Software Requirements Specification

for

<Project>

**Version 1.0 approved**

**Prepared by <Group No. 3>**

**<Section No. 5216>**

**<date created>**

**Group Members Class 2 Group 3**

1- Ritik Dhingana 19780

2- Areesha Kashif 19678

3- Saad Zuberi 18656

4- Syeda Sukaina Batool 19654

5- Syeda Faakeha Ahmed 18603

6- Hassam u din 19708

7- Hamza Naqi 15101

8- Tuba Nasir 18620

**Table of Contents**

[1. Introduction 1](#_Toc85142677)

[1.1 Purpose 1](#_Toc85142678)

[1.2 Document Conventions 1](#_Toc85142679)

[1.3 Intended Audience and Reading Suggestions 1](#_Toc85142680)

[1.4 Product Scope 1](#_Toc85142681)

[1.5 References 2](#_Toc85142682)

[2. Overall Description 2](#_Toc85142683)

[2.1 Product Perspective 2](#_Toc85142684)

[2.2 Product Functions 2](#_Toc85142685)

[2.3 User Classes and Characteristics 3](#_Toc85142686)

[2.4 Operating Environment 3](#_Toc85142687)

[2.5 Design and Implementation Constraints 4](#_Toc85142688)

[2.6 User Documentation 4](#_Toc85142689)

[2.7 Assumptions and Dependencies 4](#_Toc85142690)

[3. External Interface Requirements 4](#_Toc85142691)

[3.1 User Interfaces 4](#_Toc85142692)

[3.2 Hardware Interfaces 5](#_Toc85142693)

[3.3 Software Interfaces 5](#_Toc85142694)

[3.4 Communications Interfaces 5](#_Toc85142695)

[4. System Features 5](#_Toc85142696)

[4.1 System Feature 1: Account Management 5](#_Toc85142697)

[4.2 System Feature 2: Product Management 15](#_Toc85142698)

[4.2.1 Description and Priority 15](#_Toc85142699)

[4.2.2 Stimulus/Response Sequences 15](#_Toc85142701)

[4.2.3 Functional Requirements 16](#_Toc85142702)

[5. Other Nonfunctional Requirements 35](#_Toc85142703)

[5.1 Performance Requirements 35](#_Toc85142704)

[5.2 Safety Requirements 35](#_Toc85142705)

[5.3 Security Requirements 36](#_Toc85142706)

[5.4 Software Quality Attributes 36](#_Toc85142707)

[5.5 Business Rules 36](#_Toc85142708)

[6. Other Requirements 37](#_Toc85142709)

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
|  |  |  |  |
|  |  |  |  |

# Introduction

## Purpose

Our desktop and mobile application <name> facilitates the user by providing an online platform where they can engage in buying and selling of products. Each seller will be given credibility based on their past dealings and connections with other users. The potential buyer will be able to browse through the products alongside seller details which will assist them in making their decision. The admin will be managing the application by keeping track of sellers to ensure it is running smoothly.

## Document Conventions

Convention for main heading:

* Font: Times New Roman
* Font Style: Bold
* Font Size: 18

Convention for subheading:

* Font: Times New Roman
* Font Style: Bold
* Font Size: 14

Convention for body:

* Font: Times New Roman
* Font Style: Normal
* Font Size: 12

## Intended Audience and Reading Suggestions

* Developers
* Marketing staff
* Project managers

The document is composed of 6 sections. It includes the Introduction, the overall description, system features and non-functional requirements. Glossary is also provided for further assistance.

## Product Scope

* Each user will have to sign up to the application and can manage his profile. Users can play the role of either a buyer or seller.
* The user will add other users and give them credibility.
* A strategic graph will be constructed based on the credibility of the user.
* Buyer can contact the seller if interested in the product via an in-app messaging facility or through the seller’s contact information.
* Buyer can update the credibility of the seller after purchasing a product.
* Users can search for products they want to buy; multiple options for a particular product by different sellers along with their credibility will be displayed.
* Products can be filtered and sorted according to user requirements.
* The products uploaded by their respective added sellers will be displayed on the homepage of the user.
* Admin will be overseeing that all ethical guidelines are being followed by the users.

## References

Learning UML 2.0 A Pragmatic Introduction to UML by Russ Miles, Kim Hamilton

# Overall Description

## Product Perspective

This software is a modified version of the OLX application. The credibility given to a seller by other users is an additional feature in our application that lets a user know how reliable the seller is and makes our application more authentic. Features such as the in-app messaging platform will ensure that buyer identity and details are not disclosed to the seller while enquiring about the product. Such an application aims to provide an all-in-one solution to the users where the sellers can place their product in the open market and buyers can view a variety of products suited to their requirements. The application is available on both mobile and desktop and thus available to a wider audience.

## Product Functions

* Sign up/ login is mandatory for a user to utilize the services our application aims to provide.
* Users can place an advertisement to sell a product by adding relevant details.
* Search bar will allow the users to look for products which they intend to buy.
* Products can be filtered by location, category, prices, and other product specifications.
* Users can further sort the products based on the prices and how old or new the advertisement is.
* Related products will also be displayed next to the product the user is currently viewing.
* Customer reviews will be visible along with product and seller contact details and an add to favorites option.
* Seller can purchase different packages that will feature his products
* A favorites list will let the customers add products to a list they wish to view later.
* An inbox will store the chats with other users.
* Users can follow other users and give them a rating. Buyers will also be allowed to rate the sellers based on their experience.
* The application will further allow the users to view and manage their profile, advertisements, and the users they are following.
* Buyer can report a seller in case of any violation of ethical guidelines which will then be taken care of by the admin.

## User Classes and Characteristics

There will be two types of users who will use this product.

1. User (End-user) - they register on the system to benefit from the features that our system provides. Should be able to add friends and give them credibility points. They can also buy or/and sell different products on the platform. They can perform the following functionalities:

* Should be able to add friends and give them credibility points.
* Can accept friend requests and give them credibility points.
* Can engage in buying and selling of products.
* Can search for products and look at the details of the selected product.
* Can request customer support or/or report a user.
* Can manage their profile and can even delete their profile.

1. Administrator– will be responsible for the smooth functioning of the application. Their role includes:

* Approving products.
* Taking necessary action if any customers violate any rules; for example, someone misbehaves on the chat.
* Will look into customers’ complaints and take necessary action.
* Can remove any user if they violate rules.
* Can look at lists and details of users and products.

It is necessary to ensure that administrators have full knowledge of laws and rules; to avoid

being unfair to any user.

## Operating Environment

* Cloud-hosted NoSQL database (Firebase)/SQL server
* client/server system
* Operating system: Windows/iOS/Android
* Database: SQLPlus
* Platform: Flutter

## Design and Implementation Constraints

* Algorithm is required to devise credibility ratings for the sellers.
* The application can only run-on platforms that support JVM.
* Application design on multiple platforms may imply inconsistent user experience.
* For desktop applications, it can only run-on Windows.

## User Documentation

We will provide a video tutorial of the system/application along with some images.

* **In the video tutorial,**users will also have a link from where they can access a video tutorial about software installation and other guidelines.
* Images will contain the screenshots of different interface of our application.

## Assumptions and Dependencies

* Users only add those people as their friends whom they already know and not any random person.
* Users give credibility points fairly
* Financial Transactions to be predicated on mutual trust outside the application
* Any form of verbal/non-verbal misbehavior on the behalf of the customer/user is expected to be notified.
* Any issues regarding the credibility of the seller on the account of committing fraud/selling items that were displayed as something else must be reported
* Assuming the application will run smoothly on all the aforementioned platforms and that the optimization is taken care of by the framework of the platform.
* Assuming third party users must have minimum/recommended system requirements met on their machines required to run this application.*s*
* The users know the English language, as the user interface will be provided in English

# External Interface Requirements

## User Interfaces

*<Describe the logical characteristics of each interface between the software product and the users. This may include sample screen images, any GUI standards or product family style guides that are to be followed, screen layout constraints, standard buttons and functions (e.g., help) that will appear on every screen, keyboard shortcuts, error message display standards, and so on. Define the software components for which a user interface is needed. Details of the user interface design should be documented in a separate user interface specification.>*

## Hardware Interfaces

*<Describe the logical and physical characteristics of each interface between the software product and the hardware components of the system. This may include the supported device types, the nature of the data and control interactions between the software and the hardware, and communication protocols to be used.>*

