrator or a member. *AdminID* is a foreign key from *Administrator* and *UserID* is a foreign key from *Member* |
| *Member(UserID,NumTransaction)* | The *Member* stores the information for eWomen clients. The primary key is *UserID* |
| *Posted\_by(ProductID, UserID)* | The *Posted\_by* defines many to many relationships between *Member* and *Feedback*. This implies that users are not limited to post their feedback from the purchased product. *UserID* is a foreign key from *Member* and *ProductID* is a foreign key from *Feedback* |
| *Product(ProductID, Category)* | The *Product* stores the information for eWomen products. The primary key is *ProductID* |
| *Product\_Description( ProductID,Price, Color, Size, Brand )* | *Product\_Description* stores the information for Products description. The primary key is *ProductID* |
| *Product\_Inventory( AvailableQnty, ReturnedQnty, ProductID, SoldQnty, OnSaleQnty)* | *Product\_Inventory* stores the products stocks. The primary key is *ProductID* |
| *Product\_Tracker(ShippingID, TransID, TrackerID, UserID)* | *Product\_Tracker* stores the information for product tracking. The primary key is *TrackerID* |
| *Product\_Type(Description, Name)* | *Product\_Type* stores the information for product types. The primary key is *Name* |
| *Register(RegisterID, Transtatus)* | Register scans each product order . The primary key is *RegisterID* |
| *Registered\_User(ID,Name,Password, Status, Phone, Address,City,Province, PostalCode,County, Email)* | The *Registered\_User* contains the user account information for eWomen System. Each user has a unique id to login. The status indicates if a user is active or inactive. When a user forgets its password, an email will be used for password retrieval. The primary key is *ID* |
| *Sale(ItemID,DeliveryFee, TotalBeforeTax, TotalAfterTax)* | The *Sale* shows the product costs before and after tax and delivery fee |
| *SalesLineItem(ItemID, ProdID, Quantity, SubTotal)* | *SalesLineItem* stores the information for Sales Line Items. These are the products that are going to be purchased by Member. The primary key is *ItemID* |
| *SaleProcess\_Handler(ProcessID, ProductID, TransID )* | The *SaleProcess\_Handler* stores the information for sales processing. The primary key is *ProcessID* |
| *Searches(UserID,Name,ProductID)* | *Searches* defines many to many relationship between *Member* and *Product\_Type\Product\_Description. UserID* is a foreign key from *Member*, *ProductID* is a foreign key from *Product\_Description* and *Name* is a foreign key from *Product\_Type* |
| *Shipping(ShippingID, Address, City, Province, Country, PostalCode. ShippingDate, Status, TransID)* | *Shipping* stores the information for Delivery. The primary key is *ShippingID* |
| *ShoppingCart(ItemID, ProdID, Quantity, SubTotal )* | *ShoppingCart* stores the product orders. The primary key is *ItemID* |
| *Tax\_Manager(Country, Province, TaxID, TaxValue)* | *Tax\_Manager* stores the information for Taxes. The primary key is *TaxID* |
| *Transaction(TransID, CardID, UserID, Type, Amount, TransDate)* | *Transaction* stores the information for sales transaction. The primary key is *TransID* |
| *Transferred\_to(ItemID, RegisterID)* | *Transferred\_to* defines one-to-many relationship. Each member has only one instance of shopping cart which can be moved to Register. |
| *Updates(ProductID, ProcessID, TrackerID)* | *Updates* defines many to many relationship between *SaleProcess\_Handler* and *Product\_Inventory* / *Product\_Tracker.* |
| *Viewed\_by(TrackerID, UserID)* | *Viewed\_by* defines many to many relationship between *Member* and *Product\_Tracker*. *UserID* is foreign key from *Product\_Tracker* and *UserID* is foreign key from *Member* |
| *Verifies(Authorization\_service.CardID,Credit\_Card.CardID)* | *Verifies* defines many to many relationship between *Authorization\_Service* and *Credit\_Card*. *CardID* is a foreign key from *Authorization\_Service* and from *Credit\_Card* |

## 3.5 Design Constraints

### 3.5.1 Standards Compliance

## 3.6 Software System Attributes

### 3.6.1 Reliability

* The content of the website shall mention the product's description and price so that it proves to be reliable to all users. The content shall also be precise, accurate, and current.
* The database and other data handled shall have sufficient backup. The backups for database shall be created weekly and stored in different locations.
* The administrator shall be aware of the necessary backup and recovery procedures in case of any system failure
* The security features such as encryption of passwords and use of SSL communication protocol shall make the application more reliable in terms of privacy of information and security

### 3.6.2 Availability

* The system shall be made compatible with different browsers such as Internet Explorer, Firefox, Safari, and Netscape
* The url of the eWomen shall be included in different search engines such as Google, Yahoo, MSN, and others
* In case of server failure or database crash or any other system failures, the system shall be recovered without delay and no interruption in service due to the regular backup methods and the administrator's ability to effect immediate recovery of the system
* The system shall be kept operational and made available to users even during a possible re-design process

### 3.6.3 Security

* eWomen shall allow only registered users (Administrators and members) to make online payment and place
* Members and Administrators are authenticated using a username and password
* The passwords shall be encrypted and the encrypted password shall be stored in the database
* The system shall prompt account holders to change passwords periodically
* The system shall make use of communication protocols such as SSL for transferring sensitive information such as username, password and personal information between the client and the server
* Only the administrator shall have full access control over eWomen web application
* The system shall have sufficient protection against viruses, Trojan horses, worms, etc.

