Herb Shopping

Requirements Specification and Analysis

Version 1.0

19.11.2017

Orhan Aykut Kardeş

Berke Kağan Yıldız

Batuhan Kurt

Mert Günay

Prepared for

SE301 Software Engineering



Table of Contents

[1. Introduction 1](#_Toc498723792)

[1.1. Purpose of the System 1](#_Toc498723793)

[1.2. Scope of the System 1](#_Toc498723794)

[1.3. Objectives and Success Criteria of the Project 1](#_Toc498723795)

[1.4. Definitions, Acronyms, and Abbreviations 1](#_Toc498723796)

[1.5. Overview 2](#_Toc498723797)

[2. Current System 2](#_Toc498723798)

[3. Proposed System 2](#_Toc498723799)

[3.1. Overview 3](#_Toc498723800)

[3.2. Functional Requirements 3](#_Toc498723801)

[3.3. Nonfunctional Requirements 6](#_Toc498723802)

[Usability: 6](#_Toc498723803)

[Reliability: 6](#_Toc498723804)

[Performance: 6](#_Toc498723805)

[Supportability: 6](#_Toc498723806)

[Implementation: 6](#_Toc498723807)

[Interface: 7](#_Toc498723808)

[Packaging: 7](#_Toc498723809)

[Legal: 7](#_Toc498723810)

[3.4. System Models 7](#_Toc498723811)

[Scenarios 7](#_Toc498723812)

[Use Case Model 15](#_Toc498723813)

[Object model 29](#_Toc498723814)

[Dynamic model 30](#_Toc498723815)

[3.5. Project Schedule 39](#_Toc498723816)

[4. Glossary 42](#_Toc498723817)

[5. References 43](#_Toc498723818)

REQUIREMENTS ANALYSIS DOCUMENT

# Introduction

## Purpose of the System

This project is a shopping site where users can access herbs and herbal products online in order to reach a healthier life or to get rid of their disease. The main purpose of this project is to help the user to find herbs and herbal products that will be good for the health of the user, depending on the health condition or disease. This website, which contains several quality brands, contains products that the user is looking for. Distributors can add herbal products and herbs to the system and its information. This system also allows the user to place order which will add the items to the user’s cart and make payment for the same. As a result, this system helps the user to achieve a healthier life and cure his or her illness.

## Scope of the System

Our system is called Herbco. Herbco can be used anywhere anytime as it is a web-based application. It so simple to use; register to the system and start shopping instantly. This site is also a good sales platform for many brands and distributors. Any brand that owns quality products on the market can sell their products through this site. This system collects the people who want to live healthy and the brands that want to sell the products under the same roof.

## Objectives and Success Criteria of the Project

The success of the system depends on providing the given main set of arguments:

* Use GitHub repository effectively with your teammates.
* The design of register and login systems.
* Association between database and system design.
* The general design of system in shop to have fast, efficient system.
* The demo of system should ensure good success rate.
* Implementation should be understandable, clear and basic and efficient.

## Definitions, Acronyms, and Abbreviations

The abbreviations and definitions contained in the document are given below:

* Herbco: Herb Shopping Company
* Admin: System Admin
* Distributor: Company and brand CEO who sell their products on the site
* Manager: The manager and owner of this system
* Model: A schematic description of a system that accounts for its known or inferred properties.
* System: Any interacts by the application are considered to be done by the system.
* Efficiency: The properties of an algorithm, which is the amount of computational resources used by the algorithm.
* Service: Service is a keyword. Purpose of the service is to provide the customer with a secure payment system.

## Overview

This document contains sections;

* Introduction
* Current System
* Functional Requirements
* Non-Functional Requirements
* Scenarios
* Use Case Model
* Object Model
* Dynamic Model
* User Interface
* Glossary
* References

In introduction, we have described a good explanation of scope purpose, criteria, definitions and other explanations which help other people who want to read document. In proposed system, we have described the advantages of Herbco and what makes Herbco to unique. In functional requirement, we have described that the functions are open to end users. Functional requirements states for explaining scope of the system.

In non-functional requirement section, we have described our system’s reliability, performance and others to make the system accessible to end users. In Scenarios section, we have described use case’s descriptions and explanations. In Use case model, we have stated or use case diagram. In Object model, we have described the programming objects which will implemented later on. In Dynamic model, we have described the flow of system in time with operations and function of actors. In glossary section, we have described dictionary and unknown words in RAD.

# Current System

Our system is a very intelligent system and, we designed our system on the web. Our system differs from other sites, other sites sell their brands through the site, but in the Herbco system, distributors sell their own products on Herbco, and herbco does not sell their own brands.

Other difference of our system is that if the products added by the distributors are not liked by the users ,as soon as they are removed by the manager of the Herbco system. The communication between the system administrators, distributors and users works in a synchronous manner. The platform is running synchronously, the host and system administrator are very quick to inform.

# Proposed System

Documents the requirements elicitation and the analysis model of the new system

## Overview

Herbco is a web site application that we make to benefit to people who care about their health and sick people, with herbal products. Our system aims to make these people comfortable to find and buy quality products. The Herbco system interface has been designed so that it is understandable to everybody. All the products are in the system in a way that the user can distinguish them with the recognized images, and all the features of the product and the description are understandable.

With Herbco, users can find the products they are looking for in categories. They can also find products that they want with their product name and brand name. In addition, our customers can send feedbacks about products and brands to the manager of the herbo system. Positive or negative comments are briefly evaluated by our system administrators. Obviously the Herbco system has thought of everything it can do to please its customers .Customer will be at a higher level of product satisfaction thanks to the feedback they receive from Herbco managers.

When users enter the Herbco system, they will be able to easily sort through all products on the website. They can also make advance searches according to the brand or product that they trust. Moreover, the products in the Herbco system are also grouped by quantity and weight. The managers of the Herbco system work with customer focus by statistically reviewing the products that are popular and most sold by users, or those that are not preferred.In addition, the managers are currently in communication with the vendor for the renewal of non-staple items. Herbco system administrators are able to handle the necessary products in response to customer objections. Users can increase or decrease the number of products in their shopping carts, depending on their wishes, or they can delete the product directly.

The Herbco website always wants to distribute well-known quality products around the world to heal sick people.As a result, the Herbco system is always designed to please and heal the customers.

## Functional Requirements

***Customer Functional Requirements***

* **Sign Up**

The user registers on the system to shop through the site

* **Sign In**

The user logs in to the system with the username and password from the homepage.

* **List Products**

The user views all products from the products page.

* **View Product Detail**

The user views the product details and description via the product of their choice.

* **Advanced Search**

The user makes extended searches by name, brand and category to make it easier to find a product.

* **Select Product**

The user selects the amount and size of the product he / she wants to receive.

* **Manage Shopping Cart**

The user increases the quantity in addition to the product in the basket and either completely removes the product. In addition to user continues shopping by moves to item shelf.

* **Place Order**

User makes orders of selected products safely through this site.

* **Send Feedback**

If the user encounters a problem or wants to comment on the order, the manager reports the status.

* **Select Purchase**

The user selects the payment method for own order.

***Admin Functional Requirements***

* **Sign In;**

Admin part is different from CostumerSign in and Distributor Sign in. Admin should enter the different UML. In this page, admin can login on system with username and password. Password is given by the system database.

* **Approve Product**

Products that the distributor wants to sell are approved by the admin.

* **Update Product**

Items that are edited by the distributor are approved by the admin.

* **Delete Product**

Products that the distributor does not want to sell are deleted from the system by admin.

* **Process Feedback**

Admin compile the problems that users experience while shopping and their comments on the product.

* **Log Off**

Admin exit the system when they finished the operations on the system.

***Distributor Functional Requirements***

* **Sign Up**

Distributor registers to the system to sell products.

* **Request Password Remainder**

When distributor forgot his/her password, he/she can use this option to reset or remain the password.

* **Add New Product**

The distributor enters all the details of the product it wants to sell and presents it for admin's approval.

* **Edit Stock**

The distributor makes changes to the products on sale and offers for admin approval.

* **Delete Current Product**

Distributor offers products for admin approval that they want to delete.

* **Send Brand Page Info**

Distributor provides information about the brand while recording. At the same time, He or she presents brand information on the site

***Manager Functional Requirements***

* **Sign In**

Admin part is different from CostumerSign in and Distributor Sign in. Admin should enter the different UML. In this page, admin can login on system with username and password. Password is given by the system database.

* **Accept Brand**

Manager allows distributors to sell products according to the quality of the brand.

* **Reject Brand**

Manager does not allow sellers to sell products according to the quality of the distributors who are registered.

* **Check Statistics**

Manager gets more information about which products sell more or which sell less.

* **View Reports**

The Manager sees the mistakes that occur on the order.

## Nonfunctional Requirements

Usability: Using the Herbco website is quite simple, there is a design that everyone can understand and a global interface. Our interface is based on products and pictures, our customers are our priority to use our system without any confusion. Herbco continues to work to create a more effective interface.

Reliability: Herbco system has almost all security requirements that is %90.

Performance: Herbco uses SQLite as its database system, which has a permanent storage space. So, it reacts quite fast and at the same time about 3000 parallel users can enter our system.

Supportability: The maintenance of the Herbco system is a secure, short-lived portal that is open only for managers.

Implementation: The Herbco system is implemented in the Django and Python programming language with excellent OOP and POP features using ATOM and SQLite programs. Herbco uses SQLite database system platform in back-end , In-Front-End Herbco uses Django and its libraries .You can access it via [www.herbco.com](http://www.herbco.com).

Interface: The interface of the Herbco system is designed using Django's interface implementation and HTML, CSS files. In addition, despite the use of Interfaces, third party applications and services used in the Herbco system and disclosed to customers may not be used. There are no inherited projects that present services or interfaces.

Packaging:The system packaging will be with future versions.

Legal:Herbco is licensed with MIT License. It is provided "as is" without warranty of any kind, including, but not limited to, marketability, manufacturability, software, certain fitness for purpose, and any infringement of rights. In any unusual circumstances, administrators shall not be liable for any damage or other liability arising out of the author, authors or copyright holders, from a software contract, tort or otherwise.

## System Models

### Scenarios

**Scenario 1**

|  |
| --- |
| **Scenario name:** Customer List Product |
| **Participant actor instances:**  Mert: Customer |
| **Flow of events:**   * Firstly, Mert enters the herbco web-site as a quest. * After, that he wants to search and list the products. * Mert clicks the "List Product" button. * Then, all the products appear in order. |

**Scenario 2**

|  |
| --- |
| **Scenario name:** Customer View Product Detail |
| **Participant actor instances:**  Mert: Customer |
| **Flow of events:**   * Mert visits the herbco web-site. * He wants to buy product or products from herbco web-site. * Mert finds a product or products that he will buy. * After he listed the products, he can see the product details. * He clicks to "View Product Details" then he sees the description of the relevant product. |

**Scenario 3**

|  |
| --- |
| **Scenario name:** Customer Advanced Search |
| **Participant actor instances:**  Mert: Customer |
| **Flow of events:**   * Mert visits the herbco web-site. * He listed some products and see the product details. * Also, he can do advanced search. Mert clicks to " Advanced Search " button on list product page. * He can search the products with name, category and brand. * Finally, he can find the products easily. |

**Scenario 5**

|  |
| --- |
| **Scenario name:** Customer Select Product |
| **Participant actor instances:** Mert: Customer |
| **Flow of events:**   * Mert visits the herbco web-site. * He wants to buy product or products from herbco web-site. * Mert finds a product or products that he will buy. * After he listed the products, he clicks the "Select Product" button. * Then, he can select the product in two ways that are;   - Select quantity: Mert can select the product's quantity  - Select weight: Mert can select the weight of product   * Also, if the product that are selected by him, has no stock, the system shows   the " Out of Stock " message.   * Finally, if there are no problem, product or products are selected. |

**Scenario 6**

|  |
| --- |
| **Scenario name:** Customer Add Shopping Cart |
| **Participant actor instances:**  Mert: Customer |
| **Flow of events:**   * Firstly, Mert enters the herbco web-site. * He wants to buy a product or products. * If Mert wants to shop, he has to register or login. * After he logins, he can add the products to shop cart. * Mert finds the product and clicks "Add Shopping Cart" button. * Afther that, the products are added to shopping cart. |
|  |

**Scenario 7**

|  |
| --- |
| **Scenario name:** Customer Manage Shopping Cart |
| **Participant actor instances:**  Mert: Customer |
| **Flow of events:**   * Firstly, Mert enters the herbco web-site. * If he wants to manage his shopping cart, he has to be login. * After he logins, he can see his shopping cart. * When he clicks "Manage Shopping Cart", he can add or remove products. * Also, he can move his products to shelf, so he continues the shopping. * After that, he selects what he wants to buy, and approves the shopping cart. |
|  |

**Scenario 8**

|  |
| --- |
| **Scenario name:** Customer Place Order |
| **Participant actor instances:**  Mert: Customer |
| **Flow of events:**   * Firstly, Mert has to logs into the Herbco to place order. * After, he adds to his shop cart or manage to his shop cart, he wants to place order for his product. * Mert clicks to " Place Order " button. * Also, he can send feedback after he bought the product. * Finally**,** his order received. |

**Scenario 9**

|  |
| --- |
| **Scenario name:** Customer Select Purchase |
| **Participant actor instances:**  Mert: Customer |
| **Flow of events:**   * Firstly, Mert logs into the Herbco. * After that, he bought a product. * He orders a product and, he wants to select his payment method. * Then, he clicks to " Select Purchase " button. * He sees two payment method when he clicks to " Select Purchase " button, that are;   - pay at the door  - purchase with credit card   * He chooses the one of them, if there are no limit in his credit card or he doesn't complete the shopping in 5 minutes, the purchase page is closed, so he has to continue with shopping page. * If there are no trouble, his shopping is completed. |

**Scenario 10**

|  |
| --- |
| **Use case name:** Manager Set Ditributor Accounts |
| **Participant actor instances:**  Berke: Manager |
| **Flow of events:**   * Firstly, Berke enters the herbco web-site as a manager. * Then, he wants to set the distributor accounts. * He clicks to " Set Distributor Acoounts " button on the main page. * He can set the distributors with this page. |

**Scenario 11**

|  |
| --- |
| **Scenario name:** Manager Accept Brand |
| **Participant actor instances:**  Berke: Manager**,** Aykut: Distributorutor |
| **Flow of events:**   * After Berke logs as a manager and, clicks to set distributor accounts, he can accept brands. * Aykut sends a request to Berke for sell their products, Berke checks the Aykut's brand, * After he checks the brand, he checks the distributor's income, if the distributor's brands   are sold, manager accept the brand.   * He clicks to " Accept Brand " button. * Finally, he accepts the brand. |

**Scenario 12**

|  |
| --- |
| **Scenario name:** Manager Reject Brand |
| **Participant actor instances:**  Berke: Manager |
| **Flow of events:**   * After Berke logs as a manager and, clicks to set distributor accounts, he can also   reject the brands.   * After he checks the brand and, the brand is not appropriate; * He clicks to " Reject Brand " button. * Finally, he rejects the brand. |

**Scenario 13**

|  |
| --- |
| **Scenario name:** Manager Check Statistics |
| **Participant actor instances:** Berke: Manager |
| **Flow of events:**   * Berke enters the herbco web-site as a manager. * He wants to check statistics. * If he clicks to " Check Statistics " button on main page; * He can see which brands are sold. * So, he decides which brands are trend or which products are best-sellers. |

**Scenario 14**

|  |
| --- |
| **Scenario name:** Manager View Reports |
| **Participant actor instances:**  Berke: Manager |
| **Flow of events:**   * Firstly, Berke enters the herbco web-site as a manager. * Then, he wants to look the reports which are related to products. * He clicks to " View Reports " button on main page. * After he clicks to button, he can see some reports. * The reports include some errors, for example if customers are not satisfied   a product or there is no stock for a product.   * So, Berke fixes that problems with these reports. |