## Software Interfaces

*<Describe the connections between this product and other specific software components (name and version), including databases, operating systems, tools, libraries, and integrated commercial components. Identify the data items or messages coming into the system and going out and describe the purpose of each. Describe the services needed and the nature of communications. Refer to documents that describe detailed application programming interface protocols. Identify data that will be shared across software components. If the data sharing mechanism must be implemented in a specific way (for example, use of a global data area in a multitasking operating system), specify this as an implementation constraint.>*

## Communications Interfaces

*<Describe the requirements associated with any communications functions required by this product, including e-mail, web browser, network server communications protocols, electronic forms, and so on. Define any pertinent message formatting. Identify any communication standards that will be used, such as FTP or HTTP. Specify any communication security or encryption issues, data transfer rates, and synchronization mechanisms.>*

# System Features

*<This template illustrates organizing the functional requirements for the product by system features, the major services provided by the product. You may prefer to organize this section by use case, mode of operation, user class, object class, functional hierarchy, or combinations of these, whatever makes the most logical sense for your product.>*

## System Feature 1: Account Management

Account management is an integral part of any software/application. There are 2 types of Accounts that our application manages:

 1) Admin Account

 2) User Account

**4.1.1 User Account Management:**

**4.1.1.1 Description and Priority**

Users will have an account from which they can either sell, view the products for sale or chat with sellers. Users can report or lodge a complaint from their accounts against other Users in case of any violation. They can also communicate with the admin in case of any query or opt for customer assistance if they are facing any issue. Additionally, they can view the product-list with filters applied and can add the products to their favorites list. This sub-feature is of high priority.

**4.1.1.2 Stimulus/Response Sequences**

When users open the system/application for the first time, they must make an account first. Therefore, the first step will be to click the signup button, it will take the USER to an interface where he/she will fill the data in their respective fields. After filling all the details, the User will click the create account button. The system will verify by email verification or number verification process. After confirmation, the account will be created. Then the user will be taken to an interface in which there will be multiple options like sell products, homepage, account, help with some options. After creating the account, the User can just access the account by login with the details he/she has registered its account with.

**4.1.1.3 Functional Requirements**

**REQ 1:** The system shall allow the USER to log in

**REQ 2:** The system shall allow the USER to sign up

**REQ 3:** The system shall allow the USER to log out

**REQ 4:** The system shall allow the USER to view other users and their products which they are selling.

**REQ 5:** The system shall allow the USER to edit profile

**REQ 6:** The system shall allow the USER to delete account

**REQ 7:** The system shall allow the USER to add other USERS.

|  |  |  |
| --- | --- | --- |
| **Use case Name** | **Sign Up** | |
| Related requirements | REQ 2 | |
| Goal In context | A new user will request the system to create a new account. | |
| Preconditions | User have a valid E-mail Address and/or a Phone-number | |
| Successful End Condition | A new account is created for the User. | |
| Failed End Condition | Invalid Email address.  Password not according to the specified requirements.  Invalid Phone Number. | |
| Primary Actors | User | |
| Secondary Actors |  | |
| Trigger | User presses the Sign-up button on the Home Page. | |
| Included cases |  | |
| Main Flow | **Step** | **Action** |
|  | 1. | User clicks the Sign-up button on the Home-page |
|  | 2. | Users enter Email-Address/Phone Number, Password and fill out other fields. |
|  | 3. | User waits for a verification code via Email/Phone Number. |

|  |  |  |
| --- | --- | --- |
| **Use case Name** | **Log in** | |
| Related requirements | REQ 1 | |
| Goal In context | User requests the system to log into the application. | |
| Preconditions | An account must exist for the specified Email/Phone-Number. | |
| Successful End Condition | The user successfully signs in. | |
| Failed End Condition | Unstable Internet connection.  User has entered an incorrect Email/Phone-Number.  User has entered incorrect password | |
| Primary Actors | Admin | |
| Secondary Actors |  | |
| Trigger | User wants to access the system within privileges prescribed by his account type. | |
| Included cases | **Identity Authentication** | |
| Main Flow | **Step** | **Action** |
|  | 1. **Include:: Identity Authentication** | User clicks the log-in button on the Home-page |
|  | 2. | User enters their email-ID and password |
|  | 3. | User logs into the application. |

|  |  |  |
| --- | --- | --- |
| **Use case Name** | **Log out** | |
| Related requirements | REQ 3 | |
| Goal In context | User request to end their session from the application. | |
| Preconditions | User must be signed into the system.  User must be currently logged into the application. | |
| Successful End Condition | The user will log out from the application. | |
| Failed End Condition | Log out failed due to an ongoing event (possibly uploading a file or a document) | |
| Primary Actors | User | |
| Secondary Actors |  | |
| Trigger | User wants to exit the system. | |
| Included cases |  | |
| Main Flow | **Step** | **Action** |
|  | 1. | User clicks the log-out button on the home page. |
|  | 2. | User exits the system. |

|  |  |  |
| --- | --- | --- |
| **Use case Name** | **Delete Account** | |
| Related requirements | REQ 6 | |
| Goal In context | User requests to delete all account info and cancel account privileges on the system. | |
| Preconditions | User must be signed into the system.  User account is not suspended by the admin.  User must be logged into the application. | |
| Successful End Condition | User account is deleted. | |
| Failed End Condition | The user request is rejected. | |
| Primary Actors | User | |
| Secondary Actors |  | |
| Trigger | User asks for the removal of all his information and cancel account privileges by pressing the delete account button. | |
| Included cases | **Identity Authentication** | |
| Main Flow | **Step** | **Action** |
|  | 1. | The user clicks the Delete Account button. |
|  | 2.**Include::Identity Authentication** | System prompts the user for password in order to verify. |
|  | 3. | The user gets one final caution by the system. |
|  | 4. | User presses accepts the caution and proceeds with the deletion. |
|  | 5. | Account is deleted from the system. |

|  |  |  |
| --- | --- | --- |
| **Use Case name** | **Manage Profile** | |
| Related Requirements | REQ 5 | |
| Goal In Context | A User requests to view or edit its profile. | |
| Preconditions | User must be logged into the application. | |
| Successful End Condition | The Profile is edited by the user.  Profile is viewed by the user. | |
| Failed End Condition | The data is not updated. | |
| Primary Actors | User | |
| Secondary Actors |  | |
| Trigger | The user requests the system to edit his profile data. | |
| Included Cases |  | |
| Main Flow | **Step** | **Action** |
|  | 1. | The user requests the profile data view from the system. |
|  | 2. | The user adds or updates any information on their profile. |
|  | 3. | All the changes made are saved in the database. |

|  |  |  |
| --- | --- | --- |
| **Use Case name** | **View profile of other users** | |
| Related Requirements | REQ 4 | |
| Goal In Context | User views profile of any other user. | |
| Preconditions | The user is friends with them. | |
| Successful End Condition | User is able to view the profile of the selected user. | |
| Failed End Condition | System unable to display profile of the requested user. | |
| Primary Actors | User | |
| Secondary Actors |  | |
| Trigger | User search for username in the search bar.  User click on the username from their friends list or credible sellers list on their profile.  User clicks on a profile from the “People you may know” section. | |
| Included Cases |  | |
| Main Flow | **Step** | **Action** |
|  | 1. | The user searches for a username. |
|  | 2. | Clicks on a username. |
|  | 3. | System displays the requested user profile. |

|  |  |  |
| --- | --- | --- |
| **Use Case name** | **Identity Authentication** | |
| Related Requirements |  | |
| Goal In Context | User details entered are checked and verified. | |
| Preconditions | The user enters their email and password. | |
| Successful End Condition | System checks and either confirms or rejects user identity | |
| Failed End Condition | The system is unable to verify the details. | |
| Primary Actors | User | |
| Secondary Actors | User Credentials Database | |
| Trigger | The user enters their credentials. | |
| Included Cases |  | |
| Main Flow | **Step** | **Action** |
|  | 1 | The user enters their email and password. |
|  | 2 | The system cross checks the details from the database. |
|  | 3 | The system verifies the credentials. |
| Extension | **Step** | **Branching Action** |
|  | 1 | The credentials do not match the details in the database. |

|  |  |  |
| --- | --- | --- |
| **Use Case name** | **Forgotten Password** | |
| Related Requirements |  | |
| Goal In Context | User requests to change their password. | |
| Preconditions | Account already exists. | |
| Successful End Condition | Requested user successfully changes their password. | |
| Failed End Condition | Users are unable to change their login credentials. | |
| Primary Actors | User | |
| Secondary Actors |  | |
| Trigger | The user enters their incorrect credentials. | |
| Included Cases |  | |
| Main Flow | **Step** | **Action** |
|  | 1. | When a user enters incorrect details, they can click on “forgotten password”. |
|  | 2. | The system sends a code either to the requested person’s registered email or phone number. |
|  | 3. | The User enters the received code and then enters their new password. |
|  |  |  |

**4.1.2 Admin Account Management**

**4.1.2.1 Description and Priority:**

Admin will have an account from where he/she manages/handles the complaints registered by Users or responds to the Users’ queries. This sub-feature is of high priority.

