**REQUIREMENTS ANALYSIS DOCUMENT**

1. **Introduction**
   1. **Purpose of the System**

There are so many social media websites and most of these are frequently used. However, top social media websites don’t focus on a topic. There is no social media website about culinary art concept.

Specifically, our project satisfies the need. It provides an environment to share people’s idea or an information about gastronomy or culinary. In other words, our project creates a free platform for discussion and forming an opinion.

* 1. **Scope of the System**

The scope of this project is a web-based social media platform that has the purpose of bring together people interested in gastronomy and culinary arts. In general, system has 2 different types of user; User and Admin.

Users interacts with the website. They can share any information about gastronomy or culinary they want and optionally add photos to their posts. Users can review other users' posts and interact by liking, commenting, and following other users. Another way of interaction between users is sending direct messages.

Admin is manager of the system. Admin can delete posts or comments. User verification is done by admin.

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

The success of the project is dependent on the fulfillment of the following conditions:

* Progress and completion of the project perform in synchronization with the project schedule.
* Project covers all expectation in the scope of the project.
* During the project, change requests from the customer is implemented in the due date of the project.
  1. **Definitions, Acronyms, and Abbreviations**

**Visitor:** This is unregistered person. He/she can only access to registration and login page.

**User:** This is the end-user account. A User can share, comment or like a post. He/she can interact with another User with using direct message. He/she can follow other users.

**Admin:** This is the manager account. They can manage the website. They can delete a post or a comment if he/she decides inappropriate content about it. Also, he/she can verify of the new users’ account.

**User Verification:** This is a proof given by the admin that the owner of the account is an identified real person.

* 1. **Overview**

The rest of the documents contains current system part which explains the functionality and the problems of the current system, Proposed System part which explains functional overview of the system. It includes functional and nonfunctional requirements of the system and System Models that describe the scenarios, use cases, object model, and dynamic models for the system. Glossary part contains the names and clear definitions of the participating objects in the system.

1. **Current System**

Most of the social media website has no specific concept. In this type websites, Users creates their own content such as humor, daily news, personal using and go on. Some social media websites have specific concept like LinkedIn. However, there is no social media website about gastronomy, culinary art, recipe or food history.

In this point, we noticed that there is not a free platform which anyone can share post with their own ideas or their knowledge about gastronomy and culinary arts.

1. **Proposed System**
   1. **Overview**

The system provides a social media platform that users can register to website. After, they can share post. Also, they can comment to post, and they can like to post. They can delete their posts or comments. They can update their own profile information. They can search other users. They can interact with other users with chat. Also, admin can delete posts or comments. Also, admin can verify users' profile.

* 1. **Functional Requirements**

1. Visitor can just access to registration and login panel. He/she can register to website. During the registration, visitor fills the fields that username, name, surname, e-mail, birth date, city, password. In this fields, only age and city are optional others must be filled.
2. User can login to system. He/she can share post about any information of culinary art, recipe, regional foods or something else after the login. Additionally, user can optionally add photo to their own post. Likewise, user can delete their own post if he/she thinks that this is inappropriate or irrelevant. User can like any posts. Also, he/she can comment to any post. User can like other comments. Moreover, user can delete their own comment if he/she thinks that this is inappropriate or irrelevant. Besides all these, user can interact with other users by using Direct Message. User can update their own personal information. User can search other users.
3. Admin can delete users' post if he/she thinks that this is inappropriate or irrelevant. Also, admin can delete users' comment too if he/she thinks that this is inappropriate or irrelevant. In addition to these, admin can verify users' account.
   1. **Nonfunctional Requirements**
      1. **Usability**

* Anyone with access to the internet can easily understand and use the functions of the website through understandable interface.
* The user interface is essentially a typical social media website interface.
* Frequently Asked Questions (FAQ) is included in the website.
  + 1. **Reliability**
* User information cannot be shared without sharing permission from user.
* The maintaining is performed without restarting of the system when a failure event is happened.
* Data are periodically backing up. Therefore, a important data loose will be not possible.
* User is redirected to an error page which contains possible error reasons when an exception is happened.
* User can change their password through using email if user forgot their password.
* All passwords are encrypted when user fills and sends the registration form.
  + 1. **Performance**
* The website pages changes responsively for each device such as smartphone, tablet or computer.
* The transmission of the message to another user should be in an acceptable time since messaging between users is an action that needs to be performed quickly.
* There can be no restrictions on the online users’ number since our project is a social media platform.
* 2 seconds for registration and login and 5 seconds for searching are acceptable worst latency time.
  + 1. **Supportability**
* .png, .jpg and .jpeg extensions are supported for profile picture and picture with post.
* Developers maintain the system with monthly period for encountered exceptions.
* The system is a website. The system doesn’t have any android or IOS application.
  + 1. **Implementation**
* Our project can be used in only device which has internet access.
* The source code of the project is written in JavaScript by using NodeJS Framework.
  + 1. **Interface**
* Any other existing systems don’t interact with our project.
* Data importing is happened with registration and user action inside the website.
  + 1. **Packaging**
* Developers of the website install the system.
* Node package managers in NodeJS designates number of installations.
  + 1. **Legal**
* Our project will be licensed with public domain.
* Website HTML template is purchased.
  1. **System Models**
     1. **Scenarios**

**Scenario Name** Register

**Participating Actor** Selin: Visitor

**Instances**

**Flow of Events** 1. Selin wants to share her newly learned recipe with people

might be interested. Therefore, she wants to register on the website.