### 3.6.4 Maintainability

* The system shall use the consistent structure and layout of information throughout the application to enable ease of maintenance in the future
* All related software artifacts shall be well documented and the source code shall have well documented and appropriate comments
* The product items shall be updated regularly
* The user's feedback shall be tracked carefully as they could be useful to the improvement of the application

### 3.6.5 Portability

* The system shall be portable as it shall be compatible for use in various platforms such as different operating systems such as Windows, Mac, and Linux and it is also rendered in different browsers as mentioned above and it does not depend on any special software or system interfaces for successful execution
* The generalized abstraction of the application logic of eWomen shall make it reusable portable when moving from one environment to another

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Characteristic** | **H/M/L** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** |
| 1 | Correctness |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 2 | Efficiency |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 3 | Flexibility |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 4 | Integrity/Security |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 5 | Interoperability |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 6 | Maintainability |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 7 | Portability |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 8 | Reliability |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 9 | Reusability |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 10 | Testability |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 11 | Usability |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 12 | Availability |  |  |  |  |  |  |  |  |  |  |  |  |  |

*Definitions of the quality characteristics not defined in the paragraphs above follow.*

*• Correctness - extent to which program satisfies specifications, fulfills user’s mission objectives*

*• Efficiency - amount of computing resources and code required to perform function*

*• Flexibility - effort needed to modify operational program*

*• Interoperability - effort needed to couple one system with another*

*• Reliability - extent to which program performs with required precision*

*• Reusability - extent to which it can be reused in another application*

*• Testability - effort needed to test to ensure performs as intended*

*• Usability - effort required to learn, operate, prepare input, and interpret output*

## 3.7 Organizing the Specific Requirements

### 3.7.1 System Mode

*Some systems behave quite differently depending on the mode of operation. When organizing by mode there are two possible outlines. The choice depends on whether interfaces and performance are dependent on mode.*

### 3.7.2 User Class

*Some systems provide different sets of functions to different classes of users.*