**4.1.2.2 Stimulus/Response Sequences:**

Admin’s account already exists from the launch of the app therefore when shown the interface to access the account, admin will always choose the LOGIN button and then he/she will be taken to an interface where he/she will have to fill in the blank fields with its login credentials. After filling the login details the system will verify from the database. If any of the details is wrong, the system will show an error and if the details are correct, then the admin will be taken to the main system/application interface where he will have multiple options to choose like view users, accounts etc. Lastly, To log out of the app, the admin will have to go the accounts page, and then he will have to click LOGOUT button. After pressing that button, the system will show a CONFIRMATION prompt. If the admin has to log out, he/she will click the confirm button and will click the CANCEL button if he decides otherwise.

**4.1.2.3 Functional Requirements:**

**REQ 1:** The system shall allow the admin to log in

**REQ 2:** The system shall allow the admin to log out

**REQ 3:** The system shall allow the admin to ban user in case of violation of major privacies or policies

**REQ 4:** The system shall allow the admin to give warning/suspend the user in case of violation of minor privacies or policies

**REQ 5:** The system shall allow the admin to ban user in case of violation of major privacies or policies

**REQ 6:** The system shall allow the admin to respond to user

**REQ 7:** The system shall allow the admin to view all the user and products which they are selling

**REQ 8:** The system shall allow the admin to approve products

**REQ 9:** The system shall allow the admin to edit/view its profile

**REQ 10:** The system shall allow the admin to view all complaints

**REQ 11:** The system shall allow the admin to delete any complaint if it is handled

**REQ 12:** The system shall notify the admin whenever a user has asked for support/assistance.

**REQ 13:** The system should show an error prompt if the admin has entered wrong login credentials.

**REQ 14:** The system should show a confirmation prompt when the admin is deleting a complaint.

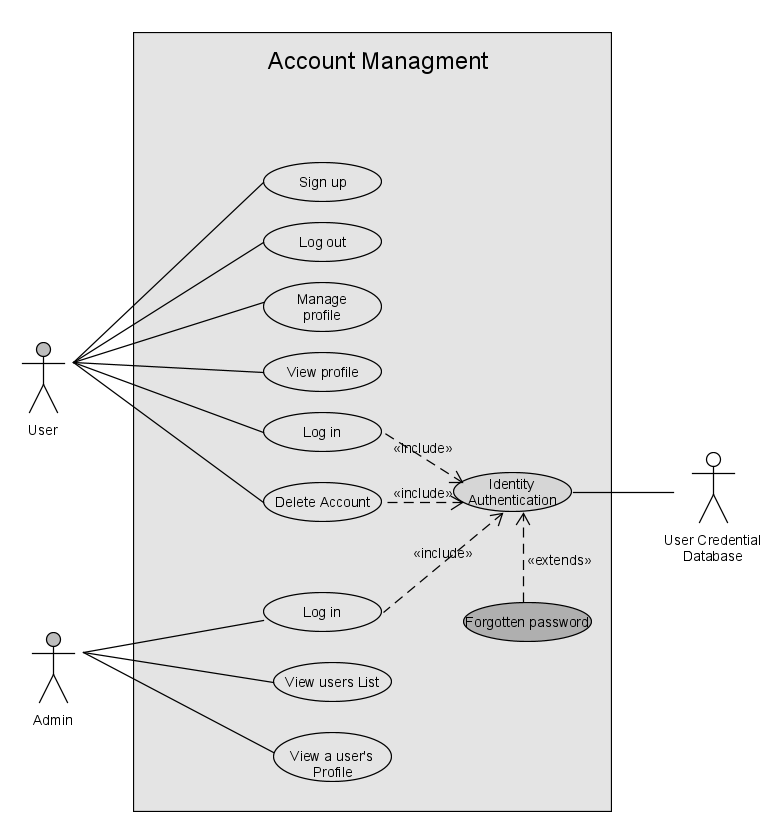
**REQ 15:** The system should show a confirmation prompt when the admin is removing the user.

**REQ 16:** The system shall allow the admin to view the user profiles

|  |  |  |
| --- | --- | --- |
| **Use case Name** | **Admin login** | |
| Related requirements | REQ 1 | |
| Goal In context | Admin requests to log into the system. | |
| Preconditions | Admin must be signed into the system | |
| Successful End Condition | The admin logs into the system. | |
| Failed End Condition | Incorrect username entered by the admin.  Incorrect password entered.  Account does not exist for that username. | |
| Primary Actors | Admin | |
| Secondary Actors |  | |
| Trigger | Admin opens admin control log on window to login. | |
| Included cases | **Identity Authentication** | |
| Main Flow | **Step** | **Action** |
|  | 1. | The admin opens the admin log on the page. |
|  | 2.**Include:: Identity Authentication** | Admin enters username and password in their respective fields. |
|  | 3. | Username and password are confirmed when the Login button is pressed. |
|  | 4. | The admin logged into the system. |

|  |  |  |
| --- | --- | --- |
| **Use Case name** | **View user list** | |
| Related Requirements | REQ 7 | |
| Goal In Context | Admin views the list of users in the system. | |
| Preconditions | Admin is logged into the system | |
| Successful End Condition | Admin views a list of all the users signed into the system. | |
| Failed End Condition | The system is unable to display the list of the users. | |
| Primary Actors | Admin | |
| Secondary Actors |  | |
| Trigger user | Admin presses the ‘View users list’ button to view all current users. | |
| Included Cases |  | |
| Main Flow | **Step** | **Action** |
|  | 1. | Admin requests the system to show a list of all the users by pressing the ‘View users list’ button. |
|  | 2. | The system fetches the data from the database. |
|  | 3. | The system displays the list of all the users which admin can view. |

|  |  |  |
| --- | --- | --- |
| **Use Case name** | **View user profile** | |
| Related Requirements | REQ 16 | |
| Goal In Context | Admin requests to view selected user’s profile. | |
| Preconditions | Admin is logged into the system | |
| Successful End Condition | Admin successfully sees a user’s profile. | |
| Failed End Condition | The system is unable to display the profile admin requested to see. | |
| Primary Actors | Admin | |
| Secondary Actors |  | |
| Trigger | Admin clicks on a username from the list of all the users.  Admin searches for a user using a search bar. | |
| Included Cases |  | |
| Main Flow | **Step** | **Action** |
|  | 1. | Admin requests the system to show a user profile by clicking on the username. |
|  | 2. | The system fetches all the public details of a user’s profile. |
|  | 3. | The system displays the whole profile of the user requested. |
|  |  |  |

****

## System Feature 2: Product Management

### Description and Priority

### The product management feature is yet another high priority feature which folds into the overall structure of sale and purchase of the products by the respective sellers. Users can add their products that they want to sell and customers can view and review the products they want to buy from a specific seller.

### Stimulus/Response Sequences

**Add products :**

To add products, users will click the "add" button. After that, users will be taken to a window where they will insert details related to the product, such as category, name, price, and picture. After adding the details, the user will click the "post" button to share the product with credible friends. Our algorithm will check if a product is in a sensitive category or not. If it is a sensitivity category, that product post will be blocked. If not, then the user will successfully share with friends.

**Buy products**:

To buy a product, the user will first click the buy button. The user will then be prompted with a window where they have products posted by their friends or the previous products they have bought. In this window, users will see other options, such as searching for a product by name. Here, users will enter the related keyword in the search field and click the search button. If the keyword relates to any product in the system, the user will have all the related products, but if not, the system will prompt a message saying "Product does exist". Users will have another option, which is to filter by category. The user will change the filter option available to him/her, and the system will display the product based on the filter the user has selected. After obtaining a product, the user will click to see details of the product, and they will be taken to a window where they can view complete details of the product. If the user wants that product, then they will click the contact seller option, where they will start a conversation to buy that product, or the user can call directly to the seller.

**Update product details**

To update product details, the user will have an edit product details option. He/she will click that option, then a new window will be shown where the user can edit details of products. After editing, the user will click the update button. If the updates are in the allowed category, then the system will update the details of that product.

**Remove product(seller)**

 To remove a product, a user(seller) will search for the product he/she wants to remove by entering a related keyword in the search field, or he/she can search by category. After getting the product he/she wants to remove, the user will click the "Remove product" option, and the system will show a message to confirm the removal. when the user confirms that the system will remove the product.

