**COMSATS University Islamabad, Abbottabad Campus**

**Department of Computer Science**

**Project Proposal**

**Cosmetic Management System**

**CSC392 Object Oriented Software Engineering**

Submitted on: <23/04/2022>

Group Members:

Talah Khan (FA20-BSE-042)

Osama Khan (FA20-BSE-047)

Sabahat Siddique (FA20-BSE-054)

Umama Noor (FA20-BSE-055)

Farhan Khan (FA20-BSE-69)

Bilal Khan (FA20-BSE-071)

Supervisor: Mukhtiar Zamin

Table of Contents

[CHAPTER 1 PROJECT PROPOSAL 7](#_Toc108456828)

[1.1 Introduction 7](#_Toc108456829)

[1.2 Vision and Business Case 7](#_Toc108456830)

[1.3 Use-Case Model 7](#_Toc108456831)

[1.4 Supplementary Specification 8](#_Toc108456832)

[1.5 Glossary 9](#_Toc108456833)

[1.6 Risk List & Risk Management Plan 9](#_Toc108456834)

[CHAPTER 2 USE CASES 9](#_Toc108456835)

[2.1 Use Case Diagram 10](#_Toc108456836)

[2.2 Brief Level Use Cases 10](#_Toc108456837)

[Talah Khan (FA20-BSE-042) 10](#_Toc108456838)

[Use Case: Create Account 10](#_Toc108456839)

[Use Case: Login Account 10](#_Toc108456840)

[Use Case: Update Profile 10](#_Toc108456841)

[Use Case: give feedback 11](#_Toc108456842)

[Use Case: view customer feedback 11](#_Toc108456843)

[Osama Khan (FA20-BSE-047) 11](#_Toc108456844)

[Use Case: search item 11](#_Toc108456845)

[Use Case: add item 11](#_Toc108456846)

[Use Case: add category 11](#_Toc108456847)

[Use Case: view item 11](#_Toc108456848)

[Use Case: view category 11](#_Toc108456849)

[Sabahat Siddique (FA20-BSE-054) 12](#_Toc108456850)

[Use Case: Purchase Order 12](#_Toc108456851)

[Use Case: return item 12](#_Toc108456852)

[Use Case: add to cart 12](#_Toc108456853)

[Use Case: replace item 12](#_Toc108456854)

[Use Case: delete item 12](#_Toc108456855)

[Umama Noor (FA20-BSE-055) 13](#_Toc108456856)

[**Use Case: place order** 13](#_Toc108456857)

[**Use Case: track order** 13](#_Toc108456858)

[**Use Case: manage order** 13](#_Toc108456859)

[**Use Case: Payment** 13](#_Toc108456860)

[**Use Case: Process Sale** 13](#_Toc108456861)

[Farhan Khan (FA20-BSE-069) 13](#_Toc108456862)

[Use Case: add stock 13](#_Toc108456863)

[Use Case: return defective 14](#_Toc108456864)

[Use Case: refill store 14](#_Toc108456865)

[Bilal Khan (FA20-BSE-071) 14](#_Toc108456866)

[Use Case: generate report 14](#_Toc108456867)

[Use Case: view sales report 14](#_Toc108456868)

[Use Case: view profit report 14](#_Toc108456869)

[Use Case: view loss report 14](#_Toc108456870)

[Use Case: edit report 14](#_Toc108456871)

[2.3 Fully Dressed Use Cases 15](#_Toc108456872)

[Talah khan (FA20-BSE-042) 15](#_Toc108456873)

[Use Case UC1: register account 15](#_Toc108456874)

[Use Case UC2: login 17](#_Toc108456875)

[Use Case UC3: update profile 19](#_Toc108456876)

[Use Case UC4: customer feedback 20](#_Toc108456877)

[Use Case UC5: view customer feedback 22](#_Toc108456878)

[Osama Khan (FA20-BSE-047) 24](#_Toc108456879)

[Use Case UC1: Search item 24](#_Toc108456880)

[Use Case UC2: Add item 26](#_Toc108456881)

[Use Case UC3: Add category 28](#_Toc108456882)

[Use Case UC4: view item 29](#_Toc108456883)

[Use Case UC5: view category 31](#_Toc108456884)

[Sabahat Siddique (FA20-BSE-054) 33](#_Toc108456885)

[Use Case UC1: purchased order 33](#_Toc108456886)

[Use Case UC2: RETURN ITEM 36](#_Toc108456887)

[Use Case UC1: Add to Cart 39](#_Toc108456888)

[Use Case UC1: Replace item 42](#_Toc108456889)

[Use Case UC1: Delete item 45](#_Toc108456890)

[Umama Noor (FA20-BSE-055) 48](#_Toc108456891)

[Use Case UC1: Place Order 48](#_Toc108456892)

[Use Case UC1: track Order 51](#_Toc108456893)

[Use Case UC1: Process sale 53](#_Toc108456894)

[Use Case UC1: manage Order 56](#_Toc108456895)

[Use Case UC1: Payment 58](#_Toc108456896)

[Farhan Khan (FA20-BSE-069) 60](#_Toc108456897)

[Bilal Khan (FA20-BSE-071) 64](#_Toc108456898)

[2.4 NetBeans Screenshots 71](#_Toc108456899)

[CHAPTER 3 Domain Model 88](#_Toc108456900)

[3.1 Introduction 88](#_Toc108456901)

[Talah Khan (FA20-BSE-042) 88](#_Toc108456902)

[Osama Khan (FA20-BSE-047) 88](#_Toc108456903)

[Sabahat Siddique (FA20-BSE-054) 90](#_Toc108456904)

[Umama Noor (FA20-BSE-055) 90](#_Toc108456905)

[Farhan Khan (FA20-BSE-069) 90](#_Toc108456906)

[Bilal Khan (FA20-BSE-071) 91](#_Toc108456907)

[CHAPTER 4 System Sequence Diagram 93](#_Toc108456908)

[4.1 Introduction 93](#_Toc108456909)

[Talah Khan (FA20-BSE-042) 93](#_Toc108456910)

[Osama Khan (FA20-BSE-047) 98](#_Toc108456911)

[Sabahat Siddique (FA20-BSE-054) 104](#_Toc108456912)

[Umama Noor (FA20-BSE-055) 110](#_Toc108456913)

[Farhan Khan (FA20-BSE-069) 116](#_Toc108456914)

[Bilal Khan (FA20-BSE-071) 117](#_Toc108456915)

[CHAPTER 5 Operation Contracts 118](#_Toc108456916)

[5.1 Introduction 118](#_Toc108456917)

[Talah Khan (FA20-BSE-042) 118](#_Toc108456918)

[Osama Khan (FA20-BSE-047) 120](#_Toc108456919)

[Sabahat Siddique (FA20-BSE-054) 126](#_Toc108456920)

[Umama Noor (FA20-BSE-055) 130](#_Toc108456921)

[Farhan Khan (FA20-BSE-069) 133](#_Toc108456922)

[Bilal Khan (FA20-BSE-071) 136](#_Toc108456923)

[CHAPTER 6 Package Diagram 138](#_Toc108456924)

[6.1 Introduction 138](#_Toc108456925)

[Talah Khan (FA20-BSE-042) 138](#_Toc108456926)

[Osama Khan (FA20-BSE-047) 138](#_Toc108456927)

[Sabahat Siddique (FA20-BSE-054) 138](#_Toc108456928)

[Umama Noor (FA20-BSE-055) 138](#_Toc108456929)

[Farhan Khan (FA20-BSE-069) 138](#_Toc108456930)

[Bilal Khan (FA20-BSE-071) 138](#_Toc108456931)

[CHAPTER 7 Interaction Diagrams 138](#_Toc108456932)

[7.1 Sequence Diagram 139](#_Toc108456933)

[Talah Khan (FA20-BSE-042) 139](#_Toc108456934)

[Osama Khan (FA20-BSE-047) 140](#_Toc108456935)

[Sabahat Siddique (FA20-BSE-054) 143](#_Toc108456936)

[Umama Noor (FA20-BSE-055) 143](#_Toc108456937)

[Farhan Khan (FA20-BSE-069) 143](#_Toc108456938)

[Bilal Khan (FA20-BSE-071) 143](#_Toc108456939)

[7.2 Communication Diagram 144](#_Toc108456940)

[Talah Khan (FA20-BSE-042) 144](#_Toc108456941)

[Osama Khan (FA20-BSE-047) 144](#_Toc108456942)

[Sabahat Siddique (FA20-BSE-054) 146](#_Toc108456943)

[Umama Noor (FA20-BSE-055) 146](#_Toc108456944)

[Farhan Khan (FA20-BSE-069) 146](#_Toc108456945)

[Bilal Khan (FA20-BSE-071) 146](#_Toc108456946)

[CHAPTER 8 Class Diagram 146](#_Toc108456947)

[8.1 Class Diagram 146](#_Toc108456948)

[Talah Khan (FA20-BSE-042) 147](#_Toc108456949)

[Osama Khan (FA20-BSE-047) 148](#_Toc108456950)

[Sabahat Siddique (FA20-BSE-054) 148](#_Toc108456951)

[Umama Noor (FA20-BSE-055) 148](#_Toc108456952)

[Farhan Khan (FA20-BSE-069) 148](#_Toc108456953)

[Bilal Khan (FA20-BSE-071) 148](#_Toc108456954)

[CHAPTER 9 Coding 148](#_Toc108456955)

[9.1 coding with explanations 148](#_Toc108456956)

[Talah Khan (FA20-BSE-042) 148](#_Toc108456957)

[Osama Khan (FA20-BSE-047) 148](#_Toc108456958)

[Sabahat Siddique (FA20-BSE-054) 148](#_Toc108456959)

[Umama Noor (FA20-BSE-055) 148](#_Toc108456960)

[Farhan Khan (FA20-BSE-069) 148](#_Toc108456961)

[Bilal Khan (FA20-BSE-071) 148](#_Toc108456962)

# CHAPTER 1 PROJECT PROPOSAL

## 1.1 Introduction

Cosmetic management system is a software used to manage the products and costumers record. It is typically used by store which need to manage the records of their customers and products. There has always been a need for efficient management of network-based system for handling customer orders.

We have decided to build as software as a solution for this problem. this project is made using JAVA programming language. It is a windows-based software. The design is made using JAVA Swing graphical user interface.

The software contains an inventory and enables an administrator to keep track of customer orders and maintain records. The admin and the shop keeper can also generate and view report regarding customer and the products. It also consists of a feedback system; feedback can be given by the customer about a certain product or overall, on how to improve the system.

## 1.2 Vision and Business Case

Availability of this system 24/7 is very much crucial for the client function. for this purpose, we are developing a cosmetic management system that is able to provide high quality access to clients database

The proposed cosmetic management system will be having a lot of functional features discussed in the use-case model section

The proposed management system will have the capability to buy, give feedback, view, review etc. this will give the client accessibility to make sure buying and managing is easy. Other than that admin will enable the user to ensure their order can be tracked.

## 1.3 Use-Case Model

• Login:

Admin will be able to Login into the system with the username and password they used while registering their account.

• Manage Products:

Admin needs to have access to their system so he/she can manage item, from this they can keep a record of all the items

• View Order Detail:

complete and proper details about the order should be visible to the admin.

• Manage Categories:

Categories are made when items are divided into sub-categories. Admin has the right to manage the categories, which also has different sub-Categories according to the requirements.

• Manage Customers:

Admin should be able to manage customer data. Record of the customers will also be maintained in the database.

• Register:

If the user wants to buy any product and they haven’t created their account, then first of all they need to register themselves.

• View Products:

User can view the products, which they want.

• Manage cart:

User can add the product into their cart or Wishlist, which they want to buy.

• Place Order:

The order will be place by the system when the user confirms their order.

• Login:

If the user wants to buy the product, then first of all, they should sign in their id, which was assigned to him/her.

## 1.4 Supplementary Specification

• The system shall be available to the user and the admin 24 hours/day.

• It should be made possible to upgrade the system while it is running.

• Users that are not allowed to view specific areas of the system will be redirected back to the login page.

• During an average load, all Web pages must download in three seconds, and in five seconds during a peak load.

• While executing a search, the system must be able to display at least 100 search results per page.

• The end user shall be able to place an order within 30 seconds.

• The system should be able to contain data of 1,000,000 users.

• The system shall accept 50 requests per second.

## 1.5 Glossary

***Key domain terminology, and data dictionary.***

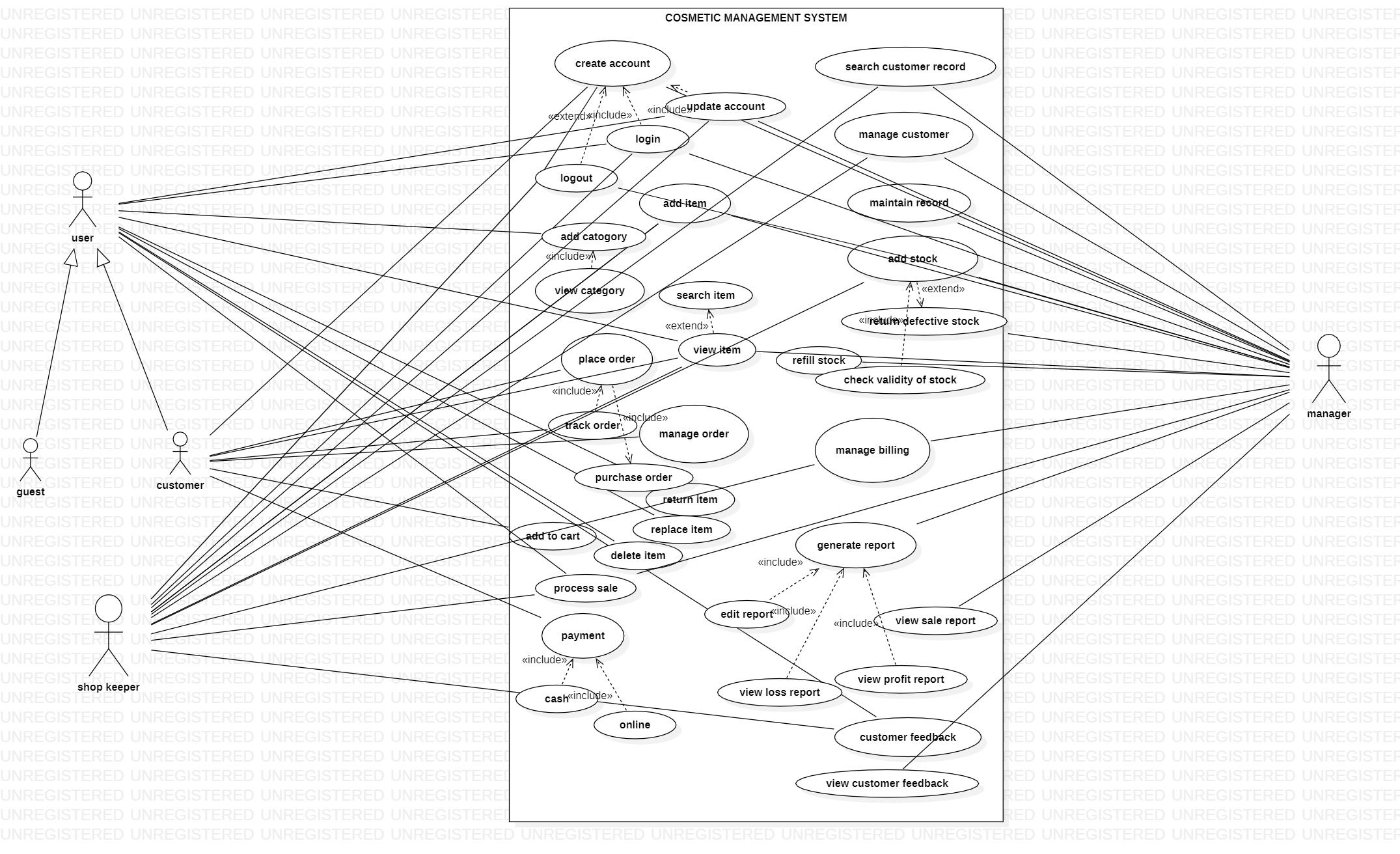
7.8. NextGen Example: A (Partial) Glossary

## 1.6 Risk List & Risk Management Plan

* + Password and information of users can be leaked.
  + Showing products which aren’t in stock.
  + Showing wrong price
  + Payment method not secure
  + Privacy not fully provided
  + Multiple users with same username
  + Software not according to ISO certification.

# CHAPTER 2 USE CASES

## 2.1 Use Case Diagram



## 2.2 Brief Level Use Cases

### Talah Khan (FA20-BSE-042)

#### Use Case: Create Account

user opens the cosmetic management system and clicks on the register button. The system takes the user to the registration page. The user is required to enter information asked by the system. The user then enters his full name, email, address, phone number and password. The system then checks if the user is already registered or not. If he is not registered the system validates and registers him as a customer.