2. Selin enters the register page on the website.

3. She enters her name, surname, username, date of birth, e-mail address and password to the given form on the website. 4. Then, she clicks the register button.

5. After that she redirects to login page and receives the message from the website as “You Registered Successfully.”.

**Scenario Name** Login

**Participating Actor** Cem: Visitor

**Instances**

**Flow of Events** 1. Cem wants to take a look at new posts on gastronomy. Therefore, enters the website to see new contents.

2. He clicks the login button on the main page of the website and receives a login form which includes username and the password fields.

3. He enters his username and password and then clicks to the “Login Button”.

4. Then, he notices the main page of the website and his name

on the right-top of the website.

**Scenario Name** ForgotPassword

**Participating Actor** Emre: Visitor

**Instances**

**Flow of Events** 1. Emre wants to log in to the website.

2. He clicks the login button on the main page of the website.

3. When he enters the login page, he realizes that he cannot remember his password.

4. He notices the 'forgot password' link under the password field and clicks on it.

5. He reaches forgot password page.

6. He writes his email address in the email field on the page and presses the submit button.

7. He receives feedback that his new password has been sent to his e-mail.

.

**Scenario Name** AccountVerification

**Participating Actor** Metin: Admin

**Instances**

**Flow of Events** 1. Metin logs into the site and sees that there were new requests from some users for account verification.

2. Metin checks the requests in the system and notices request from the famous chef Arda Türkmen.

3. Metin checks the information he sent and examines the ID photo of him. Then, Metin confirms that the account really belongs to Arda. He clicks the Accept Button on the verification request.

4. After that, verification request disappears and next the request comes to its place.

**Scenario Name** DeletePost

**Participating Actor** Canan: Admin

**Instances**

**Flow of Events** 1. Canan logs into the site as admin.

2. She enters the page where the posts shared by users were displayed.

3. She sees that is an inappropriate post for the website.

4. She clicks the Delete Button next to the post and removes the post from the website.

**Scenario Name** DeleteComment

**Participating Actor** Ceren: Admin

**Instances**

**Flow of Events** 1. Ceren logs into the site as admin.

2. She enters the page where the comments shared by users were displayed.

3. She sees that is an inappropriate comment for the website.

4. She clicks the Delete Button next to the comment and removes the comment from the website.

**Scenario Name** SharePost

**Participating Actor** Ali: User

**Instances**

**Flow of Events** 1. Ali wants to post about his idea for gastronomy into the website.

2. He enters the homepage which includes shares posted by people he follows.

3. He sees post sharing area above the posts.

4. He writes his ideas and clicks to Post Button.

5. After that, he sees that his post is shared.

**Scenario Name** SharePostWithPhoto

**Participating Actor** Yaren: User

**Instances**

**Flow of Events** 1. Yaren wants to post about his idea for gastronomy with adding related photo with her post into the website.