**Scenario 15**

|  |
| --- |
| **Scenario name:** Distributor Add New Product |
| **Participant actor instances:**  Aykut: Distributor |
| **Flow of events:**   * Aykut enters the herbco web-site as a distributor. * He wants to sell his products. * He clicks to " Add New Product " on main page. * Then, he adds the all informations that are pictures, description, qualities about   the product.   * Finally, after he adds informaions, he sends it. |

**Scenario 16**

|  |
| --- |
| **Scenario name:** Distributor Edit Stock |
| **Participant actor instances:**  Aykut: Distributor |
| **Flow of events:**   * After Aykut logs as distributor on the herbco web-site, * If Aykut's products will sold on herbco web-site, and there is no stock for his product   he can edit the stock.   * He clicks to " Edit Stock " button on main page**.** * With this button, he can edit;   - add stock for his product. |

**Scenario 17**

|  |
| --- |
| **Scenario name:** Distributor Delete Current Product |
| **Participant actor instances:**  Aykut: Distributor |
| **Flow of events:**   * After Aykut logs as distributor on the herbco web-site, * If Aykut's all products are sold or he wants to delete his product; * He clicks to " Delete Current Product " button on main page**.** * So, he can delete his every product. |

**Scenario 18**

|  |
| --- |
| **Scenario name:** Distributor Send Brand Page Info |
| **Participant actor instances:**  Aykut: Distributor**,** Berke: Manager |
| **Flow of events:**   * Firstly, Aykut logs into herbco as a distributor. * Then, Aykut wants to send a page which contains brand informations to Berke. * Aykut clicks to " Send Brand Page Info " on main page. * He fills the all blanks which are in page. * He describes his brands, when he completes the page, sents it to Berke. |
|  |

**Scenario 19**

|  |
| --- |
| **Scenario name:** Admin Approve Product |
| **Participant actor instances:**  Batuhan: Admin |
| **Flow of events:**   * Firstly, Batuhan logs into the Herbco as an admin. * Batuhan wants to approve a product that is sended by manager. * When he approves a product; * Batuhan clicks the "Approve Product" button on administrator main page. * Finally, he can click to button and approve the product. |
|  |

**Scenario 20**

|  |
| --- |
| **Scenario name:** Admin Update Product |
| **Participant actor instances:**  Batuhan: Admin |
| **Flow of events:**   * Batuhan logs into the Herbco as an admin. * Batuhan wants to update a product that is approved by manager. * When he updates a product; * Batuhan clicks the "Update Product" button on administrator main page. * So, he can change stock informations of products or if there are any mistake   about products, he can handle it. |

**Scenario 21**

|  |
| --- |
| **Scenario name:** Admin Remove Product |
| **Participant actor instances:**  Batuhan: Admin |
| **Flow of events:**   * Firstly, Batuhan logs into the Herbco as an admin. * If there are no stock for a product or any informations that belongs a product   is wrong, Batuhan will remove this product.   * Batuhan clicks the "Remove Product " button on administrator main page. * After that, he can remove any product. |
|  |

**Scenario 22**

|  |
| --- |
| **Use case name:** Admin Process Feedback |
| **Participant actor instances:**  Batuhan: Admin**,** Mert: Customer |
| **Flow of events:**   * Batuhan enters the herbco web-site as an admin. * Then if he receives feedbacks, he wants to evaluate them. * So, Batuhan clicks to " Process Feedback " button on administrator page. * After he clicks, he can see the feedbacks that are came from the Customers. * When he sees the feedbacks, he can evaluate this feedbacks for example;   - If any problem with a product, he can remove or,  - If a product is very successful, he can see it. |

**Scenario 23**

|  |
| --- |
| **Scenario name:** Sign up |
| **Participant actor instances:**  Mert: Customer, Berke: Manager, Aykut: Distributor, Batuhan: Admin |
| **Flow of events:**   * Firstly, if a visitor enters to herbco web-site. * Then, he wants to shop or wants to register herbco-web site. * He clicks to "Sign Up" button on the main page. * He fills the all fields and clicks Sign Up. * If informations are correct, and the passwords that are entered by him matches,   he can access his account.   * Finally, user account is created. |

**Scenario 24**

|  |
| --- |
| **Scenario name:** Sign In |
| |  | | --- | | **Participant actor instances:**  Mert: Customer, Berke: Manager, Aykut: Distributor, Batuhan: Admin | |
| **Flow of events:**   * A visitor enters to herbco web-site. * Then, he wants to shop or he may be customer, distributor, manager or admin. * If he wants to only visit the web-site he doesn't have to be login, but he wants to   login as an admin, distributor, manager or customer he has to be login.   * So, He clicks to "Sign In" button on the main page. * He fills the username and password fields, then clicks sign in. * If informations are correct, he will sign in, buy if the password is not correct, he can   clicks to " Remember Password " button to reset his password.   * Finally, he signs in and if he wants to log out, he clicks to "Log out " button to log   on main page. |

### Use Case Model

**Use case 1**

|  |
| --- |
| **Use case name:** Sign-up |
| **Participant actors:** Costumer, Distributor |
| **Flow of events:**   * Actor enters to the site. * Actor clicks "Sign-up" button. * Site responds by displaying the Sign-up screen. * Actor enters all required fields. * Actor clicks "Sign-up" button. * System inserts the information to the database which is provided by User. * System redirecting distributor to login page. |
| **Entry Condition:** Actor visits the URL page of the system. |
| **Exit Condition:** Actor clicks "Sign-up" button and entered information should be correct. |
| **Quality Requirements:**  1. If Actor leaves one or more fields empty, system displays a warning message, like "This area cannot be empty.”  2. The information that user entered is checking and verified by the system in real-time |

**Use case 2**

|  |
| --- |
| **Use case name:** Sign-in |
| **Participant actors:** All Actors |
| **Flow of events:**   * Actor goes on sign-in page. * System responds by displaying the login screen of the application. * Actor enters username and password to the login fields. * Actor clicks "Login" button. * If the username and password match with the username and password which is stored in database. User authenticates. * System fetches the dashboard of student. |
| **Entry Condition:** Actor visits the URL page of the system. |
| **Exit Condition:** Actor clicks "Sign-in" button and entered information should be correct. |
| **Quality Requirements:**  1. If Actor leaves one or more fields empty, system displays a warning message, like "This area cannot be empty.”  2. If the information’s checked from database are not true, system displays a warning message, like "Wrong username or password, please retry.” |

**Use Case 3**

|  |
| --- |
| **Use Case Name:** Send Brand Page Info |
| **Participant actor instances:** Distributor |
| **Flow of events:**   * This event starts with Distributor's click to the Brand Info Settings button. * Distributor is directed to the Brand Info Settings. * Distributor sets the info page of the brand. * Distributor clicks done * Brand Page Info inserted to the database and waiting for approval by manager. |
| **Entry Condition:** Distributor should be signed-in. |
| **Exit Condition:** A request is created. Or alternatively empty one is created. |
|  |
| **Quality Requirement:**  **1.** System response to the actor arrives in a few seconds. |

**Use Case 4**

|  |
| --- |
| **Use Case Name:** List Product |
| **Participant actor instances:** Customer |
| **Flow of events:**   * This event starts with Customer’s typing in search bar and click to the search button. * System check the database if the searched string is in it. * System shows to Customer the product list as searched. * Customer clicks any categories he/she desires. * System shows the customer the detail of the product. |
| **Entry Condition:** |
| **Exit Condition:** Products listed |
| **Quality Requirement:**  **1.** System response to the actor arrives in a few seconds. |

**Use Case 5**

|  |
| --- |
| **Use Case Name:** Select Product |
| **Participant actor instances:** Customer |
| **Flow of events:**   * This event starts with Customer's click to a product. * System shows the customer the detail of the product. * System put customer to the Select Quantity or Select Weight use case. |
| **Entry Condition:** |
| **Exit Condition:** Product selected |
| **Quality Requirement:**  **1.** System response to the actor arrives in a few seconds. |

**Use Case 6**

|  |
| --- |
| **Use Case Name:** Add Shopping Cat |
| **Participant actor instances:** Customer |
| **Flow of events:**   * This event starts with Customer's click to a product. * System shows the customer the detail of the product. * Customer selects the quantity or weight. * System checks from database if the product is out of stock or not. * System add the product to the shopping cart if it is in stocks. |
| **Entry Condition:** Customer should be logged. And also, should already selected a product. |
| **Quality Requirement:**   * System response to the actor arrives in a few seconds. |
| **Exit Condition:** The product is on the shopping cart, or out of stock. |

**Use Case 7**

|  |
| --- |
| **Use Case Name:** Manage Shopping Cart |
| **Participant actor instances:** Customer |
| **Flow of events:**   * This flow starts with Customer ’s clicks to the Shopping Cart button. * System shows to Customer his/her Shopping Cart. * Customer can make changes on the shopping cart and Approve Shopping Cart button. * Systems updates the shopping cart |
| **Entry Condition:** User should already have added product to shopping cart |
| **Exit Condition:** Shopping cart is updated |
| **Quality Requirement:**   * System response to the actor arrives in a few seconds. * Customer can Add/remove item in cart * Customer can move item to shelf |

**Use Case 8**

|  |
| --- |
| **Use Case Name:** Place Order |
| **Participant actor instances:** Customer |
| **Flow of events:**   * This event starts with User’s click to the Place Order button. * System gets the required fields to place an order for customer. * Customer fills the required fields and clicks the complete order button. * System checks the required fields and places the order. |
| **Entry Condition:** User should Approve Shopping Cart |
| **Exit Condition:** Order placed or canceled. |
| **Quality Requirement:**  **1.** Customer can cancel the order before complete it.  2. Customer can send feedbacks after the order is completed.  3. If customer leaves a field empty, system displays a warning message, like "This area cannot be empty.". |

**Use Case 9**

|  |
| --- |
| **Use Case Name:** Select Purchase |
| **Participant actor instances:** Customer |
| **Flow of events:**   * This event starts with Customer ’s clicks to Purchases button. * Customer selects a purchase. * System shows the information about the selected purchase. |
| **Entry Condition:** Customer should have purchased |
| **Exit Condition:** Selected groups) is deleted from users group page. |
| **Quality Requirement:**  **1.** System response to the actor arrives in a few seconds. |

**Use Case 10**

|  |
| --- |
| **Use case name:** Set Distributor Accounts |
| **Participant actors:** Manager |
| **Flow of events:**   * Manager clicks to the Set Distributor Accounts button on Distributor Accountspage. * System fetches the screen that displays the information’s of all distributors. * Manager selects a distributor. * System shows the brands of the distributor * Manager Accept or Reject a Brand. * System saves the changes to the database. * System displays a "Saved successfully!" message on a dialogue box. * System redirects manager to the Distributor Accounts page. |
| **Entry Condition:** Manager should signed-in. |
| **Exit Condition:** Manager accepts or rejects or cancels the changes. |
| **Quality Requirements:**  1. Manager can click on Cancel button to delay the accept or reject actions. |

**Use Case 11**

|  |
| --- |
| **Use case name:** Check Statistics |
| **Participant actors:** User |
| **Flow of events:**   * Manager clicks the "Check Statistics" button on Statistics page. * System displays the statistics of the site. |
| **Entry Condition:** Manager should be signed-in. |
| **Exit Condition:** Manager clicks "back" button. |
| **Quality Requirements:** None |

**Use Case 12**

|  |
| --- |
| **Use case name:** View Reports |
| **Participant actors:** Manager |
| **Flow of events:**   * Manager clicks the View Reports button on Statistics page. * Manager clicks to a distributor or a product * System shows the view reports of distributor or product that manager clicked. |
| **Entry Condition:** Manager should be signed-in |
| **Exit Condition:** Manager clicks "back" button. |
| **Quality Requirements:** None. |

**Use Case 13**

|  |
| --- |
| **Use case name:** Manage Products |
| **Participant actors:** Admin |
| **Flow of events:**   * Admin clicks any managing buttons (approve product, update product, remove product) on his/her manage products box. * System displays the products. * Admin does the managing with required boxes and fields. * System updates database with the changes made. |
| **Entry Condition:** Admin should be signed-in. |
| **Exit Condition:** Admin clicks any managing tool and does the managing. |
| **Quality Requirements:** None. |

**Use Case 14**

|  |
| --- |
| **Use case name:** Process Feedback |
| **Participant actors:** Admin |
| **Flow of events:**   * Admin clicks the Feedbacks button on admin panel page. * System shows all feedbacks to admin * Admin selects a feedback and click select button. * System displays the feedback and all information about it. * Admin process it with process it button. |
| **Entry Condition:** Admin should be signed-in. |
| **Exit Condition:** Admin clicks "Back" button and system direct to the manage products page. |
| **Quality Requirements:** None. |

**Use Case 15**

|  |
| --- |
| **Use case name:** Approve Product |
| **Participant actors:** Admin |
| **Flow of events:**   * Admin clicks Approve Product button on the Managing Box. * System shows all the products that are not approved. * Admin ticks the approve boxes of products. * Admin clicks done. * System puts the items to the site. |
| **Entry Condition:** Admin should be signed-in. |
| **Exit Condition:** Admin approves a product or click back button. |
| **Quality Requirements:** None. |

**Use Case 16**

|  |
| --- |
| **Use case name:** Remove Product |
| **Participant actors:** Admin |
| **Flow of events:**   * Admin clicks Remove Product button on the Managing Box. * System shows all the products. * Admin selects a product. * Admin clicks done. * System removes the item. |
| **Entry Condition:** Admin should be signed-in. |
| **Exit Condition:** Admin removes a product or click back button. |
| **Quality Requirements:** None. |

**Use Case 17**

|  |
| --- |
| **Use case name:** Update Product |
| **Participant actors:** Admin |
| **Flow of events:**   * Admin clicks Update Product button on the Managing Box. * System shows all the products. * Admin selects a product. * System shows the all information that can be changed. * Admin changes any of the information. * Admin clicks done * System updates the product in the database. |
| **Entry Condition:** Admin should be signed-in. |
| **Exit Condition:** Admin updates a product or click back button. |
| **Quality Requirements:** None. |

**Use Case 18**

|  |
| --- |
| **Use case name:** Send Product Form |
| **Participant actors:** Distributor |
| **Flow of events:**   * Distributor clicks Product Form page. * System shows the page with 3 buttons, Add New Product, Edit Stock and Delete Current Product * Distributor clicks any of them. * System displays a page with required information. * Distributor fills and sets the requirements and clicks done * System makes the changes in database. |
| **Entry Condition:** Distributor should be signed-in. |
| **Exit Condition:** Distributor adds, edits or delete product. |
| **Quality Requirements:** None. |

**Use Case 19**

|  |
| --- |
| **Use case name:** Add New Product |
| **Participant actors:** Distributor |
| **Flow of events:**   * Distributor clicks Add New Product button. * System displays a page with required fields * Distributor fills the required fields and clicks done. * System adds the new product in database. |
| **Entry Condition:** Distributor should be in the Product Form page. |
| **Exit Condition:** Distributor adds the product or click back button. |
| **Quality Requirements:**  **1** System displays a confirmation box after distributor clicks done. |

**Use Case 20**

|  |
| --- |
| **Use case name:** Delete Current Product |
| **Participant actors:** Distributor |
| **Flow of events:**   * Distributor clicks Delete Current Product button. * System displays a page with all products that the distributor added before. * Distributor selects a product. * Distributor clicks done. * System deletes the selected product in database. |
| **Entry Condition:** Distributor should be in the Product Form page. |
| **Exit Condition:** Distributor deletes the product or click back button. |
| **Quality Requirements:**  **1** System displays a confirmation box after distributor clicks done.  2 If the distributor has not added product before, system displays a message that says, “You don't have any product”. |