**Chat with seller**

The user (customer) will first contact the seller using the chat with the seller option. If the seller is available for chat, the conversation will start between the seller and the customer. After a successful agreement, the seller will be ready to sell the product and will deliver it through any delivery system that is accessible to both parties. Payment will be made by the customer, which is acceptable by the seller.

**Favorites List**

If a user wishes to view products in future he can add it to the Favorites list. For this the user (customer) must be logged into the system. When a user searches for a product to purchase, he/she will click to view that product in that window, from there he can click on the Favorites icon. After clicking that button, the user will successfully add his/her desired item to the list, which will then be saved in the database. Whenever a user wishes to view his or her favorite product, he can view it by clicking the "view favorites list" option available within his profile. The user will have other options for that product, such as buying it if available. A user will also be able to remove products from the list using the “remove from list’ option.

### 4.2.3 Functional Requirements

**REQ 1:** The system will allow user to buy products

**REQ 2:**The system will allow user to sell products to the system

**REQ 3:**The system will allow user to add details of the product

**REQ 4:**The system will allow product owner to remove a product.

**REQ 5:**The system will allow product owner to update details of the product

**REQ 6:**The system will allow buyer of the product to view details.

**REQ 7:**The system will allow user to search products by their category

**REQ 8:** The system will allow users to search products by typing product name in the search field.

**REQ 9:**The system will display all the related products that are searched by the user.

**REQ 10:** The system will allow user to traverse between search results.

**REQ 11:**After searching when product is not found system will notify the user.

**REQ 12:** USER will have the option to add the products to FAVORITE.

**REQ 13:** USER will have the option to view the FAVORITES.

**REQ 14:** USER will have the option to delete the products from FAVORITES.

**REQ 15:** The system shall allow the USER to message the seller if he/she has any query.

**REQ 16:** User will have the option to view chat history.

**REQ 17:** USER will have the option to END THE CHAT if his/her queries are resolved

**REQ 18:**The system will allow user to edit the details of the product

**REQ 19:**The system will allow buyer to view all the products of the user.

|  |  |  |
| --- | --- | --- |
| **Use Case name** | **Add Products to Favorites Option** | |
| Related Requirements | REQ 12 | |
| Goal In Context | A User wants to add the products in his/her favorites list. | |
| Preconditions | User must be logged into the system. | |
| Successful End Condition | Product is successfully added to the FAVORITES LIST | |
| Failed End Condition | Product is not added to the FAVORITES LIST | |
| Primary Actors | User | |
| Secondary Actors |  | |
| Trigger | Product is not added to the FAVORITES LIST | |
| Included Cases |  | |
| Main Flow | **Step** | **Action** |
|  | 1 | The User searches for products they want |
|  | 2 | They select a product |
|  | 3 | They click the Favorites button |
|  | 4 | The product is successfully added to the FAVORITES list |

|  |  |  |
| --- | --- | --- |
| **Use Case name** | **sell a product** | |
| Related Requirements | REQ 2, REQ 3 | |
| Goal In Context | A user requests the system to publish an ad for a product. | |
| Preconditions | User must be logged into the application. | |
| Successful End Condition | The User successfully places an ad for their product. | |
| Failed End Condition | The user request to place an ad was not successful.  Incomplete/Incorrect details of the product provided. | |
| Primary Actors | User | |
| Secondary Actors | Admin | |
| Trigger | User requests the system to allow them to place an ad to sell a product. | |
| Included Cases | **Approve Product** | |
| Main Flow | **Step** | **Action** |
|  | 1. | The user requests the system to post an ad to sell their product |
|  | 2. | The user enters all the details of the product they want to sell. |
|  | 3. | After entering all the details, the user presses the “Post Now” button. |
|  | 4.**Include::Approve product** | The product gets approved by the admin. |
|  | 5. | The user successfully places an ad. |

|  |  |  |
| --- | --- | --- |
| **Use Case name** | **Search products** | |
| Related Requirements | REQ 7, REQ 8, REQ 9, REQ 10, REQ 11 | |
| Goal In Context | User requests to view products available for sale. | |
| Preconditions | User is logged into their account. | |
| Successful End Condition | The user successfully sees a list of products. | |
| Failed End Condition | The user was unable to load the page.  There is no product related to the keyword searched. | |
| Primary Actors | User / Admin | |
| Secondary Actors | - | |
| Trigger | User searches for a product using the search bar.  Admin searches for a product using the search bar. | |
| Included Cases |  | |
| Main Flow | **Step** | **Action** |
|  | 1. | Enters product name in the search bar |
|  | 2. | All related products are shown to the user |
| Extension | **Step** | **Branching Factors** |
|  | 2.1 | User wants to filter/sort products. |

|  |  |  |
| --- | --- | --- |
| **Use Case name** | **Message Seller** | |
| Related Requirements | REQ 15, REQ 16, REQ 17 | |
| Goal In Context | The user sends a message to the other user through the in-app messaging platform. | |
| Preconditions | The user is logged into the application.   User is viewing product details. | |
| Successful End Condition | The message has been delivered to the other user. | |
| Failed End Condition | Unstable internet connection thereby message is not delivered.  The seller is suspended from the application. | |
| Primary Actors | User | |
| Secondary Actors |  | |
| Trigger | User wants to inquire about a product. | |
| Included Cases |  | |
| Main Flow | **Step** | **Action** |
|  | 1. | User views product details |
|  | 2. | Users click on the ‘Message seller’ option to chat with the seller. |
|  | 3. | User sends a message to the other user inquiring about the product. |

|  |  |  |
| --- | --- | --- |
| **Use Case name** | **Details of products** | |
| Related Requirements | REQ 6 | |
| Goal In Context | The user requests to view details of a specific product. | |
| Preconditions | The user has already searched for a product  a list of products is shown to the user. | |
| Successful End Condition | The user successfully sees all the details of the selected product. | |
| Failed End Condition | Product details unavailable.  The user is unable to see any details of the selected product. | |
| Primary Actors | User | |
| Secondary Actors |  | |
| Trigger | User clicks on a specific product to see its details. | |
| Included Cases |  | |
| Main Flow | **Step** | **Action** |
|  | 1. | User clicks on the product they want to further explore. |
|  | 2. | The system fetches details of the product from the database. |
|  | 3. | All the details of the selected product are presented. |
| Extension | **Step** | **Branching Factors** |
|  | 3.1 | User adds the product to favorites. |

|  |  |  |
| --- | --- | --- |
| **Use case Name** | **Filter ­­products** | |
| Related requirements | REQ 7 | |
| Goal In context | User/Admin requests for the products to displayed according to the filter/sorting option chosen by him. | |
| Preconditions | User/Admin must be logged into the system.  User/Admin has searched for a product.  Product list is displayed. | |
| Successful End Condition | Products are displayed as per user’s preference. | |
| Failed End Condition | No product falls under the filters applied. | |
| Primary Actors | User/Admin | |
| Secondary Actors | - | |
| Trigger | User/Admin chooses from filters/sorting options that are available. | |
| Included cases |  | |
| Main Flow | **Step** | **Action** |
|  | 1. | Product is searched by a user. |
|  | 2. | The user/admin applies filters and/or one of the sorting options. |
|  | 3. | Products are displayed accordingly. |

|  |  |  |
| --- | --- | --- |
| **Use case Name** | **End chat** | |
| Related requirements | REQ17 | |
| Goal In context | User who initiated the chat want to end it. | |
| Preconditions | User must be logged into the application.  Chat between the two users exist. | |
| Successful End Condition | The chat has been ended. | |
| Failed End Condition | Chat is unable to be cleared if it is flagged inappropriate by the admin or another user. | |
| Primary Actors | User | |
| Secondary Actors |  | |
| Trigger | User asks for the deletion of chat. | |
| Included cases |  | |
| Main Flow | **Step** | **Action** |
|  | 1. | The user opens the chat they want to delete. |
|  | 2. | The user presses the ‘End chat’ button found in the options menu of the chat. |
|  | 3. | The chat has been ended from the system. |

|  |  |  |
| --- | --- | --- |
| **Use case Name** | **View Favorites list** | |
| Related requirements | REQ 13 | |
| Goal In context | User requests the system to view his favorites list. | |
| Preconditions | User must be logged into the system. | |
| Successful End Condition | User can open the list.  Favourite products of a user are visible. | |
| Failed End Condition | User was not able to load the favorites page.  All favorite products of user are not displayed.  Favorites list for the user does not exist. | |
| Primary Actors | User | |
| Secondary Actors | None | |
| Trigger | The user requests the system to display its favorites list. | |
| Included cases |  | |
| Main Flow | **Step** | **Action** |
|  | 1. | The User clicks ‘My Favorites list’ within his profile. |
|  | 2. | System fetches the user favorites list from the Database. |
|  | 3. | Favorites list is visible to the user. |
| Extension | Step | Branching Factors |
|  | 4.1 | User selects a product to see its details |

