Software Requirements Specification

for

Project – 3: Creda  
Version 1.0 approved  
  
Prepared by Group No. 3

SHAHZAIB AHMED – **19175**  
TABREZ HASSAN – **19737**  
MANAHIL JAWAID - **18725**  
MUHAMMAD JAFFAR ABBAS – **19666**  
ESSA SHEIKH – **18630**  
REYYAN AHMED – **19749**   
ABDUL HADI – **19723**WASEEM SHEHZAD – **19732**

**Section No. 3  
  
14th October 2021**

**Table of Contents**

Table of Contents 0

Revision History 1

1. Introduction 2

1.1 Purpose 2

1.2 Document Conventions 2

1.3 Intended Audience and Reading Suggestions 2

1.4 Product Scope 3

1.5 References 3

2. Overall Description 4

2.1 Product Perspective 4

2.2 Product Functions 4

2.3 User Classes and Characteristics 5

2.4 Operating Environment 6

2.5 Design and Implementation Constraints 6

2.6 User Documentation 7

2.7 Assumptions and Dependencies 7

3. External Interface Requirements 7

3.1 User Interfaces 7

3.2 Hardware Interfaces 7

3.3 Software Interfaces 7

3.4 Communications Interfaces 8

4. System Features 8

4.1 Friendly User Interface 8

4.2 Credibility Metric……………………………………………………………………………..…..13

4.3 Seller/User Interaction…………………………………………………………………………….16

4.4 Easy Advertisement………………………………………………………………………………19

4.5 Administrative Features…………………………………………………………………………..23

5. Other Nonfunctional Requirements 26

5.1 Performance Requirements 26

5.2 Safety Requirements 27

5.3 Security Requirements 27

5.4 Software Quality Attributes 27

5.5 Business Rules 29

6. Other Requirements 30

Appendix A: Glossary 30

Appendix B: Analysis Models 32

Appendix C: To Be Determined List 32

**Revision History**

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

# Introduction

## Purpose

The purpose of this document is to collect and examine all the various concepts for our project “**Creda**”, which have surfaced to define the project and its requirements. Also, to obtain a better grasp of the project, we have and will predict and sort out how we expect this product will be used. In a nutshell, the purpose of this SRS document is to provide a comprehensive overview of our software product, including its specifications and objectives. The SRS document will also portray the scope of the project, like users and their requirements, tools required to develop and design the product. It will also define our client's, team's, and audience's perceptions of the product and its capabilities. The document will, however, assist the designer or developer to design and develop by adhering to the software development life cycle process.

## Document Conventions

Overall, the SRS document is drafted in **Times New Roman** fonts. The Calibri fonts are used in the Table of **Use Cases**. The parts of the project that are more significant as that of others or showing a message/name of any important tool/important announcements or claims are highlighted in **Bold** Format. In addition to it, any code syntax, UML text, or any keywords recognized/referred by the system, are drafted in Courier New. The result/output of the command will be shown in **Courier New Bold.** The optional requirements of the product will be enclosed in curly braces **{ }.** While the Boolean operations will be differentiated with ***|*** symbol. The *Italic* format will refer to the link or connection.

## Intended Audience and Reading Suggestions

The intended audience for this software requirements specification document includes the developers working on the web app, project team members, course instructors, teacher’s assistants, the client, and the users of Creda. This document contains details of how Creda works and what it will do.

Below is the overview of the document:

Section 1 - Introduction: This section contains the basic introduction to this software requirements specification document. The intended audience of this section is the same as the intended audience of this document.

Section 2 - Overall Description: This section contains the overview and high-level description of the features of the Creda software. It is intended for the client and developers.

Section 3 - External Interface Requirements:

Section 4 - System Features: The suggested audience for this section are the client and the developers working on this software. This section contains use cases, functional requirements and system features. This section helps understand the functionality of Credoment in more detail.

Section 5 - Other Non-functional Requirements: This section contains non-functional requirements of the software. It includes the security, performance, safety, software and business requirements. The intended audience for this section is the team of developers working on this software.

Section 6 – Other Requirements: This section includes other requirements such as database, legal and internationalization requirements etc.

## Product Scope

The purpose of this credible seller’s list is to allow users to buy items based on the credibility of sellers. Creda is software where users can buy and sell items. What makes it different is that users will be able to see how credible a seller is and then choose to buy from them. Users will be able to sign up and add connections who will then be given credibility based on categories such as family, friend, acquaintance, etc.