#### Use Case: Login Account

user opens the cosmetic management system and clicks on the login button. The system takes the user to the login page. The user is required to enter information asked by the system which is the username and the password. The system then checks if the user is already registered or not. If he is registered the system validates and logs him in.

#### Use Case: Update Profile

user opens the cosmetic management system and logs in into his account. The system takes the user to his dashboard or profile page. Over their user clicks on the update profile button where he can view his profile details and update them as well.

#### Use Case: give feedback

user opens the cosmetic management system and logs in into his account. He then views the products and gives feedback about them or gives feedback about the CMS.

#### Use Case: view customer feedback

user opens the cosmetic management system and logs in into his account. He then views the feedbacks given by the customer about them or the products.

### Osama Khan (FA20-BSE-047)

#### Use Case: search item

User requests to search item. System checks the request of user. User inputs search on system. System checks the request validity. After verification system allows user to relevant search item.

#### Use Case: add item

Manager/Shopkeeper requests the system to allow access to add items to stock. System first verify that Manager/Shopkeeper is registered or logged into the system. After verification system allows Manager/Shopkeeper to add items to the stock. Manager/Shopkeeper checks the database to see which items are not available. Manager contacts different companies to send items. After receiving items Manager/Shopkeeper add those items to the CMS.

#### Use Case: add category

Manager/Shopkeeper requests the system to allow access to add categories to stock. System first verify that Manager/Shopkeeper is registered or logged into the system . After verification system allows Manager/Shopkeeper to add categories to the stock. Manager/Shopkeeper checks the database to see which categories are not available. Manager/Shopkeeper add those categories to the CMS successfully.

#### Use Case: view item

Customer arrives at CMS to order item. System asks user to login. Customer requests the system to display items. System show all the items to the customer. Customer view all the items. Customer after viewing select items and request system to purchase order. System successfully show items and customer view all the items successfully.

#### Use Case: view category

Customer arrives at CMS to order item. System asks user to login. Customer requests the system to display categories. System show all the items to the customer. Customer view all the categories. Customer after viewing select categories and request system to purchase order. System successfully show categories and customer view all the categories successfully.

### Sabahat Siddique (FA20-BSE-054)

#### Use Case: Purchase Order

Customer arrives atCMS to purchase order…customer request to system to show items…customer views all the items and choose the item which he or she wants to purchase…system verify that the product is available that is purchased by customer. After verification system asks the customer for payment method**.** Customer select payment methodand purchased order successfully

#### Use Case: return item

Customer asks to return the item. System check that the return item is OK or not. System set the deadline for returning orders if the deadline pass send SORRY message to customer. If the deadline is not pass system allows to the customer to return the order. System also check return item ID and quantity

#### Use Case: add to cart

Customer arrives at CMS to add item to cart. Customers request the system to show items. System shows all the available item to the customers. Customers view all the item which they want to add to cart. System checks the availability of product. If the product available system allows the customer to add item to cart

#### Use Case: replace item

Customer wants to replace the order. System check that the replacement of the order is possible or not. System check that the demand of the customer for the new item is available in stock or not. If stock is not available send msg to customer that the stock is unavailable. If stock available system allow the customer to replace the item. System also checks replace item ID and quantity as well.

#### Use Case: delete item

User arrives at CMS to delete items. User select the item that he wants to delete. User request to system to delete selected items.

### Umama Noor (FA20-BSE-055)

#### **Use Case: place order**

Customer opens the CMS to place order. Customer requests the system to show items. System display items list to the Customer. Customer adds items to shopping cart. System display message indicate the item added to shopping cart. Customer proceeds to checkout. System ask user provide shipping and billing information. Customer provides shipping and billing information. System confirms the shipping information, process the order and ship out the items. Customer receives the items.

#### **Use Case: track order**

After the user has selected items to purchase and then order the items.The user will provide payment and shipping information. The system will respond with confirmation of the order and a tracking number that the user can use to check on order status in the future. System verifies the availability of selected product, payment from customer and then delivers or track order to the customer address.

#### **Use Case: manage order**

Manager’s responsibility is to add the items in stock and then manage the customer’s order. Customer requests the system to show the details of the product. System first verifies that customer is authorized to system. After verification system will show the details of product to the customer.

#### **Use Case: Payment**

Customer opens the CMS to place order. After placing the order customer requests for the payment process. System shows the secure payment method to the customers. Payment method involves cash on delivery or online payment through Easypaisa or ATM.

#### **Use Case: Process Sale**

A customer opens a checkout with items to purchase. The cashier uses the POS system to record each purchased item. The system presents a running total and line-item details. The customer enters payment information, which the system validates and records. The system updates inventory. The customer receives a receipt from the system and then leaves with the items.

### Farhan Khan (FA20-BSE-069)

#### Use Case: add stock

Shopkeeper and manager both have the rights to add stock in the store. Managers must determine which brands would be the best fit, from there, if he feel like it’s going to be a mutually beneficial then make an appointment to see the items. The manger must know the rules about buying internationally too. After you buy the items then Shopkeeper add the stock to the store inventory with the item name, id, price, description, and brand.

#### Use Case: return defective

The customer rights act gives you a clear early right to reject the goods that are unsatisfactory quality. The store must stop selling these items and delete the records of those items from the database of the store and then return it.

#### Use Case: refill store

This use case allows to refill the store if its empty. Once an item is less than the safety stock, then the store manager needs to place order for all the items that need to be refilled based on the available stock.

### Bilal Khan (FA20-BSE-071)

#### Use Case: generate report

This use case allow to generate report of the solid product which daily , weekly , and monthly sold products .An admin login into system and view the sales report. On the basis of sales report shopkeeper generate a report to admin which product most sold. In this report a brief detail of products .Admin check report on daily or monthly sales report.

#### Use Case: view sales report

This use case allow admin to view sales report check the daily , weekly , and monthly . In this report a brief detail of products .Admin or shopkeeper check view sales report on daily bases.

#### Use Case: view profit report

User enter into cosmetic management system and to review the profit of sold products. The user calculates profit based on the cost and sales price in use at the time.

#### Use Case: view loss report

User enter into cosmetic management system and to review the loss of sold products. The user calculates loss based on the cost and sales price in use at the time. An admin generate a report which specify loss in daily ,weekly ,monthly.

#### Use Case: edit report

User enter into cosmetic management system and edit in the report of products . Shopkeeper sold more product in old report in this he will edit in report and generate new report to customer.

## 2.3 Fully Dressed Use Cases

### Talah khan (FA20-BSE-042)

| Use Case UC1: register account |
| --- |
| **Scope**: cosmetic management system  **Level**: user goal  **Primary** **Actor**: customer  **Stakeholders and Interests**:  - Customer: want to register an account in the cosmetic management system.  - Salesperson: want to register an account in the cosmetic management system.  - Manager: want to register an account in the cosmetic management system.  **Preconditions**: User is identified and authenticated. |

**Success Guarantee** (or Postconditions): username is available, and account is registered. Account is not already registered.

**Main Success Scenario (or Basic Flow):**

1. User clicks on register an account button.
2. System redirects him to the registration page
3. System asks him to provide the details.
4. The user enters the details.
5. System then registers the account when user clicks on register
6. System then takes the user to his dashboard where he can access the store and his own profile.

**Extensions (or Alternative Flows):**

\*a. At any time when the user tries signing up:

1. user enters the username and password he used when registering his account.
2. The system then verifies his credentials when he clicks on the register button.
3. when the credentials matches then he is redirected to his profile.

\*b. At any time when the user tries signing up:

1. user enters the password and username he used when registering his account.
2. The system then verifies his credentials.
3. If the credentials don’t match, then the system displays an error either his password or username is incorrect.

**Special Requirements:**

* + 1. Text should be visible from 2 meters.
    2. Color scheme should be used which is clearly visible.
    3. Special characters should be used in username and password.
    4. It should take less than 1 minutes to register an account.
    5. Password should be display as \*\*\*\*.

**Technology and Data Variations List**:

1. Languages used is java.
2. Software used to design interface is netbeans, Gui Swing.
3. Mouse and keyboard are required.

Frequency of Occurrence: Could be nearly continuous.

| Use Case UC2: login |
| --- |
| **Scope**: cosmetic management system  **Level**: user goal  **Primary** **Actor**: customer, admin, shopkeeper  **Stakeholders and Interests**:  - Customer: want to login in the cosmetic management system.  - Salesperson: want to login in the cosmetic management system.  - Manager: want to login in the cosmetic management system.  **Preconditions**: User is identified and authenticated. |

**Success Guarantee** (or Postconditions): username and password are valid, and account is registered. User must be able to login.

**Main Success Scenario (or Basic Flow):**

1. User clicks on login button.
2. System redirects him to the log in page
3. System asks him to provide the details.
4. The user enters username, and password
5. System then takes the user to his dashboard where he can access the store and his own profile.

**Extensions (or Alternative Flows):**

\*a. At any time when the user tries to login:

1. user enters the username and password he used when registering his account.
2. The system then verifies his credentials when he clicks on the register button.
3. when the credentials matches then he is redirected to his profile.

\*b. At any time when the user tries to login:

1. user enters the password and username he used when registering his account.
2. The system then verifies his credentials.
3. If the credentials don’t match, then the system displays an error either his password or username is incorrect.

**Special Requirements:**

* + 1. Text should be visible from 2 meters.
    2. Color scheme should be used which is clearly visible.
    3. Special characters should be used in username and password.
    4. It should take less than 1 minutes to register an account.
    5. Password should be display as \*\*\*\*.

**Technology and Data Variations List**:

1. Languages used is java.
2. Software used to design interface is netbeans, Gui Swing.
3. Mouse and keyboard are required .

Frequency of Occurrence: Could be nearly continuous.

**Open Issues:**

- What are the tax law variations?

- Explore the remote service recovery issue.

- What customization is needed for different businesses?

- Must a cashier take their cash drawer when they log out?

- Can the customer directly use the card reader, or does the cashier have to do it?

| Use Case UC3: update profile |
| --- |
| **Scope**: cosmetic management system  **Level**: user goal  **Primary** **Actor**: customer, admin, shopkeeper  **Stakeholders and Interests**:  - Customer: want to update his account in the cosmetic management system.  - Salesperson: want to update his account in the cosmetic management system.  - Manager: want to update his account in the cosmetic management system.  **Preconditions**: User is identified and authenticated. |

**Success Guarantee** (or Postconditions): user should be successfully logged in into his account where he can update his profile.

**Main Success Scenario (or Basic Flow):**

1. The user clicks onto the update profile button.
2. System redirects him to that page and shows a form.
3. User enters the details that he want to change and click on update.
4. System then updates his profile.

**Extensions (or Alternative Flows):**

\*a. At any time when the user tries to update his account:

1. user enters into his account.
2. He clicks on the update profile button.
3. The changes what he wants to change and then clicks on update button.
4. System then updates his account.

\*b. At any time when the user tries signing up:

1. user enters into his account.
2. He clicks on the update profile button.
3. The changes what he wants to change and then clicks on update button.
4. System shows an error if the changes made were not appropriate.

**Special Requirements:**

* + 1. Text should be visible from 2 meters.
    2. Color scheme should be used which is clearly visible.
    3. Special characters should be used in username and password.
    4. It should take less than 1 minutes to register an account.
    5. Password should be display as \*\*\*\*.

**Technology and Data Variations List**:

1. Languages used is java.
2. Software used to design interface is netbeans, Gui Swing.
3. Mouse and keyboard are required.

Frequency of Occurrence: Could be nearly continuous.

**Open Issues:**

- What are the tax law variations?

- Explore the remote service recovery issue.

- What customization is needed for different businesses?

- Must a cashier take their cash drawer when they log out?

- Can the customer directly use the card reader, or does the cashier have to do it?

| Use Case UC4: customer feedback |
| --- |
| **Scope**: cosmetic management system  **Level**: user goal  **Primary** **Actor**: customer  **Stakeholders and Interests**:  - Customer: want to give feedback about his product or CMS.  **Preconditions**: User is identified and authenticated. |

**Success Guarantee** (or Postconditions): user should be successfully logged in into his account where he can view product and give feedback about it.

**Main Success Scenario (or Basic Flow):**

1. User clicks on the give feedback button.
2. System displays the feedback form
3. User fills the form and submits the form
4. System stores the form in the database

**Extensions (or Alternative Flows):**

\*a. At any time when the user tries to give feedback:

1. user enters into his account.
2. He views a product and decides to give it a feedback.
3. A form then opens, and he provides feedback.
4. He then submits the feedback and system shows a message that feedback is submitted successfully.

\*b. At any time when the user tries to give feedback:

1. user enters into his account.
2. He views a product and decides to give it a feedback.
3. A form then opens, and he provides feedback.
4. He then submits the feedback and system shows a message that feedback is inappropriate or not completed fully.

**Special Requirements:**

* + 1. Text should be visible from 2 meters.
    2. Color scheme should be used which is clearly visible.
    3. Special characters should be used in username and password.
    4. It should take less than 1 minutes to register an account.
    5. Password should be display as \*\*\*\*.

**Technology and Data Variations List**:

1. Languages used is java.
2. Software used to design interface is netbeans, Gui Swing.
3. Mouse and keyboard are required .

Frequency of Occurrence: Could be nearly continuous.

**Open Issues:**

- What are the tax law variations?

- Explore the remote service recovery issue.

- What customization is needed for different businesses?

- Must a cashier take their cash drawer when they log out?

- Can the customer directly use the card reader, or does the cashier have to do it?

| Use Case UC5: view customer feedback |
| --- |
| **Scope**: cosmetic management system  **Level**: user goal  **Primary** **Actor**: admin  **Stakeholders and Interests**:  - admin: want to view feedback about products or CMS.  **Preconditions**: User is identified and authenticated. |

**Success Guarantee** (or Postconditions): user should be successfully logged in into his account where he can view feedbacks given by the customers.

**Main Success Scenario (or Basic Flow):**

1. User opens the management system to login into his account.
2. System redirects him to the login page
3. System asks him to provide the details.
4. The user enters username and password.
5. System then verifies and logs him into his profile.
6. The admin then clicks on the view feedback button.
7. System redirects him to that page.
8. Admin then views the feedback given by the customer.

**Extensions (or Alternative Flows):**

\*a. At any time when the admin tries to view feedbacks:

1. user enters into his account.
2. He views feedback given by the customers,
3. He can rate them back as well when he views the feedback

\*b. At any time when the admin tries to view feedbacks:

1. user enters into his account.
2. He clicks to view the feedback given by the customers.
3. The feedback doesn’t open due to an error.
4. He then restarts the system to view the feedback.

**Special Requirements:**

* + 1. Text should be visible from 2 meters.
    2. Color scheme should be used which is clearly visible.
    3. Special characters should be used in username and password.
    4. It should take less than 1 minutes to register an account.
    5. Password should be display as \*\*\*\*.

**Technology and Data Variations List**:

1. Languages used is java.
2. Software used to design interface is netbeans, Gui Swing.
3. Mouse and keyboard are required.

Frequency of Occurrence: Could be nearly continuous.

**Open Issues:**

- What are the tax law variations?

- Explore the remote service recovery issue.

- What customization is needed for different businesses?

- Must a cashier take their cash drawer when they log out?

- Can the customer directly use the card reader, or does the cashier have to do it?

### Osama Khan (FA20-BSE-047)

| Use Case UC1: Search item Brief Level  User requests to search item. System checks the request of user. User inputs search on system. System checks the request validity. After verification system allows user to relevant search item. |
| --- |
| **Scope**: cosmetic management system  **Level**: user goal  **Primary** **Actor**: customer , Manager , Shopkeeper  **Stakeholders and Interests**:  -customer: Wants accurate, fast and relevant search  - Shopkeeper: Wants accurate, fast and relevant search  - Company: Wants to ensure that user , manager and shopkeeper are able to search the item with fast and relevant results  - Manager: Wants accurate, fast and relevant search  **Preconditions**: User is identified and authenticated. |

**Success Guarantee** (or Postconditions):

User’s Search is done. Search item is visible to user, manager, and shopkeeper

**Main Success Scenario (or Basic Flow):**

1. User opens the management system to register an account.
2. System redirects him to the registration page
3. System asks him to provide the details.
4. The user enters username, password, email and address
5. System then checks if the account is already registered or not.
6. If the account is registered the system asks to provide new information or to login
7. If the account is not registered then the system registers the account.
8. System then takes the user to his dashboard where he can access the store and his own profile.

**Extensions (or Alternative Flows):**

\*a. At any time, System fails:

1. To support recovery and correct relevant, ensure all search items are successfully searched and events can be recovered from any step of the scenario.

b\*. Search not found.