|  |  |  |
| --- | --- | --- |
| **Use Case name** | **Edit product details** | |
| Related Requirements | REQ 18 | |
| Goal In Context | The user requests the system to edit product details. | |
| Preconditions | The user has already posted that product to sell. | |
| Successful End Condition | The user successfully updates the product details. | |
| Failed End Condition | The user is unable to edit or save the changes made. | |
| Primary Actors | User | |
| Secondary Actors |  | |
| Trigger | The user wants to update or add any new detail. | |
| Included Cases |  | |
| Main Flow | **Step** | **Action** |
|  | 1 | The user requests the system to edit the details of a product he/she previously posted to sell. |
|  | 2 | The user edits all the details of the product. |
|  | 3 | The user saves all the details. |

|  |  |  |
| --- | --- | --- |
| **Use case Name** | **Remove from Favorites list** | |
| Related requirements | REQ 14 | |
| Goal In context | A user requests to remove the product from his favorites list. | |
| Preconditions | User must be logged into the application.  Favorites list should not be empty. | |
| Successful End Condition | User removes a product from favorites list. | |
| Failed End Condition | User was not able to remove the product from favorites list. | |
| Primary Actors | User | |
| Secondary Actors | None | |
| Trigger | User wishes to remove a product from the favourites list. | |
| Included cases |  | |
| Main Flow | **Step** | **Action** |
|  | 1. | The favorites list is displayed to the user. |
|  | 2. | The User selects the product which it has to remove |
|  | 3. | The User presses the ‘Remove from Favorites’ option. |
|  | 4. | The system shows a prompt to ask for confirmation |
|  | 5. | The User clicks the confirm button. |
|  | 6. | Product is removed from the list. |

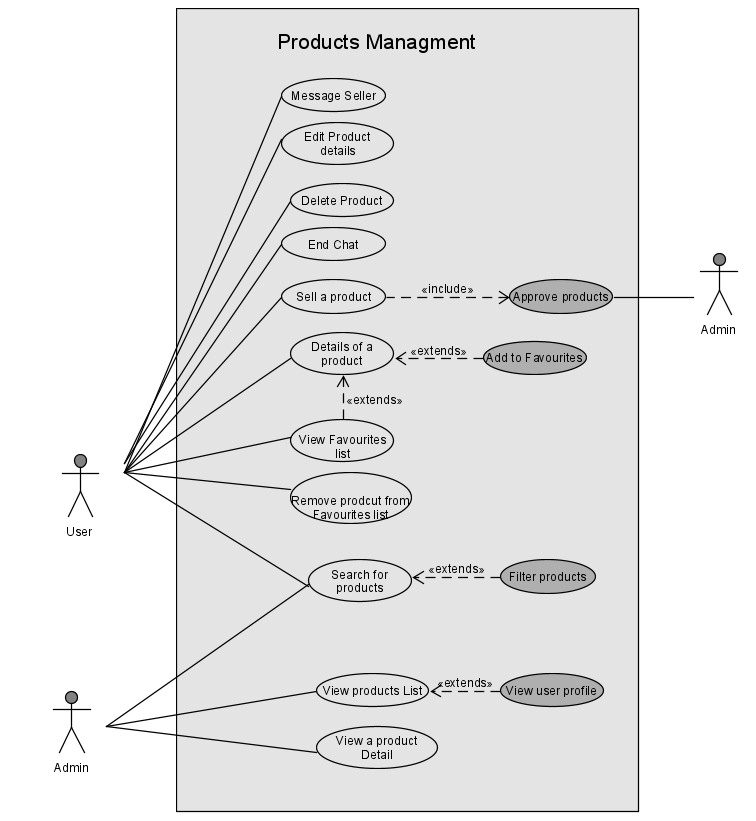
|  |  |  |
| --- | --- | --- |
| **Use Case name** | **View product-list** | |
| Related Requirements | REQ 19 | |
| Goal In Context | Admin requests to view the list of all the products posted by users. | |
| Preconditions | Admin is logged into the system. | |
| Successful End Condition | Admin views the list of all the products of a user. | |
| Failed End Condition | The system is unable to display the list of products. | |
| Primary Actors | Admin | |
| Secondary Actors |  | |
| Trigger | Admin asks the system to display a list of all the products within a user profile. | |
| Included Cases |  | |
| Main Flow | **Step** | **Action** |
|  | 1. | Admin views the list of users registered. |
|  | 2. | Admin clicks on the profile of the owner of the products |
|  | 2. | Admin requests the system to display the list of all products uploaded by the user. |
|  | 3. | The system fetches the list of all the products from the database. |
|  | 4. | The product list of user is displayed. |
| Extension | Step | Branching Factors |
|  | 2.1 | Admin requests to view the profile of the user who posted the ad. |
|  | 2.1 | System displays the profile of the user requested by the admin. |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| **Use Case name** | **View product details** | |
| Related Requirements |  | |
| Goal In Context | Admin requests to view the details of a selected product. | |
| Preconditions | Admin is logged into the system.  Admin is already on the homepage where he can view all the products  Admin is on user’s profile where product list of that specific seller is visible. | |
| Successful End Condition | Admin views all the details of a selected product. | |
| Failed End Condition | The system is unable to display the details of the selected product. | |
| Primary Actors | Admin | |
| Secondary Actors |  | |
| Trigger | Admin clicks on a product to view its details. | |
| Included Cases |  | |
| Main Flow | **Step** | **Action** |
|  | **1.** | Product is searched. |
|  | 2. | Admin click on the product to view its details. |
|  | 3. | The system brings details of the product from the database. |
|  | 4. | All the details of the selected product are displayed. |

ss

|  |  |  |
| --- | --- | --- |
| **Use Case name** | **Approve Product** | |
| Related Requirements |  | |
| Goal In Context | User requests admin for the approval of product ad. | |
| Preconditions | The admin is logged into their account. | |
| Successful End Condition | Admin approves the product.  Admin rejects the product due to some constraints. | |
| Failed End Condition | Admin is unable to take any action on a user's request to approve products. | |
| Primary Actors | Admin | |
| Secondary Actors | Reject | |
| Trigger | Admin gets a notification from the user to examine a product. | |
| Included Cases |  | |
| Main Flow | **Step** | **Action** |
|  | 1. | Admin receives a notification for approval of the product. |
|  | 2. | Admin investigates all the details of the product. |
|  | 3. | Decides whether to accept or reject the product on the basis of rules and regulations of the system. |

|  |  |  |
| --- | --- | --- |
| **Use Case name** | Delete product | |
| Related Requirements |  | |
| Goal In Context | The user requests the system to delete a product they had previously posted to sell. | |
| Preconditions | The user has already posted that product to sell. | |
| Successful End Condition | The user successfully deletes the product. | |
| Failed End Condition | The user is unable to delete the product. | |
| Primary Actors | User | |
| Secondary Actors |  | |
| Trigger | The user has sold the product. OR the user no longer wishes to sell that product. | |
| Included Cases |  | |
| Main Flow | **Step** | **Action** |
|  | 1 | The user requests the system to delete the product he/she posted previously to sell. |
|  | 2 | The user clicks “Delete product”. |
|  | 3 | The product is deleted from the database. |



**4.3 System Feature : User Credibility**

**4.3.1 Description and Priority:**

User credibility is a main feature which differentiates our system/application from other B2B applications. Unlike ratings/reviews, User’s credibility is based on trust and belief amongst Users, but it can be changed accordingly. The credibility rating will also be automatically assigned to known users of those 2 users who have added each other but all that all would be done by an algorithm, and it will vary user to user. If a friend’s credibility drops down to some defined threshold, he will no longer to visible to that user. This feature is of high priority.

**4.3.2 Stimulus/Response Sequences:**

At first when a new user is added, he sends requests to other users with that he/she also assigns a credibility rating and once the other user accepts his/her request, they can also give credibility to that user. With time it changes like when a user buys a product from another user and that user delivers a faulty product or does a fraud. Another case can be when the user is satisfied with the purchase. In this case the original buyer assigns credibility to that user who will in turn affect the total credibility rating of the user. If the credibility of a certain user drops below a defined threshold, the system kicks out the user. In this way, users assign credibility to each other.