The UI will be very intuitive so that the application is easy-to-use for the users of Creda. This will ensure that users of all ages can easily use this software without facing any difficulties. To prevent any unauthorized access, an authentication system will be added which will ensure privacy of the users.

## References

* https://dipeshagrawal.files.wordpress.com/2018/07/docuri-com\_online-project.pdf
* https://www.emvco.com/emv-technologies/src/
* https://kupdf.net/download/srs-of-e-commerce-online-book-shopping\_590796f8dc0d601466959edf\_pdf
* https://ieeexplore.ieee.org/abstract/document/9402781/
* https://www.ornl.gov/~webworks/cppr/y2001/pres/116675.pdf
* https://www.slideshare.net/hamnalatif9/olx-features-e-ecommerce
* https://thetechpanda.com/top-features-olx-standalone/17088/
* https://github.com/MertUsenmez/Software-Requirement-Specification/blob/master/SRS%20PROJECT.pdf
* https://www.slideshare.net/OnlyTechForYou/online-ecommerce-website-srs

# Overall Description

## Product Perspective

Creda is a credible seller list application that is more like OLX. Creda is an application that supports a number of functions for both sellers and buyers. The difference is Creda is based on credibility, when you add somebody as your companion, you add validity to the individual, this way the application keeps track of the user. Credibility scores make buying and selling more transparent. The application will be available for anyone with a mobile or pc with an internet connection.

## Product Functions

Creda will provide several functions each is listed below:

1. Login/Signup

* New users will need to sign up by email and set their password and user name.
* Existing users will log in to their account using their unique username and password.

1. Connection feature

* New users will give credibility scores to other users based upon their relationship.
* New users will be given suggestions by the system for connection for a better understanding of the user.
* Search option will enable users to add connections by searching.

1. Maintain records for users

* Credibility Score is maintained after every update.
* Users average time spent after login.
* Sellers store information like quantity and quality of the product.
* Report: Users can report other users.
* Users can change their own account details.

1. Easy advertising (Seller)

* Sellers can easily list their product by attaching pictures.
* Sellers can add the description and details of the product.
* Sellers can view their ads list.
* Buyers can view all the listed products by the seller.

1. Administrative Account

* Admins can add users.
* If a user is found to violate the policies then they will be removed.
* Product listings can be removed by admins if they violate the system policies.

1. Customer data management:

* After login users will be given a credibility score based on their relation.
* Users can only give credit scores to their connections.
* The data structure is a tree/graph.
* Categorization based on relation.

1. Access

* Buyer can only access the seller's credibility score from the buyer’s data.
* Admins have access to the whole data of buyers and sellers.
* Log/record the listings
* Anyone may sign up for a customer account.

1. Facility of using the service on desktop and mobile

* Seamless integration between desktop and mobile web app

## User Classes and Characteristics

The typical Creda user is simply anyone that has access to the internet and also has access to a mobile phone or computer. The person should be computer literate to be a Creda user. There are two types of Creda users:

1. Buyers:

The users who are looking to buy products from some brand that are trustworthy. Creda provides a platform to provide what buyers are looking for as buyers will be able to access the credibility scores of sellers and buy according to their needs.

1. Sellers:

The users who want to sell their products online. The seller will be able to access, credibility score of the buyer so they would know that they are not selling their product to someone who would fraud.

## Operating Environment

The Credomat will work under a Webpage based environment running on HTML, Reactjs, CSS, and database per the program needs.. Creda is website therefore will operate in all widely known browsers such as Google Chrome, Mozilla Firefox, Opera, Internet explorer etc. Further specifications for this will be elaborated under section 3.3: Software Interfaces.

The application will be made as a web application compatible to run on both a desktop as well as a smartphone.As such the only requirement to use this product would be an active internet connection.

Creda will be fetching user data from social media applications such as Facebook or LinkedIn. Since all the data will be imported from and exported to the cloud server, the user must have an active internet connection. For better understanding of this refer to FR5 in the section 4.1.3.

The basic input devices required would be a keyboard, a mouse and a monitor for a desktop system, and a touchscreen smartphone if you use Creda on your cell phone.

## Design and Implementation Constraints

The API will be integrated as a web-based application that can be accessed from a desktop, or a smartphone therefore will not be compatible with older phones.

All the data from the user will be saved on cloud storage, accessing, and updating it may suffer lag.

Since the data stored on the cloud will of course include the user’s personal information as well, it may cause security concerns or possible data malfunction.

The maintenance will be regulated on users’ end as well as the clients, as the application is based around user accommodation it will be simple to work with.