2. She enters the homepage which includes shares posted by people she follows.

3. She sees post sharing area above the posts.

4. She writes his ideas and she add a photo then she clicks to Post Button.

5. After that, she sees that his post is shared.

**Scenario Name** LikePost

**Participating Actor** Furkan: User

**Instances**

**Flow of Events** 1. Furkan enters to the website.

2. He sees the homepage which includes shares posted by people he follows.

3. Then, he sees post that he likes its content.

4. He clicks to Like Button under the related post.

**Scenario Name** DeletePost

**Participating Actor** Ahmet Can: User

**Instances**

**Flow of Events** 1. Ahmet Can enters to the website.

2. He enters that the profile page which includes shares posted by himself.

3. He notices that there is his own post that he wants to delete.

4. He clicks the Delete Button next to the post and removes the post from the website.

**Scenario Name** AddComment

**Participating Actor** Buse: User

**Instances**

**Flow of Events** 1. Buse enters to the website.

2. She sees the homepage which includes shares posted by people he follows.

3. Then, she sees post that she wants to write something to post as a comment.

4. She writes comment to post’s comment area.

5. She clicks to Comment Button under the related post.

6. After that, comment adds to post's comment area.

**Scenario Name** LikeComment

**Participating Actor** Fatih: User

**Instances**

**Flow of Events** 1. Fatih enters to the website.

2. He sees the homepage which includes shares posted by people he follows.

3. Then, he sees comment that he likes its content.

4. He clicks to Like Button near the related comment.

**Scenario Name** DeleteComment

**Participating Actor** Göksu: User

**Instances**

**Flow of Events** 1. Göksu enters to the website.

2. She sees the homepage which includes shares posted by people she follows.

3. She realizes that she has made an inappropriate comment for the website.

4. She clicks the Delete Button next to the comment and removes the comment from the website.

**Scenario Name** UpdateProfileInformation

**Participating Actor** Bora: User

**Instances**

**Flow of Events** 1. Bora enters to the website.

2. He enters that the profile page which includes shares posted by himself.

3. Then, he clicks to Settings Icon in the page.

4. He sees editable information about him, and changes fields which should be re-wrote.

5. After that, he clicks to Send Button and he redirected to homepage.

**Scenario Name** Search

**Participating Actor** Oğuzhan: User

**Instances**

**Flow of Events** 1. Oğuzhan wants to search Yaren’s account on the website.

2. He enters the homepage which includes shares posted by people he follows.

3. He writes username of Yaren in the search bar.

4. He sees accounts that Yaren’s account and some accounts that username like Yaren’s username.

5. Then, he clicks to Yaren’s account.

**Scenario Name** DirectMessage

**Participating Actor** Aykut: User

**Instances**

**Flow of Events** 1. Aykut wants to text message to Oğuz on the website.

2. He clicks to the Message Tab which includes all messages related with him.

3. Then, he clicks to Oğuz’s account.

4. He sees a panel that provides chatting with Oğuz.

5. After that he writes “hi!” to Oğuz.

**Scenario Name** FollowUser

**Participating Actor** Oğuz: User

**Instances**

**Flow of Events** 1. Oğuz wants to follow Rıdvan on the website.

2. He searches Rıdvan, and he clicks to Rıdvan’s account.

3. Then, he sees Rıdvan’s profile page.

4. He clicks to Follow Button near to Rıdvan’s username.

**Scenario Name** VerificationRequest

**Participating Actor** Rıdvan: User

**Instances**

**Flow of Events** 1. Rıdvan enters to the website.

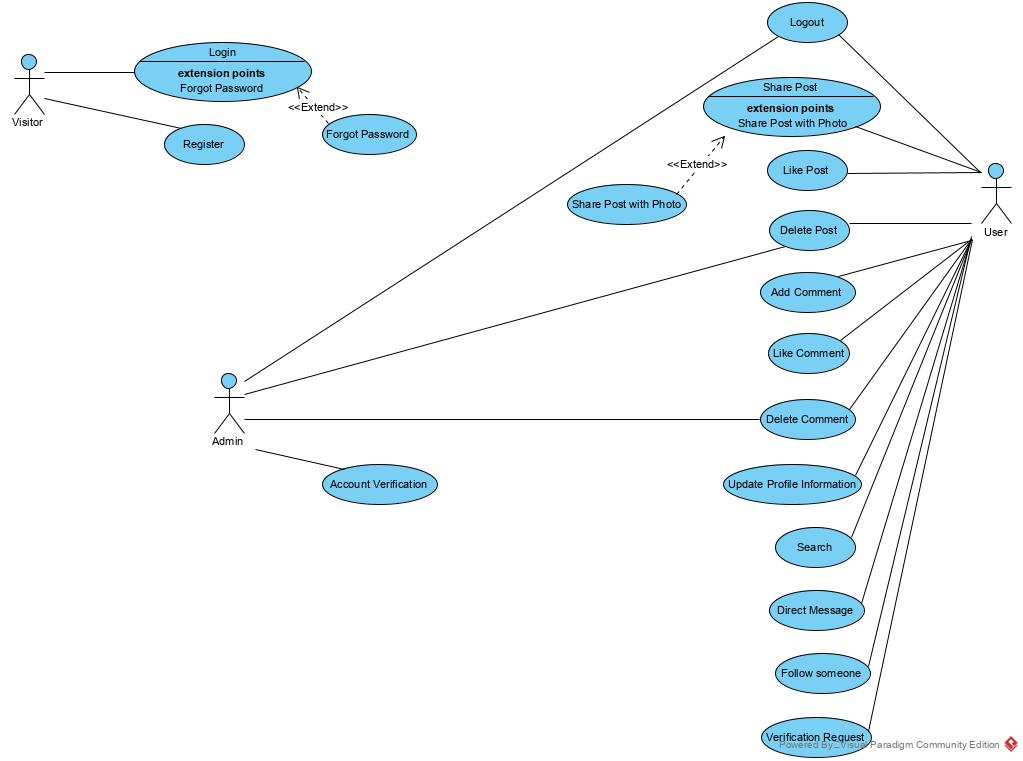
2. He enters that the profile page which includes shares posted by himself.

3. Then, he clicks to Settings Icon in the page.

4. He uploads his ID photo to website.

5. After that he clicks to Send Button.

* + 1. **Use case model**



**Use case name** Login

**Participating Actors** Initiated by Visitor

**Flow of Events** 1. Visitor clicks to “Login Button” on main page of the website.

2.Website displays a form which includes e-mail and password, both of them are required.