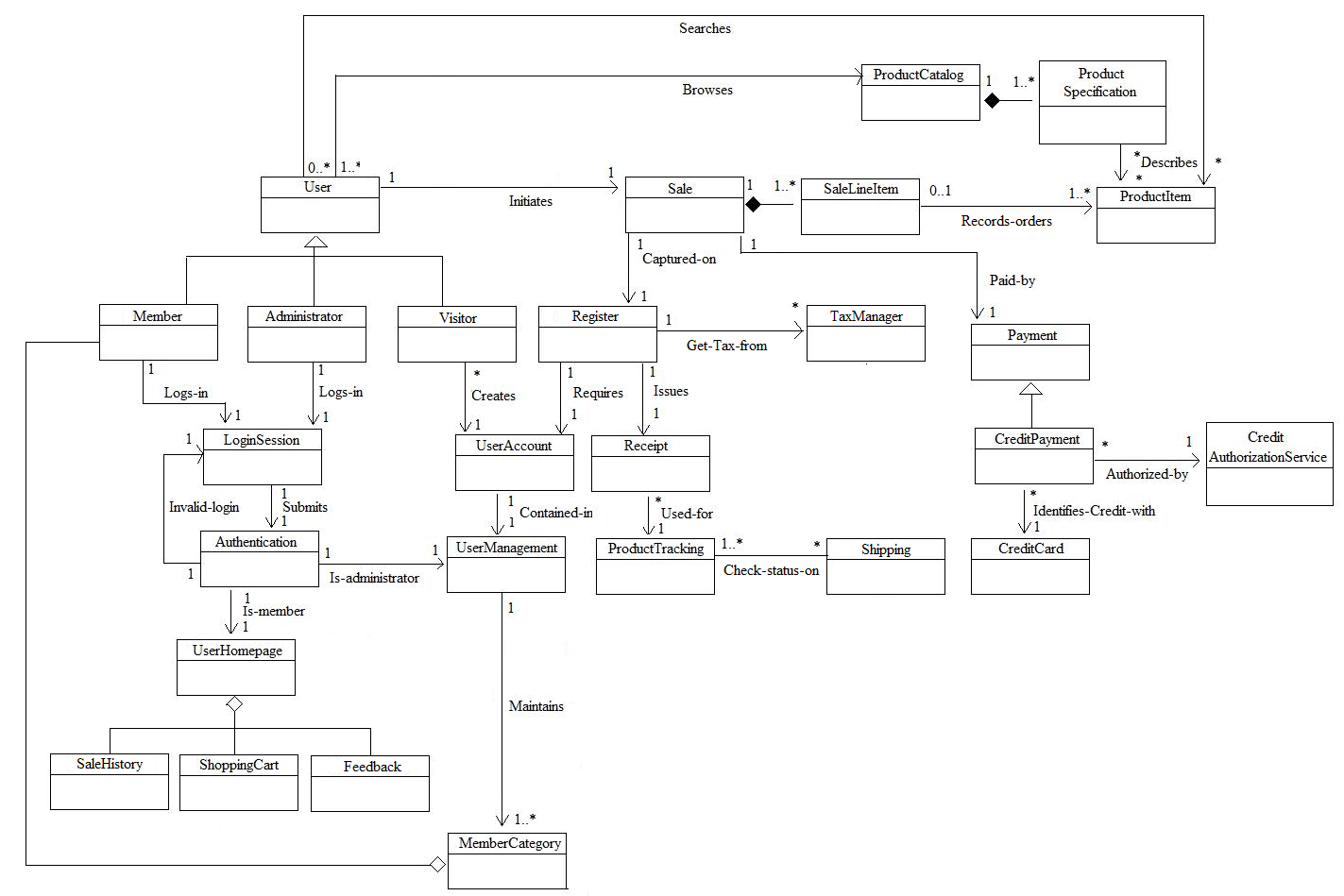
### 3.7.3 Objects

*Objects are real-world entities that have a counterpart within the system. Associated with each object is a set of attributes and functions. These functions are also called services, methods, or processes. Note that sets of objects may share attributes and services. These are grouped together as classes.*

### 3.7.4 Feature

### 3.7.4.1 Problem Domain

This section shows the problem domain. It includes textual definition of the most important domain concepts and a conceptual data model defining the relationship between the most important domain concepts. Domain model is shown below



The description of main domain concepts is given below in alphabetical order

|  |  |
| --- | --- |
| Concept | Description |
| Administrator | It manages the whole system |
| Authentication | It validates userID and password |
| CreditAuthorizationService | It is a third party system that authorizes credit payment |
| CreditCard | It contains credit card information |
| CreditPayment | It is used to pay the products |
| Feedback | It enables a Member to send a feedback of the purchased product. It is used to identify user's satisfaction on the products |
| LoginSession | It enables a user to login to the system |
| Member | It is a registered user |
| MemberCategory | It contains all members userID |
| Payment | It is used to pay the products |
| ProductCatalog | It contains all product types |
| ProductItem | It is the product |
| ProductSpecification | It contains products description |
| ProductTracking | It tracks the product for shipping purposes |
| Register | It handles all the transactions |
| Receipt | It is a proof of purchase |
| Sale | It contains sales line items |
| SaleLineItem | It contains the product lists for purchase |
| SaleHistory | It contains all the products purchased by userID |
| Shipping | It contains the information for shipping |
| ShoppingCart | It contains the product lists for orders |
| TaxManager | It contains the tax information for specific country and province |
| UserAccount | It contains the user profile |
| UserHomepage | It is a personalized page for Administrator and Member |
| UserManagement | It contains all the registered users |
| Visitor | It is a non-registered user |

### 3.7.4.2 System Overview

This section is a high level description of the intended solution. It includes a list of essential features of the system and use case diagram that defines the users and the main functions of the system.

### 

### 3.7.4.3 User Requirements

Use cases are strongly recommended to be used in documenting user requirements. The following sections are typically included in the use case descriptions:

* Name: Unique name of the use case
* Summary: Short description of the use case
* Actors: List of users and the other systems that interacts directly with a system
* Pre-conditions: Description of start situation and goals of the users
* Basic sequence: Description of special situations, violations of business rules
* Exceptions: Description of special situation, violation of business rules
* Post conditions: Description of end situation
* Priority
* Status
* Traces to requirements
* Traces to test cases

Unique Identifier: **UC-001**

|  |  |
| --- | --- |
| Name | Manage Category |
| Summary | This use case enables an Administrator to create, modify, or delete product category |
| Actors | Administrator |
| Preconditions | Administrator is signed into his\her account |
| Basic Sequence | 1. Use case begins when Administrator click Manage Category from the homepage main menu  2. The system returns a menu with the options to Add Category, Edit Category, and Delete Category  3A. Administrator selects Add Category  3A.1 The system returns a form to create a category  3A.2 Administrator enters the name and description and click Save  3A.3 The use case ends when Administrator clicks Save  3B. Administrator selects Edit Category  3B.1 The system returns the category lists  3B.2 Administrator selects a category and clicks Edit  3B.3 The system returns an editable form and the name and description for the with name and description  3B.4. Administrator modifies the name and\or description and click Save  3B.5 The use case ends when Administrator clicks Save  3C. Administrator selects Delete Category  3C.1 The system returns the category lists  3C.2 Administrator selects a category and clicks Delete  3C.3 The system outputs a confirmation dialog  3C.4 Administrator confirms the deletion  3C.5 The use case ends when Administrator confirms the deletion |
| Exception | Exception A: The Category name is not unique  If category name is not unique, then the system will output a message that the category already exists.  Exception B: Administrator clicks Cancel from Delete Confirmation  If Administrator selects Cancel from Delete Confirmation dialog, then the system will output a message that the deletion is cancelled |
| Postconditions | The system generates a message that the category task is successfully added, edited, or deleted |
| Priority | Must |
| Status | Not started |
| Traces to requirements |  |
| Traces to test cases |  |

