# Requirements Specification and

# Analysis Document

Version 1.0

Selin ATALAY

Metin Berk KARATAŞ

Nevzat Atilla ÖZDER

Alper Tunga ÇEVİK



**Table of Contents**

**Table of Contents**

**REQUIREMENTS ANALYSIS DOCUMENT[1] ...........................................................**

**3**

**1. INTRODUCTION ...........................................................................................**

**3**

**1.1.PURPOSE OF THE SYSTEM .............................................................................. 3**

**1.2.SCOPE OF THE SYSTEM ................................................................................. 3**

**1.3.OBJECTIVES AND SUCCESS CRITERIA OF THE PROJECT ................................................ 4**

**1.4.DEFINITIONS, ACRONYMS, AND ABBREVIATIONS ........................................................**

**4**

1. **5.OVERVIEW ..............................................................................................**

**5**

1. **CURRENT SYSTEM ........................................................................................ 6**
2. **PROPOSED SYSTEM .......................................................................................**

**7**

**3.1.OVERVIEW .............................................................................................. 7**

**3.2.FUNCTIONAL REQUIREMENTS ........................................................................... 8**

**3.3.NONFUNCTIONAL REQUIREMENTS ...................................................................... 9**

**3.4.SYSTEM MODEL ..........................................................................................**

**10**

**3.5.PROJECT SCHEDULE ....................................................................................**

**61**

**4.GLOSSARY ................................................................................................ 61**

**5. REFERENCES .............................................................................................** **63**

**Introduction**

* 1. **Purpose of the System**

E-commerce project is a sales site. The main goal of E-commerce is to sell the products of the distributor. It is a platform for costumers and distributors to be together. The customer can choose the appropriate one by filtering the products.

* 1. **Scope of the System**

As we have mentioned at purpose of the site, e-commerce is a sales site and its two main targets are sell products and buy products. The distributor is approved by admin at registration process.

At this sales sites, costumer can inspect products, can filter products and if the product is out of stock, customer can buy it. Costumer can add product in their basket. Customers and distributors can edit their profile. Distributors can offer their products for sale. They can add their products to the system , but They can't do that without an admin approval. When distributors want to add a product, they enter product information(name,brand,discount and such) and the product goes to admin approval. If the product is approved, it is offered for sale.

In this system, admin can products in the system for sale. No approval is required for this operation. Admin gives the approval of users who want to register to the system. If the user does not fit the system, admin will reject to register. Likewise, he/she approves or rejects the products. Admin approves or rejects distributor profile edits.

* 1. **Objectives and Success Criteria of the Project**

- To provide reliable, efficient, lossless data.

- Well association between platform and database design.

- The general design of system in order to have fast, efficient application.

- The system has implementations that are understandable, clear and efficient.

- The system should be used by the people who are related with the application.

- The system should guarantee every user’s protection of information of their data.

* 1. **Definitions, Acronyms, and Abbreviations**

- View is a visual representation of a model.

- GUI is graphical user interface.

- DB is short version of database term.

- Distributor is an actor an this sales site and approved by admin.

- Customer and visitor are an actor in a system.

* 1. **Overview**

- Rest of the RAD contains non-functional (includes usability, reliability, performance, supportability, implementation, interface, operational, packaging, and legal requirements) and functional requirements (includes high-level functionality of the system).

- System models are given. Scenarios are inside of system model section. Scenarios are telling us about details of functional requirements. Use case models, object model, dynamic model and user interface view (mockup) are the parts of system model section.

1. C**urrent System**

There is no current system for now.

**3.1 Functional Requirements**

In our system, one of our main functionalities is buying a product which is done by customer. As we mentioned earlier the customer is a user who is registered to the system. To buy a product he logs in and chooses from a list of products which one he wants to buy. He selects the number of products buying and then choose the payment style and options. He confirms the buying and he is displayed with a receipt. Our system will also support disabling buy. This happens when the customer has to change his idea and when he wants to cancel/update his request. For example, he wants to increase the number of product to buy for in the same product, so he changes his buy to edit this. The system also allows the customer to rate a product. This will be done by going to product’s page and voting it. There are ten points and the customer can give from one to ten points with ten being the best. Customer can also search for a product by name. We have included a search bar functionality and when the customer presses it he types in the name. Results matching the name are returned.

Our distributor is able to send requests to add their products to the system. He goes to the add product section and he is presented with a form. He fills in the fields which are required to register the product to the system. The fields include: product’s title, category, description, price, discount, images, product’s brand, number of products the distributor has. He then sends the request which shall be delivered to the admin’s inbox. Our system also allows the manager to be able to take down their restaurant from the system. To do this he has to send another request to the admin.

The admin’s main functionalities will be to add and disable products on the system. To add a product, he opens his inbox and he has some waiting requests sent by managers, he opens it and if he chooses to accept the request, he adds the product to the system’s database so customers can now make reservation to this restaurant. To disable product, he again opens inbox and there are some requests, he opens it and if he chooses to accept the request he disables it from the system and customers cannot buy to the product anymore.

**3.2 Nunfunctional Requirements**

**Usability**

Website should be easily available for all users. The admin panel must also be available on phone. The home page must be the same for all users. Each page must have the same body structure. Actors’ pages should be understandable.

**Reliability**

All members must have secure access to the website. Visitors should not do anything. Logging to the website should be provided with unique e-mails and passwords that are appropriate for password criteria.

**Performance**

Website should be shown in mobile phones,computers and tablets. Application should be running in more than one device simultaneously and access should be guaranteed. The website is going to be a dynamic content, so there should not be complicated queries in back end to not decrease performance.

**Supportability**

The system should be managed by admin. Developer will be responsible to provide continuance, compatibility and testability for the created program.

**Implementation**

Design of database will be implemented by using SQLite3

Django 2.1 which is a framework will be used both client and server side.

**Interface**

The system should not interact with any existing system. The system should be able to be used by a user. The user should be connected to the network to use the features of the system.

**Packaging**

Admin should install the system. Also, the system is an application so the websiteshould be uploaded on the server. All steps as a package are given within GitHub.

**Legal**

Project’s all contents are protected by the law of copyright.

**3.4.System Model**

Here, we show the high level functionalities of our system and how the users of our system interact with the system. For example, one of the functionalities my system is designed to support is to buy exam product. To do this, firstly the user must login the site. The user subsequently enters website after this he finds through product and then he clicks product and choose number of product. When he presses give order button the system opens payment page and wants him to enter payment information from user. When the payment information enters by customer the system check the payment information and then the product preparing for shipping by distributor. The customer also choose shipping address and the distributor knows that where s/he sends the product.