**Use Case 21**

|  |
| --- |
| **Use case name:** Edit Stock |
| **Participant actors:** Distributor |
| **Flow of events:**   * Distributor clicks Edit Stock button. * System displays a page with all products that the distributor added before. * Distributor selects a product. * System displays a bar with current stock number. * Distributor changes the number of stock. * Distributor clicks done. * System changes the stock of the product in database. |
| **Entry Condition:** Distributor should be in the Product Form page. |
| **Exit Condition:** Distributor edits the product stock or click back button. |
| **Quality Requirements:**  **1** System displays a confirmation box after distributor clicks done.  **2** If the distributor has not added product before, system displays a message that says, “You don't have any product”.  **3** If distributor enters anything but positive number, system shows a massage that says, “You should enter a positive number” |

**Use Case 21**

|  |
| --- |
| **Use Case Name:** Select Quantity |
| **Participant actor instances:** Customer |
| **Flow of events:**   * This event starts after Customer selects a product. * System shows the required quantity box. * Customer selects the quantity of the selected product. * System checks from database if the product is out of stock or not. |
| **Entry Condition:** Customer should select a product. |
| **Exit Condition:** Product quantity selected, or is out of stock |
| **Quality Requirement:**   1. System response to the actor arrives in a few seconds. 2. If the desired quantity is not available system shows the available quantity to customer. |

**Use Case 22**

|  |
| --- |
| **Use Case Name:** Select Weight |
| **Participant actor instances:** Customer |
| **Flow of events:**   * This event starts after Customer selects a product. * System shows the required weight box. * Customer selects the quantity of the selected product. * System checks from database if the product is out of stock or not. |
| **Entry Condition:** Customer should select a product. |
| **Exit Condition:** Product quantity selected, or is out of stock |
| **Quality Requirement:**   1. System response to the actor arrives in a few seconds. 2. If the desired quantity is not available system shows the available quantity to customer. |

**Use Case 23**

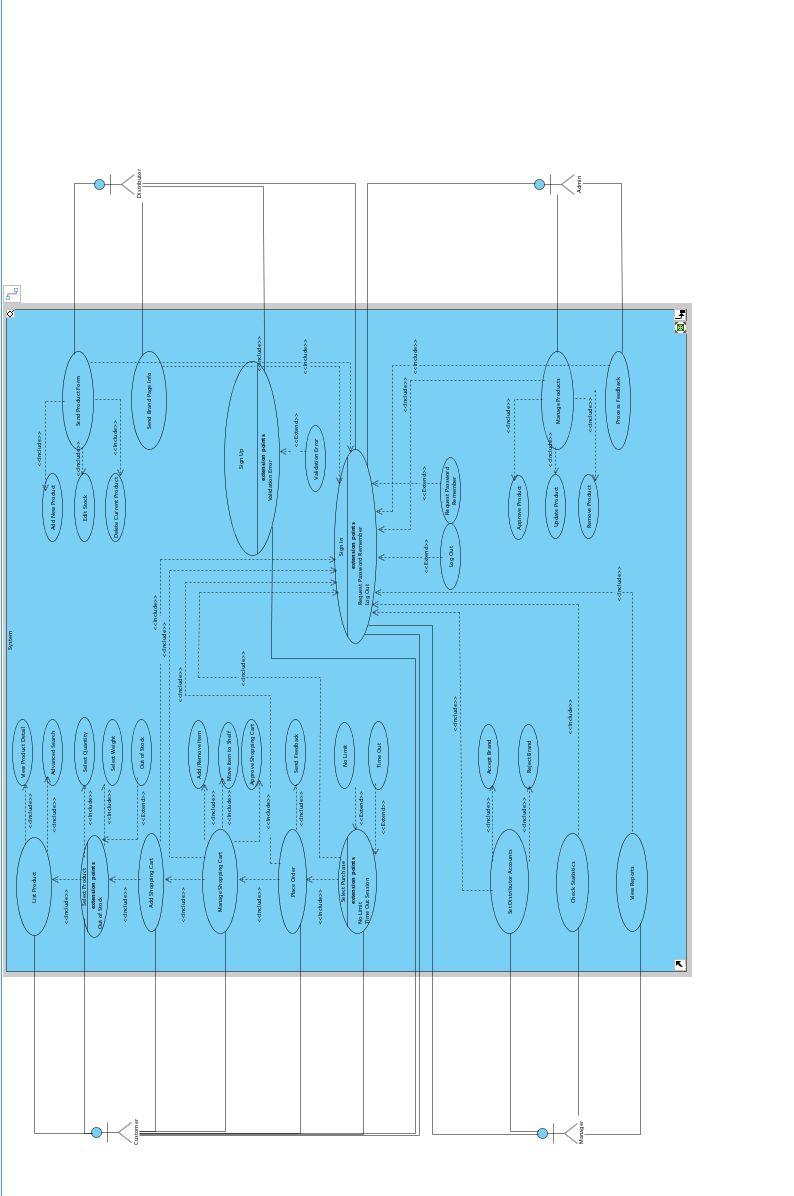
|  |
| --- |
| **Use Case Name:** Add/Remove Item |
| **Participant actor instances:** Customer |
| **Flow of events:**   * This flow starts with Customer’s click to the Shopping Cart button. * System shows the Customer's Shopping Cart. * Customer changes the quantity of the product or remove it. * Systems updates the shopping cart |
| **Entry Condition:** User should already have added product to shopping cart |
| **Exit Condition:** Shopping cart is updated |
| **Quality Requirement:**   * System response to the actor arrives in a few seconds. |

**Use Case 24**

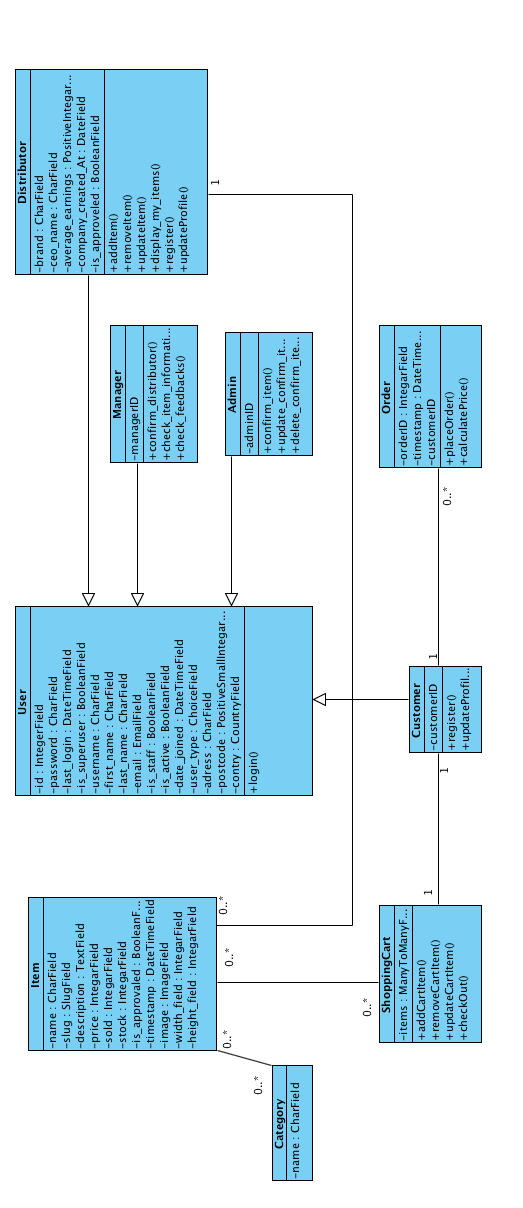
|  |
| --- |
| **Use Case Name:** Move Item to Shelf |
| **Participant actor instances:** Customer |
| **Flow of events:**   * This flow starts with Customer’s click to the Shopping Cart button. * System shows the Customer's Shopping Cart. * Customer clicks move item to shelf button. * Systems moves the item to the shelf |
| **Entry Condition:** User should already have added product to shopping cart |
| **Exit Condition:** Shopping cart is updated |
| **Quality Requirement:**   * System response to the actor arrives in a few seconds. |

**Use Case 25**

|  |
| --- |
| **Use Case Name:** Send Feedback |
| **Participant actor instances:** Customer |
| **Flow of events:**   * This flow starts after customer places an order. * Customer clicks the product that has been ordered. * System displays a send feedback button. * Customer clicks the send feedback button. * System displays a feedback page with required fields. * Customer fills the required fields and clicks done. * System adds the feedback to the database. |
| **Entry Condition:** User should already place an order. |
| **Exit Condition:** Feedback sent. |
| **Quality Requirement:**   * System response to the actor arrives in a few seconds. * System shows an error message if any of the fields that is required to fill is left blank. |