* 1. System gets signal and displays user that result not found.
  2. User probably starts new search and re-enters all items to be searched.

User continues with search (probably entering more items, viewing items purchasing order or handling payment).

**Special Requirements:**

* + 1. It should take less than 1 minutes to display results.
    2. Search should be according to Domain.
    3. Text should be visible from 2 meters.
    4. Color scheme should be used which is clearly visible.

**Technology and Data Variations List**:

1. Languages used is java.
2. Software used to design interface is netbeans, Gui Swing.
3. Mouse and keyboard are required.

Frequency of Occurrence: Could be nearly continuous

| Use Case UC2: Add item |
| --- |
| **Scope**: cosmetic management system  **Level**: user goal  **Primary** **Actor**: Manager, shopkeeper  **Stakeholders and Interests**:  - Salesperson: want to add items in the cosmetic management system.  - Manager: want to add items in the cosmetic management system.  **Preconditions**: User is identified and authenticated. |

**Success Guarantee** (or Postconditions): Accounts are registered. Items are successfully added to CMS

**Main Success Scenario (or Basic Flow):**

1. User opens the management system to log in into his account.
2. Manager/Shopkeeper requests the system to allow access to add items to stock.
3. System first verify that Manager/Shopkeeper is registered or logged into the system.
4. After verification system allows Manager/Shopkeeper to add items to the stock.
5. Manager/Shopkeeper checks the database to see which items are not available.
6. Manager contacts different companies to send items.
7. After receiving items Manager/Shopkeeper add those items to the CMS.
8. Items are successfully added.

**Extensions (or Alternative Flows):**

\*a. At any time when the Manager/Customer tries to add item:

1. Manager/Customer checks the unavailable items.
2. Manager/Customer requests system to add item.
3. System allows Manager/Customer to add items.
4. Items are added successfully

\*b. At any time when the Manager/Customer tries to add item:

1. Manager/Customer checks the unavailable items.
2. Manager/Customer requests system to add item.
3. System doesn’t allows Manager/Customer to add items.
4. Manager/Customer tries again.

**Special Requirements:**

1. Manager/Customer must be registered with the system.
2. Items should not be in stock already.

**Technology and Data Variations List**:

1. Languages used is java.
2. Software used to design interface is netbeans, Gui Swing.
3. Mouse and keyboard are required .

Frequency of Occurrence: Could be nearly continuous.

**Open Issues:**

- What are the tax law variations?

- Explore the remote service recovery issue.

- What customization is needed for different businesses?

- Must a cashier take their cash drawer when they log out?

- Can the customer directly use the card reader, or does the cashier have to do it?

| Use Case UC3: Add category |
| --- |
| **Scope**: cosmetic management system  **Level**: user goal  **Primary** **Actor**: Manager , shopkeeper  **Stakeholders and Interests**:  Salesperson: want to add items in the cosmetic management system.  - Manager: want to add items in the cosmetic management system.  **Preconditions**: User is identified and authenticated. |

**Success Guarantee** (or Postconditions): Manager/Shopkeeper should be successfully logged in into his account where he can add categories to CMS

**Main Success Scenario (or Basic Flow):**

* Manager/Shopkeeper requests the system to allow access to add categories to stock.
* System first verify that Manager/Shopkeeper is registered or logged into the system.
* After verification system allows Manager/Shopkeeper to add categories to the stock.
* Manager/Shopkeeper checks the database to see which categories are not available.
* Manager/Shopkeeper add those categories to the CMS successfully.

**Extensions (or Alternative Flows):**

\*a. At any time when the Manager/Customer tries to add category:

1. Manager/Customer checks the unavailable categories.
2. Manager/Customer requests system to add categories.
3. System allows Manager/Customer to add categories.
4. Categories are added successfully

\*b. At any time when the Manager/Customer tries to add category:

1. Manager/Customer checks the unavailable categories.
2. Manager/Customer requests system to add categories.
3. System doesn’t allows Manager/Customer to add categories.
4. User tries again.

**Special Requirements:**

1. Manager/Customer must be registered with the system.
2. Categories should not be in stock already.

**Technology and Data Variations List**:

1. Languages used is java.
2. Software used to design interface is netbeans, Gui Swing.
3. Mouse and keyboard are required .

Frequency of Occurrence: Could be nearly continuous.

**Open Issues:**

- What are the tax law variations?

- Explore the remote service recovery issue.

- What customization is needed for different businesses?

- Must a cashier take their cash drawer when they log out?

- Can the customer directly use the card reader, or does the cashier have to do it?

| Use Case UC4: view item |
| --- |
| **Scope**: cosmetic management system  **Level**: user goal  **Primary** **Actor**: Customer, Manager , Customer  **Stakeholders and Interests**:  - Customer , Manager, Customer: wants to view items on CMS.  **Preconditions**: User is identified and authenticated. |

**Success Guarantee** (or Postconditions): user should be successfully logged in into his account where he can view product.

**Main Success Scenario (or Basic Flow):**

1. Customer arrives at CMS to order item.
2. User opens the management system to login into his account.
3. System asks user to login.
4. Customer requests the system to display items.
5. System shows all the items to the customer.
6. Customer view all the items.
7. Customer after viewing select items and request system to purchase order.
8. System successfully show items and customer view all the items successfully.

**Extensions (or Alternative Flows):**

\*a. At any time when the user tries to view item

1. User enters into his account.
2. He enters item to view it.
3. System shows relevant searches.
4. User opens item to view.
5. Item successfully shown by CMS and user views it.

\*b. At any time when the user tries to view item.

1. User enters into his account.
2. He enters item to view it.
3. System shows relevant searches.
4. Item not found.
5. User is requested to view another item.

**Special Requirements:**

-Item should be present in inventory.

**Technology and Data Variations List**:

1. Languages used is java.
2. Software used to design interface is netbeans, Gui Swing.
3. Mouse and keyboard are required .

Frequency of Occurrence: Could be nearly continuous.

**Open Issues:**

- What are the tax law variations?

- Explore the remote service recovery issue.

- What customization is needed for different businesses?

- Must a cashier take their cash drawer when they log out?

- Can the customer directly use the card reader, or does the cashier have to do it?

| Use Case UC5: view category |
| --- |
| **Scope**: cosmetic management system  **Level**: user goal  **Primary** **Actor**: Customer , Manager, Customer  **Stakeholders and Interests**:  - Customer , Manager, Customer: wants to view categories on CMS.  **Preconditions**: User is identified and authenticated. |

**Success Guarantee** (or Postconditions): user should be successfully logged in into his account where he can view category.

**Main Success Scenario (or Basic Flow):**

1. Customer arrives at CMS to view category and order .
2. User login into his account.
3. Customer requests the system to display categories.
4. System shows all the categories to the customer.
5. Customer view all the categories.
6. Customer after viewing select categories and request system to purchase order.
7. System successfully show categories and customer view all the categories successfully.

**Extensions (or Alternative Flows):**

a. At any time when the user tries to view category:

1. User enters into his account.
2. He enters category to view it.
3. System shows relevant searches.
4. User opens category to view.
5. Category successfully shown by CMS and user views it.

\*b. At any time when the user tries to view category:

1. User enters into his account.
2. He enters category to view it.
3. System shows relevant searches.
4. Category not found.
5. User is requested to view another category.

**Special Requirements:**

* + 1. User should be authenticated.
    2. It should take less than 10 sec to view category
    3. Categories should be present in inventory.
    4. Text should be visible from 2 meters.
    5. Color scheme should be used which is clearly visible.

**Technology and Data Variations List**:

1. Languages used is java.
2. Software used to design interface is netbeans, Gui Swing.
3. Mouse and keyboard are required.

Frequency of Occurrence: Could be nearly continuous.

**Open Issues:**

- What are the tax law variations?

- Explore the remote service recovery issue.

- What customization is needed for different businesses?

- Must a cashier take their cash drawer when they log out?

- Can the customer directly use the card reader, or does the cashier have to do it?

### Sabahat Siddique (FA20-BSE-054)

| Use Case UC1: purchased order |
| --- |
| **Scope**: cosmetic management system  **Level**: user goal  **Primary** **Actor**: Customer  **Stakeholders and Interests**:  shopkeeper: Wants accurate, fast entry, and no payment errors, as cash drawer shortages are deducted from his/her salary.  shopkeeper: Wants sales commissions updated.  Customer: Wants purchase and fast service with minimal effort. Wants easily visible display of ordered items and prices. Wants proof of purchase to support returns.  - Company: Wants to accurately record transactions and satisfy customer interests. Wants to ensure that Payment Authorization Service payment receivables are recorded.  - Manager: Wants to be able to quickly perform override operations, and easily debug shopkeeper problems.  **Preconditions**:   * Cashier is identified and authenticated. * Check the availability of stock |

**Success Guarantee** (or Postconditions):

User successfully purchased the order. Sale is saved. Successful payment. Commissions recorded. Receipt is generated. Payment authorization approvals are recorded.

**Main Success Scenario (or Basic Flow):**

1. Customer goes to the CMS to purchase order
2. User wants to purchase the order
3. System shows the list of products to the user
4. The system provides the user with the opportunity to select the product
5. The user select product which he or she wants to purchased
6. The system retrieves the details for the chosen product including product availability, product quality and show details to user
7. The user confirms that he or she wants to purchase the order
8. System verify the order and allow user to purchased order

**Extensions (or Alternative Flows):**

\*a. At any time, Manager will manage all the details of system

\*b. At any time, System fails:

To support recovery and correct accounting, ensure all transaction sensitive state and events can be recovered from any step of the scenario.

1. manager restarts System, logs in, and requests recovery of all data
2. System recover all the information

2a. System detects anomalies preventing recovery:

* 1. System signals error to the manager, records the error, and enters a clean state
  2. Manager starts a new sale.

1a. Customer or Manager indicate to resume a suspended sale.

1. Cashier performs resume operation, and enters the ID to retrieve the sale.
2. System displays the state of the resumed sale, with subtotal.

2a. Sale not found.

* 1. System signals error to the manager
  2. manager probably starts new sale and re-enters all items.
  3. Manager continues with sale (probably entering more items or handling payment).

2a. customer successfully purchased the order

1. If the customer successfully purchased the order system request for payment method

2b. customer fails to purchased the order

1. If the customer fails to purchased the order system show message to customers that items is not available

**Special Requirements:**

. Text must be visible from 1 meter.

- user authorization response within 30 seconds 90% of the time.

- system fails then recover immediately

- Language internationalization on the text displayed.

-use color theme that is easily visible

- Id and password must required to registered the account

-Use at least one capital letter in login

-Password shows as a hidden digits.

-If wrong entry at id or password use red bar below the text

**Technology and Data Variations List**:

1. Manager entering an authorization code via the keyboard.
2. Item identifier entered by bar code laser scanner (if bar code is present) or keyboard.
3. Item identifier may be any UPC, EAN, JAN, or SKU coding scheme.
4. Credit account information entered by card reader or keyboard.
5. Credit payment signature captured on paper receipt
6. Use java language
7. Design use cases in NetBeans software using GUI swing

Frequency of Occurrence: Could be nearly continuous.

**Open Issues:**

* Unprofessional design of software
* Login failures
* No category suggestions, photos or prices on sales events
* Bad user experience
* Lack of personalization
* Missing or fake products reviews
* Missing products information
* Too complex check-out process
* Payment failures
* Lack of security and privacy leak
* Not having flexible return policy

| Use Case UC2: RETURN ITEM |
| --- |
| **Scope**: cosmetic shop management  **Level**: user level  **Primary** **Actor**: Customer  **Stakeholders and Interests**:  shopkeeper: Wants accurate, fast entry, and no payment errors, as cash drawer shortages are deducted from his/her salary.  - Customer: Wants to return item and fast service with minimal effort. Wants proof of purchase to support returns.  - Company: Wants to accurately record transactions and satisfy customer interests. Wants to ensure that Payment Authorization Service payment receivables are recorded. Wants some fault tolerance to allow sales capture even if server components (e.g., remote credit validation) are unavailable. Wants automatic and fast update of accounting and inventory.  - Manager: Wants to be able to quickly perform override operations, and easily debug Cashier problems.  **Preconditions**:   * Customer is identified and authenticated. * Customer return item if deadline of returning item is not pass |

**Success Guarantee** (or Postconditions): customer successfully return the items. receipt generate. Return payment to customer

**Main Success Scenario (or Basic Flow):**

1. Customer goes to the CMS to return item
2. Customer wants to return the item
3. Customer asks the system to return item
4. System check that the return item is OK or not
5. System set the deadline for returning orders if the deadline pass send SORRY message to customer
6. If the deadline is not pass system allows to the customer to return the order
7. Customer request to system to return payment of returned item
8. System return the payment to the customer of return product

**Extensions (or Alternative Flows):**

\*a. At any time, Manager will manage all the details of system

\*b. At any time, System fails:

To support recovery and correct accounting, ensure all transaction sensitive state and events can be recovered from any step of the scenario.

1. manager restarts System, logs in, and requests recovery of all data
2. System recover all the information

2a. System detects anomalies preventing recovery:

* 1. System signals error to the manager, records the error, and enters a clean state
  2. Manager starts a new sale.

1a. Customer or Manager indicate to resume a suspended sale.

1. Cashier performs resume operation, and enters the ID to retrieve the sale.
2. System displays the state of the resumed sale, with subtotal.

2a. Sale not found.

* 1. System signals error to the manager
  2. manager probably starts new sale and re-enters all items.
  3. Manager continues with sale (probably entering more items or handling payment).

2a. Customer fails to return item

* System send message to customer that order will not return
* System send payment back to the customer

2b. Customer successfully returns the item

* System allow the customer to return item
* System ask for the payment process

**Special Requirements:**

. Text must be visible from 1 meter.

- user authorization response within 30 seconds 90% of the time.

- system fails then recover immediately

- Language internationalization on the text displayed.

-use color theme that is easily visible

- id and password must required to registered the account

-Use atleast one capital letter in login

-Password shows as a hidden digits

-If wrong entry at id or password use red bar below the text

**Technology and Data Variations List**:

1. Manager entering an authorization code via the keyboard.
2. Item identifier entered by bar code laser scanner (if bar code is present) or keyboard.
3. Item identifier may be any UPC, EAN, JAN, or SKU coding scheme.
4. Credit account information entered by card reader or keyboard.
5. Credit payment signature captured on paper receipt
6. Use java language
7. Design use cases in NetBeans software using GUI swing

Frequency of Occurrence: Could be nearly continuous.

**Open Issues:**

* Unprofessional design of software
* Login failures
* No category suggestions, photos or prices on sales events
* Bad user experience
* Lack of personalization
* Missing or fake products reviews
* Missing products information
* Too complex check-out process
* Payment failures
* Lack of security and privacy leak
* Not having flexible return policy

| Use Case UC1: Add to Cart |
| --- |
| **Scope**: cosmetic management system  **Level**: user goal  **Primary** **Actor**: Customer  **Stakeholders and Interests**:  - Customer: Wants accurate, fast entry, and no payment errors, and add order to cart.  - Shopkeeper: Wants sales commissions updated.  - Customer: Wants purchase and fast service with minimal effort. Wants easily visible display of entered items and prices. Wants proof of purchase to support returns.  - Company: Wants to accurately record transactions and satisfy customer interests. Wants to ensure that Payment Authorization Service payment receivables are recorded  - Manager: Wants to be able to quickly perform override operations, and easily debug Cashier problems.  **Preconditions**:   * Customer is identified and authenticated. * If stock is not available customer is unable to add order to cart |

**Success Guarantee** (or Postconditions): Sale is saved. Payment is correctly calculated. Commissions recorded. Receipt is generated. Payment authorization approvals are recorded. customer successfully add order to cart

**Main Success Scenario (or Basic Flow):**

1. Customer goes to the CMS to add item to cart
2. customer wants to add products to cart
3. customer request the system to show items
4. system accept the request of customer and show products to customer
5. customer view all the products and choose item
6. customer request the system to add chosen item to cart
7. system verify that chosen item available in stock
8. after verification system allow customer to add item to cart
9. system also shows the price of items to customer
10. customer after payment add item to cart

**Extensions (or Alternative Flows):**

\*a. At any time, Manager will manage all the details of system

\*b. At any time, System fails:

To support recovery and correct accounting, ensure all transaction sensitive state and events can be recovered from any step of the scenario.

1. manager restarts System, logs in, and requests recovery of all data
2. System recover all the information

2a. System detects anomalies preventing recovery:

* 1. System signals error to the manager, records the error, and enters a clean state
  2. Manager starts a new sale.

1a. Customer or Manager indicate to resume a suspended sale.

1. Cashier performs resume operation, and enters the ID to retrieve the sale.
2. System displays the state of the resumed sale, with subtotal.