Applications and tools to be used, Reactjs, and database to be used are yet to be decided.

The application must be working in parallel with the database to avoid misinformation.

The software will be monolingual and therefore will only be available in the English language. However, shifting to a bilingual or even a multilingual platform in the future release is up for discussion.

## User Documentation

Refer to subsection 3.1: User Interface under section 3: External Interface Requirements.

## Assumptions and Dependencies

The software is designed under the assumption that it will work incoherent with third-party social media applications such as Facebook or LinkedIn, which the user will allow access to, and by which a network will be formed on the application itself by using the user's profile and making a tree of the connections.

The application will be able to keep track of your connections and regularly update them.

The credibility value shown will reflect on how strong the relationship between any two nodes (persons) in a link is.

The database works in conjunction with the seller’s inventory and functions concurrently with the application to ensure that the API always indicates the true value for the said inventory.

# 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.>*

1. **System Features**

**4.1 Friendly User Interface**

4.1.1 Description and Priority

For any service, it has become necessary to provide a user-friendly interface and make things as easy as possible for users to access in order to make your service available to a larger audience. We think of it as a high-priority feature because of its larger benefits and importance.

4.1.2 Stimulus/Response Sequences

Sign Up:

* User visits the web app.
* User enters his email address and details.
* User makes his account.
* System displays a congratulatory message for joining the platform as “Congratulations Username for joining Creda”.

Login:

* User visits the web app.
* User enters his credentials.
* User logs in to his account.
* System displays a “Welcome” message with a pleasant notification sound.

4.1.3 Functional Requirements

FR1 - Sign Up Feature:

* Users visit the web app, they will see a signup page.
* They can create an account by signing up with their emails.

**Use Case Name Sign Up**

**Related Requirements** FR 1

**Goal in Context** Person creates a new user account

**Preconditions** Valid and working email ID

**Successful End Condition** A user account is created for the user

**Failed End Condition** User account is not created

**Primary Actor** User

**Secondary Actor** None

**Trigger** The user clicks sign up to create new account

**Included Cases Check Email**

**Main Flow Step** **Action**

1. The user asks system to create a new account.
2. The user enters his details.

**3** The user’s details are checked.

include:Check Email

**4** The new user account is created.

FR2 - Login Feature:

* Users visit the web app, they will see a login page.
* Existing users can log in to their accounts by entering their credentials.

**Use Case Name Login**

**Related Requirements** FR 2

**Goal in Context** Person logs in to existing user account

**Preconditions** Existing User Account

**Successful End Condition** User logs in to his account

**Failed End Condition** User failed to login

**Primary Actor** User

**Secondary Actor** None

**Trigger** The user clicks log in to existing account

**Included Cases Check Email**

**Main Flow Step** **Action**

1. The user asks system to create a new account.
2. The user enters his details.

**3** The user’s details are checked.

include:Check Email

**4** The user logs in to his account.

FR3 - Sign Out Feature:

* Logged in users can click the sign out button to exit.

**Use Case Name Sign Out**

**Related Requirements** FR 3

**Goal in Context** Person signing out from his existing user account

**Preconditions** Existing User Account

**Successful End Condition** User signs out from his account

**Failed End Condition** User failed to sign out

**Primary Actor** User

**Secondary Actor** None

**Trigger** The user clicks sign out to existing account

**Included Cases Check Email**

**Main Flow Step** **Action**

1. The user asks system to sign out.
2. The system signs out the user.

FR4 - Add Connections Feature:

* A new user when he sets up his account and logs in on the app.
* This is a one-time feature for all users and displays some users of the app.
* The new user can add them as connections.
* Even if a new user doesn’t add any connection through this feature, there are other ways.

**Use Case Name Add Connections**

**Related Requirements** FR 3

**Goal in Context** Person adding first new connections on his profile

**Preconditions** Existing User Account

**Successful End Condition** User adds connections

**Failed End Condition** User is not able to add connections

**Primary Actor** User

**Secondary Actor** None

**Trigger** The user asks system database to connect with another user

**Included Cases None**

**Main Flow Step** **Action**

1. The user finds someone relevant to him.
2. The user adds him.

FR5 - Make Connections Page:

* This page displays suggested connections for the user by the system on the basis of the details that user has entered.
* The user can add connections from this page as per his desire.
* To have a better understanding, it would work similarly to the ‘Network’ page on LinkedIn and ‘People You May Know’ on Facebook.