Scenario 1:

Scenario Name: Login

Participant actor instances: Berk: Customer, Selin: Distributor, Atilla: Admin

The flow of events:

1. When Atilla open the e-commerce site, system starts and brings login screen
2. The username and password fields are filled. He presses the login button.
3. Both password and username match, there is no error so logging in process is successfully completed.
4. Atilla presses logout button and logout.

Scenario 1 Extensions:

2a. Atilla inputs wrong username or password so logging in the process fails. He is asked to recheck the information he provided.

2b. Atilla inputs wrong username or password so logging in the process fails. He is asked to recheck the information he provided. He realized he has forgotten his password. The system guides him to reset the password by sending a reset link to his attached email address.

2c.When Atilla tries to enter his profile page he recognize that he forgot his password. He presses forgot password button and the system opens the forget password form to him.

Scenario 2:

Scenario Name: Register

Participant actor instances: Alper: Visitor

The flow of events:

1. Alper wants to buy some products from using e-commerce website. For doing this, firstly he has to sign up the application. When he opening the application the app shows him login and register button. When he chose the register button, system redirect him to the register page.
2. The register page consists of required information. So Alper fills the username, email, first name, last name and password information about himself. He must select distributor or customer options (If he is a customer, he select customer option, otherwise he chooses distributor option. If he selected distributor option, the request will send to admin).
3. After pressing the register button if any error occurs the system give him an information about it and want to fix them.
4. After fixing issues, he logs in the system.

Scenario Register Extensions:

3. Alper may be enter short username, password or he may be enter a username, or an email already taken.

Scenario 3:

Scenario name: Delete Account

Participant actor instances: Berk: Customer, Selin: Distributor

Flow of events:

1. Berk has decided that he wants to delete his account. He logs in using his username and password.
2. He goes to view his profile where he can make changes and updates.
3. He located the delete account button and taps it.
4. Finally, his account become inactive and he must login to activate again.
5. He logs out successfully.

Scenario Extensions:

1.1. Berk enters the wrong username or password. He is asked to re-enter the password.

4.1. Account is not deleted due to system error. He is asked to try again later.

Scenario 4:

Scenario name: Update Profile

Participant actor instances: Berk: Customer, Selin: Distributor

Flow of events:

1. Berk has some changes on his personal information (email, gender, first name and last name) and he wants to update this changes on the e-commerce web site.
2. He opening web site and pressing profile.
3. He presses change profile button.
4. When he changes his information he presses save button.

Scenario Extensions:

4.1 If there are a problem about changed information the system return a fail message and give feedback to user like invalid password.

Scenario 5:

Scenario Name: Distributor Approve

Participant actor instances: Atilla: Admin

Flow of events:

1. Atilla logs in the system.
2. Clicks distributor pending list button and she sees the distributors waiting for approval on the opened page.
3. She clicks the 'approve' button next to the name of the appropriate person.
4. Distributor is became approved.

Scenario 6:

Scenario Name: Reject Distributor

Participant actor instances: Atilla: Admin

Flow of events:

1. Atilla logs in the system.
2. Clicks disributor pending list button and she sees the distributos waiting for approval on the opened page.
3. He clicks the ‘reject’ button next to the name of the unsuitable person.

Scenario 7:

Scenario Name: View Distributor Pending List

Participant actor instances: Atilla: Admin

Flow of events:

1. Atilla logs in the system.
2. Clicks distributor pending list button.
3. He sees all waiting for approval.

Scenario 8:

Scenario Name: View Product Detail

Participant actor instances: Atilla: Admin, Berk: Customer, Selin: Distributor

Flow of events:

1. He logs in the system.
2. Atilla clicks on the product.
3. He sees product’s price, the amount of stock, the score, discount and the distributor of the product if any.

Scenario 9:

Scenario Name: Add Product

Participant actor instances: Atilla: Admin

Flow of events:

1. He logs in the system.
2. Atilla clicks add product button.
3. He enters product’s tittle, category, brand, description, price, discount, stock and images.
4. Atilla clicks submit button (If distributor registers this product, she waits for admin approval. If admin approve this product, product is become avaible).

Scenario 10:

Scenario Name: Approve Product

Participant actor instances: Atilla: Admin

Flow of events:

1. Atilla logs in the system.
2. Clicks product pending list button and she sees the products waiting for approval on the opened page.
3. He clicks the 'approve' button next to the name of the appropriate product.
4. Product is avaible.

Scenario 11:

Scenario Name: Reject Product

Participant actor instances: Atilla: Admin

Flow of events:

1. Atilla logs in the system.
2. Clicks product pending list button and he sees the products waiting for approval on the opened page.
3. He clicks the ‘reject’ button next to the name of the unsuitable product then he fill a text about why this product was rejected. After clicking ‘send’ button the message is emailed to the corresponding distributor.

Scenario 12:

Scenario Name: Monitor Product Pending List

Participant actor instances: Atilla: Admin

Flow of events:

1. Atilla logs in the system.
2. Clicks product pending list button.
3. He sees all waiting for approval.

Scenario 13:

Scenario name: Update Product

Participant actor instances: Atilla: Admin, Selin: Distributor

Flow of events:

1. Selin logs in the system.
2. She clicks product list.
3. She change product’s title, category, brand, description, price, discount, stock, weight or images.
4. She clicks update button.
5. If she enters something invalid, the system will display an error message.
6. Product is updated.

Scenario 14:

Scenario Name: Change Password

Participant Actor: Berk: Customer, Selin: Distributor

Flow of events:

1. Berk wants to change his own e-commerce account site password
2. Berk sign in his own e-commerce account and presses profile button
3. Then he chooses change password option and the system opens him change password screen.
4. He enters old password, new password and new password again fields and presses save button
5. In the end system save the new password to the database and returns to him a message like password change correctly.

Scenario Change Password Extensions:

4. He may enter invalid password for change or he may enter his old password wrong.

2. He may fail to enter his own username and password correctly.

Scenario 15:

Scenario Name: Monitor User Details

Participant Actor: Berk: Customer, Selin: Distributor

Flow of events:

1. Berk wants to check his information on the e-commerce site and if there is a problem he wants to change it
2. He enters e-commerce site with his login information
3. He finds show profile button and presses it
4. The system opens profile page to him

Scenario Monitor User Details Extension:

2. He may fail to enter his own username and password correctly.

Scenario 16:

Scenario Name: Rate Product

Participant Actor: Berk: Customer

Flow of events:

1. Berk bought a bag from e-commerce web site and he likes it
2. For support the distributor of bag he wants to rate the bag and she opens web site.
3. He sign in the site
4. He chooses the bag from the my purchases tab and the system redirect her to bag’s own page.
5. Berk vote the bag and he logs out from the site.