2a. Sale not found.

* 1. System signals error to the manager
  2. manager probably starts new sale and re-enters all items.
  3. Manager continues with sale (probably entering more items or handling payment).

2a. Customer fails to add items to cart

* System send message to customer that order unable to add to cart
* System send payment back to the customer

2b. Customer successfully add order to cart

* System allow the customer to add order to cart
* System ask for the payment process

**Special Requirements:**

. Text must be visible from 1 meter.

- user authorization response within 30 seconds 90% of the time.

- system fails then recover immediately

- Language internationalization on the text displayed.

-use color theme that is easily visible

- id and password must required to registered the account

-Use atleast one capital letter in login

-Password shows as a hidden digits

-If wrong entry at id or password use red bar below the text

**Technology and Data Variations List**:

1. Manager entering an authorization code via the keyboard.
2. Item identifier entered by bar code laser scanner (if bar code is present) or keyboard.
3. Item identifier may be any UPC, EAN, JAN, or SKU coding scheme.
4. Credit account information entered by card reader or keyboard.
5. Credit payment signature captured on paper receipt
6. Use java language
7. Design use cases in NetBeans software using GUI swing

Frequency of Occurrence: Could be nearly continuous.

**Open Issues:**

* Unprofessional design of software
* Login failures
* No category suggestions, photos or prices on sales events
* Bad user experience
* Lack of personalization
* Missing or fake products reviews
* Missing products information
* Too complex check-out process
* Payment failures
* Lack of security and privacy leak
* Not having flexible return policy

| Use Case UC1: Replace item |
| --- |
| **Scope**: cosmetic management system  **Level**: user goal  **Primary** **Actor**: Customer  **Stakeholders and Interests**:  - Customer: Wants accurate, fast entry, and no payment errors, and replace item  - Shopkeeper: Wants sales commissions updated.  - Customer: Wants purchase and fast service with minimal effort. Wants easily visible display of entered items and prices. Wants proof of purchase to support returns.  - Company: Wants to accurately record transactions and satisfy customer interests. Wants to ensure that Payment Authorization Service payment receivables are recorded.  - Manager: Wants to be able to quickly perform override operations, and easily debug problems.  **Preconditions**:   * Cashier is identified and authenticated. * If stock is not available customer is unable to replace item |

**Success Guarantee** (or Postconditions): Sale is saved. Tax is correctly calculated. New items are updated. Commissions recorded. Receipt is generated. Payment authorization approvals are recorded. customer successfully replace the item

**Main Success Scenario (or Basic Flow):**

1. Customerarrives at CMS to replace item
2. Customer wants to replace item
3. Customer request the system to show other items so he or she can replace item with another item
4. System accept the request of customer and shows other items to replace
5. Customer view all items and choose another ite which he or she wants to replace
6. System checks that new items is available is stock
7. If stock available system allows the customer to replace item
8. Is stock is not available system also have to show msg to customer
9. System also shows price of product against new item
10. If new item is less in price than other item than system return payment to customer
11. If chosen item is greater in price than replace item so system asks for payment from customer
12. Customer pays for the product and successfully replace the item

**Extensions (or Alternative Flows):**

\*a. At any time, Manager will manage all the details of system

\*b. At any time, System fails:

To support recovery and correct accounting, ensure all transaction sensitive state and events can be recovered from any step of the scenario.

1. manager restarts System, logs in, and requests recovery of all data
2. System recover all the information

2a. System detects anomalies preventing recovery:

* 1. System signals error to the manager, records the error, and enters a clean state
  2. Manager starts a new sale.

1a. Customer or Manager indicate to resume a suspended sale.

1. Cashier performs resume operation, and enters the ID to retrieve the sale.
2. System displays the state of the resumed sale, with subtotal.

2a. Sale not found.

* 1. System signals error to the manager
  2. manager probably starts new sale and re-enters all items.
  3. Manager continues with sale (probably entering more items or handling payment).

2a. Customer fails to replace item

* System send message to customer that order will not replace
* System send payment back to the customer

2b. Customer successfully replace the item

* System allow the customer to replace item
* System ask for the payment process

**Special Requirements:**

. Text must be visible from 1 meter.

- user authorization response within 30 seconds 90% of the time.

- system fails then recover immediately

- Language internationalization on the text displayed.

-use color theme that is easily visible

- id and password must required to registered the account

-Use atleast one capital letter in login

-Password shows as a hidden digits

-If wrong entry at id or password use red bar below the text

- …

**Technology and Data Variations List**:

1. Manager entering an authorization code via the keyboard.
2. Item identifier entered by bar code laser scanner (if bar code is present) or keyboard.
3. Item identifier may be any UPC, EAN, JAN, or SKU coding scheme.
4. Credit account information entered by card reader or keyboard.
5. Credit payment signature captured on paper receipt
6. Use java language
7. Design use cases in NetBeans software using GUI swing

Frequency of Occurrence: Could be nearly continuous.

**Open Issues:**

* Unprofessional design of software
* Login failures
* No category suggestions, photos or prices on sales events
* Bad user experience
* Lack of personalization
* Missing or fake products reviews
* Missing products information
* Too complex check-out process
* Payment failures
* Lack of security and privacy leak
* Not having flexible return policy

| Use Case UC1: Delete item |
| --- |
| **Scope**: cosmetic management system  **Level**: user goal  **Primary** **Actor**: Customer  **Stakeholders and Interests**:  - Customer: Wants accurate, fast entry, and no payment errors  - Salesperson: Wants sales commissions updated. And able to delete item from stock  - Customer: Wants purchase and fast service with minimal effort. Wants easily visible display of entered items and prices. Wants proof of purchase to support returns.  - Company: Wants to accurately record transactions and satisfy customer interests. Wants to ensure that Payment Authorization Service payment receivables are recorded  - Manager: Wants to be able to quickly perform override operations, and easily debug problems.  **Preconditions**:   * shopkeeper is identified and authenticated * shopkeeper can delete item if stock is short. |

**Success Guarantee** (or Postconditions): Sale is saved. Tax is correctly calculated. Accounting and Inventory are updated. Commissions recorded. Receipt is generated. Payment authorization approvals are recorded.

**Main Success Scenario (or Basic Flow):**

1. user check the CMS
2. user checks the stock availability
3. if the stock is not available user have to be delete item from CMS
4. user request to system to delete that item
5. after verification system allow the user to delete item

**Extensions (or Alternative Flows):**

\*a. At any time, Manager will manage all the details of system

\*b. At any time, System fails:

To support recovery and correct accounting, ensure all transaction sensitive state and events can be recovered from any step of the scenario.

1. manager restarts System, logs in, and requests recovery of all data
2. System recover all the information

2a. System detects anomalies preventing recovery:

* 1. System signals error to the manager, records the error, and enters a clean state
  2. Manager starts a new sale.

1a. Customer or Manager indicate to resume a suspended sale.

1. Cashier performs resume operation, and enters the ID to retrieve the sale.
2. System displays the state of the resumed sale, with subtotal.

2a. Sale not found.

* 1. System signals error to the manager
  2. manager probably starts new sale and re-enters all items.
  3. Manager continues with sale (probably entering more items or handling payment).

2a. Customer fails to delete item

* System send message to customers that order is not available
* System send payment back to the customer

2b. manager successfully delete the item

* System allow the user to delete item

**Special Requirements:**

. Text must be visible from 1 meter.

- user authorization response within 30 seconds 90% of the time.

- system fails then recover immediately

- Language internationalization on the text displayed.

-use color theme that is easily visible

- id and password must required to registered the account

-Use atleast one capital letter in login

-Password shows as a hidden digits

-If wrong entry at id or password use red bar below the text

**Technology and Data Variations List**:

1. Manager entering an authorization code via the keyboard.
2. Item identifier entered by bar code laser scanner (if bar code is present) or keyboard.
3. Item identifier may be any UPC, EAN, JAN, or SKU coding scheme.
4. Credit account information entered by card reader or keyboard.
5. Credit payment signature captured on paper receipt
6. Use java language
7. Design use cases in NetBeans software using GUI swing

Frequency of Occurrence: Could be nearly continuous.

**Open Issues:**

* Unprofessional design of software
* Login failures
* No category suggestions, photos or prices on sales events
* Bad user experience
* Lack of personalization
* Missing or fake products reviews
* Missing products information
* Too complex check-out process
* Payment failures
* Lack of security and privacy leak
* Not having flexible return policy

### Umama Noor (FA20-BSE-055)

| Use Case UC1: Place Order |
| --- |
| **Scope**: cosmetic management system  **Level**: user goal  **Primary** **Actor**: customer  **Stakeholders and Interests**:  - Customer: Wants purchase and fast service with minimal effort. Wants easily visible display of ordered items and prices.  - Company: wants to ensure that order is successfully placed by the customer.  - Manager: Wants to be able to quickly perform override operations, and easily debug customer problems.  **Preconditions**  User is identified and authenticated.   * User has selected the items to be purchased. |

**Success Guarantee** (or Postconditions):

Customer place order successfully.

The order will be placed in the system.

**Main Success Scenario (or Basic Flow):**

1.Customer opens the CMS to place the order.

2.Customer send requests the system to show the list of products.

3. System shows the list to the customer.

4. Customer searches the item they want and selects the products from the list and add product to cart.

5.Customer requests the system to place order of selected items and also asks for the payment method.

6. System accept the order that customer request and checks that product is available or not and deliver product.

7.Customer receives the product and pay for it.

**Extensions (or Alternative Flows):**

\*a. At any time, Manager will manage all the details of system

\*b. At any time, System fails:

To support recovery and correct accounting, ensure all transaction sensitive state and events can be recovered from any step of the scenario.

1. manager restarts System, logs in, and requests recovery of all data
2. System recover all the information

2a. System detects anomalies preventing recovery:

1. System signals error to the manager, records the error, and enters a clean state
2. Manager starts a new sale.

1a. Customer or Manager indicate to resume a suspended sale.

1. Cashier performs resume operation, and enters the ID to retrieve the sale.
2. System displays the state of the resumed sale, with subtotal.

2a. Sale not found.

1. System signals error to the manager
2. manager probably starts new sale and re-enters all items.
3. Manager continues with sale (probably entering more items or handling payment).

2a. customer successfully placed the order

1. If the customer successfully placed the order system request for payment method

2b. customer fails to place the order

1. If the customer fails to place the order system show message to customers that items is not available or out of stock.

**Special Requirements:**

. Text must be visible from 1 meter.

- user authorization response within 30 seconds 90% of the time.

- system fails then recover immediately

- Language internationalization on the text displayed.

-use color theme that is easily visible

- Id and password must required to registered the account

-Use at least one capital letter in login

-Password shows as a hidden digits.

-If wrong entry at id or password use red bar below the text

**Technology and Data Variations List**:

1. Manager entering an authorization code via the keyboard.
2. Item identifier entered by bar code laser scanner (if bar code is present) or keyboard.
3. Item identifier may be any UPC, EAN, JAN, or SKU coding scheme.
4. Credit account information entered by card reader or keyboard.
5. Credit payment signature captured on paper receipt
6. Use java language
7. Design use cases in NetBeans software using GUI swing.

**Open Issues:**

* Login failures
* Bad user experience
* Missing products information
* Complex check-out process
* Payment failures
* Not having flexible return policy

| Use Case UC1: track Order |
| --- |
| **Scope**: cosmetic management system  **Level**: user goal  **Primary** **Actor**: customer  **Stakeholders and Interests**:  - Customer: Wants purchase and fast service with minimal effort. Wants easily visible display of ordered items and prices.  *- Company: wants to ensure that order is successfully placed by the customer.*  *- Manager: Wants to be able to quickly perform override operations, and easily debug customer problems.*  **Preconditions**  User is identified and authenticated. |

**Success Guarantee** (or Postconditions):

The user will have a tracking ID for the order.

**Main Success Scenario (or Basic Flow):**

1.Customer selects the item from the item list to place the order.

2. System will request for payment.

3.The user will provide payment and shipping information.

4.The system will respond with confirmation of the order and a tracking number that the user can use to check on order status in the future. After verification order will be delivered to the customer’s address.

**Extensions (or Alternative Flows):**

\*a. At any time, Manager will manage all the details of system

\*b. At any time, System fails:

To support recovery and correct accounting, ensure all transaction sensitive state and events can be recovered from any step of the scenario.

1. manager restarts System, logs in, and requests recovery of all data
2. System recover all the information

2a. System detects anomalies preventing recovery:

1. System signals error to the manager, records the error, and enters a clean state
2. Manager starts a new sale.

1a. Customer or Manager indicate to resume a suspended sale.

1. Cashier performs resume operation, and enters the ID to retrieve the sale.
2. System displays the state of the resumed sale, with subtotal.

2a. Sale not found.

1. System signals error to the manager
2. manager probably starts new sale and re-enters all items.
3. Manager continues with sale (probably entering more items or handling payment).

2a. customer enters the address, order-id and order will be tracked.

2b. if order-id is invalid then user will not be able to track the order.

**Special Requirements:**

. Text must be visible from 1 meter.

- user authorization response within 30 seconds 90% of the time.

- system fails then recover immediately

- Language internationalization on the text displayed.

-use color theme that is easily visible

- Id and password must required to registered the account

-Use at least one capital letter in login

-Password shows as a hidden digits.

-If wrong entry at id or password use red bar below the text

**Technology and Data Variations List**:

1. Manager entering an authorization code via the keyboard.
2. Item identifier entered by bar code laser scanner (if bar code is present) or keyboard.
3. Item identifier may be any UPC, EAN, JAN, or SKU coding scheme.
4. Credit account information entered by card reader or keyboard.
5. Credit payment signature captured on paper receipt
6. Use java language
7. Design use cases in NetBeans software using GUI swing.

**Open Issues:**

* Login failures
* Bad user experience
* Missing products information
* Complex check-out process
* Payment failures
* Not having flexible return policy

| Use Case UC1: Process sale |
| --- |
| **Scope**: cosmetic management system  **Level**: user goal  **Primary** **Actor**: Cashier  **Stakeholders and Interests**:  Cashier: Wants accurate, fast entry, and no payment errors, as cash drawer shortages are deducted from his/her  salary.  - Cashier: Wants accurate, fast entry, and no payment errors, as cash drawer shortages are deducted from his/her  salary.  - Cashier: Wants accurate, fast entry, and no payment errors, as cash drawer shortages are deducted from his/her  salary.  - Cashier: Wants accurate, fast entry, and no payment errors, as cash drawer shortages are deducted from his/her  salary.  - Cashier: Wants accurate, fast entry, and no payment errors, as cash drawer shortages are deducted from his/her salary.  - Customer: Wants purchase and fast service with minimal effort. Wants easily visible display of ordered items and prices.  - Company: wants to ensure that order is successfully placed by the customer.  - Manager: Wants to be able to quickly perform override operations, and easily debug customer problems.  **Preconditions**  Cashier is identified and authenticated |

**Success Guarantee** (or Postconditions):

Receipt is generated. Payment authorization approvals are recorded.

Sale is saved. Tax is correctly calculated. Accounting and Inventory are

updated. Commissions recorded. Receipt is generated. Payment authorization approvals are recorded

Sale is saved. Tax is correctly calculated. Accounting and Inventory are

updated. Commissions recorded. Receipt is generated. Payment authorization approvals are recorded

Sale is saved. Tax is correctly calculated. Accounting and Inventory are

updated. Commissions recorded. Receipt is generated. Payment authorization approvals are recorded

Sale is saved. Tax is correctly calculated. Accounting and Inventory are

updated. Commissions recorded. Receipt is generated. Payment authorization approvals are recordedSale is saved. Receipt is generated. Payment authorization approvals are recorded.

**Main Success Scenario (or Basic Flow):**

1. Customer arrives at POS checkout with goods and/or services to purchase.
2. Cashier starts a new sale.
3. Cashier enters item identifier.
4. System records sale line item and presents item description, price, and running total. Price calculated from a set of price rules.
5. System presents total with taxes calculated.
6. Cashier tells Customer the total, and asks for payment.
7. Customer pays and System handles payment.
8. System logs completed sale and sends sale and payment information to the external Accounting system (for accounting and commissions) and Inventory system (to update inventory).
9. System presents receipt.
10. Customer leaves with receipt and goods.

**Extensions (or Alternative Flows):**

\*a. At any time, Manager requests an override operation:

1. System enters Manager-authorized mode.

2. Manager or Cashier performs one Manager-mode operation. e.g., cash balance change, resume a suspended sale on another register, void a sale, etc.

3. System reverts to Cashier-authorized mode.

\*b. At any time, System fails:

To support recovery and correct accounting, ensure all transaction sensitive state and events can be recovered from any step of the scenario.