**Use Case Name Making Connections through Suggestions**

**Related Requirements** FR 4

**Goal in Context** Person adding new connections on his profile

**Preconditions** Existing User Account

**Successful End Condition** User adds connections

**Failed End Condition** User is not able to add connections

**Primary Actor** User

**Secondary Actor** None

**Trigger** The user asks system database to connect with other user

**Included Cases None**

**Main Flow Step** **Action**

1. The user heads over to Make Connections Page.
2. The user finds users relevant to him.
3. The user adds them.

FR6 - Search Option:

* All Users will have the search option
* They can use it to find the people they know.
* They can add them to their connection’s list.

**Use Case Name Making Connections through Search**

**Related Requirements** FR 5

**Goal in Context** Person adding new connections on his profile

**Preconditions** Existing User Account

**Successful End Condition** User adds connections

**Failed End Condition** User is not able to add connections, User doesn’t find desired connection

**Primary Actor** User

**Secondary Actor** None

**Trigger** The user asks system database to connect with another user

**Included Cases None**

**Main Flow Step** **Action**

1. The user uses search option.
2. The user type keywords to find a particular user.
3. The user scrolls through the list of users matching keywords.
4. The user finds the user he wanted.
5. The user adds him.

**4.2 Credibility Metric**

4.2.1 Description and Priority

It is the most important feature of our web app and distinguishes our app from other buying and selling platforms. Each user will have a credibility score which may or may not be the same for all users. For e.g. X has 9 credibility in Y’s world, but X can have 7 credibility for Z because each user will have a personalized environment or in short each user has his own world.

4.2.2 Stimulus/Response Sequences

User assigns relation with other user:

* If he assigns a family relation to another user, then a thanking message “Thank You for adding Family Members” is displayed for some moments.
* If he assigns a family relation to another user, then a thanking message “Thank You for adding Friends” is displayed for some moments.
* If he assigns a family relation to another user, then a thanking message “Thank You for adding Acquaintance” is displayed for some moments.

4.2.3 Functional Requirements

FR7 - Graph/Tree Data Structure (TBD):

* The app thinks of creating a world for every user.
* We can achieve this either through tree data structure or graph data structure.
* The user would be the root node and his connections would act as child nodes.
* This would be happening at the backend when the user is adding connections thus use cases of FR 4, FR 5, and FR 6 would be having it running at backend.

FR8 - Categorization of Relations:

* Whenever a user adds a connection, the user will get an option list.
* The app would give users three options to identify relations from that list.
* The user needs to identify the relation from the following list :

1. Family
2. Friend
3. Acquaintance

**Use Case Name Identifying Relationship with Connection**

**Related Requirements** FR 7

**Goal in Context** Person adding new connections on his profile and identifying relation

**Preconditions** Existing User Account, User adding connection

**Successful End Condition** User adds connections with relation specified

**Failed End Condition** User is not able to add connections and identify relationship

**Primary Actor** User

**Secondary Actor** Another user

**Trigger** The user asks system database to connect with other user

**Included Cases None**

**Main Flow Step** **Action**

1. The user is adding a connection.
2. The system asks user to identify the relationship.
3. The system provides a list of relationships.
4. The user successfully identifies the relation with connection.

FR9 - Credibility Score by User:

* During the connection addition process, the user will assign a credibility score to his connection.
* The system saves it.
* The system uses it according to the network of other users to display personalized credibility for every other user.

**Use Case Name Assigning Credibility to Connection**

**Related Requirements** FR 8

**Goal in Context** Person adding new connections on his profile and assigning credibility score

**Preconditions** Existing User Account, User adding connection

**Successful End Condition** User adds connections with assigned credibility score

**Failed End Condition** User is not able to add connections and assign credibility

**Primary Actor** User

**Secondary Actor** Another user

**Trigger** The user asks system database to connect with another user

**Included Cases None**

**Main Flow Step** **Action**

1. The user is adding a connection.
2. The system asks user to assign the credibility to the user.
3. The user assigns a credibility score.
4. The system saves it and uses it according to different conditions.

FR10 - Computational Formula (TBD):

* This formula will be used to determine the credibility of a seller who has a product in which the user is interested.
* Credibility would be computed on the basis of the number of  nodes (connections) involved between the user and seller.
* Credibility would also depend on the type of relation of those nodes.
* After reaching a certain level of graph/tree, a computational formula(which is under working) would be used for better calculation of credibility.
* This would be happening at the backend when the user is adding connections thus use cases of FR 4, FR 5, FR 6, FR 7, and FR 8 would be having it running at backend.