**4.3.3 Functional Requirements:**

**REQ 1:**The system shall allow a user to add other users.

**REQ 2:**The system shall allow a user to accept or deny the request of other users.

**REQ 3:**The system shall allow a user to give credibility to other users.

**REQ 4:**Users shall have the option to change the credibility of other users in case of fraud.

**REQ 5:**The system should show a confirmation prompt when a user changes the credibility of other users.

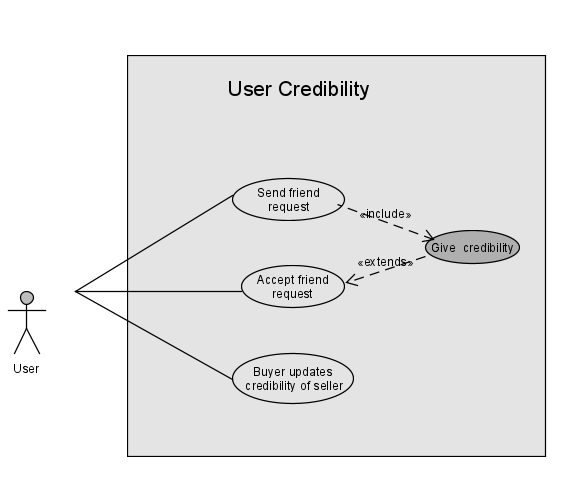
**REQ 6:**The system should remove a user if his/her credibility drops below a defined threshold.

|  |  |  |
| --- | --- | --- |
| **Use Case name** | **Send Friend Request** | |
| Related Requirements | REQ 1, REQ 3, REQ 5 | |
| Goal In Context | A user sends a friend request to a user. | |
| Preconditions | The user must be logged into their account. | |
| Successful End Condition | The user ends up sending a friend request to a user and giving credibility to them. | |
| Failed End Condition | The user is unable to send a friend request. | |
| Primary Actors | User | |
| Secondary Actors |  | |
| Trigger | User requests the system to allow them to send a friend request to a user. | |
| Included Cases | **Give Credibility** | |
| Main Flow | **Step** | **Action** |
|  | 1. | The user searches for a person to send a friend request. |
|  | 2. | The user sends a friend request to the user. |
|  | 3. **Include::Give Credibility** | The user also gives credibility to the user he/she just sends the friend request to. |

|  |  |  |
| --- | --- | --- |
| **Use Case name** | **Accept Friend Request** | |
| Related Requirements | REQ 2, REQ 3, REQ 5 | |
| Goal In Context | A user accepts a friend request. | |
| Preconditions | The user must be logged into their account. | |
| Successful End Condition | The user ends up accepting/rejecting a friend request to a user and giving credibility to them. | |
| Failed End Condition | The user is unable to take any action on the friend request received. | |
| Primary Actors | User | |
| Secondary Actors |  | |
| Trigger | A user receives a friend request. | |
| Included Cases | **Give Credibility** | |
| Main Flow | **Step** | **Action** |
|  | 1. | The user receives a friend request. |
|  | 2. | The user opens their profile and looks at their public details. |
|  | 3. | The user accepts/rejects the friend request. |
| Extension | **Step** | **Branching Action** |
|  | 1. | The user accepts the friend request. |
|  | 2. **Include::Give Credibility** | The system asks to give credibility to the friend. |

|  |  |  |
| --- | --- | --- |
| **Use Case name** | **Buyer updates credibility of the seller** | |
| Related Requirements | REQ 4, REQ 5 | |
| Goal In Context | A user gives credibility right after the business is done based on the product received. | |
| Preconditions | The business transaction that took place must be completed before a person could give new credibility. | |
| Successful End Condition | The system gets a valuable response from the buyer to feed in the system to calculate credibility. | |
| Failed End Condition | The system gets a valuable response from the buyer to feed in the system to calculate credibility. | |
| Primary Actors | User | |
| Secondary Actors |  | |
| Trigger | The buyer gets a notification telling him to give credibility to the seller from which he/she purchased a product. | |
| Included Cases |  | |
| Main Flow | **Step** | **Action** |
|  | 1. | The buyer gets the notification to rate the seller. |
|  | 2. | The buyer leaves his remarks. |
|  | 3. | The buyer rates the punctuality and quality of the work using appropriate rating scale. |
|  | 4. | The seller gets a notification about the buyer’s remarks |

|  |  |  |
| --- | --- | --- |
| **Use Case name** | Give Credibility | |
| Related Requirements | REQ3 | |
| Goal In Context | The user gives credibility to a user upon sending or accepting a friend request. | |
| Preconditions | The user is logged into their account. | |
| Successful End Condition | The user successfully gives credibility to another user. | |
| Failed End Condition | The user is unable to give credibility to another user. | |
| Primary Actors |  | |
| Secondary Actors |  | |
| Trigger | The user sends a friend request or accept a friend request. | |
| Included Cases |  | |
| Main Flow | Step | Action |
|  | 1 | The system ask user to give credibility to another user he wishes to add as their friend. |
|  | 2 | The user adds the credibility which is saved in the database. |

****

**4.4 System Feature : Customer Assistance**

**4.4.1 Description and Priority:**

User assistance is the feature which provides help related to buying a product, assistance to new users how to use the application.User assistance provides FAQ’s, chat with admin, and chat with seller options. To access the assistance feature, users must be logged into the system. If a user (customer) wants to buy a product, he/she will select the respective product and initiate chat by clicking the chat with seller option and writing a message. If a user (seller) is online, he/she will respond to the customer that way. Then the conversation starts in the chat box. This feature is of medium priority.

**4.4.2 Stimulus/Response Sequences:**

FAQs are another type of assistance provided by the application. If a user wants to ask a question, he or she clicks "help". It will prompt a FAQ’s window, where the most frequently asked questions are already answered by the admin. If the query of the user is already present there, he/she can click that question, and then the user will get a detailed answer to that query. If a user's question is not present, then the user will click the chat with admin option which will also be there at the header of the application where the user will write their query. After that, the admin will get notification, then he/she will suggest a solution in the chat box or try to help the user.

**4.4.3 Functional Requirements:**

**REQ 1:** The system will provide different support such as FAQ’s and Chat with Admin.

**REQ 2:** The user can select the support type he needs.

**REQ 3:** The system will display FAQs when user clicks FAQ’s

**REQ 4:** The system will start a conversation when the user goes for Chat with Admin option.

**REQ 5:** The Current user can share details of the other user who has misbehaves with current user or goes against privacies and policies

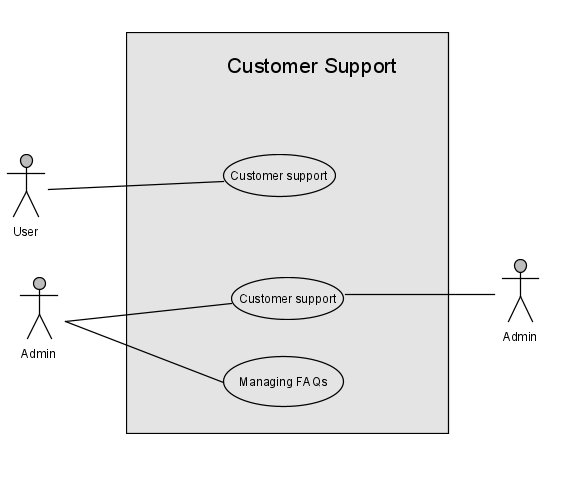
**REQ 6:** The system will allow users to chat with another user.