1. Cashier restarts System, logs in, and requests recovery of prior state.

2. System reconstructs prior state.

2a. System detects anomalies preventing recovery:

1. System signals error to the Cashier, records the error, and enters a clean state.

2. Cashier starts a new sale.

1a. Customer or Manager indicate to resume a suspended sale.

1. Cashier performs resume operation, and enters the ID to retrieve the sale.

2. System displays the state of the resumed sale, with subtotal.

2a. Sale not found.

1. System signals error to the Cashier.

2. Cashier probably starts new sale and re-enters all items.

3. Cashier continues with sale (probably entering more items or handling payment).

**Special Requirements:**

. Text must be visible from 1 meter.

- user authorization response within 30 seconds 90% of the time.

- system fails then recover immediately

- Language internationalization on the text displayed.

-use color theme that is easily visible

- Id and password must required to registered the account

-Use at least one capital letter in login

-Password shows as a hidden digits.

-If wrong entry at id or password use red bar below the text

**Technology and Data Variations List**:

1. Manager entering an authorization code via the keyboard.
2. Item identifier entered by bar code laser scanner (if bar code is present) or keyboard.
3. Item identifier may be any UPC, EAN, JAN, or SKU coding scheme.
4. Credit account information entered by card reader or keyboard.
5. Credit payment signature captured on paper receipt
6. Use java language
7. Design use cases in NetBeans software using GUI swing.

**Open Issues:**

* Login failures
* Bad user experience
* Missing products information
* Complex check-out process
* Payment failures
* Not having flexible return policy

| Use Case UC1: manage Order |
| --- |
| **Scope**: cosmetic management system  **Level**: user goal  **Primary** **Actor**: customer, manager  **Stakeholders and Interests**:  - Customer: Wants purchase and fast service with minimal effort. Wants easily visible display of ordered items and prices.  - Company: wants to ensure that order is successfully placed by the customer.  - Manager: Wants to be able to quickly perform override operations, and easily debug customer problems.  **Preconditions**  User is identified and authenticated.  Check the availability of products. |

**Success Guarantee** (or Postconditions):

Orders are successfully managed by the manager.

**Main Success Scenario (or Basic Flow):**

1.Manager will manage the products and check if stock is available or not.

2. Customer wants to buy the item and request the system to show the details of the items.

3.System will show the details of the products to the customer.

**Extensions (or Alternative Flows):**

\*a. At any time, Manager will manage all the details of system

\*b. At any time, System fails:

To support recovery and correct accounting, ensure all transaction sensitive state and events can be recovered from any step of the scenario.

1. manager restarts System, logs in, and requests recovery of all data
2. System recover all the information

2a. System detects anomalies preventing recovery:

1. System signals error to the manager, records the error, and enters a clean state
2. Manager starts a new sale.

1a. Customer or Manager indicate to resume a suspended sale.

1. Cashier performs resume operation, and enters the ID to retrieve the sale.
2. System displays the state of the resumed sale, with subtotal.

2a. Sale not found.

1. System signals error to the manager
2. manager probably starts new sale and re-enters all items.
3. Manager continues with sale (probably entering more items or handling payment).

**Special Requirements:**

. Text must be visible from 1 meter.

- user authorization response within 30 seconds 90% of the time.

- system fails then recover immediately

- Language internationalization on the text displayed.

-use color theme that is easily visible

- Id and password must required to registered the account

-Use at least one capital letter in login

-Password shows as a hidden digits.

-If wrong entry at id or password use red bar below the text

**Technology and Data Variations List**:

1. Manager entering an authorization code via the keyboard.
2. Item identifier entered by bar code laser scanner (if bar code is present) or keyboard.
3. Item identifier may be any UPC, EAN, JAN, or SKU coding scheme.
4. Credit account information entered by card reader or keyboard.
5. Credit payment signature captured on paper receipt
6. Use java language
7. Design use cases in NetBeans software using GUI swing.

**Open Issues:**

* Login failures
* Bad user experience
* Missing products information
* Complex check-out process
* Payment failures
* Not having flexible return policy

| Use Case UC1: Payment |
| --- |
| **Scope**: cosmetic management system  **Level**: user goal  **Primary** **Actor**: customer, manager  **Stakeholders and Interests**:  - Customer: will pay and buy order  - Manager: will receive payment  **Preconditions**  User is identified and authenticated. |

**Success Guarantee** (or Postconditions):

Payment is done by customer successfully and order is purchased.

**Main Success Scenario (or Basic Flow):**

1.Customer place an order and requests for the payment.

2.System displays the payment method to the customers.

3.System allows the customer to pay via cards or cash on delivery.

4.Customer pays for the ordered items and Manager will receive the payment.

**Extensions (or Alternative Flows):**

\*a. At any time, Manager will manage all the details of system

\*b. At any time, System fails:

To support recovery and correct accounting, ensure all transaction sensitive state and events can be recovered from any step of the scenario.

1. manager restarts System, logs in, and requests recovery of all data
2. System recover all the information

2a. System detects anomalies preventing recovery:

1. System signals error to the manager, records the error, and enters a clean state
2. Manager starts a new sale.

1a. Customer or Manager indicate to resume a suspended sale.

1. Cashier performs resume operation, and enters the ID to retrieve the sale.
2. System displays the state of the resumed sale, with subtotal.

2a. Sale not found.

1. System signals error to the manager
2. manager probably starts new sale and re-enters all items.
3. Manager continues with sale (probably entering more items or handling payment).

2a.Customer gives payment through the card and card is valid then order will be placed successfully.

2b. if card is invalid or there is no enough amount in the card then customer will try another method.

**Special Requirements:**

. Text must be visible from 1 meter.

- user authorization response within 30 seconds 90% of the time.

- system fails then recover immediately

- Language internationalization on the text displayed.

-use color theme that is easily visible

- Id and password must required to registered the account

-Use at least one capital letter in login

-Password shows as a hidden digits.

-If wrong entry at id or password use red bar below the text

**Technology and Data Variations List**:

1. Manager entering an authorization code via the keyboard.
2. Item identifier entered by bar code laser scanner (if bar code is present) or keyboard.
3. Item identifier may be any UPC, EAN, JAN, or SKU coding scheme.
4. Credit account information entered by card reader or keyboard.
5. Credit payment signature captured on paper receipt
6. Use java language
7. Design use cases in NetBeans software using GUI swing.

**Open Issues:**

* Login failures
* Bad user experience
* Missing products information
* Complex check-out process
* Payment failures
* Not having flexible return policy

### Farhan Khan (FA20-BSE-069)

Fully dress Use case

Use Case UC1: Add stock

**Scope**: cosmetic management system

**USE Case:** Add stock

Level: User-goal

Actor: manager, shopkeeper

Stack holder:

**Manager:** add stock to the store manage all the stock.

**Shopkeeper:** Also have the right to Add stock in the store.

**Pre-condition:**

Manager and shopkeeper both log in successfully in the store app. check the available space and customer requirements.

**Post-condition:**

The stock is recorded with date and time, store is updated, report of the stock generated, and the payments records are updated.

**Main success scenario:**

The manager and shopkeeper both login to the cosmetic store app According to the available space and customer requirements manager add stock in the cosmetic store All the payments are done from manager side. The stock is stored with date and time. The report of the stock is generated through generate report The record of the stock is saved, and store is updated.

**Extensions (or Alternative Flows):**

\*a. At any time when the Manager tries to add stock:

1. Manager/Shopkeeper checks the unavailable stock.
2. Manager/shopkeeper requests system to add stock.
3. System allows Manager/shopkeeper to add stock.
4. Stocks are added successfully.

**Special Requirements:**

1. Manager/shopkeeper must be registered with the system.
2. Stock should not be in store already.
3. Color scheme should be used which is clearly visible.

**Technology and Data Variations List**:

In this project we Design use cases in NetBeans IDE using GUI swing by java language.

**USE Case: Return defective**

Use Case UC1: Return defective

**Scope**: cosmetic management system

**Level:** user goal

**Actor:** manager

Stack holder:

**Manager:** Stop using the items which are defective and return the defective items from where he purchased.

**Pre-condition:**

The system is working well and the manager login to the system from its domain.

**Post-condition:**

Update the stock record with date and time and generate fresh report of the store

**Main success scenario:**

The manager login to the store app from its domain. The manager checks the stock and if the items are defective, then delete it from the store database. Manager contacts the seller from where he/she buy the items and return it to them.

**Extensions (or Alternative Flows):**

a. At any time when the Manager/shopkeeper tries to return defective:

1. Manager checks the defective stock.
2. Manager requests system to delete defective stocks.
3. System allows Manager to delete defective stocks.
4. Stocks return successfully.

**Special Requirements:**

1. Manager must be registered with the system.
2. Color scheme should be used which is clearly visible.

**Technology and Data Variations List**:

In this project we Design use cases in NetBeans IDE using GUI swing by java language.

**Open Issues:**

* Login failures
* Not having flexible return policy

**USE Case: Refill Store**

Use Case UC1: Refill store

**Scope**: cosmetic management system

**Level:** user goal

**Actor:** Manager

Stack holder:

**Manager:** See the available space in the store warehouse place order for all the item which need to be refilled according to the customer requirement.

**Pre-condition:**  Manager login to the store app from its domain and see the items quantity and available space

**Post-condition:** The items is stored, and the store is updated with date and time, and report are generated with date and time.

**Main success scenario:**

Manager login to the cosmetic app from its domain. Manager checks the stock and if the store is empty than he/she will place order for the items which is required in the store. Report of the required items is generated and the manager act according to the report and refill the store. The items information stored in the database with its id, name, brand, price and the store continue its selling.

**Extensions (or Alternative Flows):**

a. At any time when the Manager tries to refill store:

1. Manager checks the store inventory.
2. Manager requests system to generate report of the item which are missing.
3. System allows Manager to see the report.
4. Store refill successfully.

**Special Requirements:**

1. Manager/shopkeeper must be registered with the system.
2. Color scheme should be used which is clearly visible.

**Technology and Data Variations List**:

In this project we Design use cases in NetBeans IDE using GUI swing by java language.

### Bilal Khan (FA20-BSE-071)

**Fully dressed use case: Generate Report**

|  |
| --- |
| **Scope:** Generate Report  **Level:** User goal  **Primary Actor :** Shopkeeper  **Stack Holder & Interests:**  Shopkeeper: wants to know about the monthly sales report ,most sold products ,Return products and number of visited customers, most sales time.  **Pre-Condition**: Shopkeeper is successfully logged in. Database System is running.  **Post-condition:** The requested report is generated and saved into a CSV file with full details. A printable PDF file is created and the final report is generated on the screen.  **Main Success Scenario:**  **1.**Shopekeeper login into System.  **2. S**tarts the generate report process.  3. Entered The required information query.  4.The system displays the data on the screen and ask for confirmation.  5. After conformation the report is display on the screen and exsorted in pdf form.  **Extensions :**  If the data is fails because of missing data or other related-data so their will be error occurs  so   1. Admin starts backup and recovery process. 2. Recovers all the data of customers. 3. And in the last restart the report process.   **Special Requirements:**   1. The generated report must be in a proper format. 2. The data must be in a sequences.   **Tecnology and data Variation List:**  1.The generated report must be in pdf or CSV or in both cases.    **Frequency of Occurrence:**  Requested By shopkeeper .  As request by |

**Fully dressed level: view sale report**

|  |
| --- |
| **Scope:** View sales Report  **Level:** User goal  **Primary Actor :** Admin  **Stack Holder & Interests:**  Admin wants to know about the monthly sales report, most sold products ,Return products and number of visited customers.  **Pre-Condition**: Admin is successfully logged into system.  **Post-condition:** The requested report is generated and saved into a CSV file with full details. A printable PDF file is created and the final report is generated on the screen. Admin view the report about products of most sold defected and return .  **Main Success Scenario:**  **1.**Admin login into System .  **2. S**tarts the view report process .  3. Entered The required information query.  4.The system displays the data on the screen and ask for confirmation.  5. After conformation the report is display on the screen and exsorted in pdf form.  **Extensions :**  If the data is fails because of missing data or other related-data so their will be error occurs  so   1. Admin starts backup and recovery process. 2. Recovers all the data of customers. 3. And in the last restart the report process.   **Special Requirements:**   1. The view report must be in a proper format. 2. The data must be in a sequences.   **Tecnology and data Variation List:**  1.The generated report must be in pdf or CSV or in both cases.  **Frequency of Occurrence:** |

**Fully Dressed :View profit report**

**Scope:** View profit report

**Level: User goal**

**Primary Actor :** Shopkeeper

**Stack Holder & Interests:**

**Shopkeeper:** Shopkeeper generated a report of profit .

**Manager :**  After a generated report manager order more products .

**Owner:** In last owner want to review the report which generated by shopkeeper .

**Pre-Condition:**  The system is working well and both the manager and shopkeeper login to the system from its domain.

**Post-condition:** The requested report is generated and saved into a CSV file with full details. A printable PDF file is created and the final report is generated on the screen.

**Main Success Scenario:**

**1.**Shopekeeper login into System.

**2. S**tarts the generate report process.

3. Entered The required information query.

4.The system displays the data on the screen and ask for confirmation.

5. After conformation the report is display on the screen and exsorted in pdf form.

**Extensions :**

If the data is fails because of missing data or other related-data so their will be error occurs

so

1. Admin starts backup and recovery process.
2. Recovers all the data of customers.
3. And in the last restart the report process.

**Special Requirements:**

5.Manager/shopkeeper must be registered with the system.

6.Color scheme should be used which is clearly visible.

**Tecnology and data Variation List:**

In this project we Design use cases in NetBeans IDE using GUI swing by java language.

**Frequency of Occurrence:**  Requested By shopkeeper .

As request by

**Fully Dressed :View loss report**

**Scope:** View loss report

**Level: User goal**

**Primary Actor :** Shopkeeper

**Stack Holder & Interests:**

**Shopkeeper:** Shopkeeper generated a report of loss .

**Manager :**  After a generated report manager return that products which is going in loss .

**Owner:** See the report send by manager of loss and order the manager to make sales on products.

**Pre-Condition:**  The system is working well and both the manager and shopkeeper login to the system from its domain.

**Post-condition:** The requested report is generated and saved into a CSV file with full details. A printable PDF file is created and the final report is generated on the screen.

**Main Success Scenario:**

**1.**Shopekeeper login into System.

**2. S**tarts the generate report process.

3. Entered The required information query.

4.The system displays the data on the screen and ask for confirmation.

5. After conformation the report is display on the screen and exsorted in pdf form.

**Extensions :**

If the data is fails because of missing data or other related-data so their will be error occurs

so

1. Admin starts backup and recovery process.
2. Recovers all the data of customers.
3. And in the last restart the report process.

**Special Requirements:**

1. Manager/shopkeeper must be registered with the system.
2. Color scheme should be used which is clearly visible.

**Tecnology and data Variation List:**

1.The loss report must be in pdf or CSV or in both cases.

**Frequency of Occurrence:**  Requested By shopkeeper .

**Fully Dressed :Edit report**

**Scope:** Edit report

**Level:** User goal

**Primary Actor :** Shopkeeper

**Stack Holder & Interests:**

**Shopkeeper:** Shopkeeper generated a report of edit.

**Pre-Condition:**  The system is working well and both the manager and shopkeeper login to the system from its domain.

**Post-condition:** Update the report of edit with date and time and send report to the customer.

**Main Success Scenario:**

**1.**Shopekeeper login into System.

**2. S**tarts edit report process.

3. Entered The required information query.

4.The system displays the data on the screen and ask for confirmation.

5. After conformation the report is display on the screen and exsorted in pdf form.

**Extensions :**

If the data is fails because of missing data or other related-data so their will be error occurs

so

1. Admin starts backup and recovery process.
2. Recovers all the data of customers.
3. And in the last restart the report process.

**Special Requirements:**

1. Manager/shopkeeper must be registered with the system.
2. Color scheme should be used which is clearly visible.

**Tecnology and data Variation List:**

In this project we Design use cases in NetBeans IDE using GUI swing by java language.

.