3. Visitor writes his/her own e-mail and password and then, clicks “Login Button” on the page.

4.Website checks his/her e-mail and password after verified them, it displays home page.

**Entry condition -** The Visitor registered to the website before.

**Exit condition -** Visitor notices his/her name and surname on the top-right menu of the website.

**Use case name** Register

**Participating Actors** Initiated by Visitor

**Flow of Events** 1. Visitor opens the register page on the website.

2. Website responds to the visitor by

displaying a register form which is included username, name, surname, e-mail, birth date, city and password, all fields are required except city and birth date.

3. Visitor fills form all of the fields on opening page and clicks register button.

4. Website displays a message as “You Registered Successfully.”.

**Entry condition -** The Visitor fills all fields with given format and his/her e-mail is not registered to the website before.

**Exit condition -** The Visitor received a message from the website about successful registration.

**Use case name** ForgotPassword

**Participating Actors** Initiated by Visitor

**Flow of Events** 1. User clicks to “Forgot Password” button on login page of the website.

2. The website displays a form with an e-mail field.

3. User writes his/her own e-mail and then, clicks the submit button on the page.

4. Website checks his/her e-mail and sends a new password via e-mail.

**Exit condition -** The user sees a message that the mail was sent successfully.

**Use case name** AccountVerification

**Participating Actors** Initiated by Admin

**Flow of Events** 1. Admin clicks to “Verification Requests” section on admin page of the website.

2. The website displays a page that includes account verification requests from users.

3. Admin clicks the Accept button next to verification request.

4. The website puts a verification sign next to his/her username and remove the request from the list.

**Entry condition -** The Admin is logged in to the website with his/her username and password.

**Exit condition -** Admin sees the request has been removed from the list.

**Use case name** DeleteComment

**Participating Actors** Initiated by Admin

**Flow of Events** 1. Admin clicks to “Comments” section on admin page of the website.

2. The website displays a page that includes comments are written by users.

3. Admin clicks the delete button next to the comment.

4. The website removes that comment from the list and website.

**Entry condition -** The Admin is logged in to the website with his/her username and password.

**Exit condition -** Admin sees that the comment was successfully removed.

**Use case name** DeletePost

**Participating Actors** Initiated by Admin

**Flow of Events** 1. Admin clicks to “Posts” section on admin page of the website.

2. The website displays a page that includes posts are created by users.

3. Admin clicks the delete button next to the post.

4. The website removes the post from the list and website.

**Entry condition -** The Admin is logged in to the website with his/her username and password.

**Exit condition -** Admin sees that the post was successfully removed.

**Use case name** SharePost

**Participating Actors** Initiated by User

**Flow of Events** 1. User enters to homepage of the website.

2. The website displays the homepage which includes shares posted by people he/she follows.

3. User writes post in the post sharing area on the upper side of the homepage then he clicks to Post Button.

4. The website displays his/her post on the homepage.

**Entry condition -** The User is logged in to the website with his/her username and password.

**Exit condition -** User sees that the post which is shared by himself/herself.

**Use case name** SharePostWithPhoto

**Participating Actors** Initiated by User

**Flow of Events** 1. User enters to homepage of the website.

2. The website displays the homepage which includes shares posted by people he/she follows.

3. User writes post with adding a photo in the post sharing area on the upper side of the homepage then he clicks to Post Button.