**4.3 Seller/User Interaction**

4.3.1 Description and Priority

Facilitating interactions between the user and the seller through easy access to necessary information is a key feature of our program. The main purpose of the application is to provide users with a platform where they can buy and sell products with the assurity that the people they are dealing with are reliable and trustworthy. This feature focuses on making the communication process easy and hassle free.

4.3.2 Stimulus/Response Sequences

* In FR13, high credibility score will be displayed with a green color
* In FR14, low credibility score will be displayed with a red color
* In FR13, product prices will be displayed in bold

4.3.3 Functional Requirements

FR11 - Product views:

* The User will be displayed a list of all the available products
* The products must not be from seller who are not a part of the users Tree/Graph
* The products will be displayed with their name and price
* Chat option will be available to the user

**Use Case Name View Product Details**

**Related Requirements FR11, FR12, FR13**

**Goal in Context** A User wishes to view details regarding a listing

**Preconditions** User must be login into his/her account

**Successful End Condition** A product is displayed with a credibility score, and all other information, for the user to view

**Failed End Condition** An error message stating “the product has been de-listed”

**Primary Actor** User

**Secondary Actor** User (As Seller)

**Trigger** The User clicks on a product from a list of products

**Included Cases View Seller Details, View Seller’s Credibility Score**

**Main Flow Step** **Action**

1. The User clicks on a product from a list of products
2. Seller’s Contact details are retrieved
3. Seller’s credibility score is retrieved
4. Product details are retrieved
5. The Product pages opens up with all the information

FR12 - Seller’s Contact Details:

* The User will be displayed a specific sellers’ details
* The details will include his phone number and a brief “about me”
* The details must also include the seller’s credibility score

**Use Case Name View Seller’s Details**

**Related Requirements FR12**

**Goal in Context** Retrieving the seller’s contact info for a user

**Preconditions** User must be logged in into his/her account

**Successful End Condition** Sellers Contact information is retrieved

**Failed End Condition** An error message stating “Could not retrieve contact info”

**Primary Actor** User

**Secondary Actor** User (As Seller)

**Trigger** The User triggers this event by either viewing a product listing or a seller’s page

**Included Cases None**

**Main Flow Step** **Action**

1. The User triggers the event
2. Sellers contact info is retrieved

FR13 - Credibility Score:

* The credibility score must be assigned to every connection for a user
* It will be used to assess a seller’s credibility
* It must be displayed alongside every listing

**Use Case Name View Seller’s Credibility Score**

**Related Requirements FR13**

**Goal in Context** Retrieving a seller credibility score

**Preconditions** a Tree/Graph Exists for the User

**Successful End Condition** retrieve credibility score

**Failed End Condition** state message “A connection does not exist”

**Primary Actor** User

**Secondary Actor** User (As Seller)

**Trigger** The user views seller details or a product listing

**Included Cases View Connection**

**Main Flow Step** **Action**

1. View connection list
2. Search for the specific user
3. Retrieve his credibility score
4. Pass the score to the trigger function

FR14 - Report:

* When viewing a product listing, a user should have an option to report
* When the user chooses to report a listing, he should be presented with a report form
* the form should include options stating reasons why he chooses to report the listing
* He may choose to describe his reasons

**Use Case Name Report**

**Related Requirements FR14**

**Goal in Context** For a user to report a listing to the admin

**Preconditions** A listing must exist to report

**Successful End Condition** The listing is reported

**Failed End Condition** The listing is not reported

**Primary Actor** User

**Secondary Actor** Admin, User (As Seller)

**Trigger** The User clicks onto the report button on a listing

**Included Cases View Product Details**

**Main Flow Step** **Action**

1. A report form opens up for the user
2. The user is prompted to select a reason
3. The user may add a description if he wishes
4. The report is sent to the admin

**4.4 Easy Advertisement**

4.4.1 Description and Priority

The process of listing a product to the application must be as hassle free as possible. Ease of doing business on the application is of utmost priority, thus during its development must be taken as such, the application must be inviting towards both people wishing to sell and buy.