Unique Identifier: **UC-002**

|  |  |
| --- | --- |
| Name | Monitor Feedback |
| Summary | This use case enables the Administrator to validate the content of a feedback |
| Actors | Administrator |
| Preconditions | A Member posted a new feedback from the purchased product and  Administrator receives an email for new product feedback |
| Basic Sequence | 1. Use case begins when Administrator clicks Manage Feedback from its homepage main menu  2. The system returns the feedback lists  3. Administrator selects a new feedback and clicks Open  4. The system displays the feedback information  5. Administrator validates the feedback content and clicks Passed  6. The use case ends when the Administrator clicks Passed |
| Exception | Exception A: Administrator clicks Failed  If Administrator clicks Failed, then the system will inform the member who posted the feedback that it did not passed from the verification |
| Postconditions | The system generates a message that the feedback is passed from verification |
| Priority | Must |
| Status | Not started |
| Traces to requirements |  |
| Traces to test cases |  |

Unique Identifier: **UC-003**

|  |  |
| --- | --- |
| Name | Update User Account |
| Summary | This use case enables the Administrator or member to update user accounts |
| Actors | Administrator, Member |
| Preconditions | The Administrator or member is signed in to the system |
| Basic Sequence | 1. Use case begins when the Administrator or Member click Update Profile from homepage main menu  2. The system returns an editable form of user's profile  3. Administrator or member updates its profile and click Save  4. Use case ends when Administrator or member clicks Save |
| Exception | Exception A: User clicks Cancel  If a user clicks cancel, then the system will not save the changes made in the profile. |
| Postconditions | The system generates a message that the updated profile is successfully saved. |
| Priority | Must |
| Status | Not started |
| Traces to requirements |  |
| Traces to test cases |  |

Unique Identifier: **UC-004**

|  |  |
| --- | --- |
| Name | Get Authentication |
| Summary | This use case enables administrator or member to access his\her account based on the verification of the userid and password entries |
| Actors | Administrator, Member |
| Preconditions | 1. An administrator has a full access account  2. A member has an account in eWomen system |
| Basic Sequence | 1. Use case beigns when the administrator or member goes to the Login page  2. The system returns login form with entries for userID and password  3. The administrator or member enters its userID and password and click the LogIn button  4. The system verifies the userID and password  5. The system activates the administrator or the member's profile  6. Use case ends when the administrator or member access his\her account |
| Exception | Authentication failed:  Condition:  A1: An administrator or member enters an invalid userID or password  A2: The system displays a message indicating that the values entered are incorrect and asks the administrator or member to re-enter the userID and password. |
| Postconditions | If unable to login, user cannot access its account and Shopping Cart and e-payment are disabled.  If login is successful, administrator or member will be able to access his\her account |
| Priority | Must |
| Status | Not started |
| Traces to requirements |  |
| Traces to test cases |  |

Unique Identifier: **UC-005**

|  |  |
| --- | --- |
| Name | Manage ShoppingCart |
| Summary | This use case enables a member to update the product items that are placed in the shopping cart |
| Actors | Member |
| Preconditions | The shopping cart contains at least one product item |
| Basic Sequence | 1. Use case begins when a Member selects the shopping cart from the homepage main menu  2. The system displays the list of product items  3A. Member selects a product item and clicks Edit  3A.1. The system returns an editable form for product item  3A.2. The member modifies the number of quantity of the product that he\she wishes to buy and clicks Save  3A.3. The use case ends when the Member clicks Save  3B. Member selects a product item and clicks Delete  3B.1. The system displays a confirmation dialog  3B.2. The Member clicks Yes  3B.3 The use case ends when a user clicks Yes |
| Exception | Exception A: The Member clicks Cancel from product item updates  If a Member clicks Cancel, then the system will output a message that the product update is cancelled  Exception B: The Member clicks No from Delete Confirmation  If a Member clicks No, then the system will output a message that the deletion is cancelled |
| Postconditions | The system generates a message that the changes is successfully saved |
| Priority | Must |
| Status | Not started |
| Traces to requirements |  |
| Traces to test cases |  |

Unique Identifier: **UC-006**

|  |  |
| --- | --- |
| Name | Online Payment |
| Summary | This use case enables a Member to purchase a product |
| Actors | Member, TaxManager, PaymentAuthorizationService |
| Preconditions | The shopping cart is not empty and the Member is signed into the system |
| Basic Sequence | 1. Use case begins when the Member clicks Submit from the shopping cart page  2. The system displays Review Orders which outputs the sales line items.  3. The Member reviews the sales line items and click Next  4. The system gets the applicable taxes from Tax Manager and computes the total  5. The system displays the Confirm Orders page  6. The user validates the total amount of the product before and after tax and then clicks Next  7. The system returns a form for credit card information.  The Member filled up all the fields and click Next  8. Payment Authorization Service authorizes the credit card payment  9. The system displays the receipt id and the transaction date  10. The use case ends when the system generates a receipt |
| Exception | Exception A: Member entered an incorrect expire date in the credit card information  If the expire date is incorrect, then the system will output a message that the expire date is invalid  Exception B: Payment is not accepted  If Payment Authorization did not authorize the credit payment, then the system will output a message that the payment is failed. |
| Postconditions | The system started the shipping process |
| Priority | Must |
| Status | Not started |
| Traces to requirements |  |
| Traces to test cases |  |