4. The website displays his/her post with added photo on the homepage.

**Entry condition -** The User is logged in to the website with his/her username and password.

**Exit condition -** User sees that the post which is shared by himself/herself.

**Use case name** LikePost

**Participating Actors** Initiated by User

**Flow of Events** 1. User enters to homepage of the website.

2. The website displays the homepage which includes shares posted by people he/she follows.

3. User clicks to Like Button which is related with a post on the homepage.

4. The website displays post’s like button color is changed as red and like number is incremented by 1.

**Entry condition -** The User is logged in to the website with his/her username and password.

**Exit condition -** User sees that he/she liked a post.

**Use case name** DeletePost

**Participating Actors** Initiated by User

**Flow of Events** 1. User enters to profile page of the website.

2. The website displays the profile page which includes shares posted by himself/herself.

3. User clicks to Delete Button which is related with his/her post on the profile page.

4. The website displays that post is removed from the homepage and user’s profile page.

**Entry condition -** The User is logged in to the website with his/her username and password.

**Exit condition -** User sees that he/she removed own post.

**Use case name** AddComment

**Participating Actors** Initiated by User

**Flow of Events** 1. User enters to homepage of the website.

2. The website displays the homepage which includes shares posted by people he/she follows.

3. User writes a comment to post’s comment area on the homepage and clicks to Comment Button under the related post.

4. The website displays that user’s comment under the related post’s comment area.

**Entry condition -** The User is logged in to the website with his/her username and password.

**Exit condition -** User sees that he/she added comment to a post.

**Use case name** LikeComment

**Participating Actors** Initiated by User

**Flow of Events** 1. User enters to homepage of the website.

2. The website displays the homepage which includes shares posted by people he/she follows.

3. User clicks to Like Button which is related with a comment in a post on the homepage.

4. The website displays comment’s like button color is changed as red and like number is incremented by 1.

**Entry condition -** The User is logged in to the website with his/her username and password.

**Exit condition -** User sees that he/she liked a post’s comment.

**Use case name** DeleteComment

**Participating Actors** Initiated by User

**Flow of Events** 1. User enters to homepage of the website.

2. The website displays the homepage which includes shares posted by people he/she follows.

3. User clicks to Delete Button which is related with his/her comment in a post on the homepage.

4. The website displays that comment is removed from the homepage.

**Entry condition -** The User is logged in to the website with his/her username and password.

**Exit condition -** User sees that he/she removed own comment.

**Use case name** UpdateProfileInformation

**Participating Actors** Initiated by User

**Flow of Events** 1. User enters to profile page of the website.

2. The website displays the profile page which includes shares posted by himself/herself.