4.4.2 Stimulus/Response Sequences

* In FR16, after a product has been added, a message stating “Product has been listed” will be displayed
* In FR17, after a product has been Altered, a message stating “Product has been altered” will be displayed
* In FR15, each listing will be displayed in a rectangular fashion

4.4.3 Functional Requirements

FR15 - View List Of Products:

* A list of all the products, based on his connections, must be displayed when the user logs in
* If the user has no connection, it must display a message stating so
* the list must include the name and price of the products
* the list may also include a picture of the product
* the user must be able to interact with a listing in such a way that when he clicks on it, he is taken to a page with all the details of that listing

**Use Case Name View Product List**

**Related Requirements FR15**

**Goal in Context** the User will have a list of products from his personal connections with a link to the view product page

**Preconditions** a Tree/Graph Exists for the User

**Successful End Condition** A view of multiples products from the user’s connections are displayed

**Failed End Condition** Print the message “no listings available”

**Primary Actor** User

**Secondary Actor** Users (As Seller)

**Trigger** The User logins to his account

**Included Cases View Product Details**

**Main Flow Step** **Action**

1. The user’s tree/graph is retrieved
2. The tree is traversed to identify all the sellers
3. Product details listed by these sellers are retrieved
4. A list of these products is displayed

**Use Case Name Chat**

**Related Requirements FR11**

**Goal in Context** For the user to have a platform to chat with the seller

**Preconditions** A listing must exist

**Successful End Condition** The seller and user communicate with each another

**Failed End Condition** State the message “the seller has disabled chat feature”

**Primary Actor** User

**Secondary Actor** User (As Seller)

**Trigger** The User clicks onto the chat button on a listing

**Included Cases View Product Details**

**Main Flow Step** **Action**

1. FAQ window opens up for the user
2. The user is displayed a list of frequently asked question, their answers, and a chat option
3. If he chooses the chat option a chat window opens for the user
4. The seller responds and a connection is established

FR16 - Add New Product:

* The user must have an option to sell on his home page
* He should be presented with fields to input data in
* The user must not leave the name and price field empty
* The product should be available to other users right after it is added

**Use Case Name List Product**

**Related Requirements FR16**

**Goal in Context** create a new product for the user to list on his page

**Preconditions** User must be logged into his/her account

**Successful End Condition** A new product will be created

**Failed End Condition** The product will not be created

**Primary Actor** User

**Secondary Actor** None

**Trigger** The User clicks on a sell button on his home page

**Included Cases None**

**Main Flow Step** **Action**

1. The user adds name, price, details, and description
2. A new product is created
3. This product is available for other users

FR17 - Alter Product:

* The user must be able to update a product after he has posted it
* All fields can be altered
* Once updated, the product should be listing with the new altered data

**Use Case Name Alter Product**

**Related Requirements FR17**

**Goal in Context** alter details in an existing product

**Preconditions** A product listing must exist for the user

**Successful End Condition** product details will be altered

**Failed End Condition** the product details shall remain the same

**Primary Actor** User

**Secondary Actor** User (AS SELLER)

**Trigger** The User clicks on a change button for a product

**Included Cases None**

**Main Flow Step** **Action**

1. Alter page will pop up
2. User selects the fields he wishes to alter
3. He changes the details for the said field
4. Confirm changes
5. Product details are altered

FR18 - Seller’s Page:

* A user should be able to view all of the seller’s listings
* The page must also include the seller’s credibility score
* It must also include the seller’s contact details
* All of the listings must include their names and prices

**Use Case Name View Seller’s Details**

**Related Requirements FR18**

**Goal in Context** A page where all the listings from a particular seller are available

**Preconditions** A Seller must exist with at least one listing

**Successful End Condition** A list of products from a particular seller are displayed

**Failed End Condition** print the message “No listings available from this user”

**Primary Actor** User

**Secondary Actor** User (As Seller)

**Trigger** The User clicks onto a particular seller

**Included Cases View Seller’s Credibility Score, View Seller’s Details**

**Main Flow Step** **Action**

1. The user clicks onto a particular seller from the sellers listing
2. Seller’s credibility score is retrieved
3. Seller’s product listings are retrieved
4. The seller’s contact details are retrieved
5. The seller’s details with his credibility score are displayed on the top with his/her product listings at the bottom

**4.5 Administrative Features**

4.5.1 Description and Priority

To maintain records and manage users, administrative privileges must be offered to the client. For this purpose, administrative features must be included in the application. These features must also be protected as they cannot be used by ordinary users, thus only an administrative account shall be granted these features.