Scenario Rate Product Extensions:

3. She may fail to enter his own username and password correctly.

5. She can vote just once , if she vote the bag earlier she can’t repeat it again

Scenario 17:

Scenario Name: Image Upload

Participant Actor: Berk: Customer

1. Berk wants to upload his profile picture on the site
2. He opens the web site main page and login the site
3. He finds profile button and clicks it
4. When the profile page opened the chooses image button for update image
5. Then he chooses new picture which he wants as his profile picture and clicks save button.

Scenario 18:

Scenario Name: Monitor Products for Waiting Deletion

Participant actors: Admin: Atilla

Flow of events:

1. When a distributor wants to delete his/her own product on the site he/she must sends a request to Admin
2. When Admin wants to give answer to this deletion request and opens his own admin account on web site
3. Admin chooses waiting request tab and he sees all waiting requests
4. He chooses one of waiting request and then he decides to approve or reject that request and presses the button according to this.

Scenario Monitor Products for Waiting Deletion Extensions:

2. He may fail to enter his own username and password correctly.

Scenario 19:

Scenario Name: Filter Product

Participant actor instances: Berk: Customer

Flow of events:

1. Berk enters web site, Berk uses search bar on the top of the screen to and type the name or brand of the product and clicks on search button.
2. A. Products that have search key are listed. Berk click on the product for more information.

B. Products that have search key are listed. Berk click on add cart and product will appear on Berk’s cart. Login screen appears.

C. The search key not found. Berk enters another key to search and click on try again button to search.

Scenario 20:

Scenario Name: Cart Monitoring

Participant actor instances: Berk: Customer

Flow of events:

1. Berk enters on web site login to the system. Click on cart symbol cart which opens cart page.
2. 2a. Berk sees products that he added to the cart earlier. He change quantity one product from quantity “+” button.
3. 2b. Berk has no item on cart, so web page displays empty cart symbol.

Scenario 21:

Scenario Name: Adding Product to Cart

Participant actor instances: Berk: Customer

Flow of events:

1. Berk enters on web site login to the system.
2. 2a. Search an product and click on add cart button.
3. 2b. Select an product from homepage and click on product it leads him to product description page. He clicks on add cart to add item to cart.
4. 2c. Ali finds product from categories menu and click on add cart button to add it to cart.

Scenario 22:

Scenario Name: Deleting Product from Cart

Participant actor instances: Berk: Customer

Flow of events:

* 1. Berk enters on web site login to the system.
  2. He clicks on cart symbol to open cart.
  3. He clicks “X” button to remove product from cart. Item successfully removed from cart.

Scenario 23:

Scenario Name: Order checkout

Participant actor instances: Berk: Customer

Flow of events:

1. Berk enters on web site login to the system.
2. He clicks on cart and sees products he already added to cart.
3. He clicks on checkout button on the left.
4. He selects address information that he already filled up earlier.
5. He complete order by clicking confirm.
6. He monitor complete order symbol and order no.

Scenario 24:

Scenario Name: Order Status Check

Participant actor instances: Berk: Customer

Flow of events:

1. Berk enters on web site login to the system.
2. He opens on his account dashboard.
3. He clicks on orders tab.
4. He monitor situation of his order with its details and products in it.

Scenario 25:

Scenario Name: Address Creation

Participant actor instances: Berk: Customer

The flow of events:

1. Berk enters on web site login to the system.
2. 2a. Berk want to checkout an order opens cart and clicks on checkout button.
3. 2b. Berk opens his account’s dashboard.
4. Berk clicks on address informations addition button.

Scenario Address Creation Extensions

4. Fills the informations about country, city open address and mail code and finish with save button.

5a. Clicks on set for delivery address and bill address buttons to continue.

Scenario 26:

Scenario Name: Address Edit

Participant actor instances: Berk: Customer

Flow of events:

1. Berk enters on web site login to the system.
2. 2a. Berk want to checkout an order opens cart and clicks on checkout button.
3. 2b. Berk opens his account’s dashboard.
4. Berk clicks on address informations edition button.
5. Edits the informations about country, city open address and mail code and finish with save button.

**Use Case 1:**

|  |
| --- |
| *Use case name:* Login (Low Priority) |
| *Participant actors:* Customer, Admin or Distributor |
| *Flow of events:*   1. User first enters to the application. 2. The system presents the “login form” to the user. 3. User enters username or email into username text field on the screen, also enters own password into password text field on the screen. Lastly, the user sends a request to the system by using login button on the screen to be logged in. 4. The system checks the username and password so that the system allows the user to login. Then application redirects to home page. |
| *Entry Condition:* The user enters to login screen. |
| *Exit Condition:* The user is logged in, or user has received an explanation indicating why he/she could not login.  Forget password screen opens onto screen. |

**Use Case 2:**

|  |
| --- |
| *Use case name:* Register (Low Priority) |
| *Participant actors:* Initiated by visitor |
| *Flow of events:*   1. The person is open the application and see the register button and clicks it. 2. The application opens the register page. 3. The person is filling username, e-mail, password, confirm password, gender fields about himself/herself. After that he choose his role for this application (Role of user must be customer or distributor) 4. The person presses the register button and it will send a request to the system. |
| *Entry Condition:* The user enters to register screen. |
| *Exit Condition:* The user is registered as a customer or distributor, if any error occurs the system has send an error message about why he/she couldn’t register (can be invalid email or username in already use) |

**Use Case 3:**

|  |
| --- |
| *Use case name:* Update Profile (Low Priority) |
| *Participant actors:* Initiated by customer, distributor |
| *Flow of events:*   1. The user sends a request to the system to edit his informations that are firstname, lastname, email or gender. 2. The user edits the fields and submits the form by pressing the update button. 3. The application updates his informations. |
| *Entry Condition:* The user is logged in to the application. He chooses edit profile button on the profile screen. |
| *Exit Condition:* Information about the user is updated or the customer receive an explanation indicating why he could not update his profile. |

**Use Case 4:**

|  |
| --- |
| *Use case name: Delete Account* (Low Priority) |
| *Participant actors:* Initiated by customer, distributor |
| *Flow of events:*   1. The user opens profile page. 2. He presses ‘delete account’ button. 3. The system ask confirmation, if user confirm deletion the system validates and disable the user’s account. |
| *Entry Condition:* The user press the ‘delete account’ button. |
| *Exit Condition:* Account has been disabled successfully or system returns a failure message. |

**Use Case 5:**

Use Case Name: Distributor Approve

Participant actor instances: Admin

Flow of events:

1. Click distributor pending list button.
2. Sees the distributors waiting for approval on the opened page.
3. Selects the contact to be approved.
4. Clicks the approve button next to the name of the appropriate person.
5. Distributor is avaible.

Entry Conditions

1. Admin logged in the system.
2. Admin displays the secreen that list of pending.

Exit Conditions

1. Distributor is added to the system.

Exceptional Case

1. If administrator does not select any request of distributor, and click approve system displays a warning message, like "You should choose a request".