3. User clicks to Settings Icon in the page.

4. The website displays that editable information about him/her.

5. User changes fields which should be re-wrote and he/she clicks to Send Button.

6. The website updates user’s information and redirects User to homepage.

**Entry condition -** The User is logged in to the website with his/her username and password.

**Exit condition -** User sees that he/she changed his/her profile information.

**Use case name** Search

**Participating Actors** Initiated by User

**Flow of Events** 1. User enters to homepage of the website.

2. The website displays the homepage which includes shares posted by people he/she follows.

3. User writes another user’s username to Search Bar which is right-top of the homepage.

4. The website displays accounts that contain searched word in their usernames.

5. User clicks to any user’s account.

6. The website displays profile page of clicked account.

**Entry condition -** The User is logged in to the website with his/her username and password.

**Exit condition -** User sees searched account.

**Use case name** DirectMessage

**Participating Actors** Initiated by User

**Flow of Events** 1. User enters to homepage of the website.

2. The website displays the homepage which includes shares posted by people he/she follows.

3. User clicks to Messages Tab that upper side of the homepage.

4. The website displays all messages that related him.

5. User clicks on message with one of the users.

6. The website displays a panel that provides to chatting.

7. User types to another user.

8. The website transmits message to another user.

**Entry condition -** The User is logged in to the website with his/her username and password.

**Exit condition -** User types to another user.

**Use case name** FollowUser

**Participating Actors** Initiated by User

**Flow of Events** 1. User enters to homepage of the website.

2. The website displays the homepage which includes shares posted by people he/she follows.

3. User writes another user’s username to Search Bar which is right-top of the homepage.

4. The website displays accounts that contain searched word in their usernames.

5. User clicks to any user’s account.

6. The website displays profile page of clicked account.

7. User clicks to Follow Button near to clicked account’s username.

8. The website stores the information of this following.

**Entry condition -** The User is logged in to the website with his/her username and password.

**Exit condition -** User follows another user.

**Use case name** VerificationRequest

**Participating Actors** Initiated by User

**Flow of Events** 1. User enters to profile page of the website.

2. The website displays the profile page which includes shares posted by himself/herself.

3. User clicks to Settings Icon in the page.

4. The website displays a panel that user can send his/her own ID photo to proof he/she is a real person.

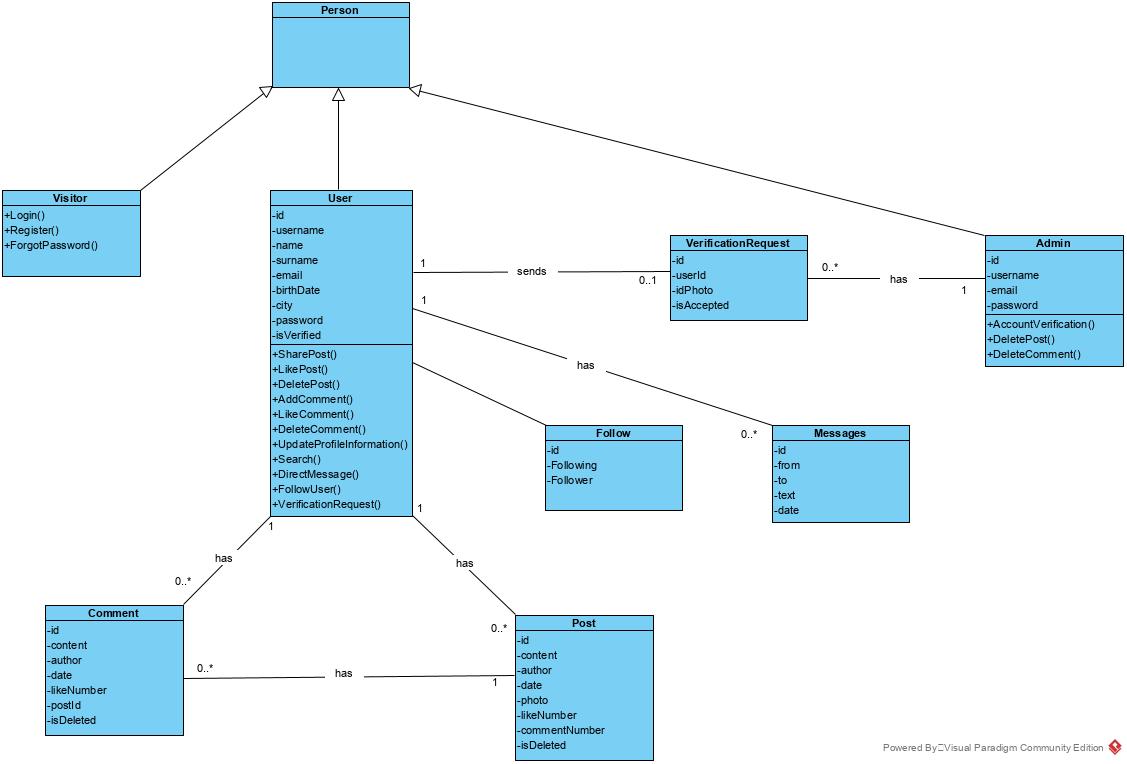
5. User uploads his ID photo to website and clicks to Send Button.

6. The website saves the request and sends to admin. After that, the website redirects User to homepage.

**Entry condition -** The User is logged in to the website with his/her username and password.

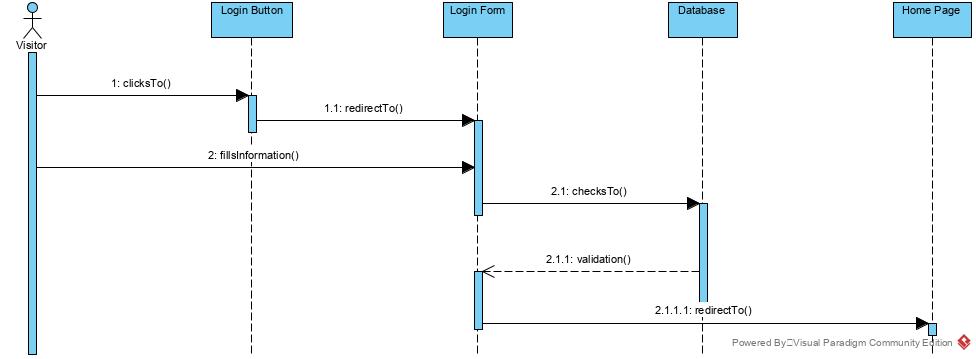
**Exit condition -** User send own ID photo to website as a request to proof he/she is a real person.