**Frequency of Occurrence:**  Requested By shopkeeper .

### 2.4 NetBeans Screenshots

FARHAN KHAN (FA20-BSE-069)

* + 1. ADD STOCK

Graphical user interface

Description automatically generated

* + 1. RETURN DEFECTIVE

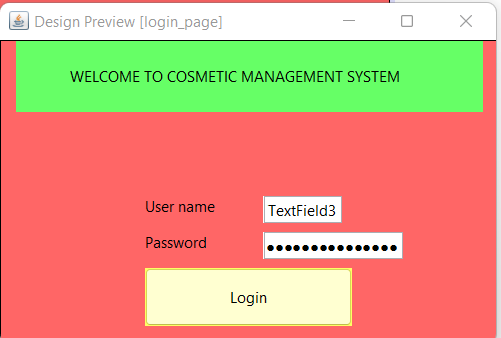
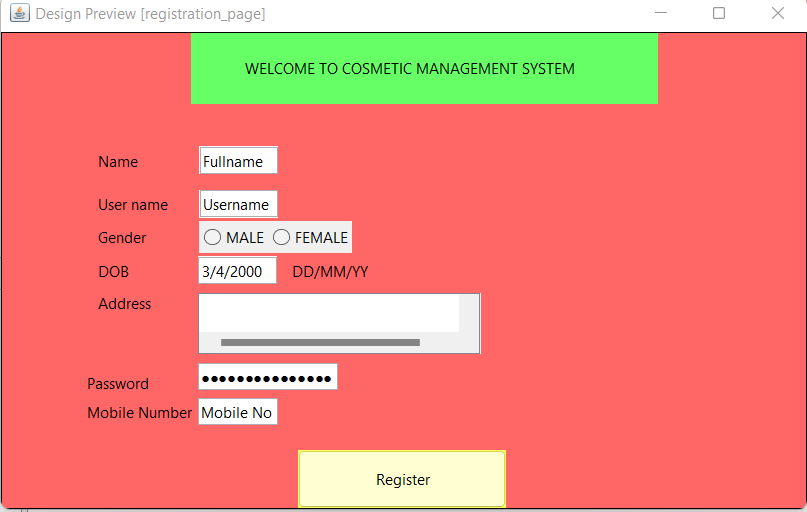
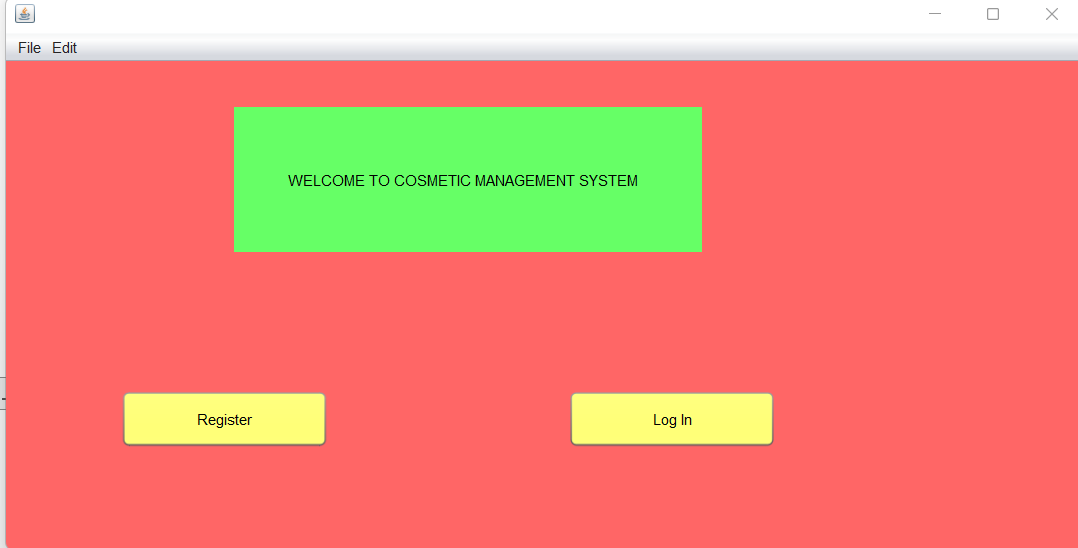
Graphical user interface

Description automatically generated

* + 1. REFILL STORE

Graphical user interface, application

Description automatically generated



Graphical user interface

Description automatically generated Graphical user interface, application

Description automatically generated

Graphical user interface, website

Description automatically generated

Graphical user interface, application

Description automatically generated

Graphical user interface, website

Description automatically generated

Graphical user interface

Description automatically generated Graphical user interface

Description automatically generated Graphical user interface

Description automatically generated Graphical user interface, application

Description automatically generated

Graphical user interface

Description automatically generated

Table

Description automatically generated

Graphical user interface, application, Word

Description automatically generated

Table

Description automatically generated

Table

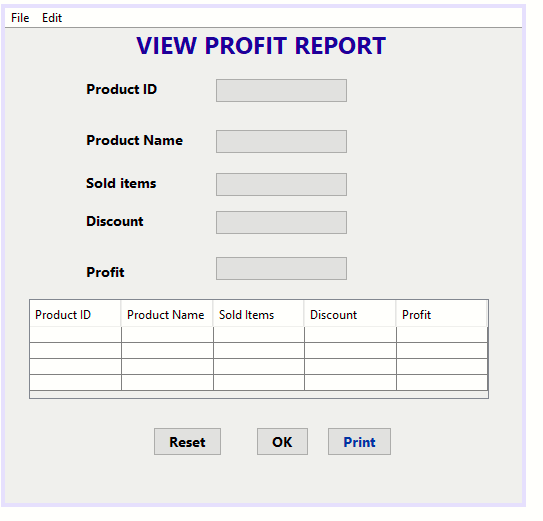
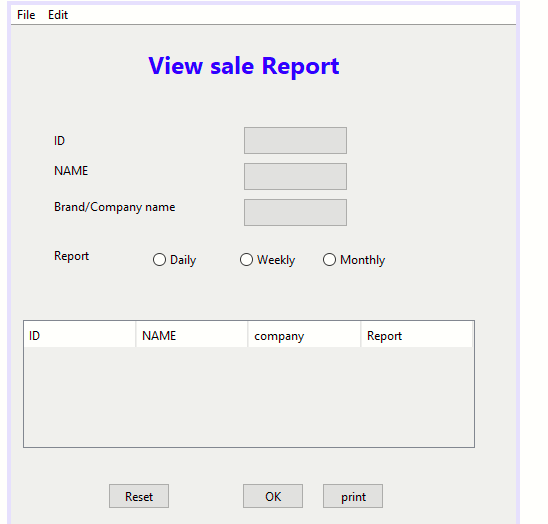
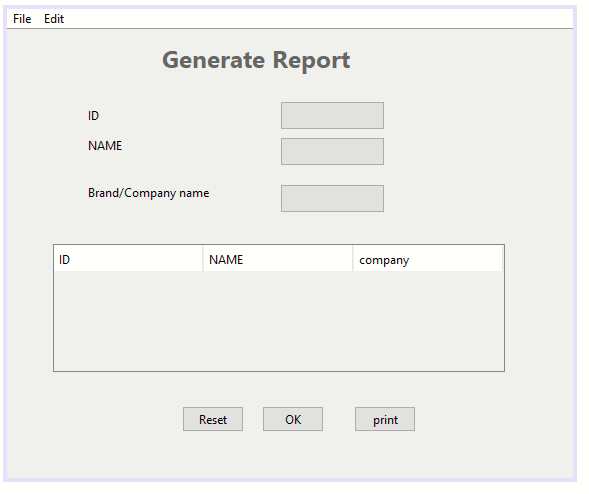
Description automatically generated

Graphical user interface, application

Description automatically generated

Graphical user interface, application

Description automatically generated



# CHAPTER 3 Domain Model

## 3.1 Introduction

### Talah Khan (FA20-BSE-042)

A picture containing text, map, receipt, document

Description automatically generated

### Osama Khan (FA20-BSE-047)

Diagram

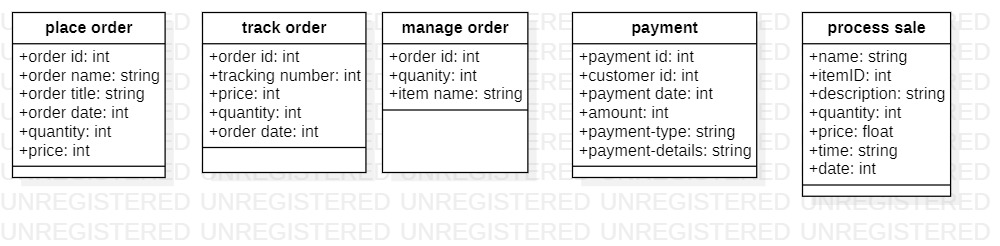
Description automatically generated

### Sabahat Siddique (FA20-BSE-054)

Diagram

Description automatically generated

### Umama Noor (FA20-BSE-055)



### Farhan Khan (FA20-BSE-069)

Diagram, schematic

Description automatically generated

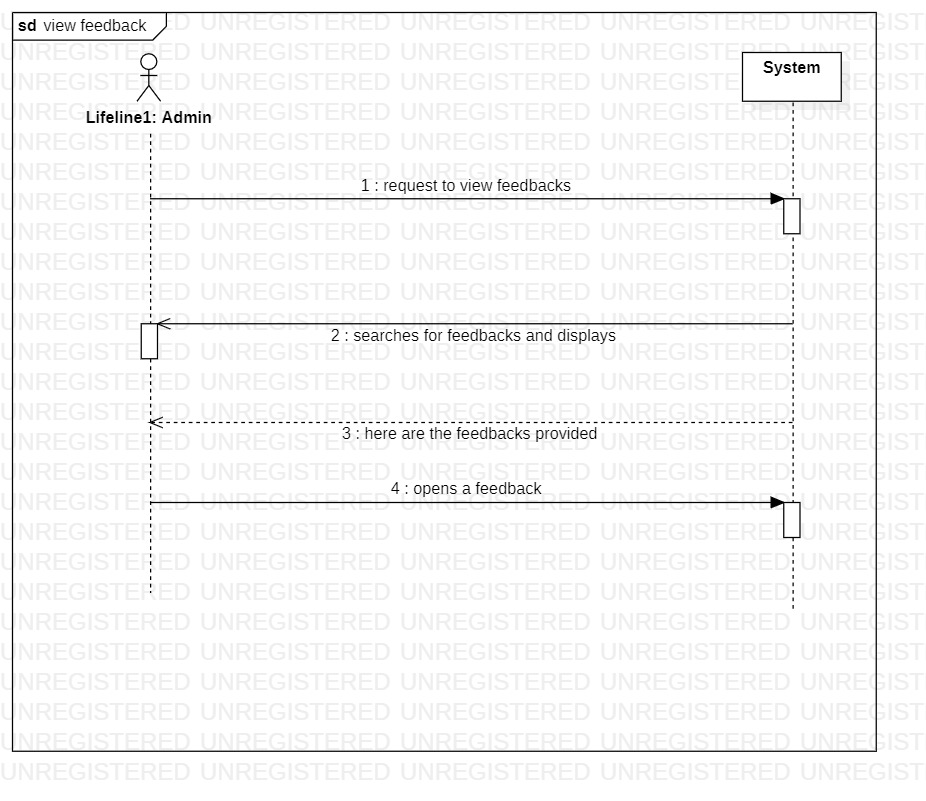
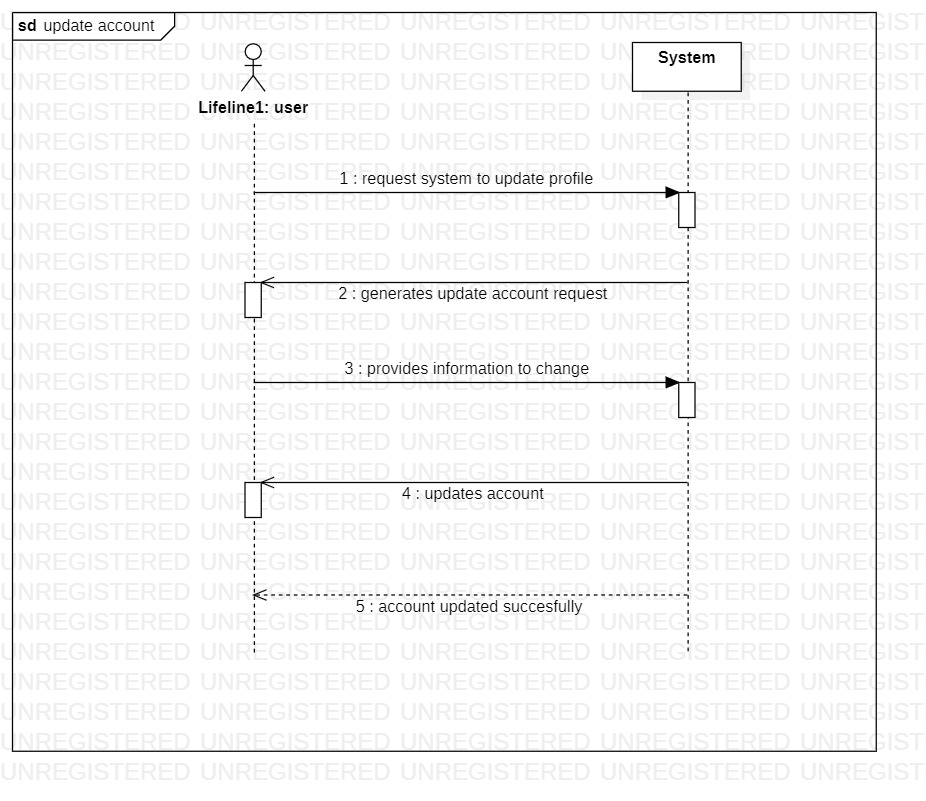
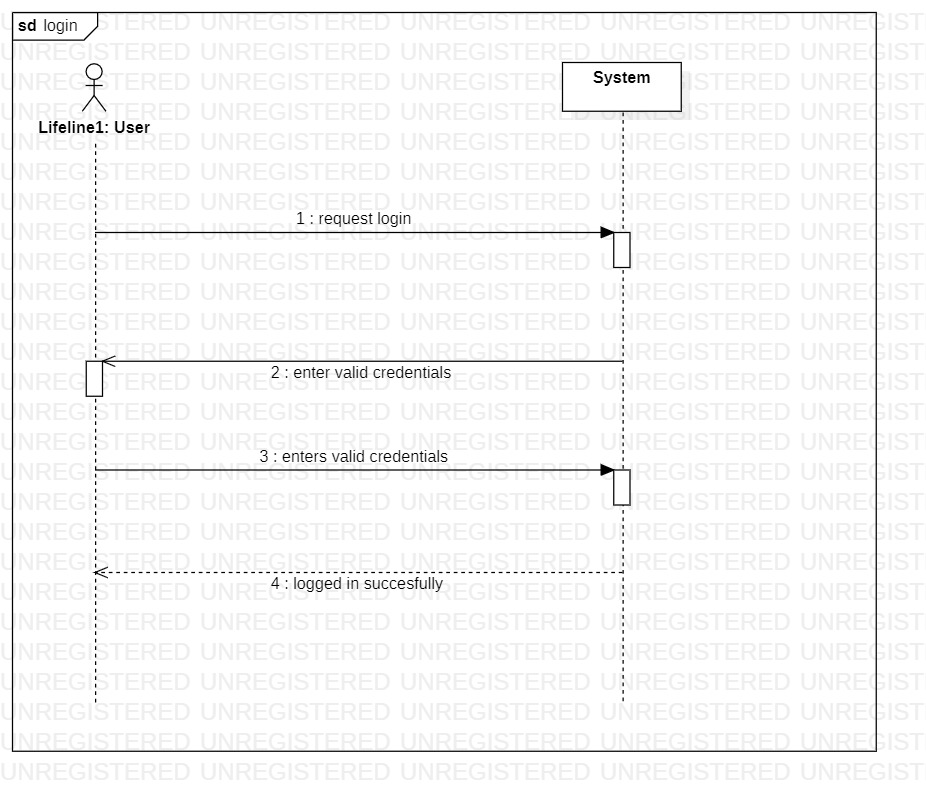
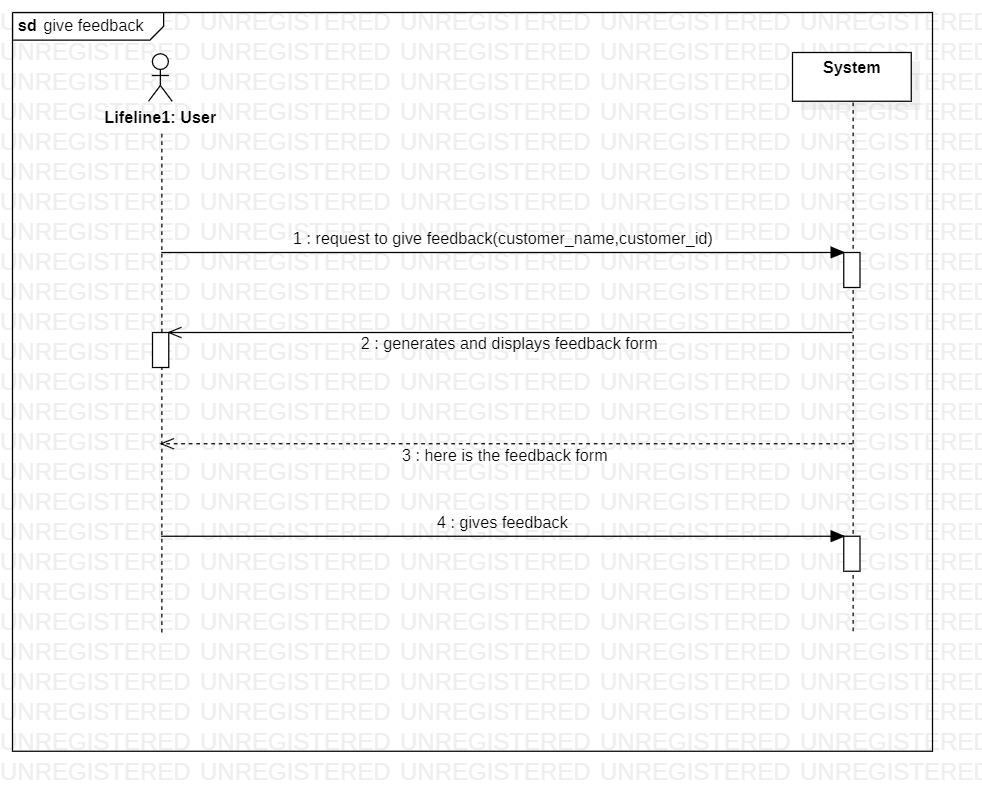
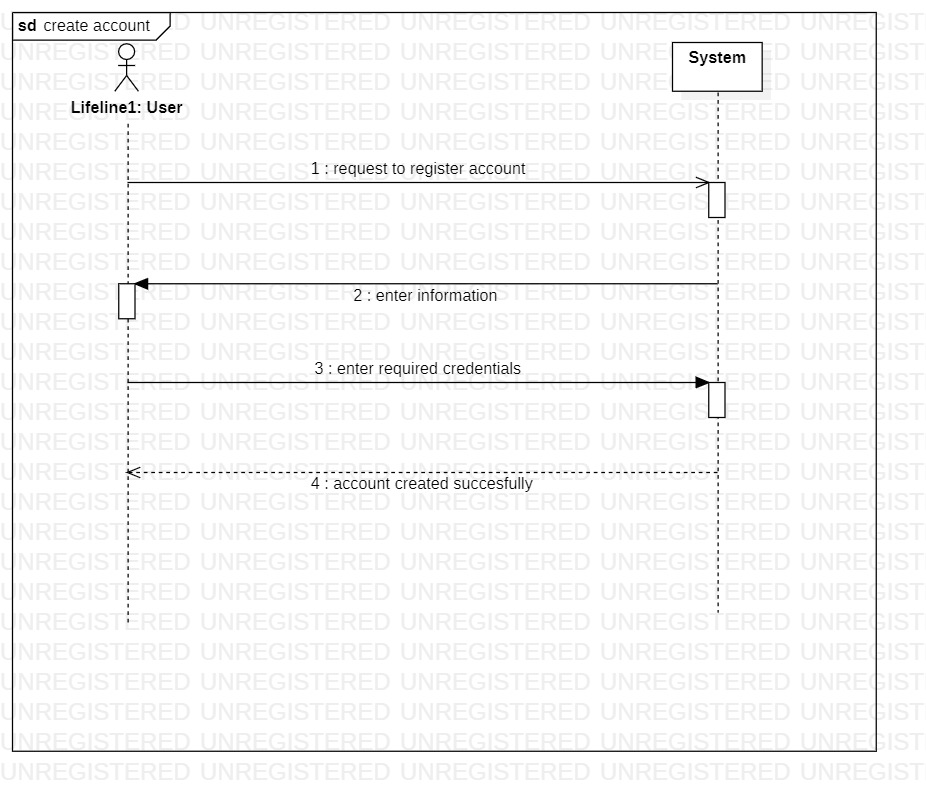
### Bilal Khan (FA20-BSE-071)