**Use Case 6 :**

Use Case Name: Distributor Reject

Participant actor instances: Admin

Flow of events:

1. Clicks distributor pending list button.
2. Sees the distributors waiting for approval on the openned page.
3. Selects the contact to be rejected.
4. Clicks the reject button.

Entry Conditions

1. Admin logged in the system.
2. Admin display the screen that list of pending.

Exit Condition

1. Distributor is not added to the system.

Exceptional Case

1. If administrator does not select any request of distributor, and click reject system displays a warning message, like "You should choose a request".

**Use Case**

Use Case Name: Add Product

Parcipant actor instances: Distributor

Flow of events:

1. Clicks add product button.
2. Enters product’s tittle,cateogary,brand,description, price, discount, stock, weight and images.
3. Clicks submit button.
4. Waits admin approval.

Entry Conditions

1. Admin logged in the system.
2. Admin display the screen that list of product.

Exit Condition

1. Product is created.

Exeptional Case

1. If administrator does not select any request of product, and click reject system displays a warning message, like "You should choose a request".

**Use Case 8**

Use Case Name: Approve Product

Participant actor instances: Admin

Flow of events:

1. Clicks product pending list button
2. Sees the products waiting for approval.
3. Clicks approve button.
4. Product is avaible.

Entry Conditions

1. Admin logged in the system.
2. Admin displays the secreen that list of pending.

Exit Conditions

1. Product added to the system.

Exponential case

1. If administrator does not select any request of product, and click approve system displays a warning message, like "You should choose a request".

**Use Case 9**

Use Case Name: Reject Product

Participant actor instances: Admin

Flow of events:

1. Clicks product pending list button.
2. Sees the products waiting for approval.
3. Clicks reject button.
4. Product is avaible.

Entry Conditions

1. Admin logged in the system.
2. Admin displays the secreen that list of pending.

Exit Conditions

1. Product rejected to the system.

Exponential case

1. “If administrator does not select any request of product, and click reject system displays a warning message, like "You should choose a request".

**Use Case 10**

Use Case Name: Product Update

Participant actor instances: Admin, Distributor

Flow of events:

1. Clicks product list.
2. Change product’s title, category, brand, description, price, discount, stock, weight or images.
3. Clicks update button.

Entry Conditions

1. They logged in the system.
2. Displays the secreen that list of product.

Exit Conditions

1. Products update to the system.

Exponential Case

1. If enters something invalid, the system will display an error message.

**Use Case 11:**

Use Case Name: Filter Product

Participant actor instances: Customer

The flow of events:

1. Customer type the name or brand of the product and click search button.

2. Customer found the product and can view details without being logged in to the system.[Product not found]

Exit Condition:

* Customer found the product successfully.

Exceptional Cases:

[Product not found 404]: If the product not found customer will see a page that indicates unsuccessful search and a button which leads different search.

**Use Case 12**

Use Case Name: Cart Operations

Participant actor instances: Customer

Flow of events:

1. Customer click on cart symbol on left top of the page.

2. Customer can monitor all the products in his/her cart can change quantity of any of them[Product out in stock]. Customer will also be able to remove any of these items from cart. Customer will also able to go directly detailed description of the product. [No items in cart]

Entry Condition:

* Customer logged in to the system.

Exit Condition:

* Customer monitor his/her cart items.
* Customer able to remove any item in cart.
* Customer able to add more quantity to any of product.

Exceptional Cases:

[No items in cart]: If there is no product in cart empty cart symbol will appear.

[Product out in stock]: If there is no more product in stocks to add the quantity of a product an error will appear to user.

**Use Case 13**

Use Case Name: Add a Product To Cart

Participant actor instances: Customer

Flow of events:

1a. Customer able to add any product to cart from any search result or any of the categories or from homepage. For all them he/she just click on add cart button.[Product out in stock]

1b. Customer able to add any product to cart from product’s detailed view by clicking add cart button.[Product out in stock]

1. Customer will see increase on cart symbol.

Entry Condition:

* Customer logged in to the system.

Exit Condition:

* Customer will see the product he/she added on cart.

Exceptional Cases:

[Product out in stock]:If there is no more product in stocks to add the quantity of a product an error will appear to user.

**Use Case 14:**

Use Case Name: Order Checkout

Participant actor instances: Customer

Flow of events:

1. Customer will click on checkout button.

2. Customer will check address and click on confirm order button. [Address not selected][Address not added]

3. Order no and confirmed order symbol will appear.

Entry Condition:

* Customer logged in to the system.
* Customer added products to the cart.
* Customer opened the cart.

Exit Condition:

* Successfully finishing order checkout.

Exceptional Cases:

[Address not selected]:If addresses not selected customer has to select any of addresses for both delivery address and bill address by clicking on set for delivery address or set for bill address buttons.

[Address not added]:If customer didn’t add any address earlier, he/she will have to add a address by using address list at the current page.

**Use Case 15**

Use Case Name: Order Status Check

Participant actor instances: Customer

Flow of events:

1. Customer will click on profile picture on the top right corner and will open user dashboard.

2. He/she will click on order details in left side menu.

3. He/she will able to see orders in ascending order of date they created.

Entry Condition:

* Customer logged in to the system.
* Customer successfully order anything.

Exit Condition:

* Customer monitor previous orders status as list.

**Use Case 16:**

Use Case Name: Address Operations

Participant actor instances: Customer

The flow of events:

1. Customer will see list of its recorded address as list.

2a. Customer will able to add any address by clicking “+Address” button and fills the informations that needed to be fulfilled.

2b. Customer will able to edit any address by clicking pen symbol on the left of any of these addresses and changing the inputs that come after opening.[Address not added]

2c. Customer will able to delete any address by clicking “X” symbol of any of these addresses.[Address not added]

Entry Condition:

* Customer logged in to the system.
* Customer opened his dashboard.

Exit Condition:

* Customer successfully added or successfully edited or successfully deleted.

Exceptional Cases:

[Address not added]:If customer didn’t add any address earlier, he/she will have to add a address by using address list at the current page.

**Use Case 17**

Use Case Name: Password Change

Participant actor instances: Customer, Distributor

Flow of events:

1. Customer wants to change his/her own e-commerce account site password

2. Customer/Distributor sign in his/her own e-commerce account and presses profile button

3. Customer/Distributor chooses change password option and the system opens him change password screen.

4. The user enters old password, new password and new password again fields and presses save button

Entry Condition:

* Customer logged in to the system.

Exit Condition:

* Customer/Distributor changes his/her own password.

Exceptional Cases:

* The system returns an error about invalid password .

**Use Case 18:**

Use Case Name: Monitoring User Details

Participant actor instances: Customer , Distributor

The flow of events:

1. Customer wants to monitor his/her own e-commerce account page

2. Customer/Distributor sign in his/her own e-commerce account and presses profile button

3. After checking the details of his/her he/she logs out from the system.

Entry Condition:

* Customer logged in to the system.