* + 1. **Object model**

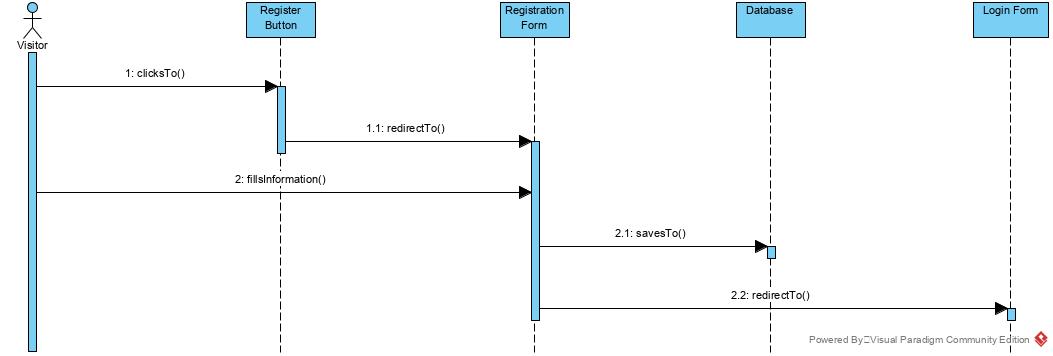
****

* + 1. **Dynamic model**

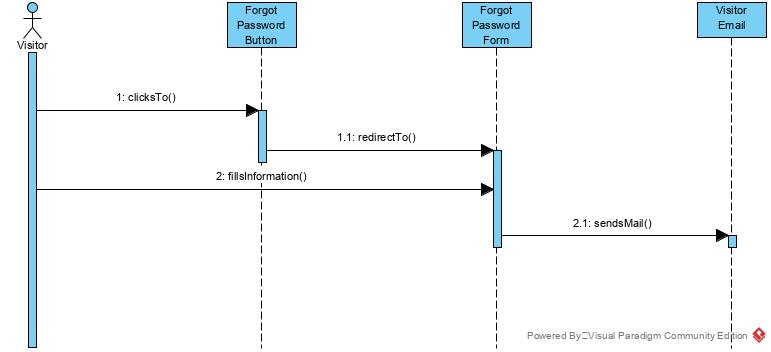
**LOGIN:**

****

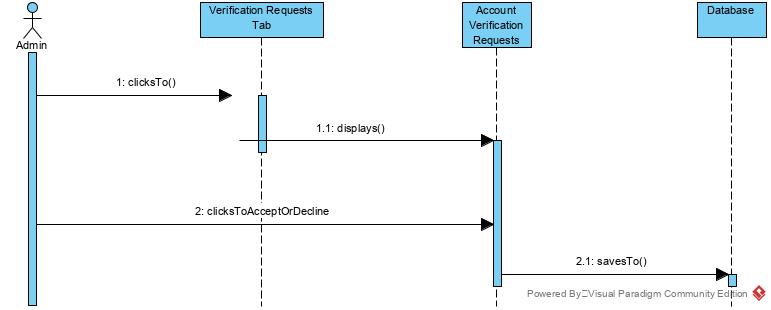
**REGISTER:**

****

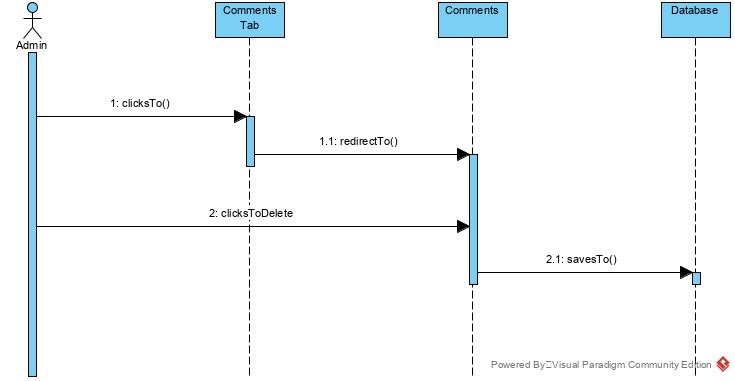
**FORGOT PASSWORD:**

****

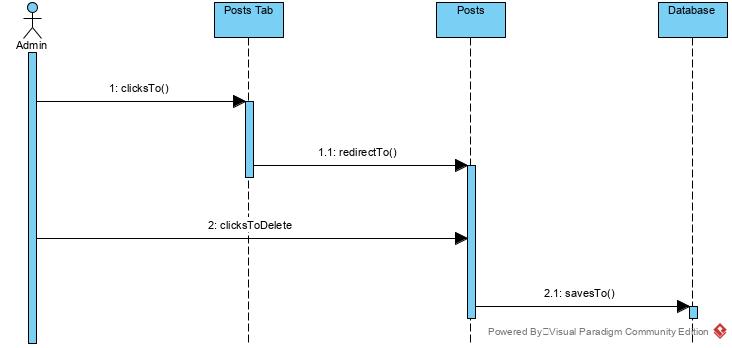
**ACCOUNT VERIFICATION:**

****

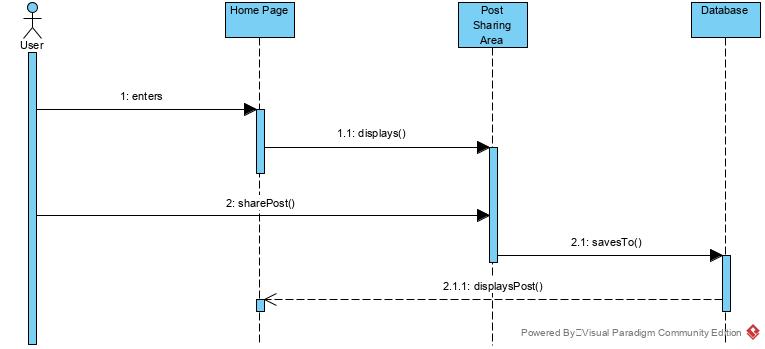