# CHAPTER 4 System Sequence Diagram

## 4.1 Introduction

### Talah Khan (FA20-BSE-042)



### Osama Khan (FA20-BSE-047)

**Searching item**

**Table

Description automatically generated**

**Add item**

Table

Description automatically generated

**Add Category**

**A picture containing table

Description automatically generated**

**View item**

**Graphical user interface, application, table, Excel

Description automatically generated**

**View category**

**A picture containing table

Description automatically generated**

### Sabahat Siddique (FA20-BSE-054)

Table

Description automatically generatedTable

Description automatically generatedTable

Description automatically generatedTable

Description automatically generatedDiagram, table

Description automatically generated with medium confidence

### Umama Noor (FA20-BSE-055)

Table

Description automatically generatedTable

Description automatically generatedTable

Description automatically generatedTable

Description automatically generatedTable

Description automatically generated

### Farhan Khan (FA20-BSE-069)

Chart

Description automatically generated

Table

Description automatically generated

Table

Description automatically generated

### Bilal Khan (FA20-BSE-071)

Table

Description automatically generated

# CHAPTER 5 Operation Contracts

## 5.1 Introduction

### Talah Khan (FA20-BSE-042)

|  |  |
| --- | --- |
| Contract CO 1: enter information | |
| Operation | enter information required |
| Cross References: | Use Cases: Register Account |
| Preconditions: | User is not registered |
| Postconditions: | Account is successfully registered |

|  |  |
| --- | --- |
| Contract CO 2: Enter required credentials | |
| Operation | Enter required credentials |
| Cross References: | Use Cases: register account |
| Preconditions: | User is not registered |
| Postconditions: | User account is successfully created |

|  |  |
| --- | --- |
| Contract CO 3: request login | |
| Operation | request login |
| Cross References: | Use Cases: login |
| Preconditions: | Account is created |
| Postconditions: | User is logged in successfully |

|  |  |
| --- | --- |
| Contract CO 4: enter valid credentials | |
| Operation | enter valid credentials |
| Cross References: | Use Cases: login |
| Preconditions: | User requests to login |
| Postconditions: | User is successfully logged in |

|  |  |
| --- | --- |
| Contract CO 5: request to update profile | |
| Operation | request to update profile |
| Cross References: | Use Cases: update profile |
| Preconditions: | User is logged in |
| Postconditions: | Update form generated |

|  |  |
| --- | --- |
| Contract CO 6: provide information to change | |
| Operation | provide information to change |
| Cross References: | Use Cases: update profile |
| Preconditions: | Update form is generated |
| Postconditions: | Profile is updated |

|  |  |
| --- | --- |
| Contract CO 7: request to give feedback | |
| Operation | request to give feedback |
| Cross References: | Use Cases: give feedback |
| Preconditions: | User is logged in |
| Postconditions: | Feedback form is generated |

|  |  |
| --- | --- |
| Contract CO 8: gives feedback | |
| Operation | Gives feedback |
| Cross References: | Use Cases: give feedback |
| Preconditions: | Feedback form is generated |
| Postconditions: | Feedback is saved |

### Osama Khan (FA20-BSE-047)

**Operation contract Searching item**

|  |
| --- |
| **Contract CO5: Request to searching item** |

|  |  |
| --- | --- |
| **Operation** | **User request to searching item** |
| **Cross reference** | **UC searching item** |
| **Precondition** | User login into the system |
| **Post condition** | User request is successfully done. |

|  |
| --- |
| **Contract CO1: searching Item** |

|  |  |
| --- | --- |
| **Operation** | **Searching item (item name, item id)** |
| **Cross reference** | **UC searching item** |
| **Precondition** | **User login into the system**  **User is identified and authenticated.** |
| **Post condition** | **User search is done**  **Search item is visible to user, manager, shopkeeper** |

**Operation contract: Add Item**

|  |
| --- |
| **Contract CO5: User request to add item** |

|  |  |
| --- | --- |
| **Operation** | **User request to add item** |
| **Cross reference** | **UC add item** |
| **Precondition** | **User login into the system**  **User is identified and authenticated.** |
| **Post condition** | **Request is successfully done.** |

|  |
| --- |
| **Contract CO2: Add Item** |

|  |  |
| --- | --- |
| **Operation** | **Add item (item name, item quality, item id, item quantity)** |
| **Cross reference** | **UC Add item** |
| **Precondition** | **User login into the system**  **User is identified and authenticated.** |
| **Post condition** | **Shopkeeper and manager add item. Item are Add successfully and visible for all** |

**Operation contract: Add Category**

|  |
| --- |
| **Contract CO5: User request Add Category** |

|  |  |
| --- | --- |
| **Operation** | **User request to add category** |
| **Cross reference** | **UC Add category** |
| **Precondition** | **User login into the system**  **User is identified and authenticated.** |
| **Post condition** | **Request is successfully done.** |

|  |
| --- |
| **Contract CO3: Add category** |

|  |  |
| --- | --- |
| **Operation** | **Add Category (category name, category quality, Category id, Category quantity)** |
| **Cross reference** | **UC Add Category** |
| **Precondition** | **User login into the system**  **User is identified and authenticated.** |
| **Post condition** | **Manager/Shopkeeper to add categories to the stock. Manager/Shopkeeper add those categories to the CMS successfully.** |

**Operation contract: View item**

|  |
| --- |
| **Contract CO5: User request view item** |

|  |  |
| --- | --- |
| **Operation** | **User request to view item.** |
| **Cross reference** | **UC View item** |
| **Precondition** | **User login into the system**  **User is identified and authenticated.** |
| **Post condition** | **User request is successful.** |

|  |
| --- |
| **Contract CO4: view Item** |

|  |  |
| --- | --- |
| **Operation** | **User request to view item** |
| **Cross reference** | **UC View item** |
| **Precondition** | **User login into the system**  **User is identified and authenticated.** |
| **Post condition** | **System successfully shows items. User manager and shopkeeper view all the items successfully.** |

**Operation contract: View Category**

|  |
| --- |
| **Contract CO5: User Request to view Category** |

|  |  |
| --- | --- |
| **Operation** | **User request to view Category** |
| **Cross reference** | **UC View category** |
| **Precondition** | **User login into the system**  **User is identified and authenticated.** |
| **Post condition** | **User request is successfully done.** |

|  |
| --- |
| **Contract CO5: view Category** |

|  |  |
| --- | --- |
| **Operation** | **User request to view Category**  **User Category (category name.id)** |
| **Cross reference** | **UC View category** |
| **Precondition** | **User login into the system**  **User is identified and authenticated.** |
| **Post condition** | **System successfully shows category. User manager and shopkeeper view all the category successfully.** |

### Sabahat Siddique (FA20-BSE-054)

|  |  |
| --- | --- |
| operation | Request to show item |
| Cross references | Purchase order |
| Preconditions: | user can view item if user is registered, and stock is available |
| Postconditions: | Select item and proceed to checkout |

|  |
| --- |
| Contract co2: request to show item |

|  |
| --- |
| Contract co: Select item and proceed to checkout |

|  |  |
| --- | --- |
| operation | Select item and proceed to checkout |
| Cross reference | Purchase order |
| Preconditions: | User select item if stock is available |
| Postconditions: | Order purchased successfully |

|  |  |
| --- | --- |
| operation | user request to place item |
| Cross reference | Replace item |
| Preconditions: | System allows the user to replace item if stock is available and must login to system |
| Postconditions: | Select new item to replace |

|  |
| --- |
| Contract co: user request to replace item |

|  |  |
| --- | --- |
| operation | Select new item to replace |
| Cross reference | Replace item |
| Preconditions: | User can select new item if stock is available |
| Postconditions: | Item replace successfully |

|  |
| --- |
| Contract co: Select new item to replace |

|  |
| --- |
| Contract co: user request to show item |

|  |  |
| --- | --- |
| operation | user request to show item |
| Cross reference | Add to cart |
| Preconditions: | System shows item to user if stock is available |
| Postconditions: | Select item and add to cart |

|  |
| --- |
| Contract co: Select item and add to cart |

|  |  |
| --- | --- |
| operation | Customer can select item to add to cart |
| Cross reference | Add to cart |
| Preconditions: | Customer can select item if they are available in stock |
| Postconditions: | Add item to cart successfully |

|  |
| --- |
| Contract co: user request to return item |

|  |  |
| --- | --- |
| operation | user request to return item |
| Cross reference | Return item |
| Preconditions: | Customer can return item if customer is login to customer |
| Postconditions: | Select request type |

|  |
| --- |
| Contract co: Select request type |

|  |  |
| --- | --- |
| operation | Select request type |
| Cross reference | Return item |
| Preconditions: | User can select request type if user is login to system |
| Postconditions: | Item return successfully |

|  |
| --- |
| Contract co: user request to delete item |

|  |  |
| --- | --- |
| operation | user request to delete item |
| Cross reference | Delete item |
| Preconditions: | User can delete item if they are login to system |
| Postconditions: | Select the item to delete |

|  |
| --- |
| Contract co: Select the item to delete |

|  |  |
| --- | --- |
| operation | User Selects the item to delete |
| Cross reference | Delete item |
| Preconditions: | User can delete item if they must registered there self to system |
| Postconditions: | User successfully delete item |

### Umama Noor (FA20-BSE-055)

|  |
| --- |
| Contract CO 1: user request to place order |

|  |  |
| --- | --- |
| **Operation** | user request to place order |
| **Cross reference** | Use Case: Place order |
| **Preconditions** | System checks that Stock is available to place order |
| **Postconditions** | User view and select items |

|  |
| --- |
| Contract CO2: user view and select items |

|  |  |
| --- | --- |
| **Operation** | user view and select items |
| **Cross reference** | Use Case: Place order |
| **Preconditions** | System check that the user is verify and stock is available to select items |
| **Postconditions** | Ordered place successfully |

|  |
| --- |
| Contract CO 3 : user request to manage order |

|  |  |
| --- | --- |
| **Operation** | user request to manage order |
| **Cross reference** | Use Cases: Manage order |
| **Preconditions** | System checks the availability of order to be managed |
| **Postconditions** | user select items to be managed |

|  |
| --- |
| Contract CO 4: user select items to be managed |

|  |  |
| --- | --- |
| **Operation** | user select items to be managed |
| **Cross reference** | Use Cases: Manage order |
| **Preconditions** | System checks the availability of order to be managed |
| **Postconditions** | Items managed suucessfully |

|  |
| --- |
| Contract CO 5: user request to track order |

|  |  |
| --- | --- |
| **Operation** | user request to track order |
| **Cross reference** | Use Cases: Track order |
| **Preconditions** | User must have a tracking id |
| **Postconditions** | Provide tracking details |

|  |
| --- |
| Contract CO 6: Provide tracking details |

|  |  |
| --- | --- |
| **Operation** | Provide tracking details |
| **Cross reference** | Use Cases: Track order |
| **Preconditions** | User must have a tracking id |
| **Postconditions** | Order track suucessfully |

|  |
| --- |
| Contract CO 7: user request for payment |

|  |  |
| --- | --- |
| **Operation** | user request for payment |
| **Cross reference** | Use Cases: Payment |
| **Preconditions** | User must have online account for payment |
| **Postconditions** | user select payment method |

|  |
| --- |
| Contract CO 8: user select payment method |

|  |  |
| --- | --- |
| **Operation** | user select payment method |
| **Cross reference** | Use Cases: Payment |
| **Preconditions** | User must have online account for payment |
| **Postconditions** | Payment is done |

|  |
| --- |
| Contract CO 9:make new sale |

|  |  |
| --- | --- |
| **Operation** | Make new sale |
| **Cross reference** | Use Cases: Process Sale |
| **Preconditions** | There is a sale underway |
| **Postconditions** | * A sale instance was created * Sale was associated with register |

|  |
| --- |
| Contract CO 10: enter item |

|  |  |
| --- | --- |
| **Operation** | enter item(itemID, quantity) |
| **Cross reference** | Use Cases: Process Sale |
| **Preconditions** | There is a sale underway |
| **Postconditions** | * A salesLineItem was created * salesLineItem was associated with current sale |

|  |
| --- |
| Contract CO 11: endSale |

|  |  |
| --- | --- |
| **Operation** | endSale() |
| **Cross reference** | Use Cases: Process Sale |
| **Preconditions** | There is a sale underway |
| **Postconditions** | Sale is complete |

|  |
| --- |
| Contract CO 12: makePayment |

|  |  |
| --- | --- |
| **Operation** | makePayment(amount: Money) |
| **Cross reference** | Use Cases: Process Sale |
| **Preconditions** | There is a sale underway |
| **Postconditions** | * A payment instance p was created * p was associated with current sale * The current sale was associated with the   store |