Unique Identifier: **UC-007**

|  |  |
| --- | --- |
| Name | Post Feedback |
| Summary | This use case enables the |
| Actors | Member |
| Preconditions | Member has recently purchased a product |
| Basic Sequence | 1. Use case begins when the Member clicks Sale History from the homepage main menu  2. The system returns a list of purchased products  3. User selects a produt item and click Post Feedback  4. The system returns a form with comment and rate fields  5. The member enters its comment and rate and clicks Send  6. The use case ends when a Member clicks Send |
| Exception | Exception A: User clicks Cancel  If a user clicks Cancel, then the system will output a message that sending feedback is cancelled. |
| Postconditions | The system generates a message that the feedback is succesfully sent |
| Priority | Must |
| Status | Not started |
| Traces to requirements |  |
| Traces to test cases |  |

Unique Identifier: **UC-008**

|  |  |
| --- | --- |
| Name | Track Product |
| Summary | This use case enables the Member to track the shipping status of recently purchased product |
| Actors | Member |
| Preconditions | Member has a receipt from the purchased product |
| Basic Sequence | 1. Use case begins when the Member selects Track Product from the homepage main menu  2. The system returns a form with a field for receipt id  3. The member enters the receipt id and click Track  4. The use case ends when a user clicks Track |
| Exception | Exception A: User entered an incorrect receipt id  If a user enters an incorrect receipt id then the system will output a message that nothing is found in the system |
| Postconditions | The system displays the shipping status of the product |
| Priority | Must |
| Status | Not started |
| Traces to requirements |  |
| Traces to test cases |  |

Unique Identifier: **UC-009**

|  |  |
| --- | --- |
| Name | Browse Product Type |
| Summary | This use case enables the Member or Visitor to browse the product types in the system |
| Actors | Member, Visitor |
| Preconditions | Member or Visitor has internet connection |
| Basic Sequence | 1. Use case begins when the Member or Visitor points the mouse cursor to browse  2. The system displays different product types  3. The Member or Visitor clicks a product type  4. The system displays the categories in the product type 5. The Member or Visitor selects a category  6. The System displays the products in the category  7. The Member or Visitor clicks a product  8. The system displays the description description  9. The use case ends when a system displays the product description |
| Exception | Exception A: User clicks a product that belongs to another product\_type  If a Member or Visitor clicks a product that belongs from another product type, then the system will close the current product view and open the selected product |
| Postconditions | The system displays the product |
| Priority | Must |
| Status | Not started |
| Traces to requirements |  |
| Traces to test cases |  |

Unique Identifier: **UC-010**

|  |  |
| --- | --- |
| Name | Search Product |
| Summary | This use case enables the Member or Visitor to search a product |
| Actors | Member, Visitor |
| Preconditions | The Member or Visitor has internet connection |
| Basic Sequence | 1. Use case begins when the Member or Visitor types a keyword in the search field and clicks Search  2. The system displays the searched the results that matches the keyword  3. The use case ends when the system outputs the search result |
| Exception | Exception A: The user keyword is mispelled  If a keyword has incorrect word, then the system will output a message that nothing is found |
| Postconditions | The system displays the number of hit results |
| Priority | Must |
| Status | Not started |
| Traces to requirements |  |
| Traces to test cases |  |

Unique Identifier: **UC-011**

|  |  |
| --- | --- |
| Name | Obtain Help |
| Summary | This use case enables the Member or Visitor to get help from the system |
| Actors | Member, Visitor |
| Preconditions | The Member or Visitor has internet connection |
| Basic Sequence | 1. Use case begins when the Member or Visitor clicks the help icon  2. The system retrieves the help for that page and field.  3. The use case ends when the system displays the help information |
| Exception | Exception A: The Member or Visitor clicks an inactive hyperlink  If a Member or Visitor clicks an inactive link, then the system will output that the page is not found |
| Postconditions | The system displays the online help |
| Priority | Must |
| Status | Not started |
| Traces to requirements |  |
| Traces to test cases |  |

Unique Identifier: **UC-012**

|  |  |
| --- | --- |
| Name | Create Account |
| Summary | This use case enables a Visitor to create a user account |
| Actors | Visitor |
| Preconditions | The Visitor does not have an existing account in the system |
| Basic Sequence | 1. Use case begins when the Visitor clicks Create Account in the the system  2. The system outputs a form for creating user account  3. Visitor filled up all the fields and clicks Join  4. Use case ends when a Visitor clicks Join |
| Exception | Exception A: The Visitor clicks Cancel  If a visitor clicks Cancel, then the system will output a message that create account is cancelled  Exception B: The Visitor enters a userID that already exists in the system  If a Visitor entered a userID that already exists in the system, then the system will output an error message that the userID is not unique. |
| Postconditions | The system generates an output message that the account is successfully created |
| Priority | Must |
| Status | Not started |
| Traces to requirements |  |
| Traces to test cases |  |

### 3.7.5 Stimulus

*Some systems can be best organized by describing their functions in terms of stimuli.*

### 3. 7.6 Response

This section shows UML Sequence Diagrams for eWomen system which illustrates the events from actors and possible intersystem events.