**REQ 7:** The chat between users will be removed after 30 days.

|  |  |  |
| --- | --- | --- |
| **Use Case name** | **Customer Support** | |
| Related Requirements | REQ 1, REQ 2, REQ 3, REQ 4, REQ 5 | |
| Goal In Context | The user requests help from the admin. | |
| Preconditions | User must be logged into the system. | |
| Successful End Condition | The user successfully launches their complaint or can ask their query from the admin. | |
| Failed End Condition | The user is unable to ask for help. | |
| Primary Actors | User | |
| Secondary Actors |  | |
| Trigger | User requests the system for the Customer Support form. | |
| Included Cases |  | |
| Main Flow | **Step** | **Action** |
|  | 1. | User requests the system for Customer Support form. |
|  | 2. | The user fills the form and attaches any relevant proof (if required). |
|  | 3. | The user sends the form to the admin. |

|  |  |  |
| --- | --- | --- |
| **Use case Name** | **Customer Support (Admin)** | |
| Related requirements |  | |
| Goal In context | Any query of customer is resolved after contacting the admin. | |
| Preconditions | Administrator must be logged into the system.  User must send request/ask a query. | |
| Successful End Condition | The admin successfully resolves the user’s issue. | |
| Failed End Condition | The admin is unable to help the user. | |
| Primary Actors | Admin | |
| Secondary Actors | User | |
| Trigger | Admin requests the system to view and responds to the query of user. | |
| Included cases |  | |
| Main Flow | **Step** | **Action** |
|  | 1. | The admin gets notified and requests the system to view the query. |
|  | 2. | The admin suggests a solution related to the issue. |
|  | 3. | The system sends a notification to the user by email. |

|  |  |  |
| --- | --- | --- |
| **Use Case name** | **Managing Frequently Asked Questions (FAQs)** | |
| Related Requirements | REQ 1 | |
| Goal In Context | Admin manages the FAQs section. | |
| Preconditions | Administrator must be logged into the system. | |
| Successful End Condition | Admin successfully views or edits any FAQ. | |
| Failed End Condition | The admin is unable to view the FAQs list OR is unable to make any changes. | |
| Primary Actors | Admin | |
| Secondary Actors |  | |
| Trigger | Admin requests the system to view or edit any FAQ. | |
| Included Cases |  | |
| Main Flow | **Step** | **Action** |
|  | 1. | Admin requests the system to view or add/update any FAQ. |
|  | 2. | Admin either views the list of the FAQs displayed or add/edit FAQs. |
|  | 3. | Any changes made by the admin are saved in the database. |



**4.5 System Feature: Breach of Rules**

**4.5.1 Description and Priority:**

User violations and regulations are a very important aspect of the software pertaining to regulatory affairs and making the system more inclusive and creating a friendly environment online. The following are the types of User Violations are to be due to misdemeanor of the misdemeanor in mutual dealing, or it could also be due to legislative violations made on behalf of the seller by purposefully selling a particular product that violates certain ethical or legal laws.

Users will have the privilege to file complaints and report any case of misdemeanor, misbehavior, harassment, and misleading information coming from the seller strictly within the chat service provided by the application. They can also report the seller if they are selling a product that violates certain ethical or legal laws.

**4.5.2 Stimulus/Response Sequences:**

If the user receives any form of misbehavior of any kind in the conduct of the seller they are in contact with strictly within the parameters of the chat feature provided by the application, they are given the privilege to report the said seller to the system admin and will also have the option to write a detailed complaint and attached the screenshots of the chat as well. In the cases of seller product violation where the seller is selling a product or a category of products which are not legally mandated such as prescription medication, drugs, prohibited beverages, oil that requires license to sell, etc. and if they are being sold under a different product category with a different name, such cases will be reported by other users who will lodge a complaint in the REPORT section of the seller profile, and will attach formidable proof which will be forwarded to the system administrators. The administrators upon receiving the complaint report can warn, remove products, revoke seller privileges or completely terminate the account at their jurisdiction. The user will also have the option to call-back any allegations of misdemeanor they have made on a seller.

**4.5.3 Functional Requirements:**

**REQ1:** There must be a thread of communication between the user and the violator.

**REQ2:** The thread of communication must be in the form of a text chat, phone calls will not be

entertained due to fabrication.

**REQ3:** The thread of communication must be within the chat service provided by the application.

**REQ4:** The lodged complaint must be reported with formidable evidence for conviction and must contain the appropriate information of how a legislative violation or a misdemeanor in mutual dealing is being taken place by the seller.

**REQ5:** The proof of misdemeanor/violation or lack thereof must be clearly mentioned in the complaint filed by the user.

**REQ6:** The system shall warn the violator before the report has been reviewed by the system admin

**REQ7:** The user shall also be given rights to revoke their complaint/allegation against the violator.

**REQ8:** The system shall allow the admin to suspend/terminate the account of the violator in case of lodged complaints being authentic.

**REQ9:** The system shall restore account privileges of the violator in case of suspension after a fine has been paid or a said number of days have been passed.

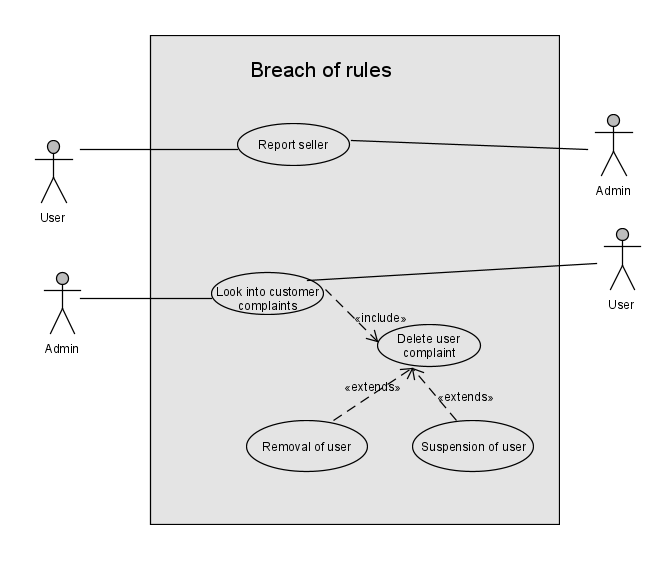
|  |  |  |
| --- | --- | --- |
| **Use case Name** | **Report Seller** | |
| Related requirements | REQ 4, REQ 5 | |
| Goal In context | User requests the system to report to the admin about another user in case of rule violation. | |
| Preconditions | User must be logged into the system.  The other user must have violated any rule/privacy. | |
| Successful End Condition | Report against the seller is successfully submitted. | |
| Failed End Condition | Report against seller has failed to submit. | |
| Primary Actors | User | |
| Secondary Actors | Admin | |
| Trigger | User requests the system to allow him/her to submit the report. | |
| Included cases |  | |
| Main Flow | Step | Action |
|  | 1. | User clicks on another user’s profile. |
|  | 2. | User clicks the REPORT SELLER button. |
|  | 3. | User writes the reason. |
|  | 4. | User clicks the SUBMIT button. |
|  | 5. | The notification goes to the admin by email. |

|  |  |  |
| --- | --- | --- |
| **Use case Name** | **Delete User Complaint** | |
| Related requirements |  | |
| Goal In context | Admin requests the system to delete the complaint which has been resolved. | |
| Preconditions | Administrator must be logged into the system.  There must be a complaint.  The complaint is resolved/handled. | |
| Successful End Condition | The admin successfully deletes the complaint. | |
| Failed End Condition | The admin is unable to delete the complaint. | |
| Primary Actors | Admin | |
| Secondary Actors |  | |
| Trigger | Admin requests the system to delete the user’s complaint. | |
| Included cases | **Look into customer complaints** | |
| Main Flow | Step | Action |
|  | 1. | Admin views the complaint. |
|  | 2. | Admin confirms if the complaint is handled/resolved. |
|  | 3. | Admin deletes the complaint. |

|  |  |  |
| --- | --- | --- |
| **Use case Name** | **Look into customer complaints** | |
| Related requirements | REQ 2, REQ 3 | |
| Goal In context | Admin requests the system to view the customer complaints. | |
| Preconditions | Administrator must be logged into the system.  User must create a complaint. | |
| Successful End Condition | The admin warns the other user who is involved in complaint.  Suspends/Removes the account of the user in case of valid complaint. | |
| Failed End Condition | Complaint was fake. | |
| Primary Actors | Admin | |
| Secondary Actors | User | |
| Trigger | Admin requests the system to view the user complaint. | |
| Included cases |  | |
| Main Flow | Step | Action |
|  | 1. | Admin views the complaint. |
|  | 2. | Admin confirms if the complaint is valid. |
|  | 3. | Warns the other user who is involved in the complaint. |
|  | 4. | If user does not respond, admin suspends the account of user who is involved in complaint. |
|  | 5. | Admin removes the user if the issue is not resolved. |
|  | 6.    **Include::Delete user complaint** | Admin deletes the user complaint after resolving it. |
| Extension | Step | Branching Factors |
|  | 4.1 | Admin suspends the user. |
|  | 5.1 | Admin suspends the user. |

|  |  |  |
| --- | --- | --- |
| **Use case Name** | **Removal Of User** | |
| Related requirements | REQ 8 | |
| Goal In context | Admin requests the system to remove the user from the system. | |
| Preconditions | Administrator must be logged into the system.  User violates major privacies and policies of the system  OR  The credibility of the user falls below the defined threshold. | |
| Successful End Condition | The admin successfully removes the user. | |
| Failed End Condition | The admin fails to remove the user. | |
| Primary Actors | Admin | |
| Secondary Actors |  | |
| Trigger | Admin requests the system to remove the user.  The system is notified when the credibility of the user drops below the defined threshold. | |
| Included cases |  | |
| Main Flow | Step | Action |
|  | 1. | Admin gets information of the user who violated the major privacies and policies of the system. |
|  | 2. | Admin presses DELETE USER button. System prompts to confirm removal of users. |
|  | 3. | Admin confirms the deletion. |
|  | 4. | The system successfully removes the user. |

|  |  |  |
| --- | --- | --- |
| **Use case Name** | **Suspension of user** | |
| Related requirements | REQ8, REQ7 | |
| Goal In context | Admin requests the system to suspend the user from the system. | |
| Preconditions | Administrator must be logged into the system.  Users violate minor policies of the systems.  Administrator must have given warning to the user before. | |
| Successful End Condition | The admin successfully suspends the user. | |
| Failed End Condition | The admin is unable to suspend the user. | |
| Primary Actors | Admin | |
| Secondary Actors |  | |
| Trigger | Admin requests the system to remove the user. | |
| Included cases |  | |
| Main Flow | Step | Action |
|  | 1. | Admin gets information of the user who violated policies or added unauthenticated products. |
|  | 2. | Admin presses the SUSPEND USER button. System prompts to confirm account suspension of the user. |
|  | 3. | Admin confirms the account suspension. |
|  | 4. | The system successfully suspends the user account. |