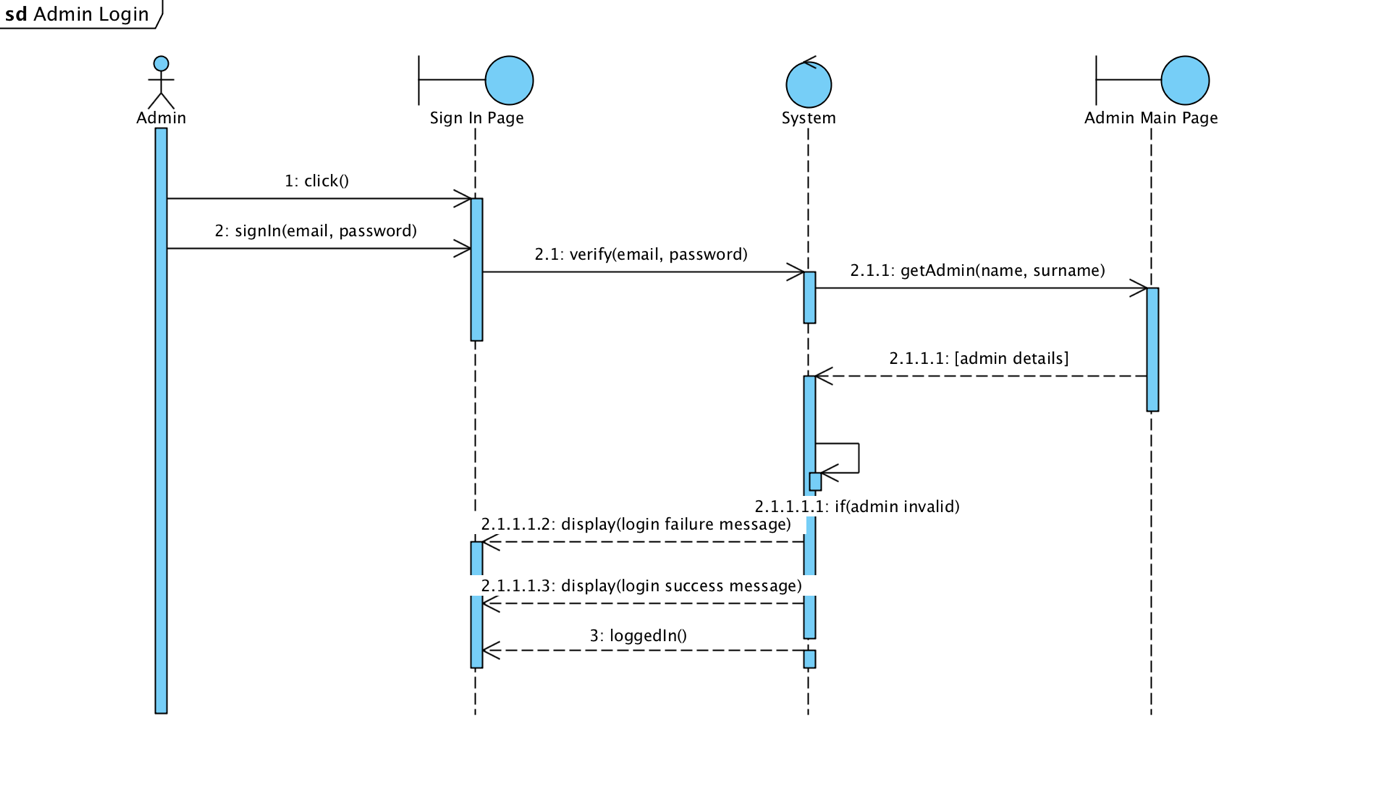


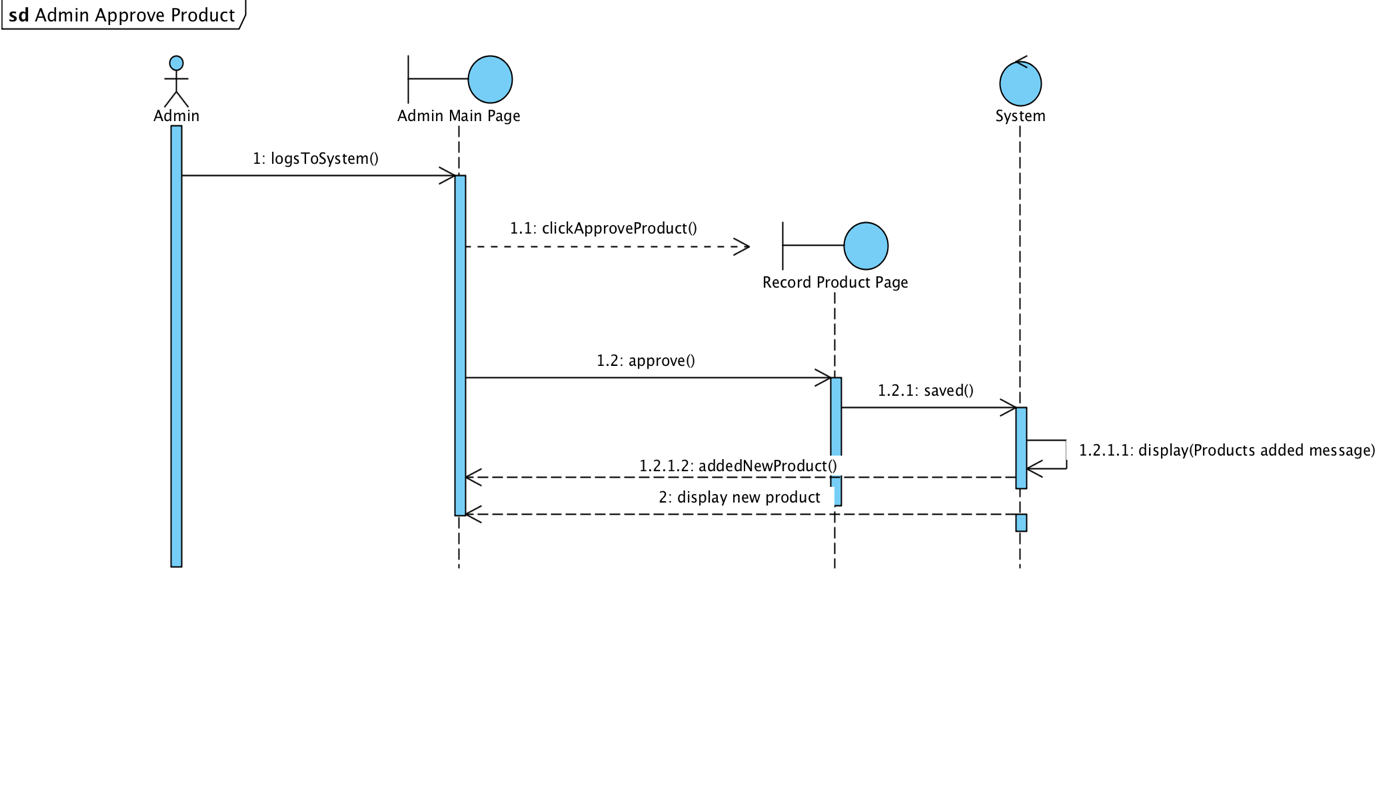
### Object model

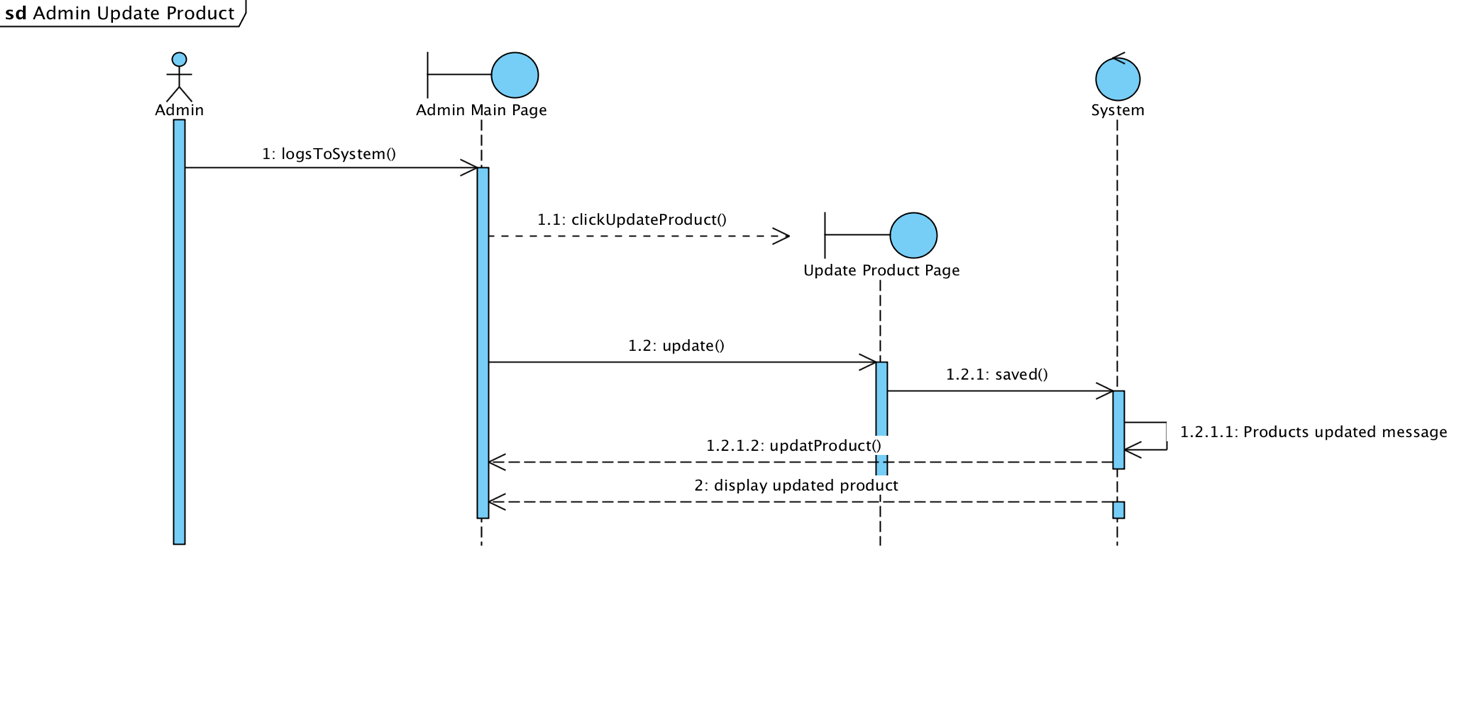


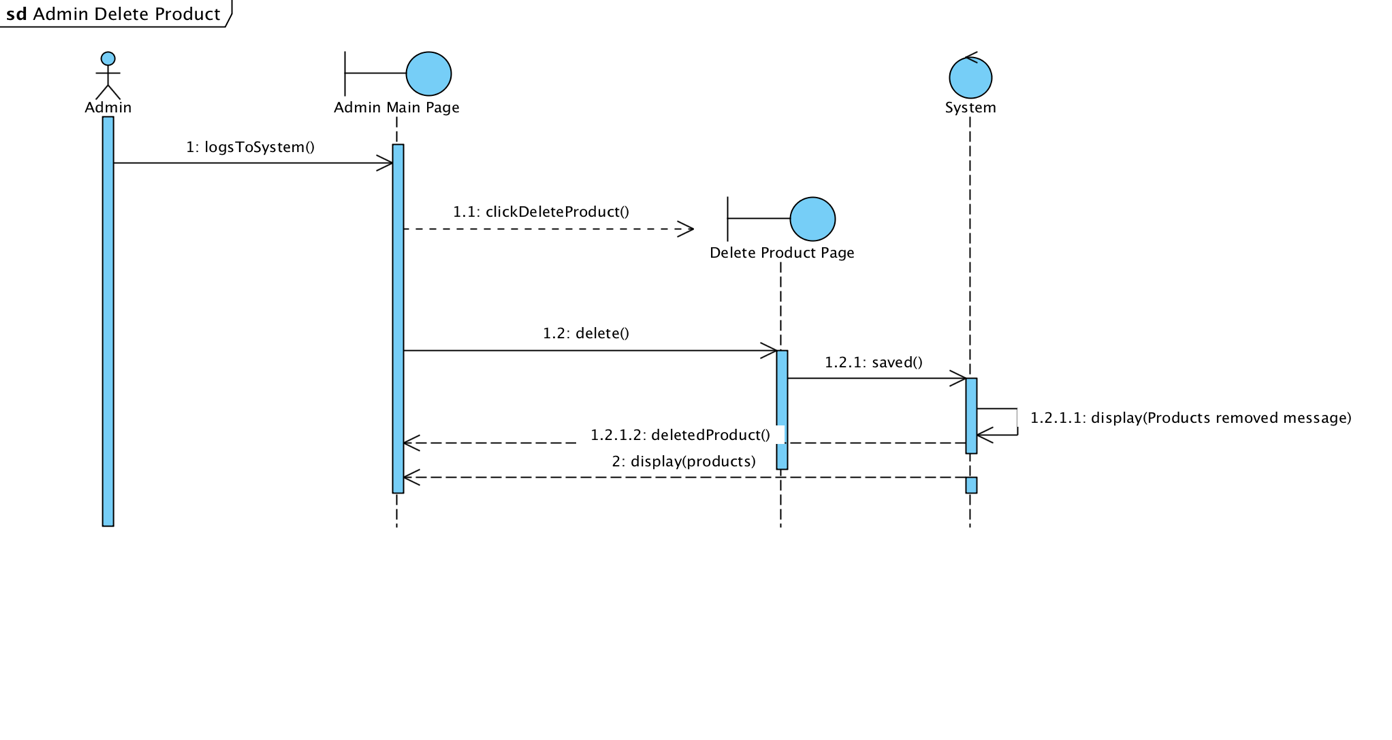
### Dynamic model

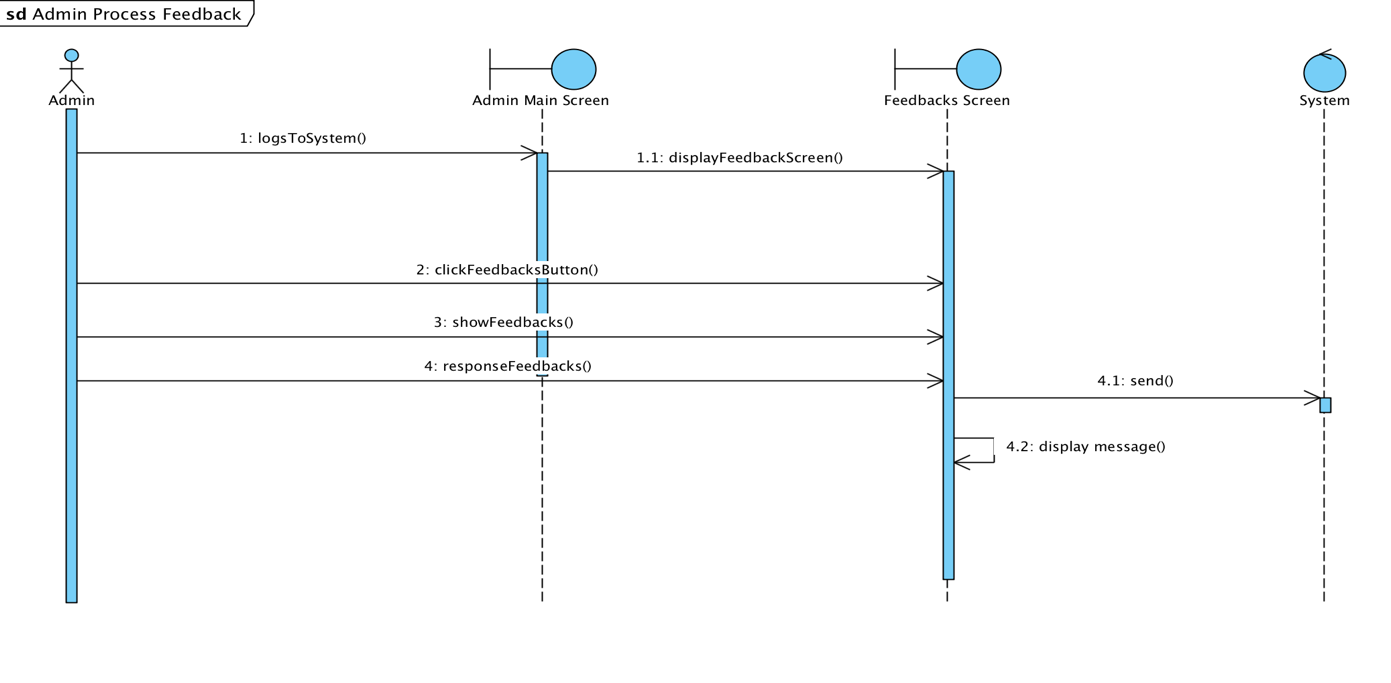
Admin Dynamic Model



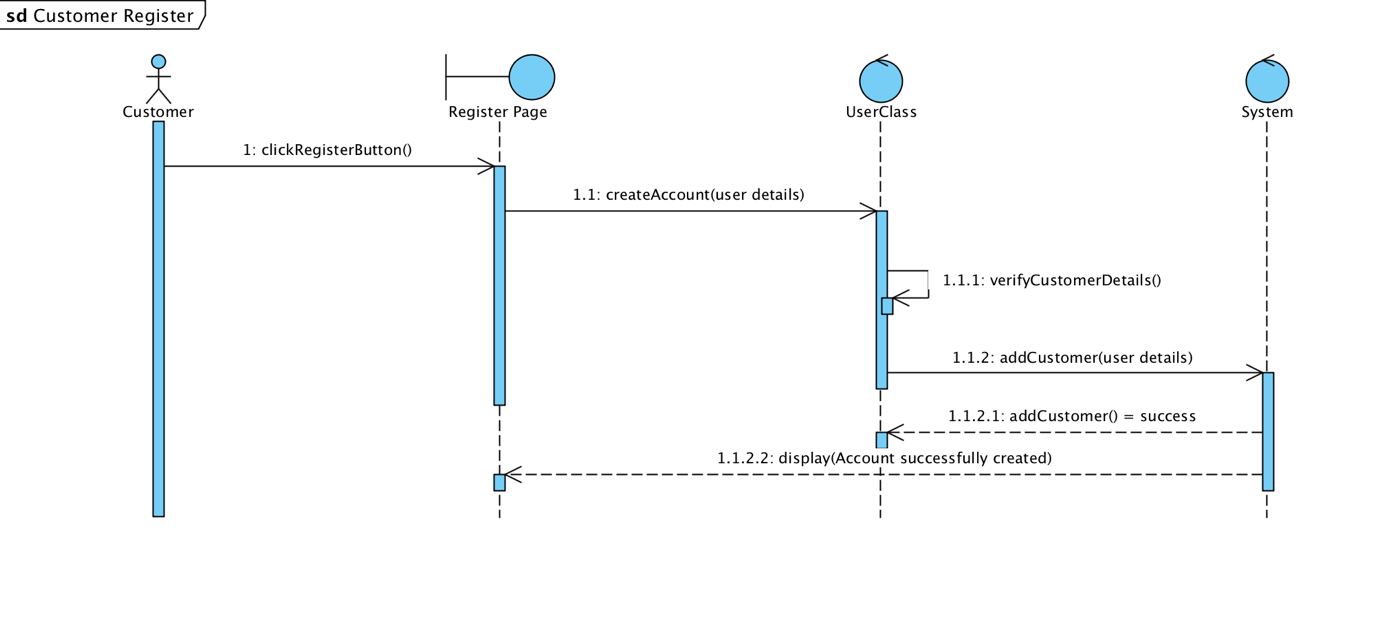


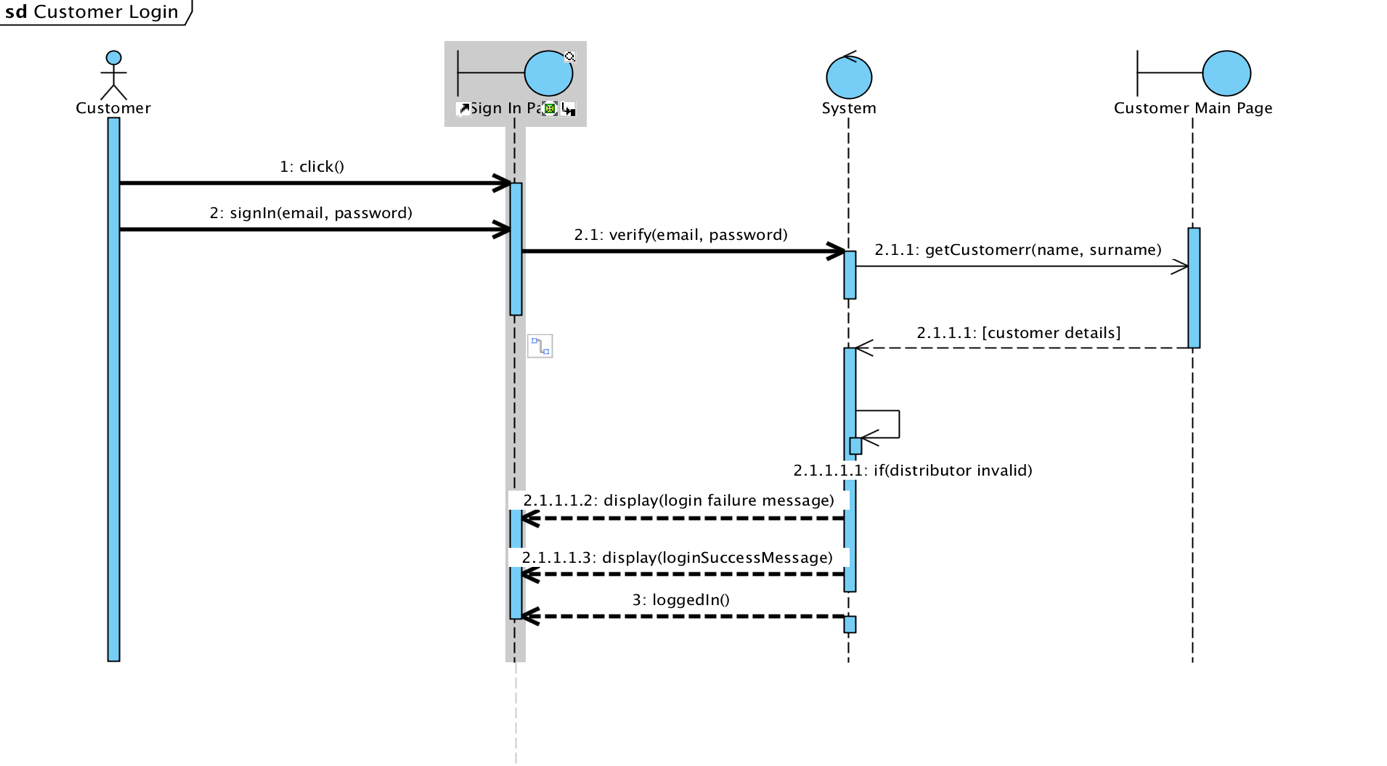


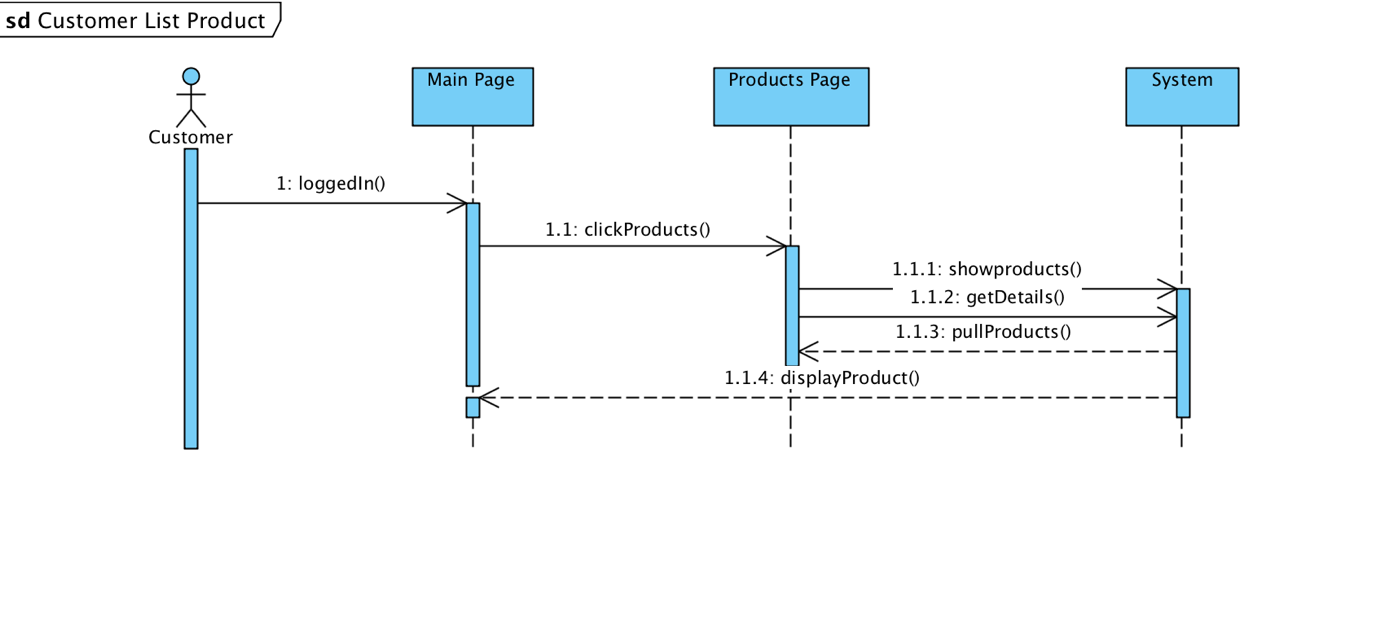


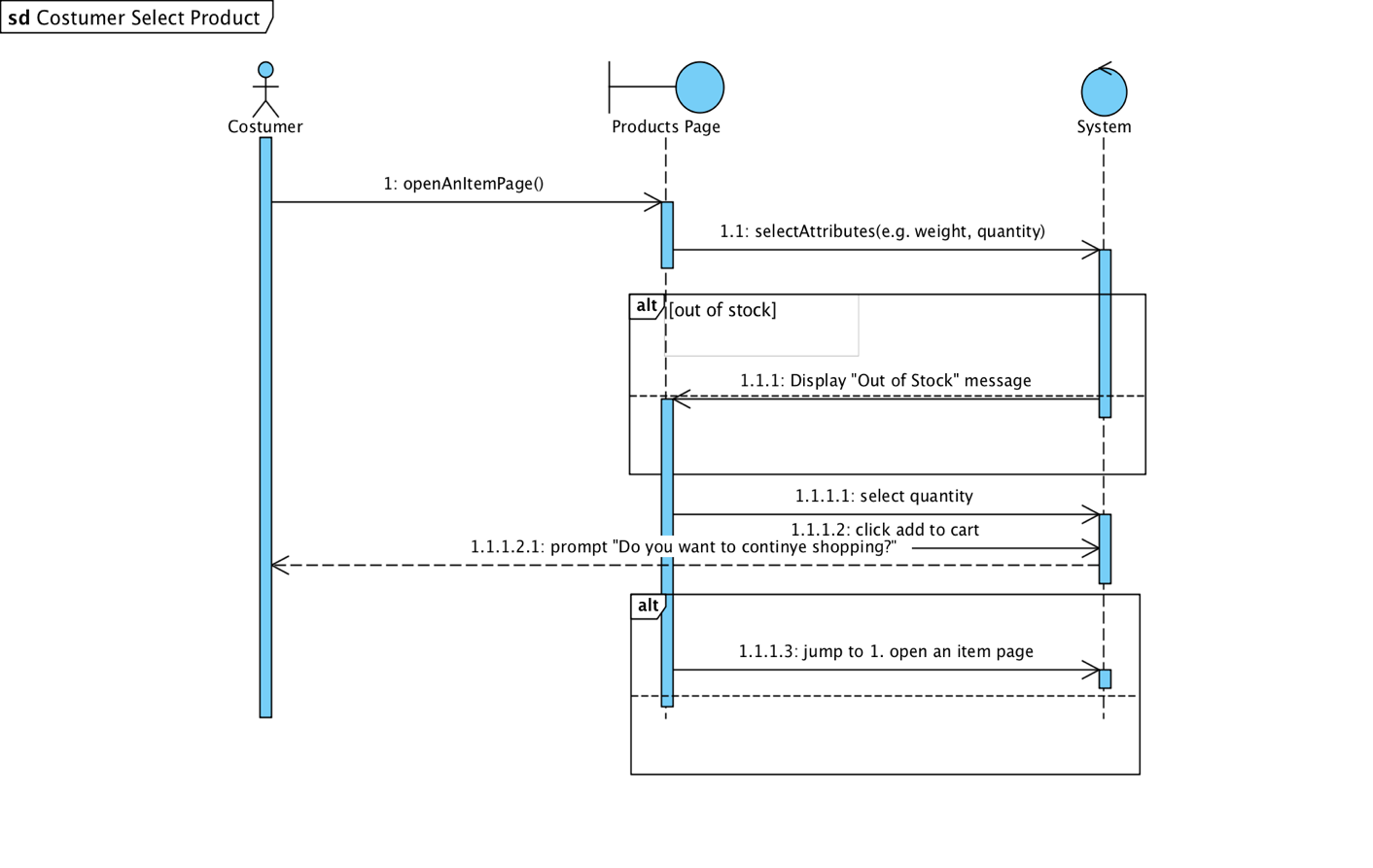


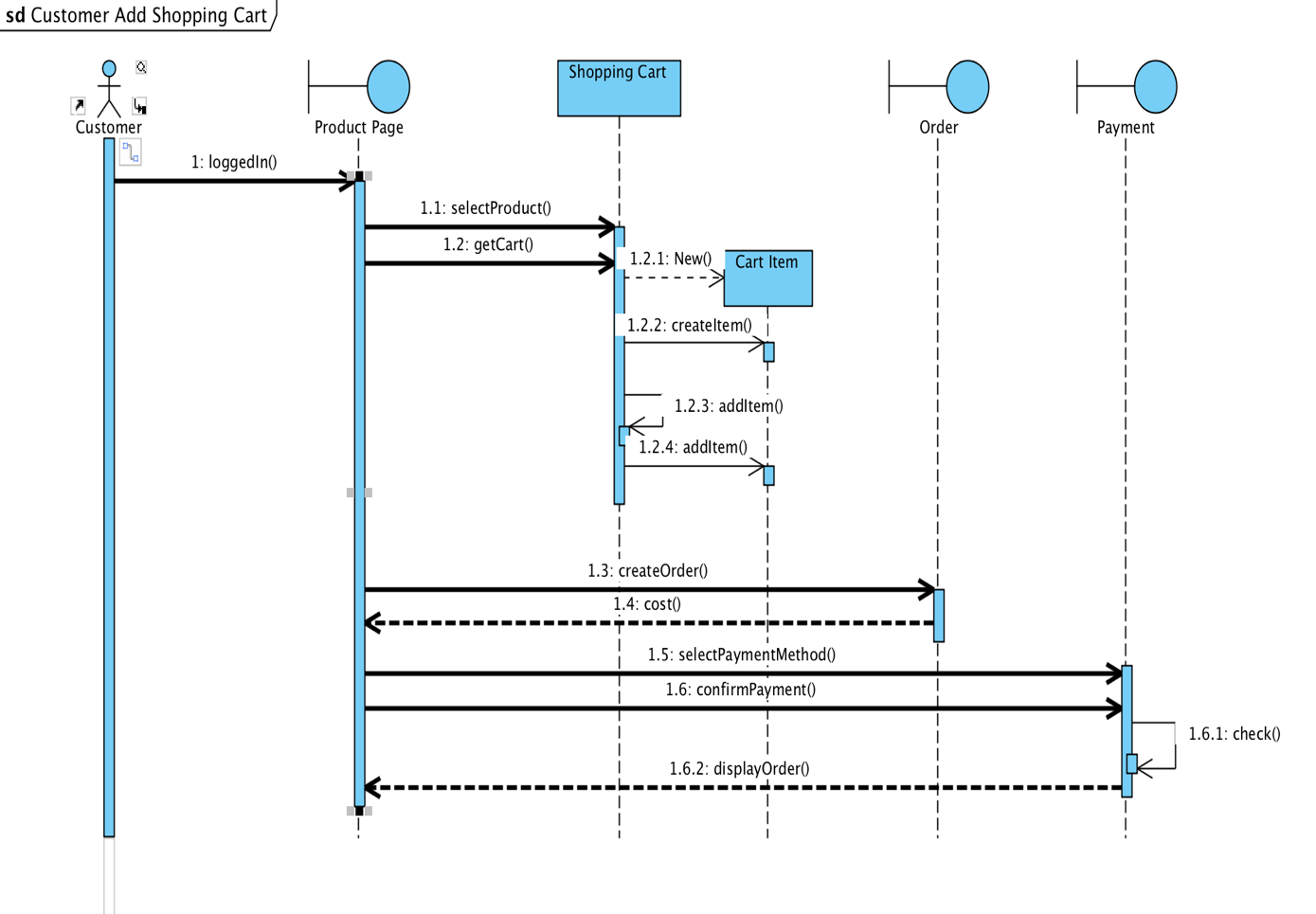
Costumer Dynamic Model



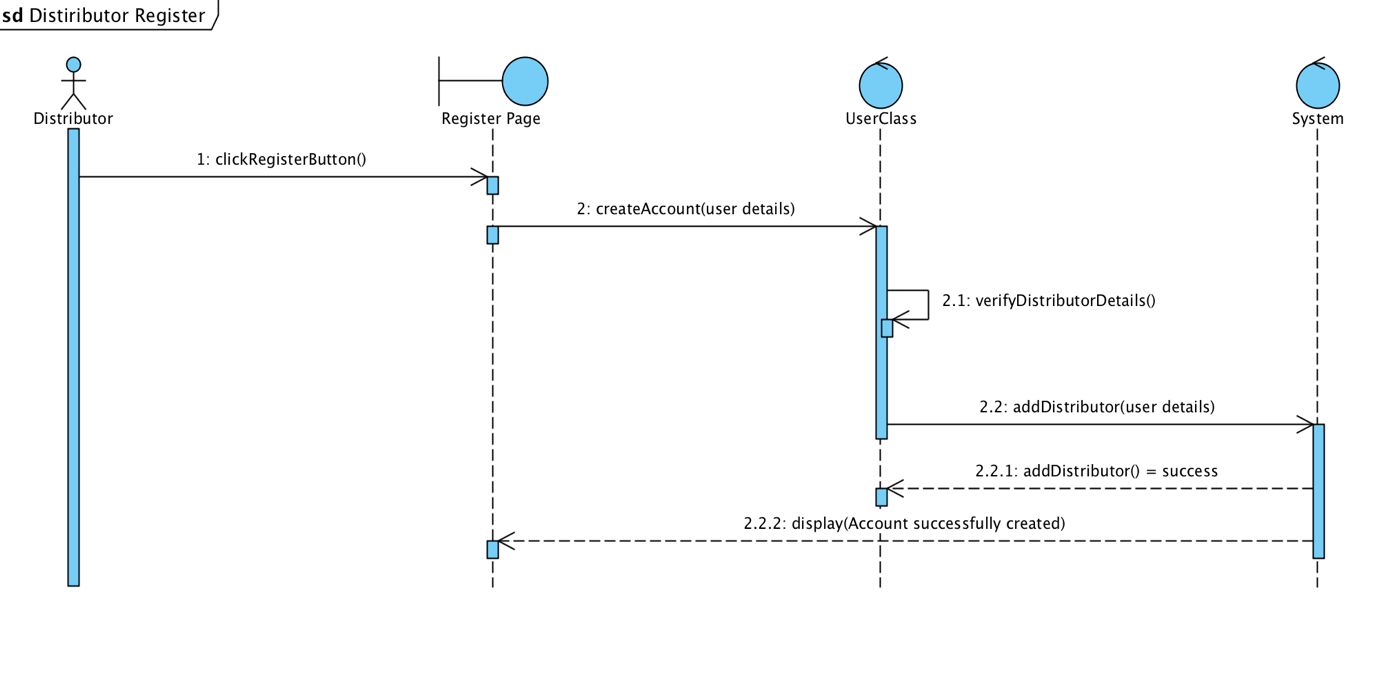


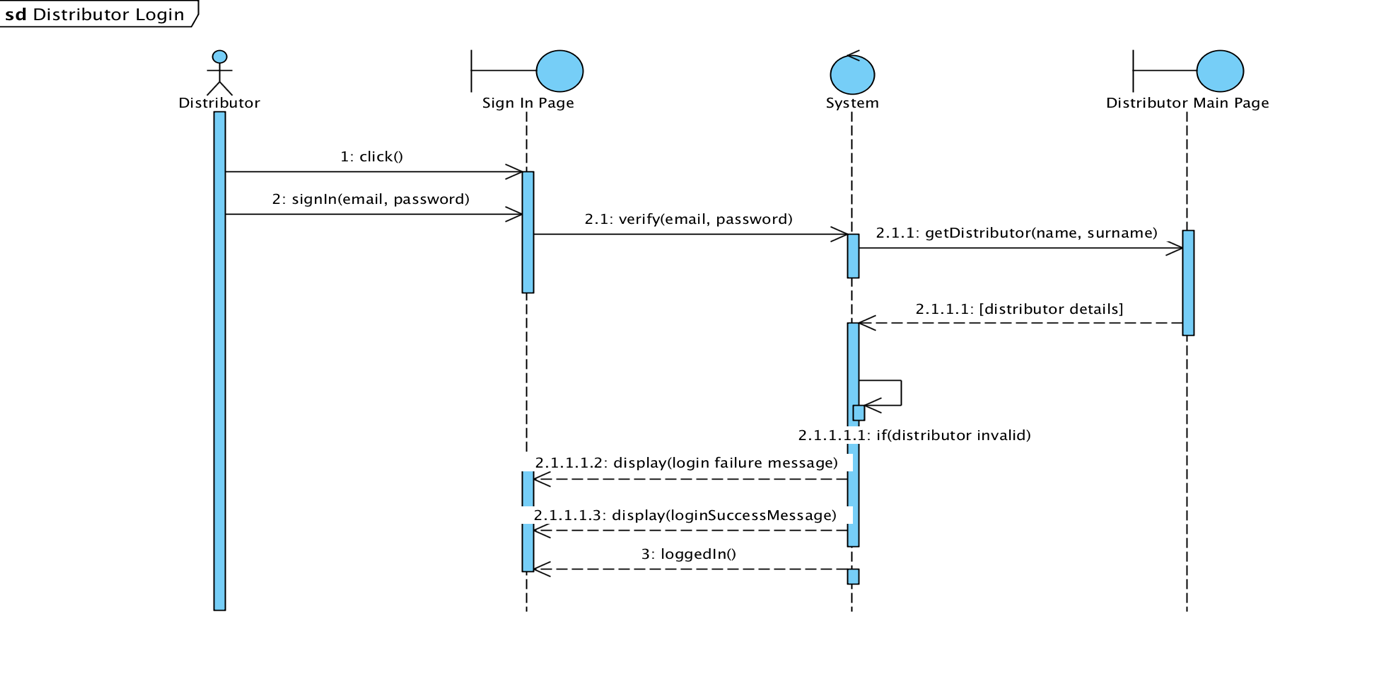


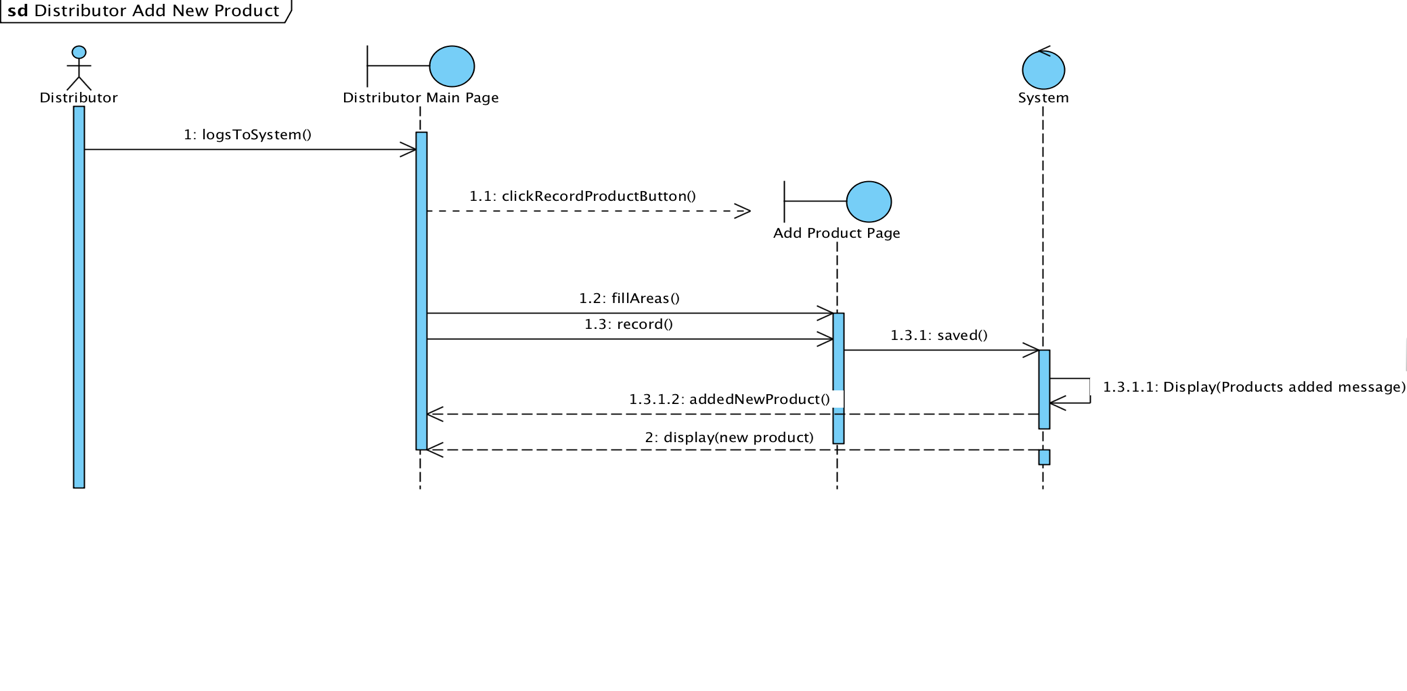