### Farhan Khan (FA20-BSE-069)

|  |  |  |
| --- | --- | --- |
| CONTRACT 1: ADD STOCK | |  |
| Operation | Add Stock (item, quantity, price) | |
| Cross References | Use Case: Add Stock | |
| Pre-Condition | Manager and shopkeeper login into the system  Check the available space | |
| Post-Condition | The store is updated with date and time and new report of the items is generated | |

|  |  |  |
| --- | --- | --- |
| CONTRACT 2: RETURN DEFECTIVE | |  |
| Operation | Return Defective (item\_name, item\_id, quantity, price, brand, categories) | |
| Cross References | Use Case: Return Defective | |
| Pre-Condition | The system is working well and the manager login to the system from its domain | |
| Post-Condition | Update the stock record with date and time and generate fresh report of the store. | |

|  |  |  |
| --- | --- | --- |
| CONTRACT 3: REFILL STORE | |  |
| Operation | Refill Store (Name, Product\_id , quantity, price, brand, categories) | |
| Cross References | Use Case: Refill Store | |
| Pre-Condition | See the available space in the store warehouse place order for all the item which need to be refilled according to the customer requirement. | |
| Post-Condition | The items are stored, and the store is updated with date and time, and report are generated with date and time. | |

### Bilal Khan (FA20-BSE-071)

|  |
| --- |
| **Contract CO1:** **View sales report** |

|  |  |
| --- | --- |
| Operation | View sales Report (Monthly,weekly,daily) |
| Cross reference | Use Case: View sales report |
| Pre-condition | User login successfully |
| Post-condition | View sale report is visible to manager,shopkeeper |

|  |
| --- |
| **Contract CO1:** **Duration Enter** |

|  |  |
| --- | --- |
| Operation | Duration enter(Monthly,weekly,daily) |
| Cross reference | Use Case:Duration entered |
| Pre-condition | User enter duration successfully |
| Post-condition | Duration is visible to manager , shopkeeper |

|  |
| --- |
| **Contract CO1:** **View report** |

|  |  |
| --- | --- |
| Operation | View Report |
| Cross reference | Use Case View report |
| Pre-condition | User login successfully |
| Post-condition | View report is visible to manager |

|  |
| --- |
| **Contract CO1:** **Request for login** |

|  |  |
| --- | --- |
| Operation | Request for login |
| Cross reference | Use Case request for login |
| Pre-condition | User login successfully |
| Post-condition | Customer is successfully logged in. Database System is running. |

# CHAPTER 6 Package Diagram

## 6.1 Introduction

### Talah Khan (FA20-BSE-042)



### Osama Khan (FA20-BSE-047)

### Sabahat Siddique (FA20-BSE-054)

### Umama Noor (FA20-BSE-055)

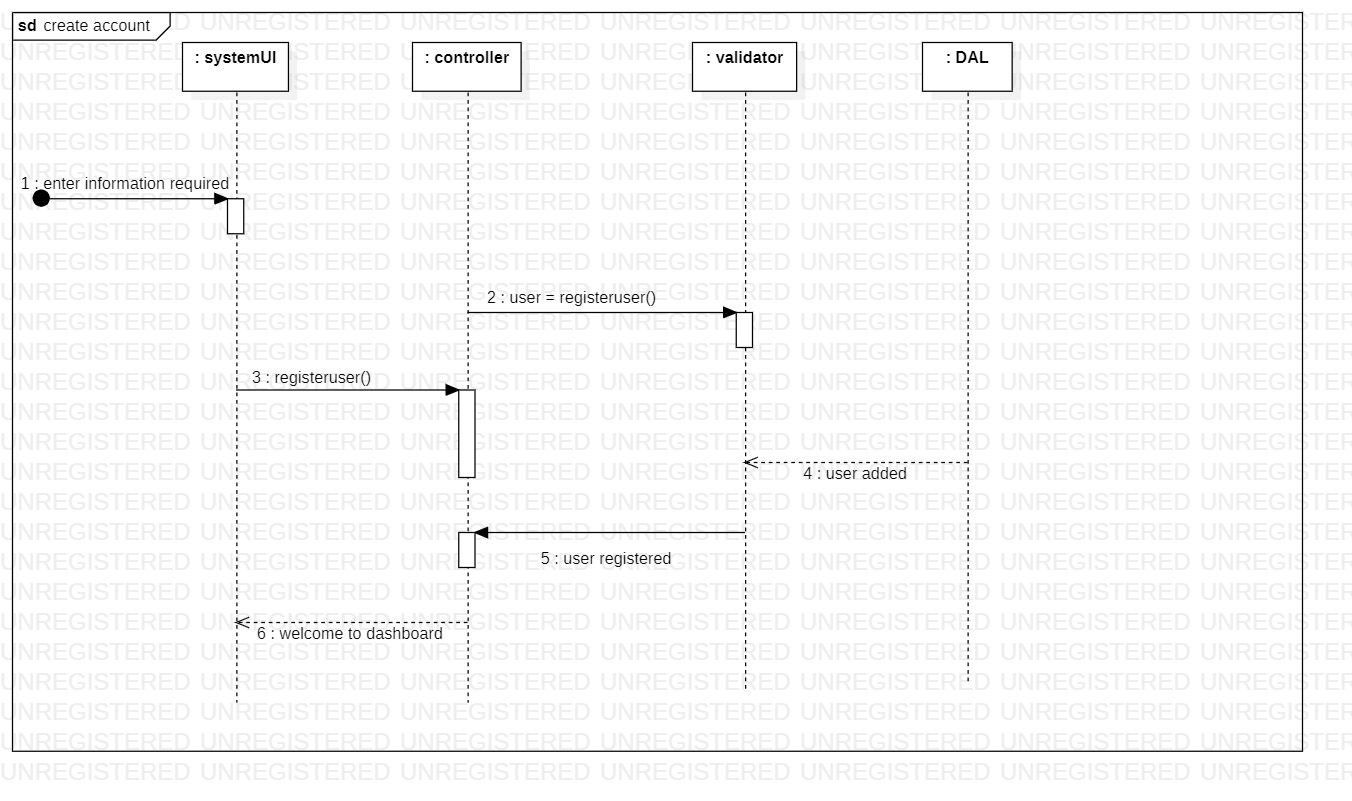
### Farhan Khan (FA20-BSE-069)

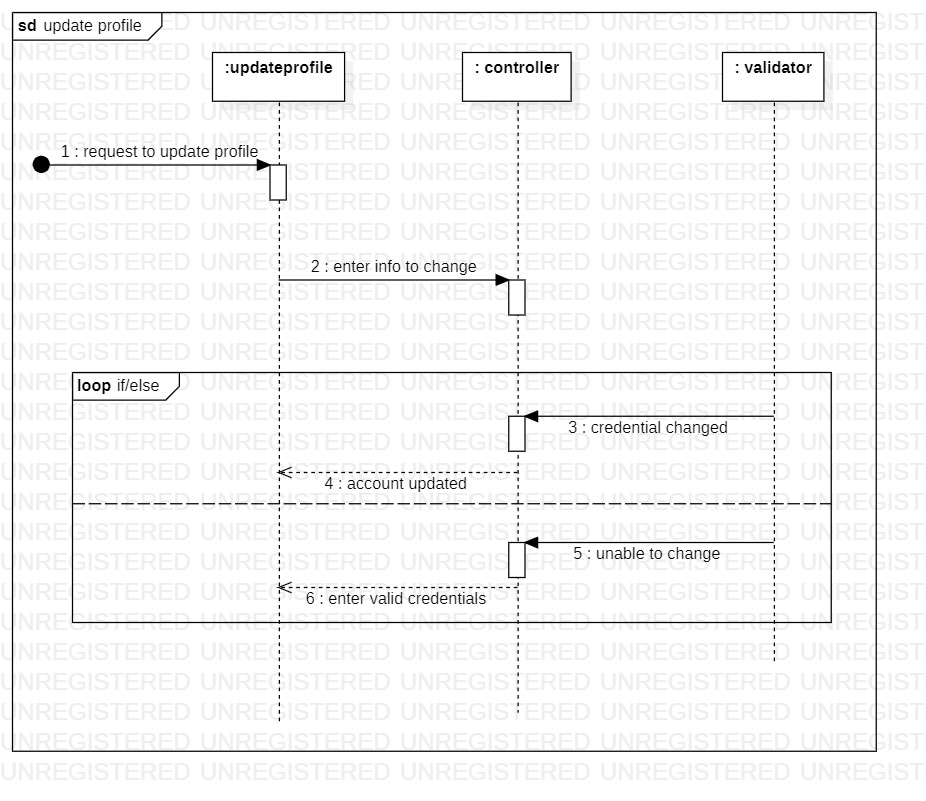
### Bilal Khan (FA20-BSE-071)

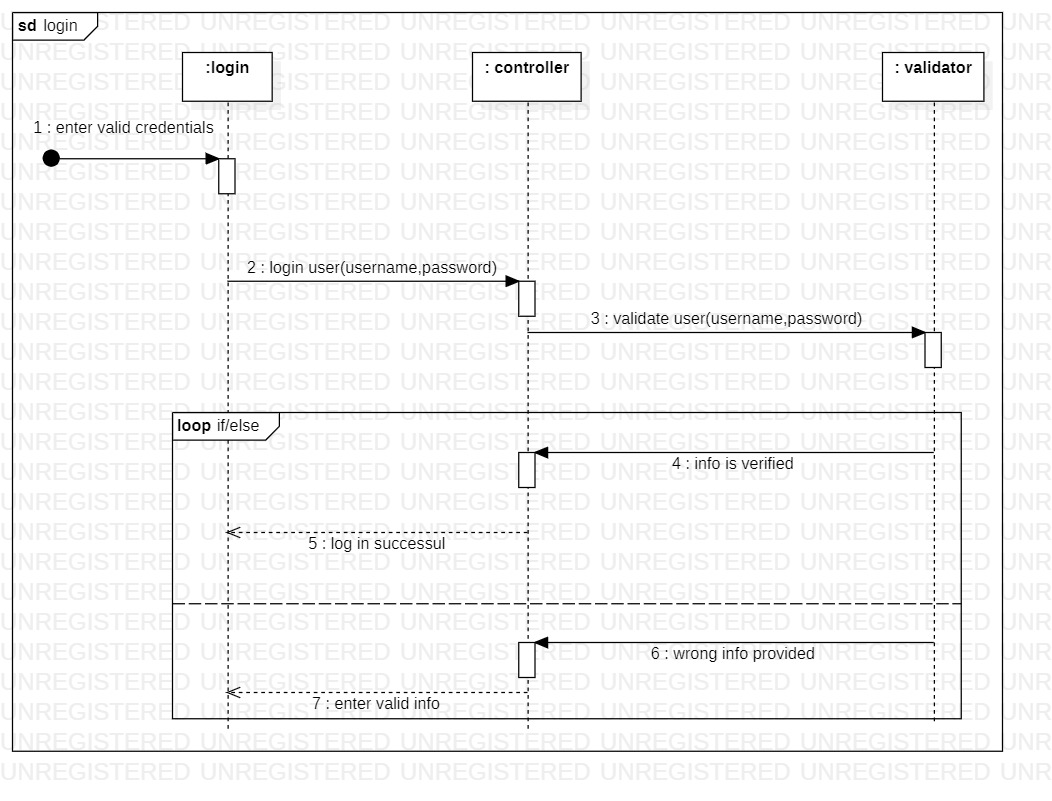
# CHAPTER 7 Interaction Diagrams

## 7.1 Sequence Diagram

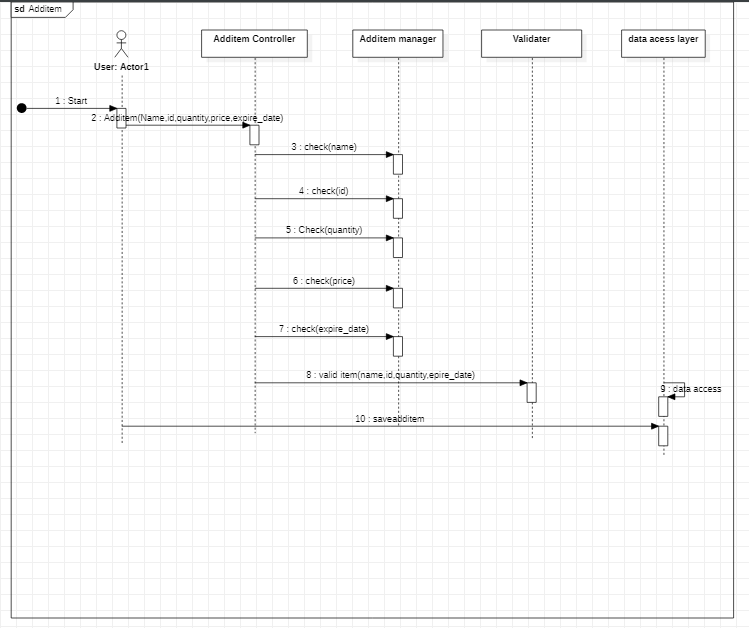
### Talah Khan (FA20-BSE-042)







### Osama Khan (FA20-BSE-047)



Chart

Description automatically generated

Chart

Description automatically generated with medium confidence

Diagram

Description automatically generated

A picture containing chart

Description automatically generated

### Sabahat Siddique (FA20-BSE-054)

### Umama Noor (FA20-BSE-055)

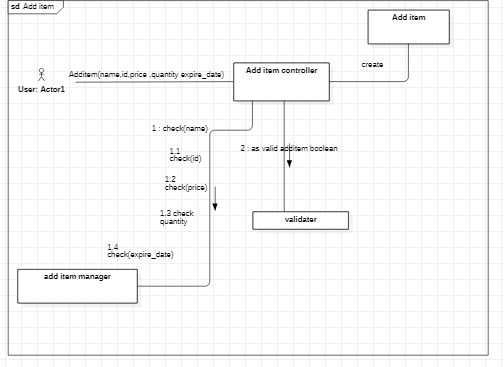
### Farhan Khan (FA20-BSE-069)

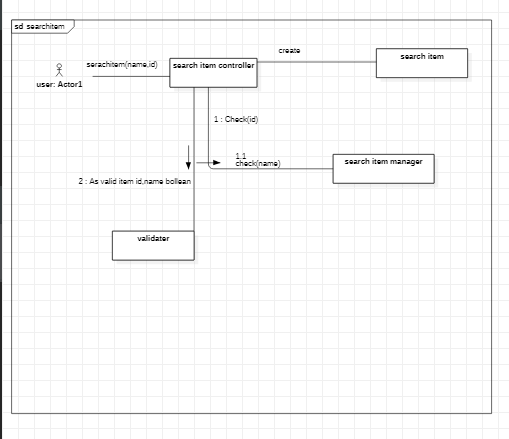
### Bilal Khan (FA20-BSE-071)

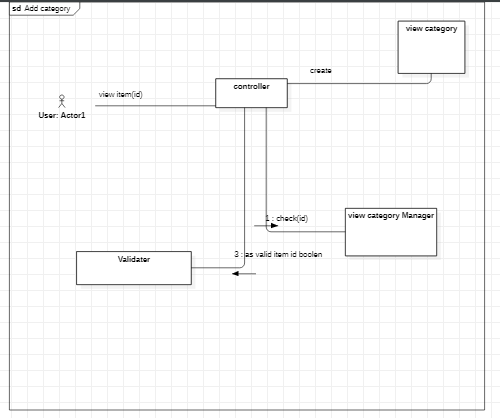
## 7.2 Communication Diagram

### Talah Khan (FA20-BSE-042)

### Osama Khan (FA20-BSE-047)







### Sabahat Siddique (FA20-BSE-054)

### Umama Noor (FA20-BSE-055)

### Farhan Khan (FA20-BSE-069)

### Bilal Khan (FA20-BSE-071)

# CHAPTER 8 Class Diagram

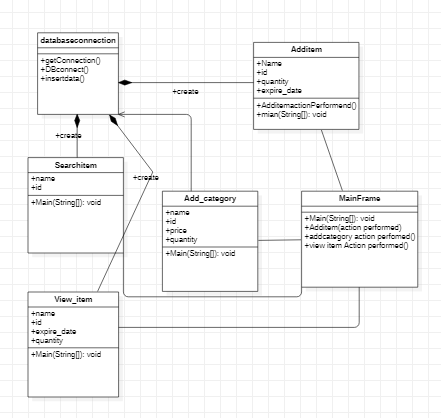
## 8.1 Class Diagram

### Talah Khan (FA20-BSE-042)

Diagram

Description automatically generated

### Osama Khan (FA20-BSE-047)



### Sabahat Siddique (FA20-BSE-054)

### Umama Noor (FA20-BSE-055)

### Farhan Khan (FA20-BSE-069)

### Bilal Khan (FA20-BSE-071)

# CHAPTER 9 Coding

## 9.1 coding with explanations

### Talah Khan (FA20-BSE-042)

### Osama Khan (FA20-BSE-047)

### Sabahat Siddique (FA20-BSE-054)

### Umama Noor (FA20-BSE-055)

### Farhan Khan (FA20-BSE-069)

### Bilal Khan (FA20-BSE-071)