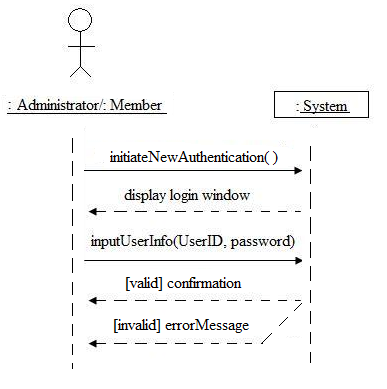


Figure: Sequence Diagram for Authentication

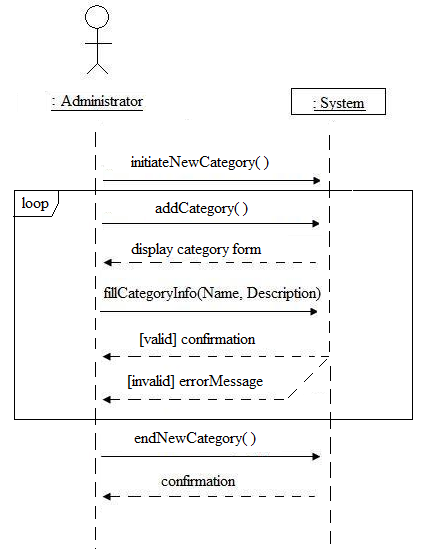


Figure: Sequence Diagram for Create Category

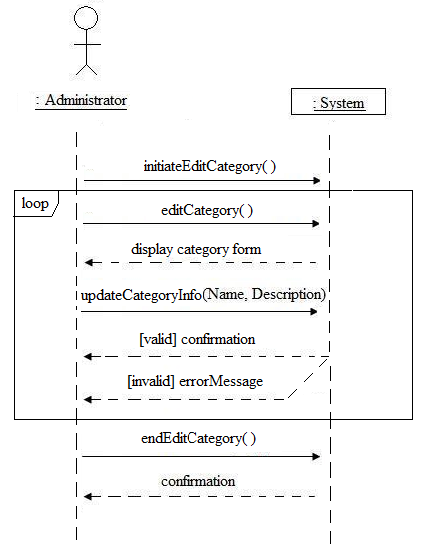


Figure: Sequence Diagram for Edit Category

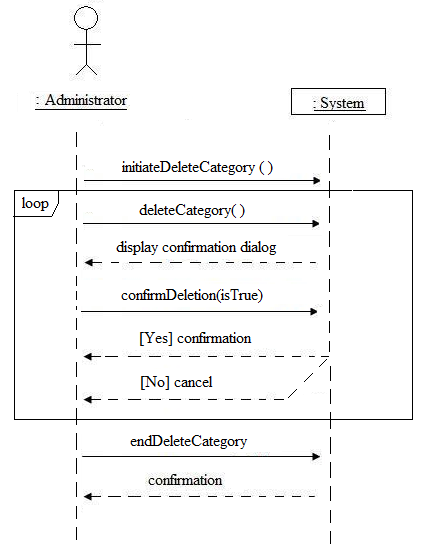


Figure: Sequence Diagram for Delete Category

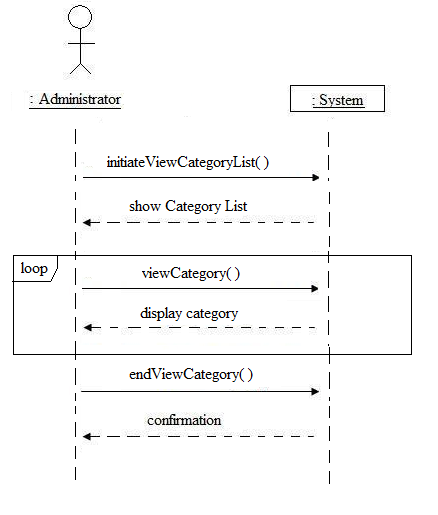


Figure: Sequence Diagram for View Category

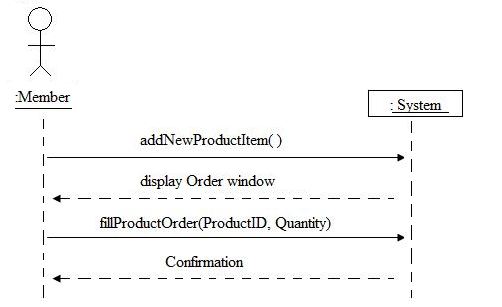
****

Figure: Sequence Diagram to add an order to ShoppingCart

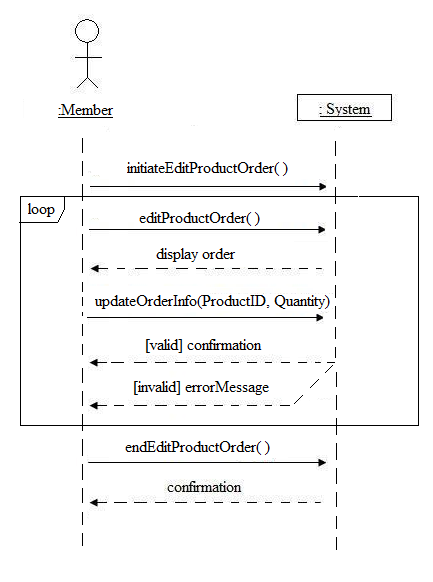
****

Figure: Sequence Diagram for Edit Order from ShoppingCart

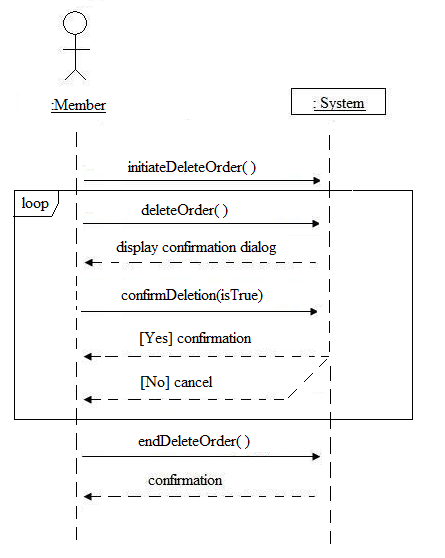
****

Figure: Sequence Diagram for Delete Order from ShoppingCart

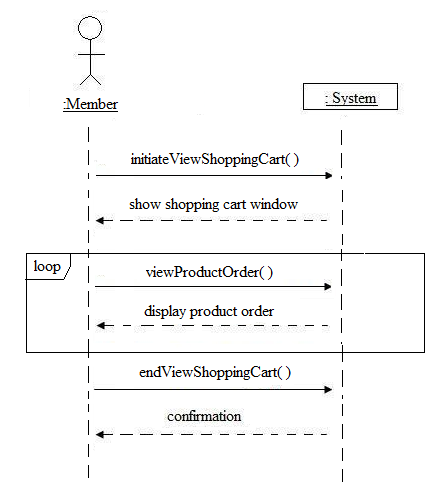
****

Figure: Sequence Diagram for View Order from ShoppingCart

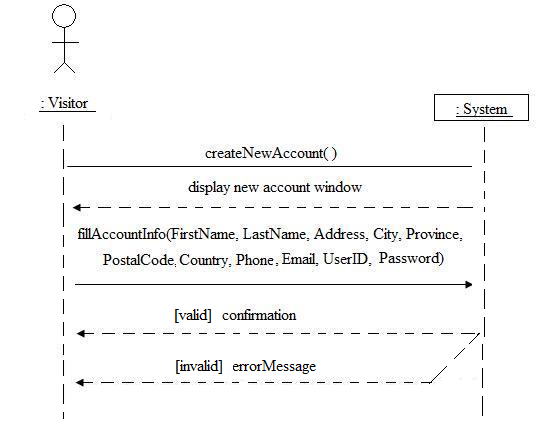