Exit Condition:

* User details monitors on screen

Exceptional Cases:

The user can enters user information wrong.

**Use Case 19:**

Use Case Name: Rate Product

Participant actor instances: Customer

The flow of events:

1. Customer wants to rate a product on the e-commerce site

2. Customer sign in his/her own e-commerce account and presses profile button

3. Customer chooses the product which is he/she wants to rate.

Entry Condition:

* Customer logged in to the system.
* Customer must choose a product and must to open the product’s own page

Exit Condition:

* User rates the product

Exceptional Cases:

The user can enters user information wrong.

The user can rate just once. If he/she tries second time he/she can’t.

**Use Case 20:**

Use Case Name: Image Upload

Participant actor instances: Customer

The flow of events:

1. Customer wants to change profile image on the e-commerce site

2. Customer sign in his/her own e-commerce account and presses profile button

3. Customer profile button.

4. Customer finds image button and presses it and the system opens the image change page to him.

Entry Condition:

* Customer logged in to the system.
* Customer must opens his/her own profile page

Exit Condition:

* User changes profile picture .

Exceptional Cases:

The user can enters user information wrong.

**Use Case 21:**

Use Case Name: Monitoring Products for Waiting Deletion

Participant actor instances: Admin

The flow of events:

1. When a distributor wants to delete his/her own product on the site he/she must sends a request to Admin

2. When Admin wants to give answer to this deletion request and opens his own admin account on web site

3. Admin chooses waiting request tab and he sees all waiting requests

4. He chooses one of waiting request and then he decides to approve or reject that request and presses the button according to this.

Entry Condition:

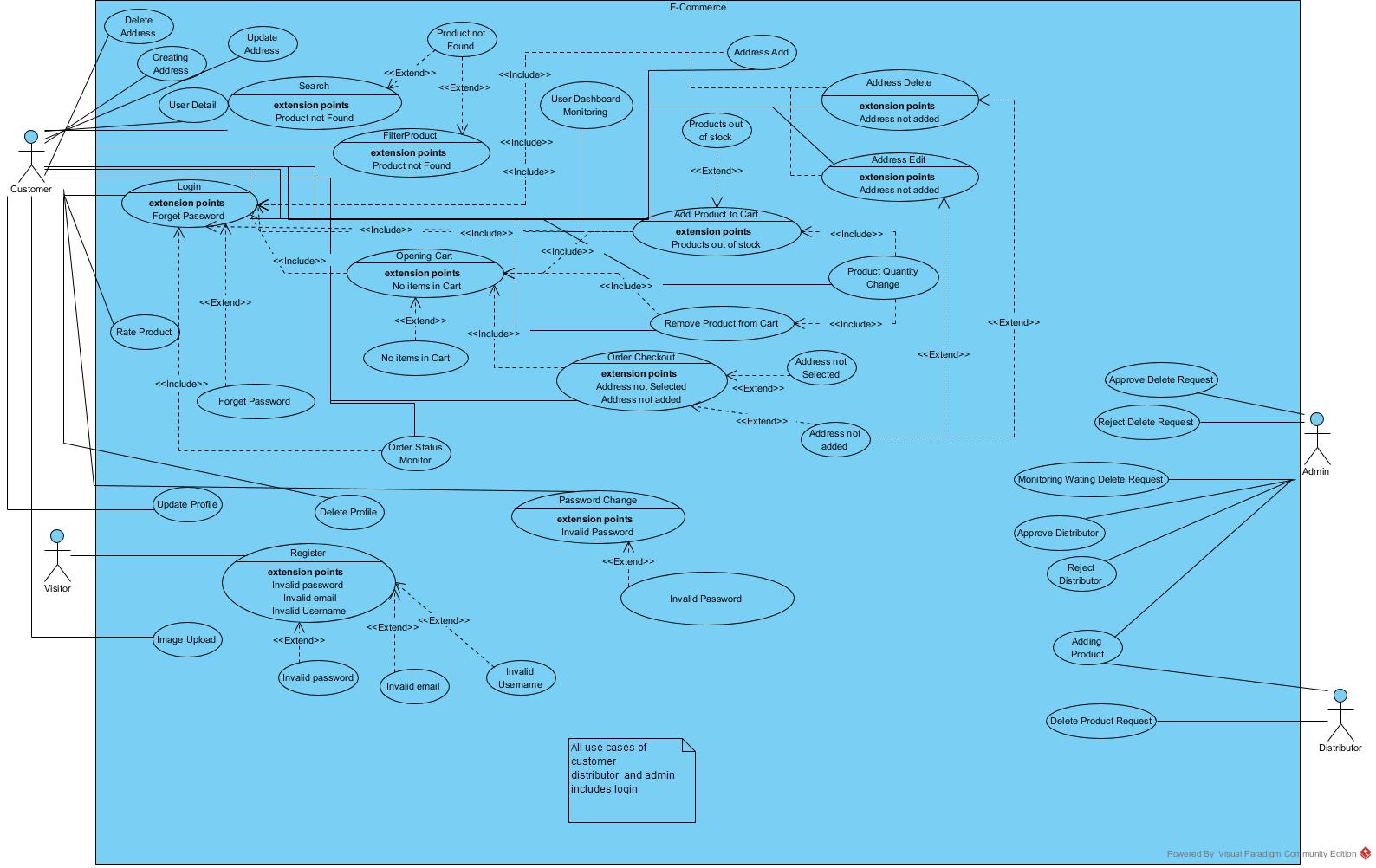
* Customer logged in to the system.
* Customer must opens waiting deletion request page

Exit Condition:

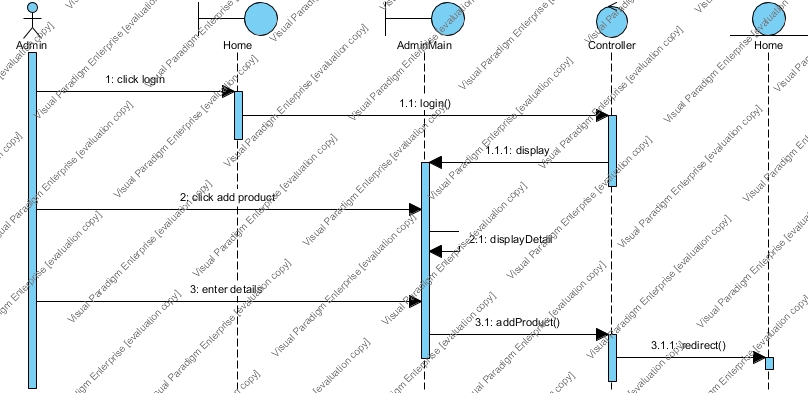
* The waiting request answered as approve or reject