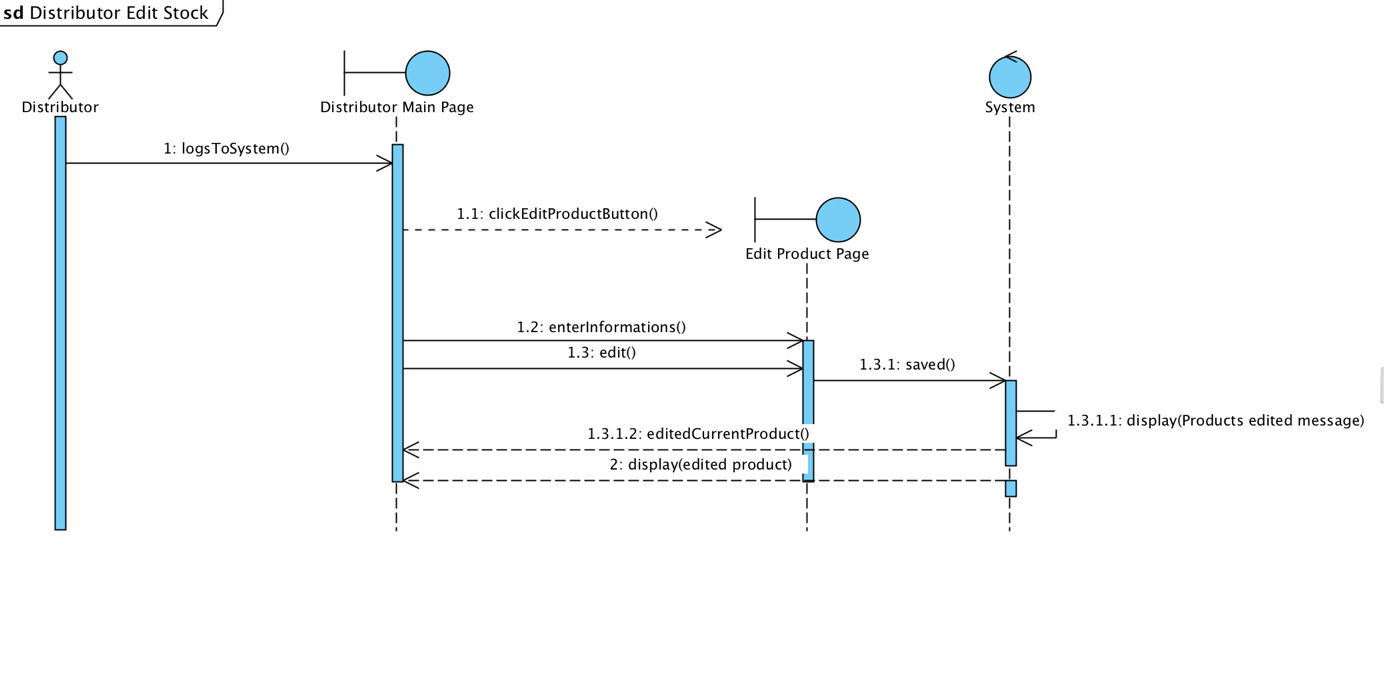


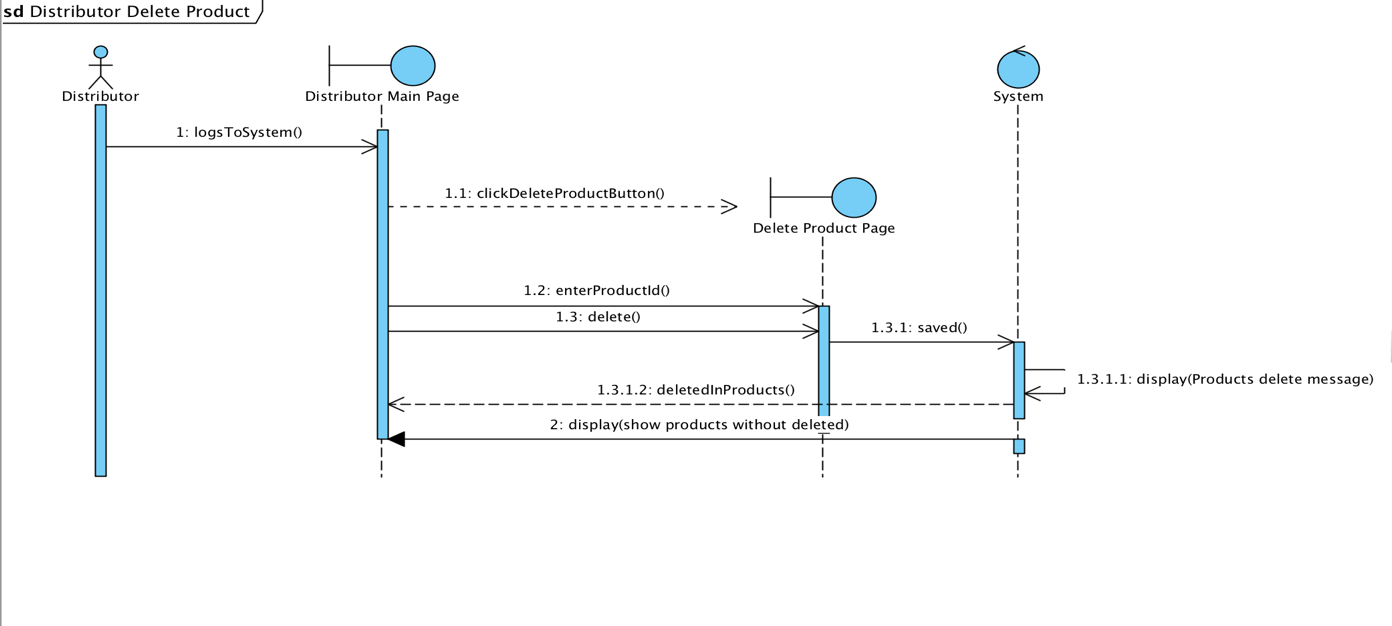
Distributor Dynamic Model



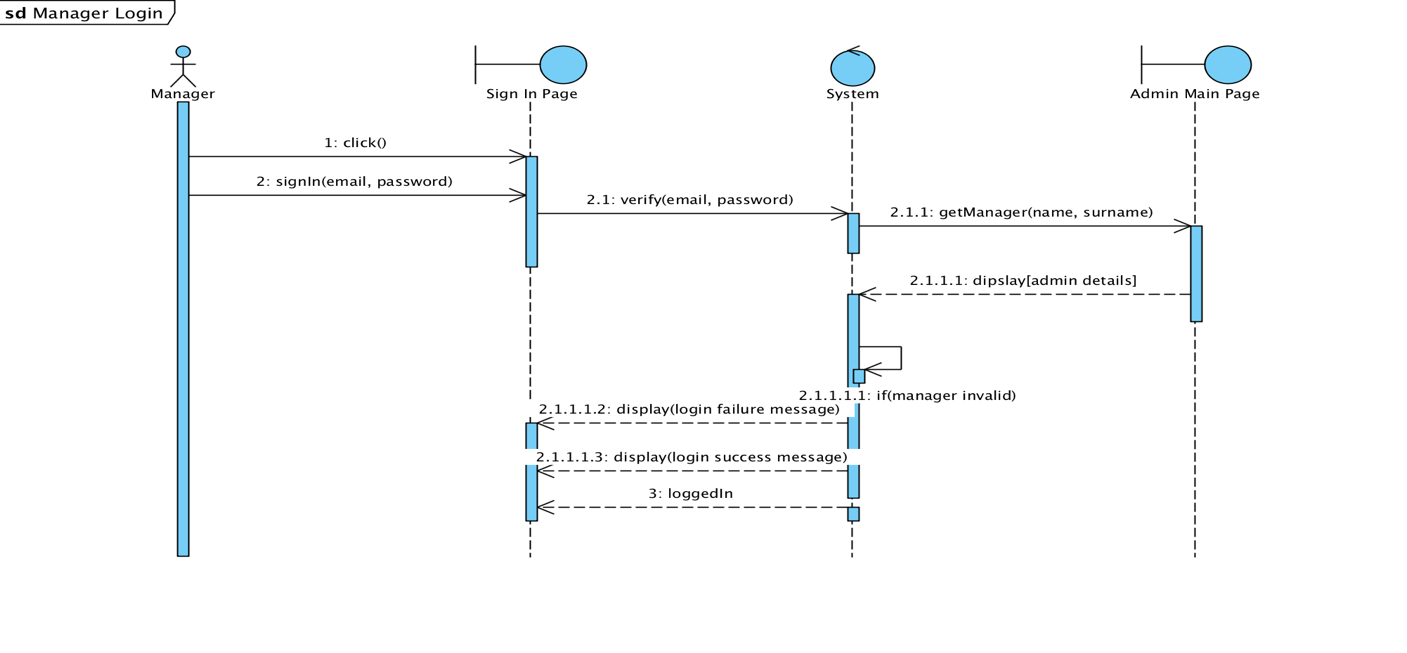


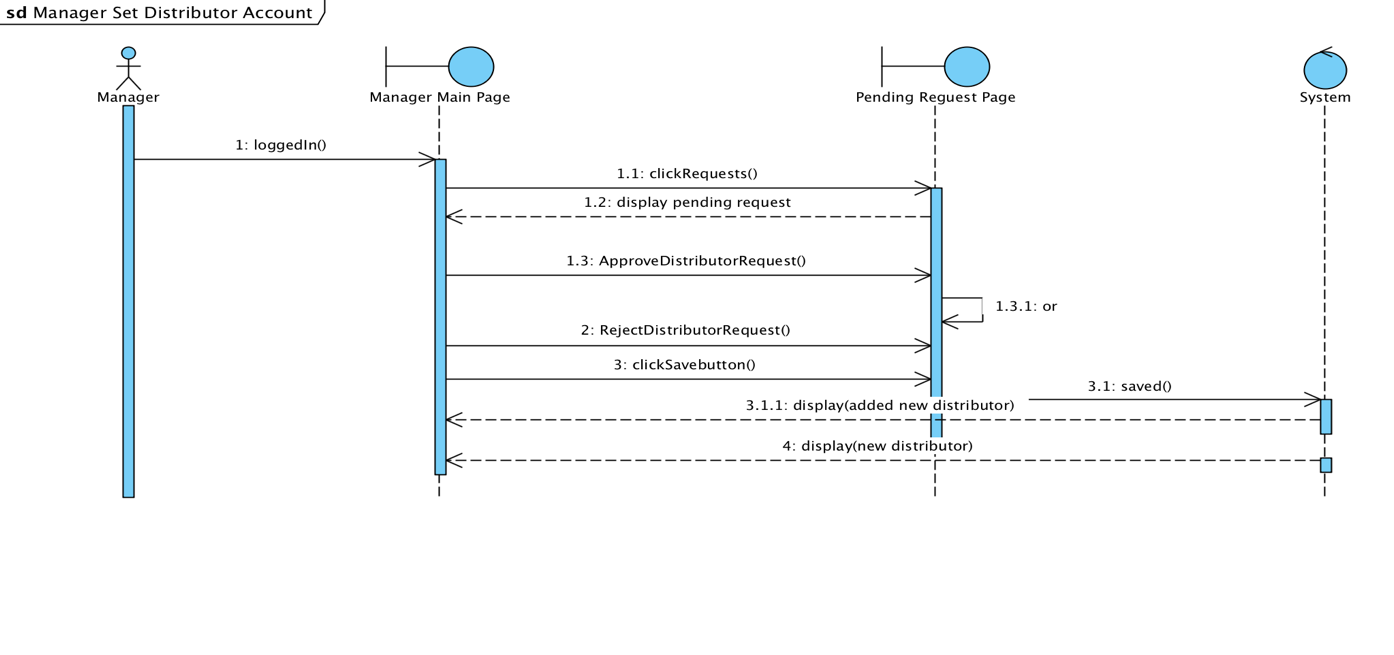


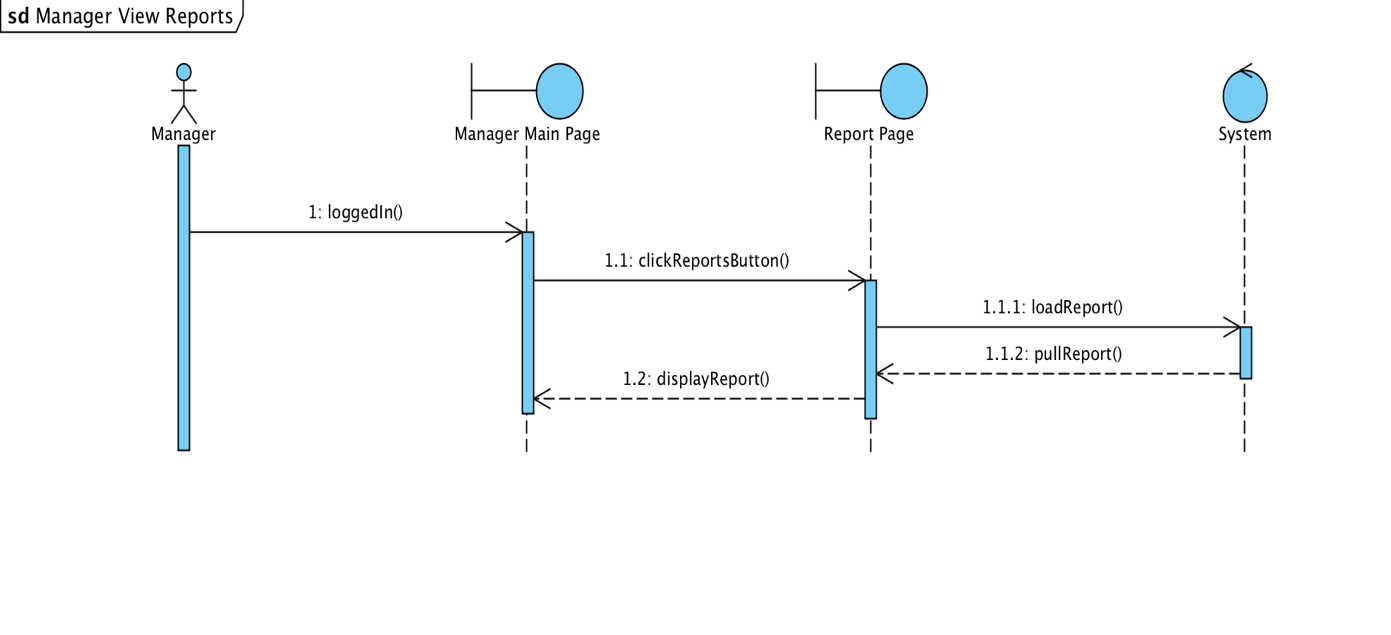


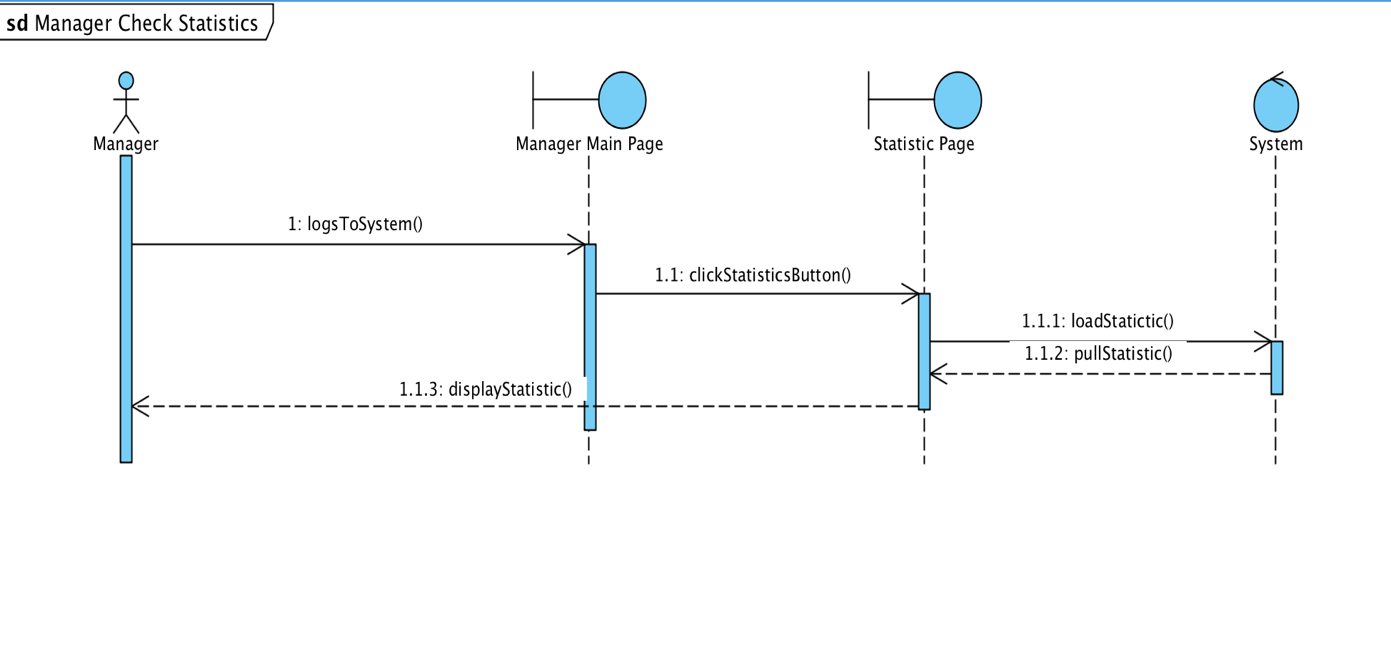


Manager Dynamic Models



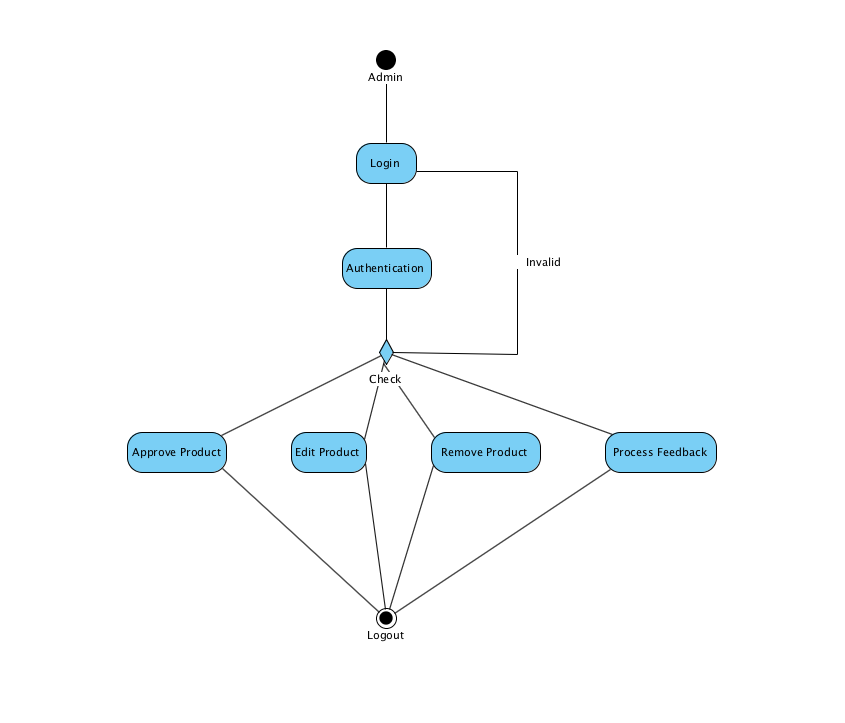




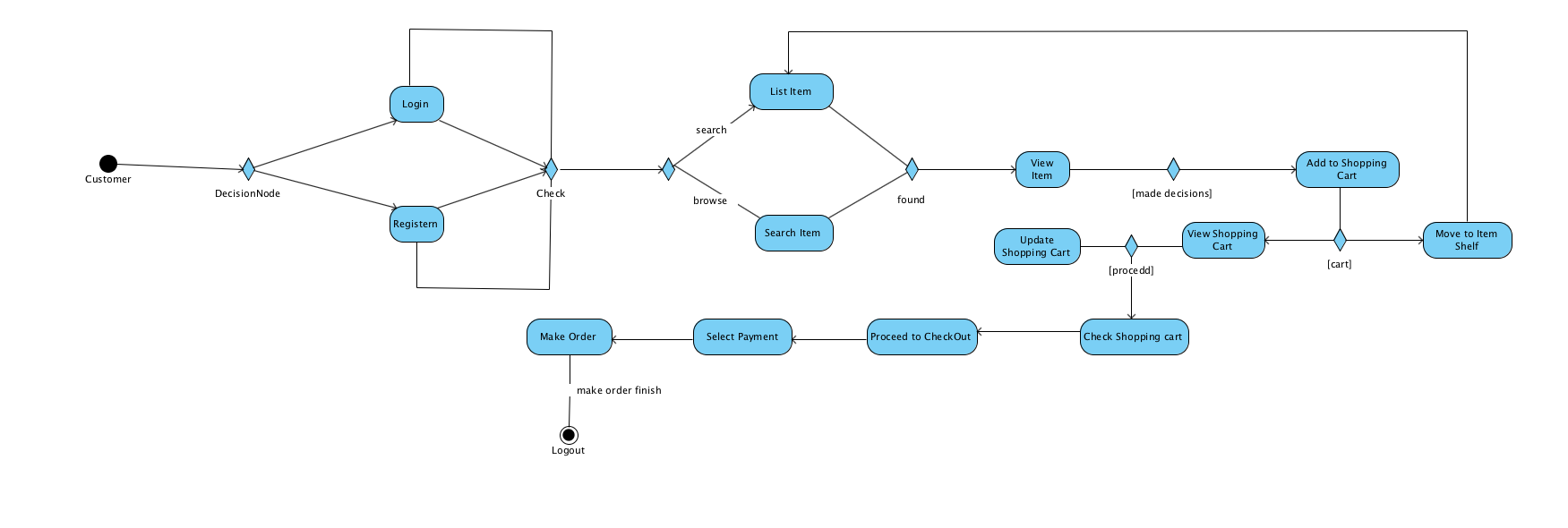


**Activity Diagrams**

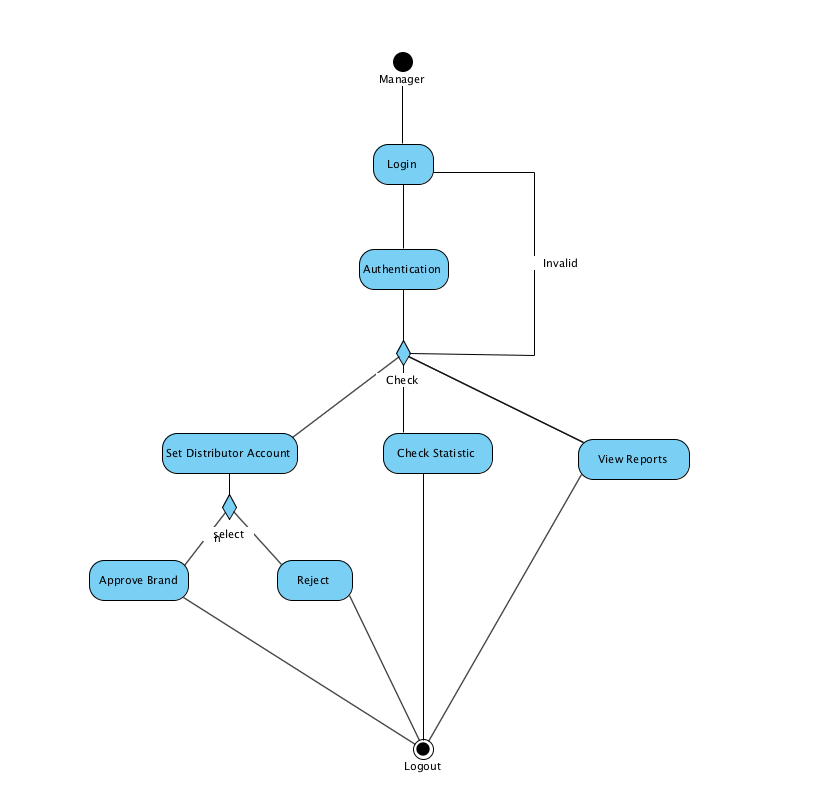
*Admin Activity Diagram*



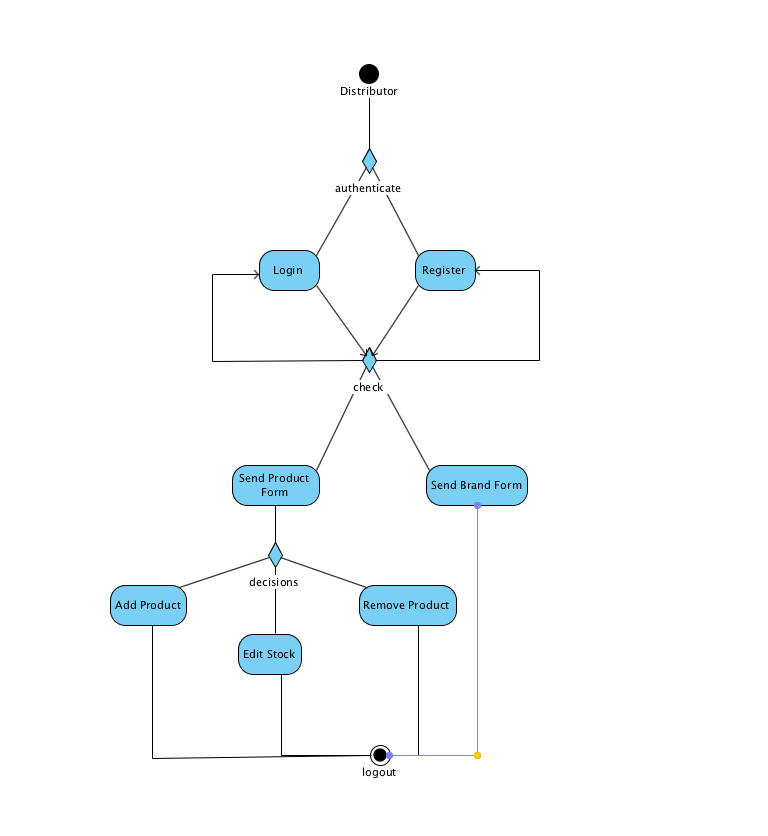
*Customer Activity Diagram*



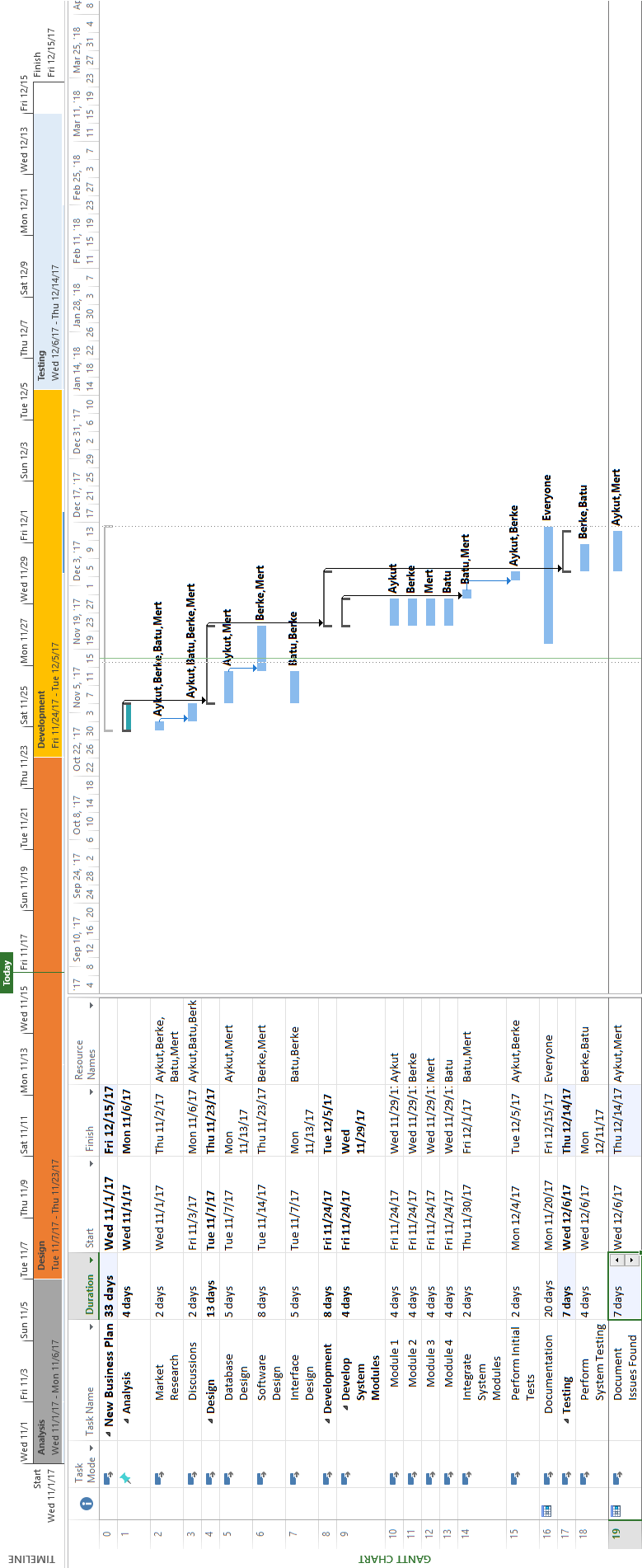
*Manager Activity Diagram*