4.5.2 Stimulus/Response Sequences

* In FR19, if a suspended user tries to login he will be presented with the message “Your account has been suspended” in red
* In FR19, if a banned user tries to login he will be presented with the message “Your account has been banned from this site” in red
* In FR20, if a users listing is delisted, he will be presented with the message “One of your product’s has been delisted”

4.5.3 Functional Requirements

FR19 - Remove Users:

* An administrator shall have the right to remove any user from the app
* The administrator will be able to search for a user using his username
* He may choose to impose a temporary suspension or ban the account for life

**Use Case Name Remove Users**

**Related Requirements** FR 19

**Goal in Context** Administrator can remove the users due to policy violation

**Preconditions** Existing User Account, Admin Login, Admin receives report of user violating the policy

**Successful End Condition** Administrator removes user

**Failed End Condition** Administrator is not able to remove user

**Primary Actor** Administrator

**Secondary Actor** User

**Trigger** The administrator clicks remove user option

**Included Cases None**

**Main Flow Step** **Action**

1. The admin receives a report.
2. The admin investigates if the policy violation report is legitimate.
3. If found report to be true, admin removes user.
4. Else, the admin dismisses the report.

FR20 - Remove Listing:

* An administrator shall have the right to remove any listing from the app
* The administrator will be presented with listing that have been reported by other users
* he may choose to de-list the listing or product

**Use Case Name Remove Product Listing**

**Related Requirements** FR 20

**Goal in Context** Administrator can remove the product listing due to policy violation

**Preconditions** Existing User Account, Admin Login, Admin receives report of product listing violating the policy

**Successful End Condition** Administrator removes product listing

**Failed End Condition** Administrator is not able to remove product listing

**Primary Actor** Administrator

**Secondary Actor** None

**Trigger** The administrator clicks remove product listing

**Included Cases None**

**Main Flow Step** **Action**

1. The admin receives a report.
2. The admin investigates if the policy violation report is legitimate.
3. If found report to be true, admin removes user.
4. Else, the admin dismisses the report.

FR21 - Maintain Records:

* An administrator will have alteration rights to privileged data
* He will have access to the list of all the users and products

**The Final diagram of Use Cases is attached below:**

Diagram

Description automatically generated

# 5. Other Nonfunctional Requirements



## Performance Requirements

PR 1: In the case of an increase in website visitors if it exceeds 500 visitors the requirement is to handle the visitors with good response time, whether the website traffic is low or high.

PR 2: an estimated time will be provided to the user when the user updates his/her details. No message will be provided for response time greater than 0.1 and less than 1.0. This will be dealt with in real-time.

PR 3: For 95% of all response time should be less than 10.0 seconds.

PR 4: less than 1% acceptable error rate is allowed during measurement of response times.

## Safety Requirements

SR 1: in case of system failure or loss of data we will have database backup.

SR 2: A user must not give credibility of 10 to an acquaintance, because it’s a safety constraint of a function.

SR 3: Our system will detect data loss of type X at least Y percent of the time.

## Security Requirements

SR 1: Our system will have a secure database for end-users’ data.

SR 2: Users of the website will not have access to change anything other than their profile.

SR 3: Every user will have access constraints because in our system there will be different types of users in terms of categories (family, friend, acquaintance).

## Software Quality Attributes

This section specifies the required system quality factors that are not related to the specific functional requirements documented in the use case model

#### 5.4.1 Accessibility

This is associated with Creda being accessible from any device it is accessed with. As it is a web application, therefore, it will be ensured that it is accessible via any web browser be it from a personal computer or a smartphone. Creda will be accessible regardless of the medium it is accessed with. Also, any graphical user interfaces of the Creda shall use adequate font size to be usable by persons with limited visual acuity.

#### 5.4.2 Correctness

As the system of Creda will constantly be improved and certain bugs will also be detected and fixed constantly. Therefore, our system will make sure that our system continues to work for every user even if it contains some bugs which are then fixed.

#### 5.4.3 Maintainability

Our system ensures the ease with which the system can be maintained:

Our system will permit the swapping and upgrade of the web application systems without downtime.

#### 5.4.4 Portability

This subsection specifies the following requirements associated with the ease with which the system can be moved from any environment (Personal computers, smartphones, etc).

This web application will be accessible via:

* Web Browsers in PCs
* Web Browsers in Smartphones/Tablets

### 5.4.5 Reliability

The following requirements are associated with reliability.

The fact is that there can be no manipulation done with the credibility score of a user given to him/her by fellow users. The system of Creda will make sure that this isn’t possible.

### 5.4.6 Safety

The following requirements are associated with the degree to which the system does not directly or indirectly (e.g., via inactivity) cause accidental harm to life or property or loss of money or data of the user.

* Creda shall not accidentally lose user account information.Creda shall not accidentally lose information regarding who.
* Creda shall not accidentally lose information regarding the credibility or relationships of the users

### 5.4.7 Robustness

This subsection specifies the following requirements associated with the degree to which the system continues to properly function under abnormal circumstances.