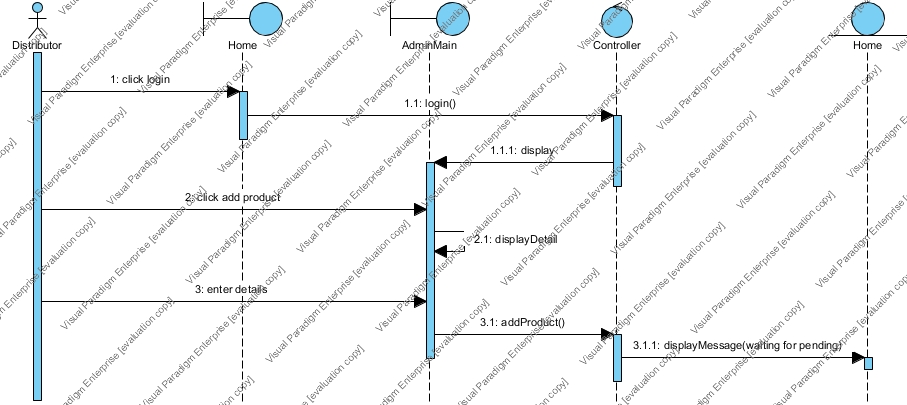
Exceptional Cases:

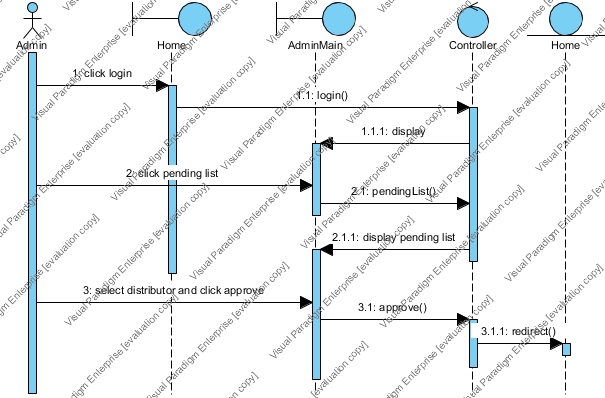
The user can enters user information wrong.