Figure: Sequence Diagram for Create New Account

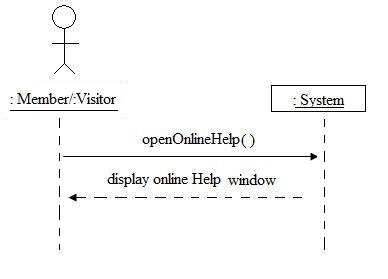


Figure: Sequence Diagram for Obtain Help

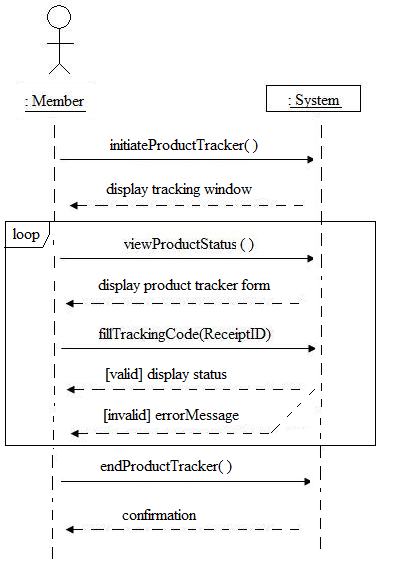


Figure: Sequence diagram for Product Tracking

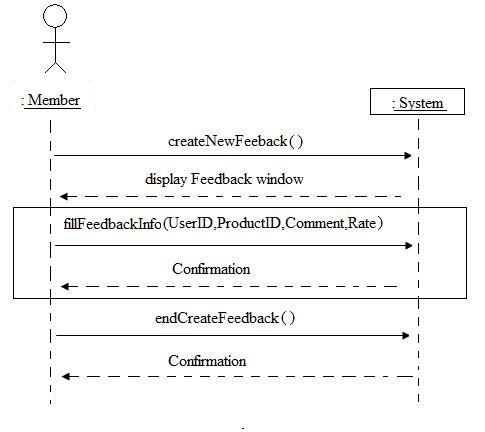


Figure: Sequence Diagram for Post Feedback

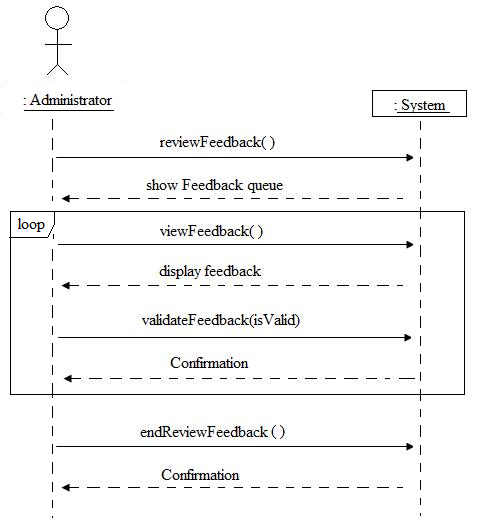


Figure: Sequence Diagram for Monitor Feedback

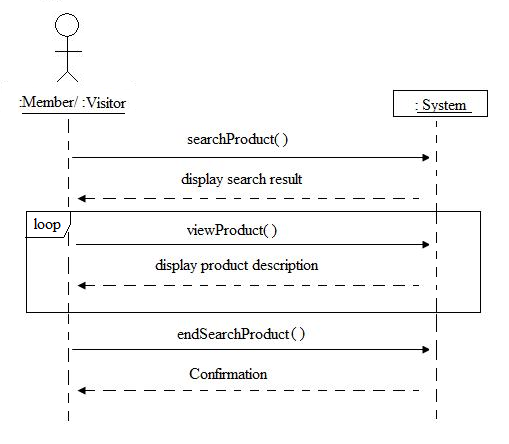


Figure: Sequence Diagram for Search

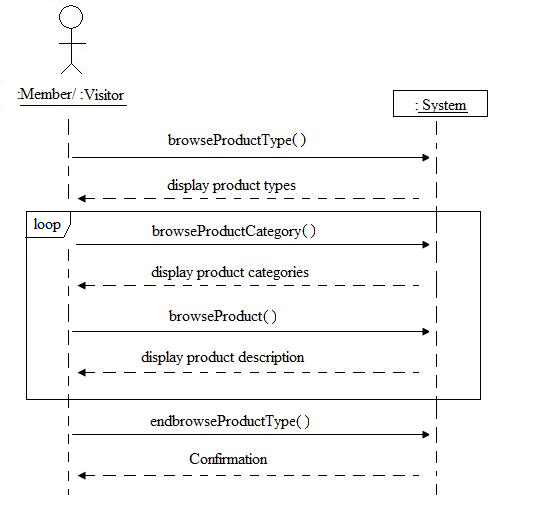


Figure: Sequence Diagram for Browse

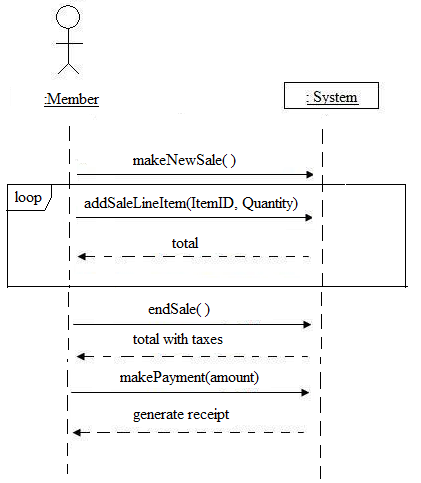


Figure: Sequence Diagram for new Sale

### 3.7.7 Functional Hierarchy

*When none of he above organizational schemes prove helpful, the overall functionality can be organized into a hierarchy of functions organized by either common inputs, common outputs, or common internal data access. Data flow diagrams and data dictionaries can be use dot show the relationships between and among the functions and data.*

## 3.8 Additional Comments

# Change Management Process

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Changes | Author |
| 0.1 | Jan. 26, 2013 | Initial Document Creation | Mischell Flor |
| 0.2 | Feb 05, 2013 | Added Use Cases Section | Mischell Flor |
| 0.3 | Feb. 07, 2013 | Draft Final Scope | Mischell Flor  Tanvin Hossaine |

# Document Approvals

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | Signature | Date |
| Product Owner | Dr.Dssouli |  |  |
| Developers | Mischell Flor  Tanvin Hossaine |  |  |
| QA Testers | Deepak Chandrashekar  Thang Tran |  |  |

# Supporting Information

Appendix B - Test Guidelines.

The following aapendix gives quick guidelines to help QA to create a more complete test specification.