* Creda should gracefully handle invalid input (i.e., detect invalid input, request valid input, and not crash) from all externals e.g., the human actors.
* Creda should gracefully handle failures (i.e., provide notify the system operator, and not crash).

#### 5.4.8 Auditability

The system supports independent auditing of its user interactions. Creda will maintain a record of each user inquiry, response, and any interaction amongst the users.

#### 5.4.9 Interoperability

This requirement is associated with the ease with which the system can be integrated with another system (e.g., browsers). Creda shall interoperate with any browser it’s accessed which include but are not restricted to Google Chrome, Safari, Opera Mini, Microsoft edge

### 5.4.11 Performance

This subsection specifies the following requirements associated with the speed with which the system shall function. Creda shall provide users with the service at the fastest pace possible without any significant delays.

### 5.4.12 Accuracy

The following requirements are concerned with the degree of correctness of the system’s outputs:

* Values of money shall be correct to the nearest penny.
* Values of time of the interaction amongst the buyer and seller shall be correct to the nearest second.

## Business Rules

**The business rules comprise over the following principles:**

* It is mandatory for the sellers to enter their contact details. Without it, they won’t be allowed to proceed to sell anything as their contact details will be shared with potential customers.
* The user’s contact details won’t be disclosed to anyone. They will have to enter their contact details, but their contact details would be kept private as no one will be able to access their contact details. This is to maintain the privacy of the user.
* Only the changes approved by the user will be implemented on their profile. Any change won’t be implemented until it’s approved by the user.
* When a user establishes a connection with another user that other user won’t automatically establish a connection with this user who initiated the connection. He/She will have a choice to establish a connection with this user. Only if this user chooses to revert the connection, they both will establish a connection.

# Other Requirements

The other requirement for this software includes the database. Creda will be using a database TBD to manage user data. No special legal requirements exist for this software. Moreover, Creda will handle errors to ensure that there is no loss of data, and the downtime period will not be long. For system requirements refer to section 4, there are no additional requirements of creda. The other requirement can be database, say, Oracle 19c, which we will be using to store data for the app.

**Appendix A: Glossary**

|  |  |
| --- | --- |
| Software development life cycle process | Software Development Life Cycle (SDLC) is a process used by the software industry to design, develop and test high quality softwares. The SDLC aims to produce high-quality software that meets or exceeds customer expectations, reaches completion within time and cost estimates. |
| UML (Unified Modelling Language) | Unified Modelling Language. The Unified Modelling Language is a general-purpose, developmental, modelling language in the field of software engineering that is intended to provide a standard way to visualize the design of a system. |
| Boolean operations | Boolean Operators are simple words (AND, OR, NOT or AND NOT) used as conjunctions to combine or exclude keywords in a search, resulting in more focused and productive results. Using these operators can greatly reduce or expand the number of records returned. |
| External Interface Requirements | External interface requirements are types of functional requirements. They're important for embedded systems. And they outline how your product will interface with other components. |
| Functional requirements | A functional requirement defines a function of a system or its component, where a function is described as a specification of behavior between inputs and outputs. |
| Non-functional requirement | A non-functional requirement is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviors. They are contrasted with functional requirements that define specific behavior or functions |
| UI ( User Interface ) | The user interface (UI) is the point of human-computer interaction and communication in a device. This can include display screens, keyboards, a mouse and the appearance of a desktop. It is also the way through which a user interacts with an application or a website. |
| Credibility Score | The credibility score is a score given to every user by another user with whom he/she builds a connection with. This score depicts the level of credibility this person carries according to the users he/she has built a connection with. |
| HTML (Hypertext Markup Language) | The Hypertext Markup Language, or HTML is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets and scripting languages such as JavaScript. |
| Reactjs | React is a free and open-source front-end JavaScript library for building user interfaces or UI components. |
| API (application programming interface) | An application programming interface is a connection between computers or between computer programs. It is a type of software interface, offering a service to other pieces of software. A document or standard that describes how to build or use such a connection or interface is called an API specification. |
| Cloud storage | Cloud storage is a cloud computing model that stores data on the Internet through a cloud computing provider who manages and operates data storage as a service. ... This gives you agility, global scale and durability, with “anytime, anywhere” data access. |
| Stimulus/Response Sequences | Stimulus: User requests to create a new course in a specific quarter. Response: System provides a form for the user to enter the course data. |

**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**

* Computational Formula
* Data Structure to be Tree or Graph
* Database to be used
* User verification and authentication