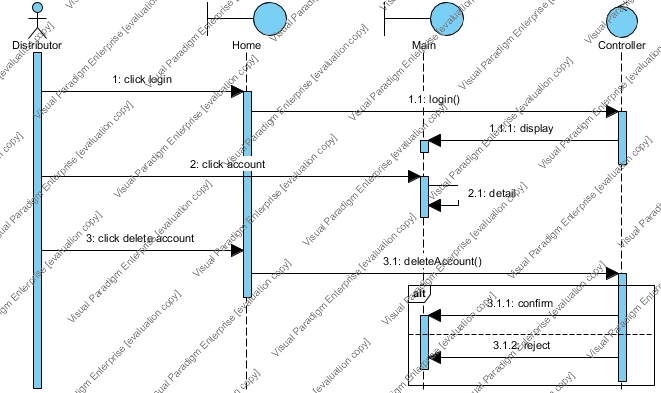


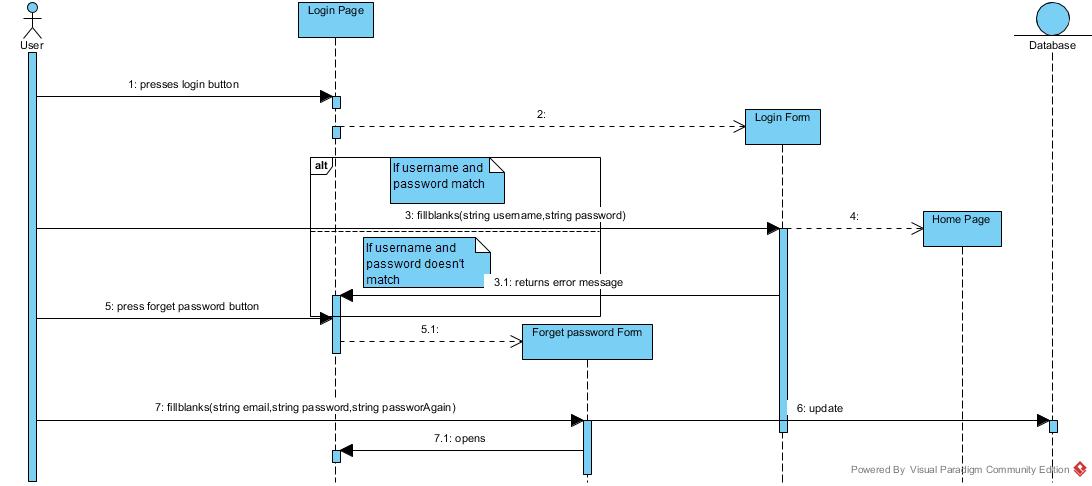
SEQUENCE DIAGRAM

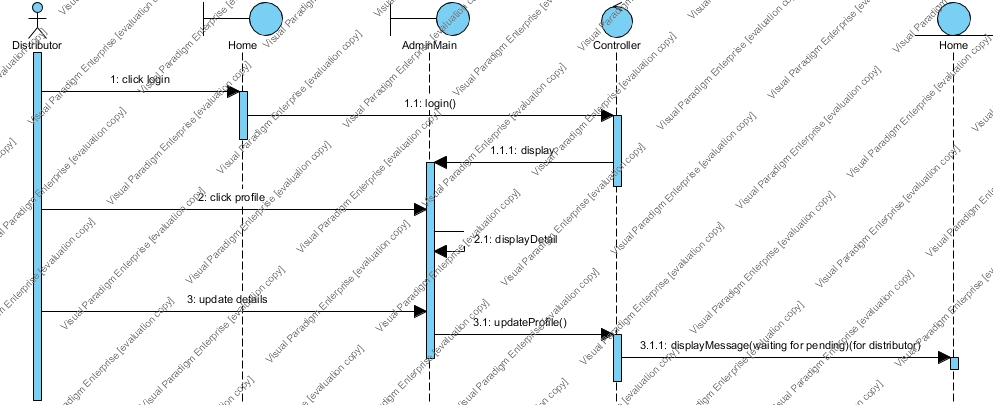


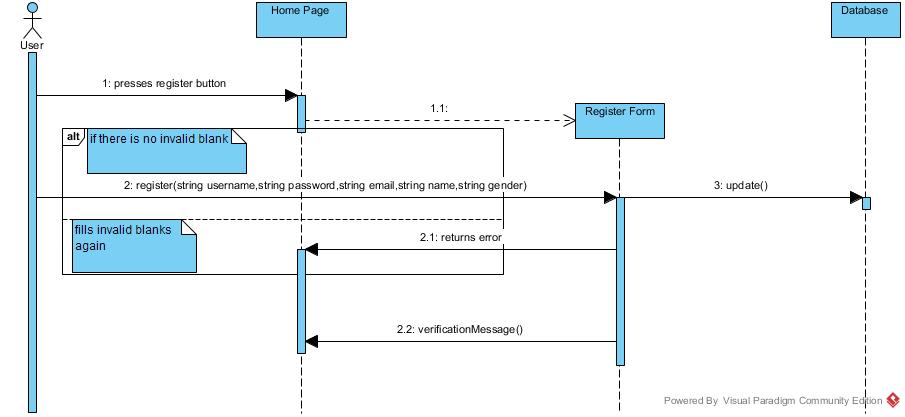


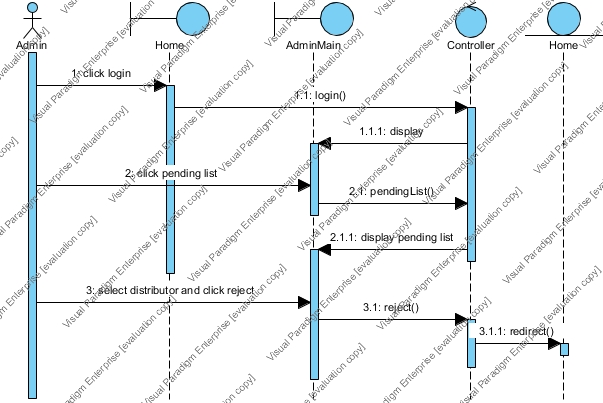


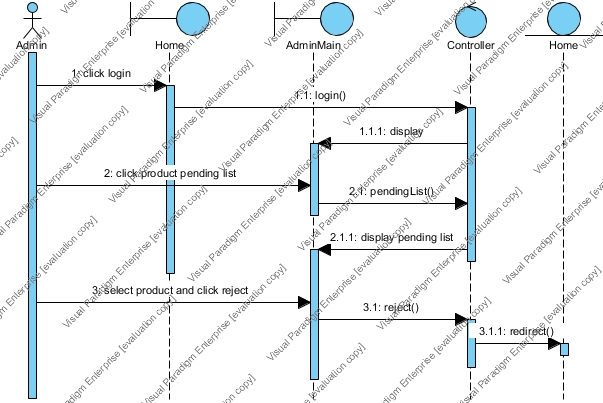


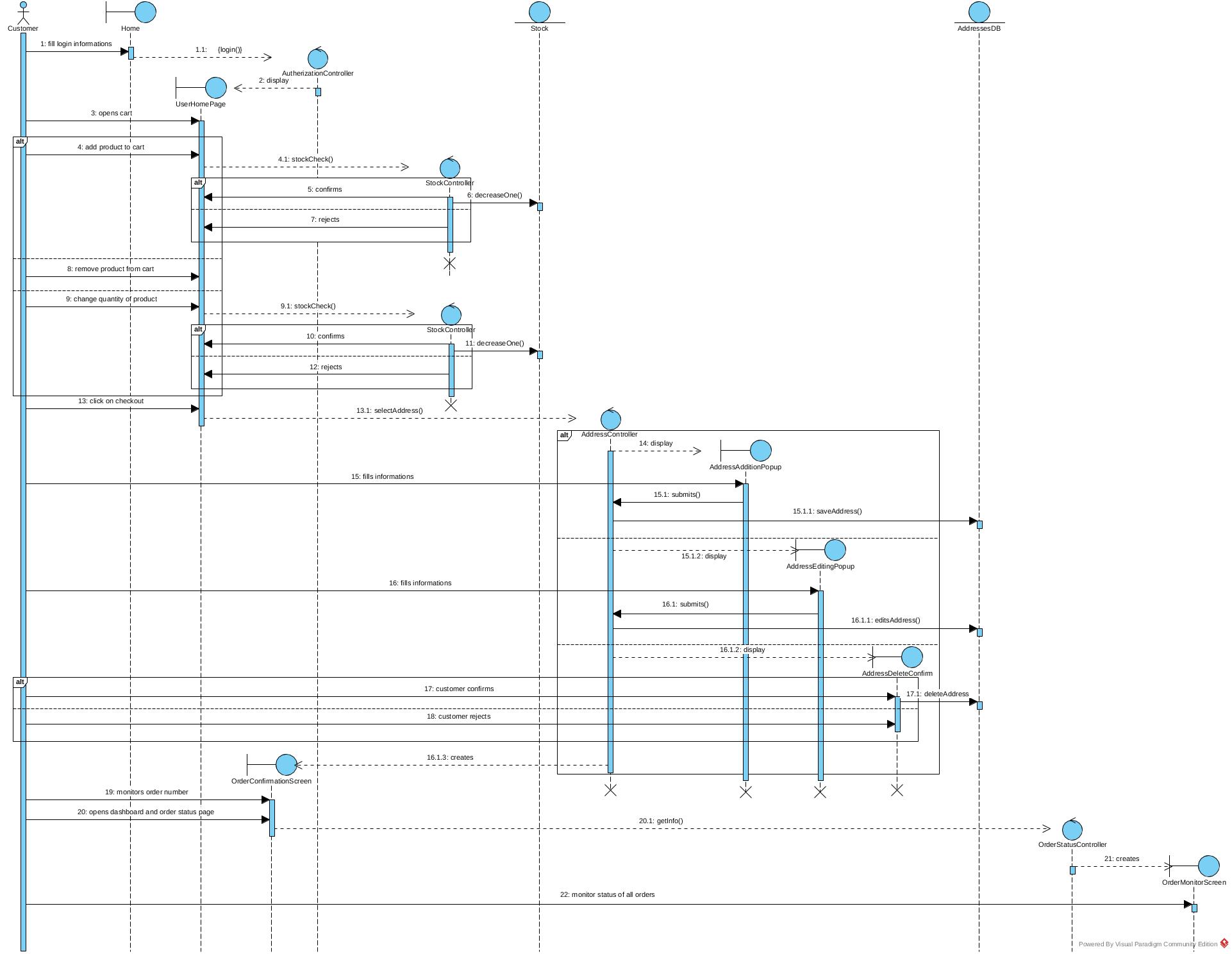


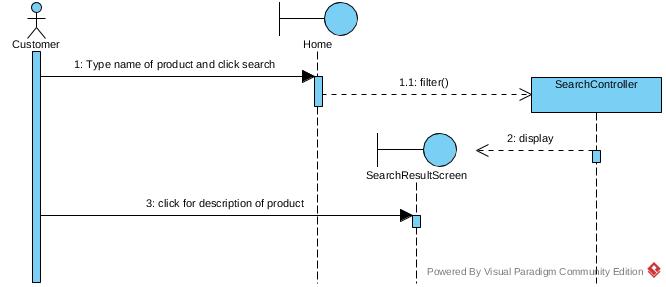


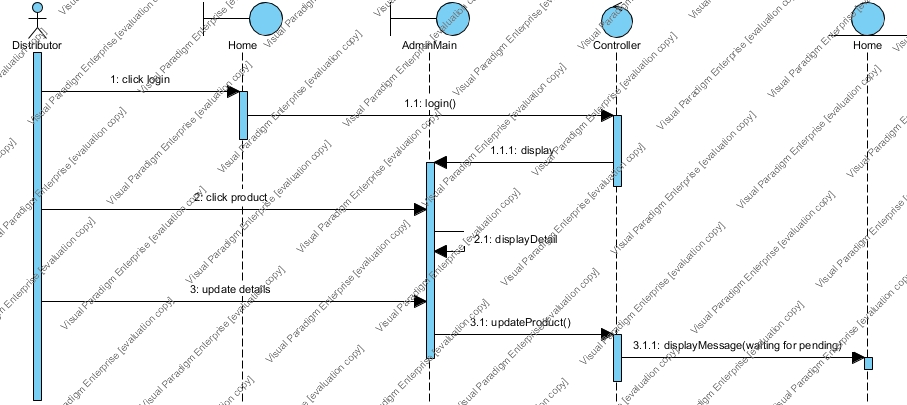


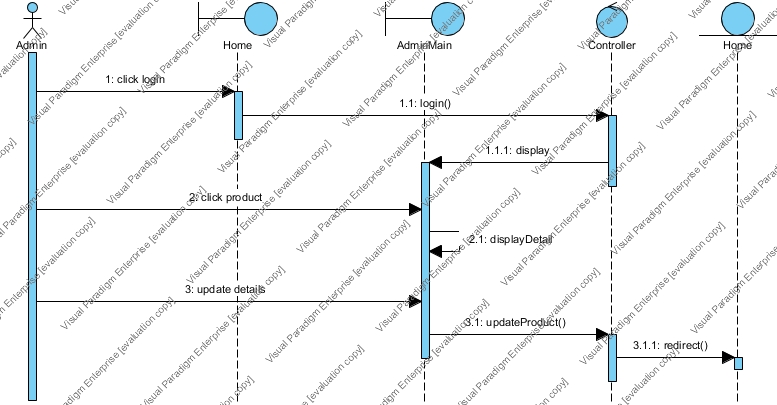


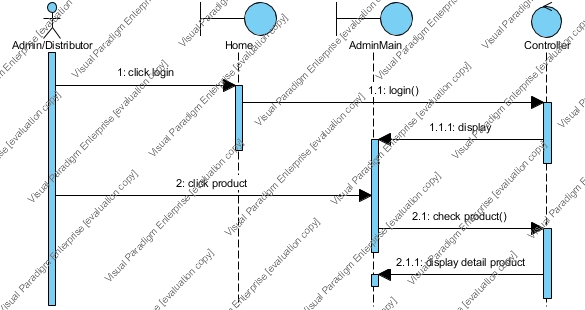


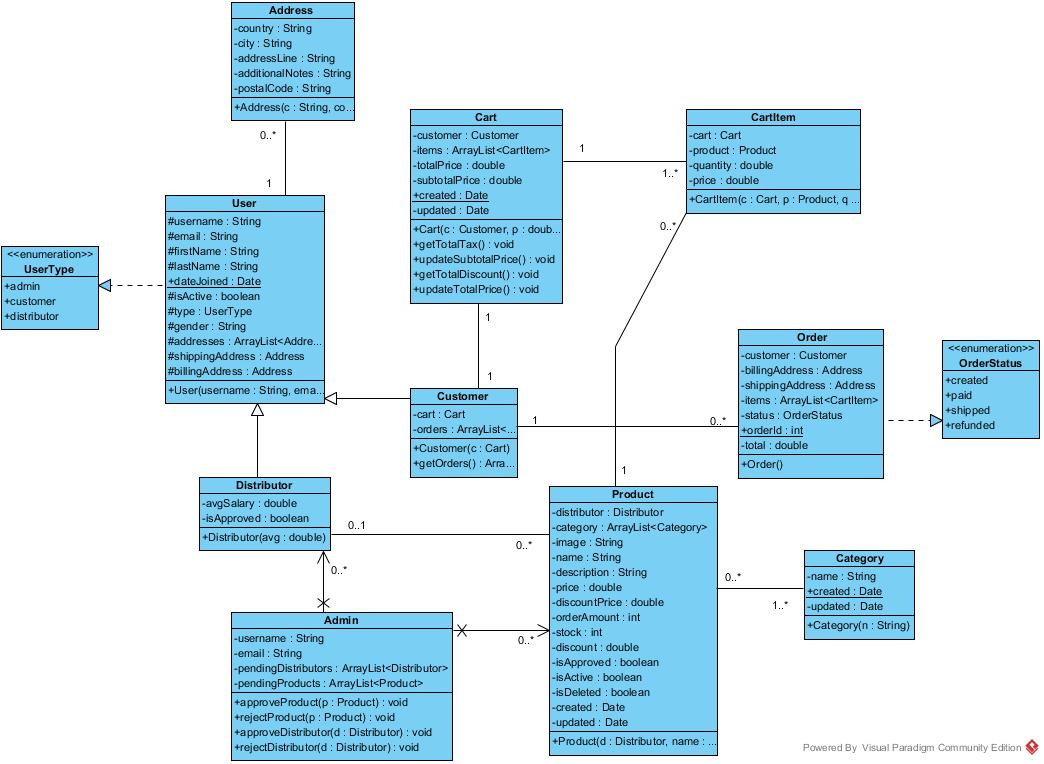










Object Diagram 

**Glossary**

* **Admin :** Actor of e-commerce website
* **Distributor:** Actor of e-commerce website
* **Customer:** Actor of e-commerce website
* **GitHub:** A web-based hosting service, mostly used for computer code.
* **Django:** A framework for web design, developed by Python.
* **Feedback:** The answer of the system for the operation.
* **Framework:** An abstraction of software functionality.
* **Database:** The collection of large data set/stack.
* **Python:** It is a modern programming language.
* **SQLite3:** A compact free database that can easily be used to create and use a database.
* **Template:** A guide of making patterns.

**References**

This subsection should:

•Provide a complete list of all documents referenced elsewhere in the RAD, or in a separate, specified document.

•Identify each document by title, report number - if applicable - date, and publishing organization.

•Specify the sources from which the references can be obtained.

The following is an example of listing a book in this section. Check the text to see how it is cross referenced (The whole document is based on [1]).

* Bruegge B. & Dutoit A.H.. (2010). *Object-Oriented Software Engineering Using UML, Patterns, and Java*, Prentice Hall, 3rd ed.