****

# Other Nonfunctional Requirements

*Since our application is a modified version of OLX, the estimates for this section have been made in reference to it.*

## Performance Requirements

* Response Time – Application login/signup should not exceed 6 seconds while executing 10 simultaneous logins. Time for loading the application for 90% of users should not be more than 2-3 seconds. Messages should be delivered instantly to the users. In cases of high workload, response time should not exceed 8 seconds.
* Workload – The application must support at least 20,000 users. It must be designed to cope up with peak workload that will be from around 6 pm to 12 pm.

|  |  |  |
| --- | --- | --- |
| **Scenario** | Daily Requests | Pages |
| **Login** | 1000 | Login, Homepage, 40% Exit |
| **View/Edit Profile** | 500 | Login, Homepage, My Profile, 40% Exit |
| **Search User** | 700 | Login, Homepage, User-List, 20% Exit |
| **View Searched User** | 850 | Login, Homepage, User-List, User Profile,20% Exit |
| **View Connected User** | 300 | Login, Homepage, Account, Followers/Following, User Profile, 20% Exit |
| **Search Product** | 3000 | Login, Homepage, Product-List, 20% Exit |
| **View Product** | 5000 | Login, Homepage, Product-List, Product, 50% Exit |
| **In-app Chatting** | 700 | Login, Homepage, Product-List, Product, Inbox, 70% Exit` |
| **Post Product Ad** | 1000 | Login, Homepage, Category, Details, 70% Exit |

* Scalability – Our application must be able to support 300,000 users when workload increases due to a rise in the number of users. The response times, however, should remain the same.

## Safety Requirements

* After account creation, verification will be done to ensure that the user is authentic and associated with the email address or phone number.
* Constraints to be set on sensitive products such as hazmat suits/oil/medical supplies or the ones that have legal jurisdictions and require certain permits/licenses to sell.

## Security Requirements

* The database should be secured from malicious cyber-attacks and illegitimate access by strengthening physical database security through cameras, locks as database is being managed in an on-premises data center.
* All the files, backup, cache, and API communication should be encrypted. Therefore, only the authorized users with the encryption key will be able to access the critical data.
* In order to maintain privacy, the messages shared between the users through the in-app messaging platform should be encrypted.
* Sensitive data shouldn’t be written in the application log or included in the backup.
* The Application should store user credentials through a keychain (IOS)or key store (Android/windows) so that users can set a password that is difficult to break, thus increasing the security of their accounts.
* App Transport security and SSL pinning should be enabled. This secures the application from man-in-middle attacks when communicating over a public network.

## Software Quality Attributes

* **Maintainability**: The maintenance will be done once a month to update the software with new functionalities and debug the code to remove defects
* **Confidentiality**: The seller details will not be disclosed to the buyer and vice-versa while communicating through the in-app messaging platform.
* **Accessibility**: impaired individuals will be assisted though the use of readable font, high-contrast colors, mouse-over labels and voice assistant.
* **Usability**: Users can perform main actions on the interface such as add to favorites list, filter, sort, search for products/users, chat with the seller, etc. with just a maximum of two clicks. Users can upload up to 10 photos with the maximum size of 7 MB/photo. Accepted formats are .jpg, .png. Error can be generated during signup/login and when posting an advertisement for a product, but the error rate should not exceed 5% on submission.
* **Reliability**: The software has an 80% reliability for a month.
* **Efficiency**: The loading of a page should not go beyond 2-3 seconds. Product should be visible to the users as soon as the seller uploads the advertisement for it. The in-app messaging should be able to send messages instantly.
* **Integrity**: User details that contain personal info, credibility and ratings shall be backed up to cloud every month in order to avoid any data loss.
* **Robustness**: Erroneous input data shall be handled accordingly. In case of device crash, data can be restored from the cloud.
* **Flexibility**: The seller can upload ads for multiple products. Although only one product detail can be viewed at a time. To ensure that the content is consistent across mobile and desktop applications, responsive user interface is provided.

## Business Rules

* Each user shall be registered via a valid email or phone number.
* Only one account is allowed per email or phone number.
* Price can only be set in Pakistani Rupees.
* Sharing sensitive information among users is not encouraged. Users will be responsible for any consequences that may follow.
* The application is only to provide buyers and sellers a platform to engage with each other. Any issues with the product will not be our responsibility.
* At least one image must be uploaded along with the product advertisement.
* Product advertisements that have been up for more than 5 years will be removed.

# Other Requirements

*<Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project.>*

**Appendix A: Glossary**

**SRS:** It stands for Software Requirements Specification

**Product Scope:** The characteristics and functions that define a product, service, or outcome determine the product scope.

**Product Perspective:** In response to stakeholder demands or assumptions in neighboring systems, they determine the product's functional behavior, characteristics, and restrictions.

**Functional Requirements:** A Functional Requirement (FR) is a statement that describes the service that the programme must provide. It refers to a software system or a component of one.

**Other Nonfunctional Requirements:** System characteristics like security, reliability, performance, maintainability, scalability, and usability are defined by nonfunctional requirements (NFRs). They guarantee the system's usability and efficacy.

**System Feature:** A feature is a subset of a software system's capability that fulfils a need, reflects a design decision, and offers a possible configuration option.

**Software Stimulus:**  System architects may use STIMULUS' modelling language to combine defined textual requirements, state machines, and block diagrams in a fully integrated simulation environment.

**Misdemeanor:** A misdemeanor is considered a crime of lesser seriousness, and a felony one of greater seriousness.

**FAQs:** It stands for Frequently Asked Questions

**Prompt:** A prompt is defined as a cue provided to someone to assist them recall what they're going to say, or anything that triggers another event or action to happen.

**Credibility:** The characteristic of being believed in and trusted.

**Assistance:** The act of assisting someone else by sharing work

**Stakeholders:** A stakeholder is someone who has an interest in a firm and may influence or be influenced by it.

**Requirements Specification:** A Requirement Specification is a list of all the requirements that must be imposed on the product's design and verification. Other information required for the product's design, verification, and maintenance is also included in the specification

**Assumptions and Dependencies:**  **Assumptions** are items that you presume are in place and contribute to the project's success. **Dependencies -** Other projects or triggers on which your project is dependent, or which benefit from your project's outcomes**.**

**Implementation Constraints:** Placement and routing are controlled by implementation restrictions. Thus are just propagated and made available to the implementation tools, and they are not directly helpful to XST.

**Hardware Interface:** The connectors, sockets, cables, and electrical impulses that flow via each line between the CPU and a peripheral device or communications network are defined by a hardware interface.

**Software Interface:** The languages, codes, and messages that programmes use to interact with one other and with the hardware are known as software interfaces (programming interfaces).

**Communications Interface:** An electrical circuit that allows one machine to interact with another machine, generally according to a set of standards.

**Intended Audience:** The group of persons for whom a service or product is developed is referred to as the intended audience.

**Appendix B: Analysis Models**

*<Optionally, include any pertinent analysis models, such as data flow diagrams, class diagrams, state-transition diagrams, or entity-relationship diagrams*.>

**Appendix C: To Be Determined List**

1- After purchasing a product, a customer can give written reviews to the seller based on their experience.

2- To increase product popularity, seller can also purchase packages to feature his products.

3- Facilities like voice assistant and readable fonts will be incorporated in the application.

4- System will notify a user once the product he searched for is available for sale.