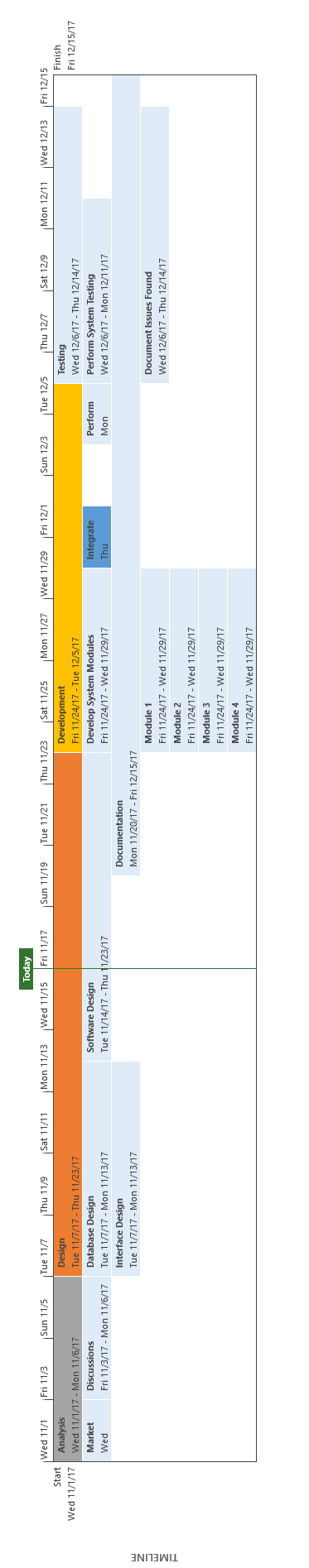


*Distributor Activity Diagram*



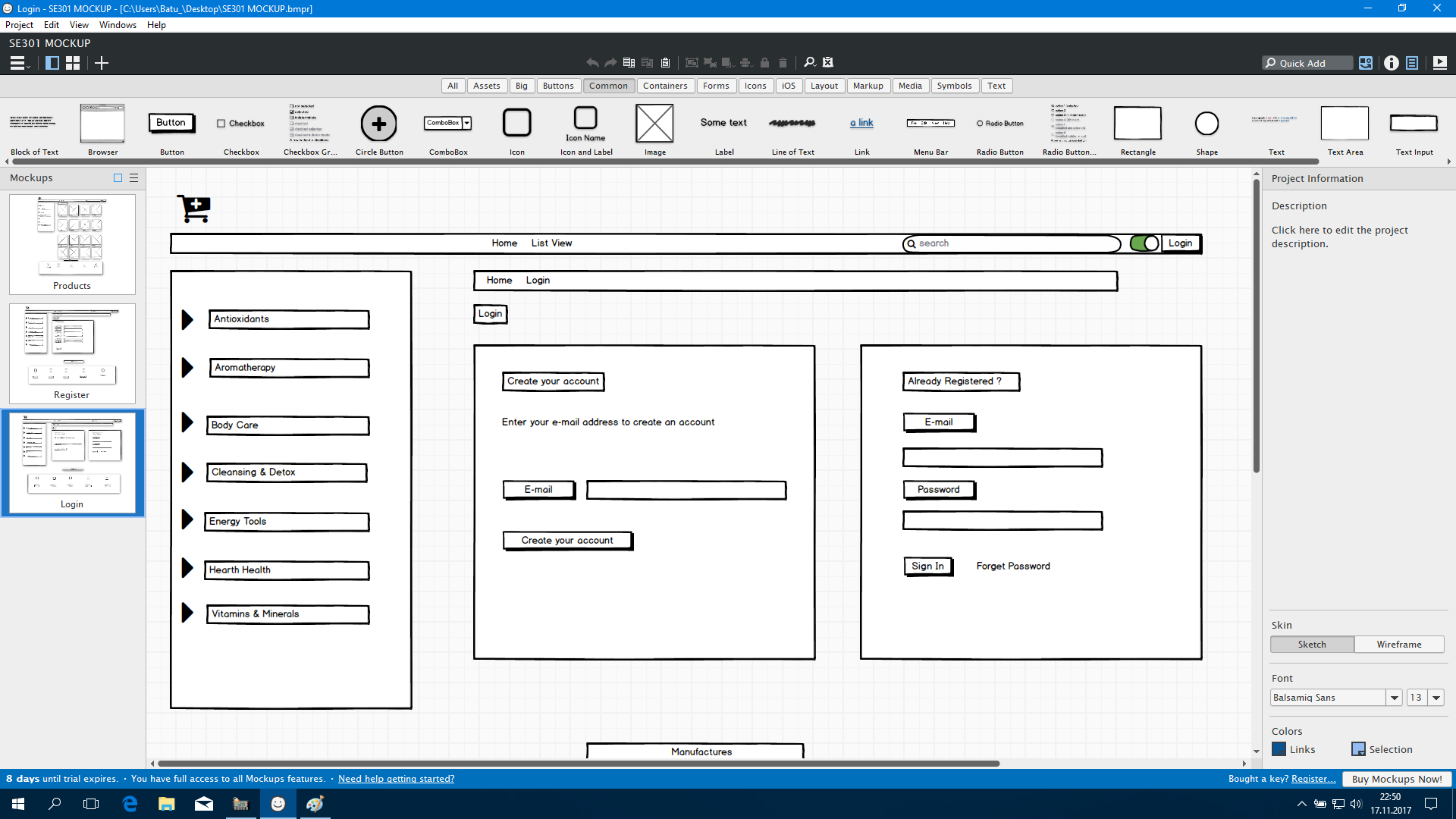
## Project Schedule



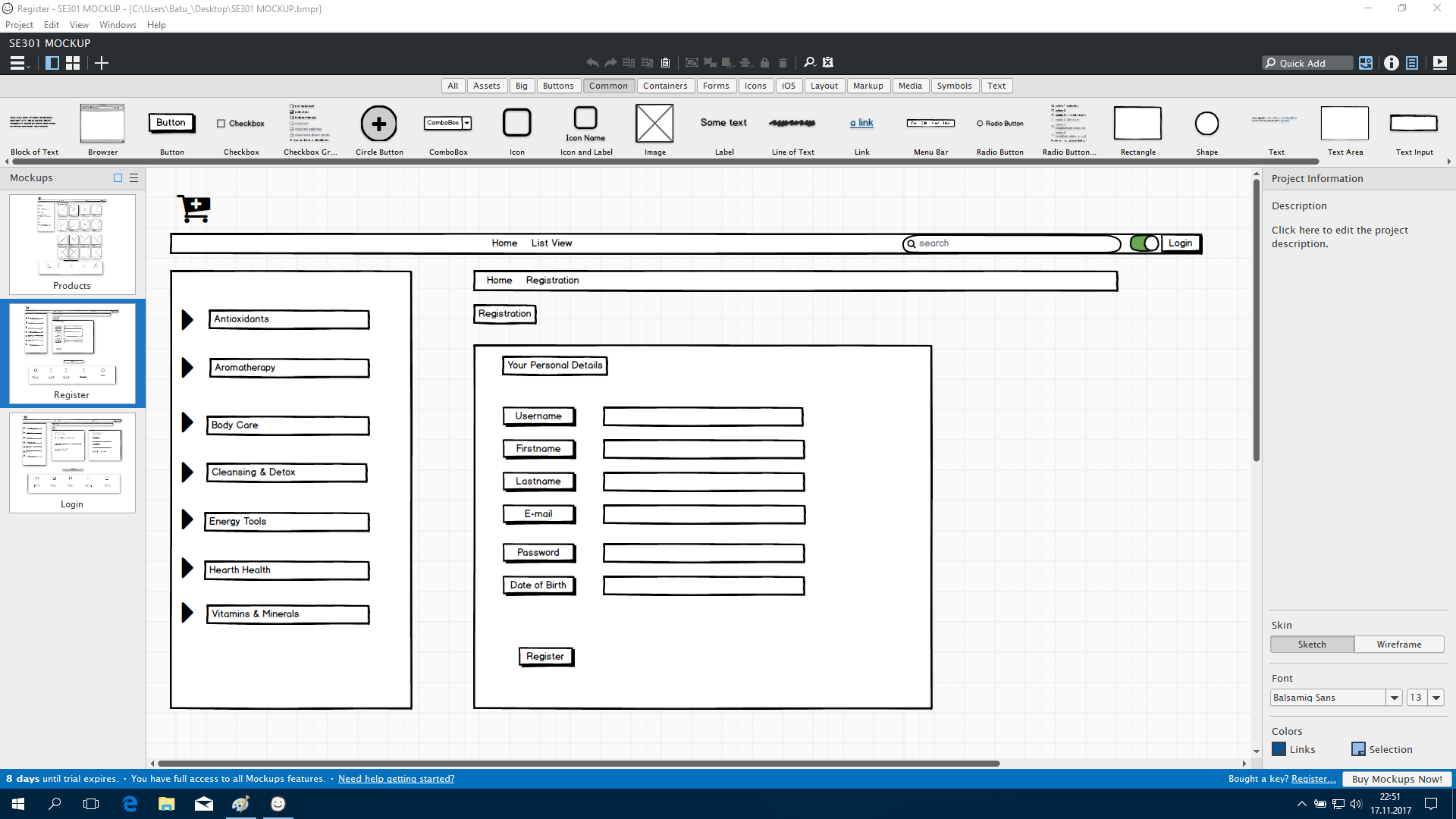


**User Interfaces**

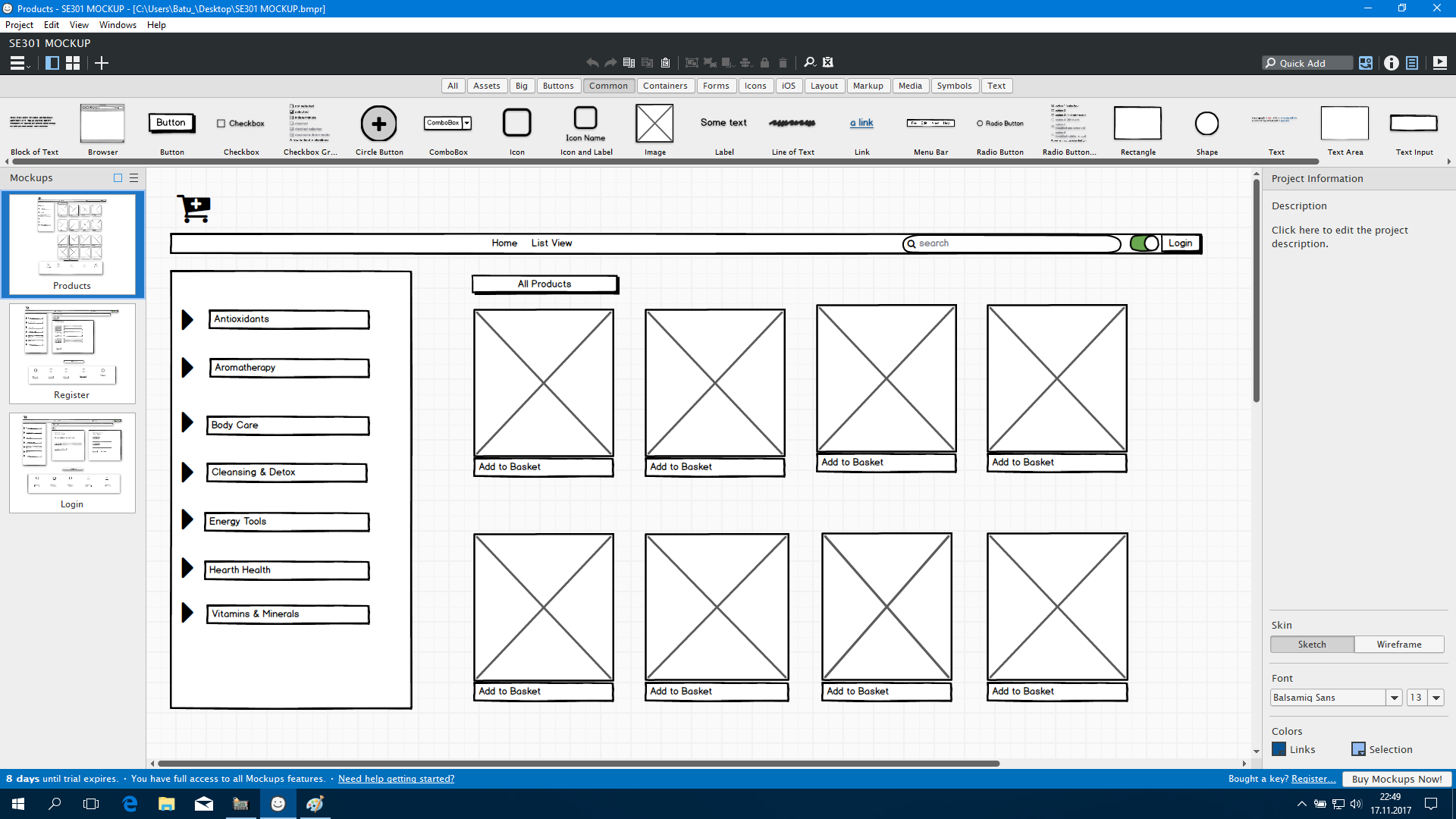
*Costumer Login*

**

*Costumer Register*

**

*Costumer View Product*

**

# Glossary

**Distributor:** Company and brand CEO who sell their products on the site

**Customer:** Person who processes in the system

**Manager:** The manager and owner of this system

**List Product:** It is where the user sees all the products.

**View Product Details:** It is where the user sees all the features of a product.

**Add to Shopping Cart:** It is where the user sees all the features of a product.

**Manage Shopping Cart:** The area where the user organizes the shopping cart.

**Move to Item Shelf:** The user can continue shopping with this option.

**Send Feedback:** The user can report the problems experienced during shopping with this option to the authorities.

**Manage Products:** The administrator deletes and updates the product attachments with this option

**Process Feedback:** The admin collects to follow the comments from the users.

**Set Brand Page Info:** Brand CEO enters the brand information when registering.

**Set Product:** Brand CEO enters the brand information when registering.

**View Reports:** The manager displays error reports that the users are relaying.

**Check Statistic:** The manager will see which brand and which product's sales ratio is high.

**Manage Products:** The manager manages product requests from the CEO.

# References

1. Bruegge B. & Dutoit A.H.. (2010). *Object-Oriented Software Engineering Using UML, Patterns, and Java*, Prentice Hall, 3rd ed.
2. <https://www.uml-diagrams.org/class-diagrams-overview.html>
3. <https://www.uml-diagrams.org/class-reference.html>
4. <https://www.ibm.com/developerworks/architecture/library/ar-analpat/> (IBM)