**DELETE COMMENT:**

****

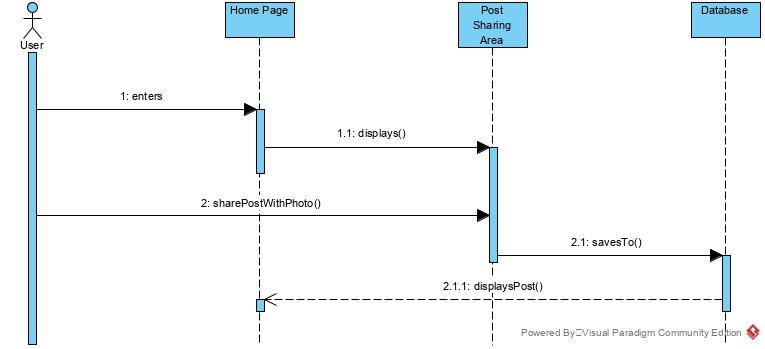
**DELETE POST:**

****

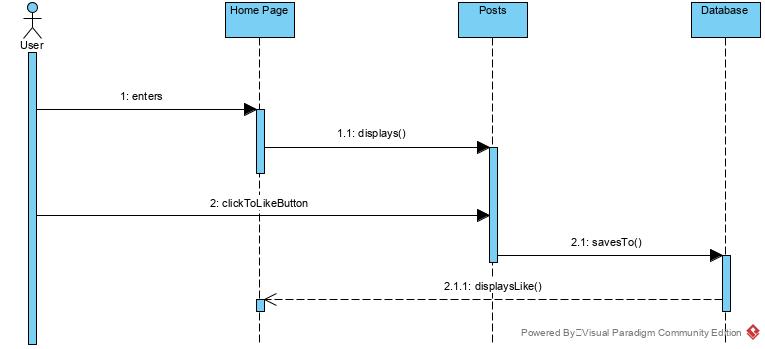
**SHARE POST:**

****

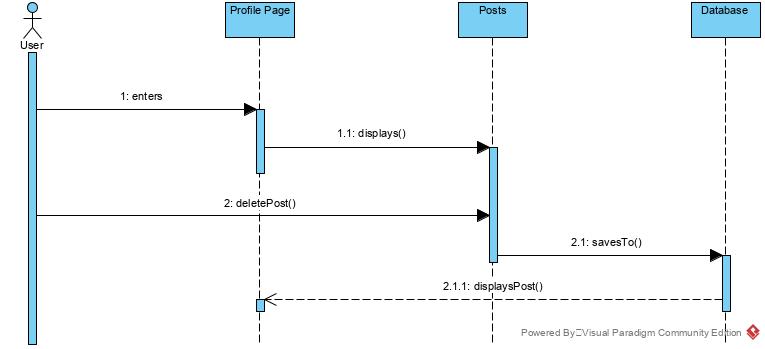
**SHARE POST WITH PHOTO:**

****

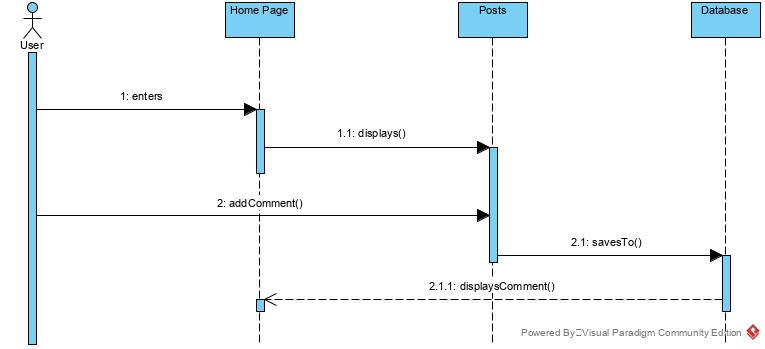
**LIKE POST:**

****

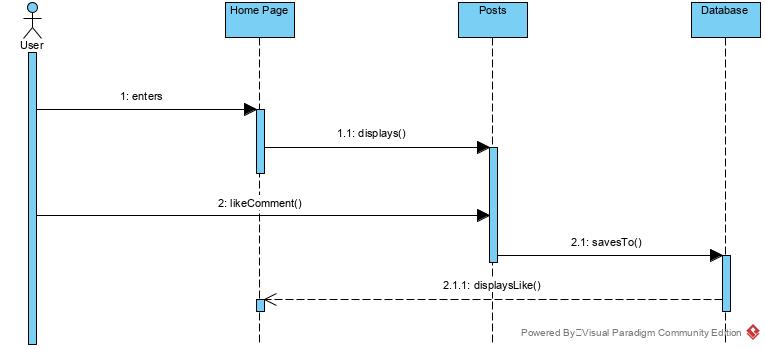
**DELETE POST:**

****

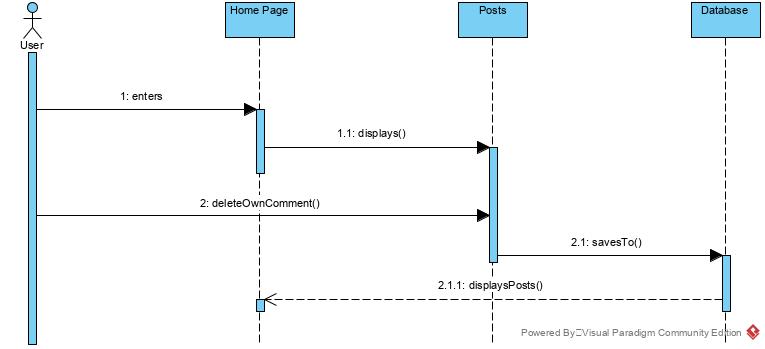
**ADD COMMENT:**

****

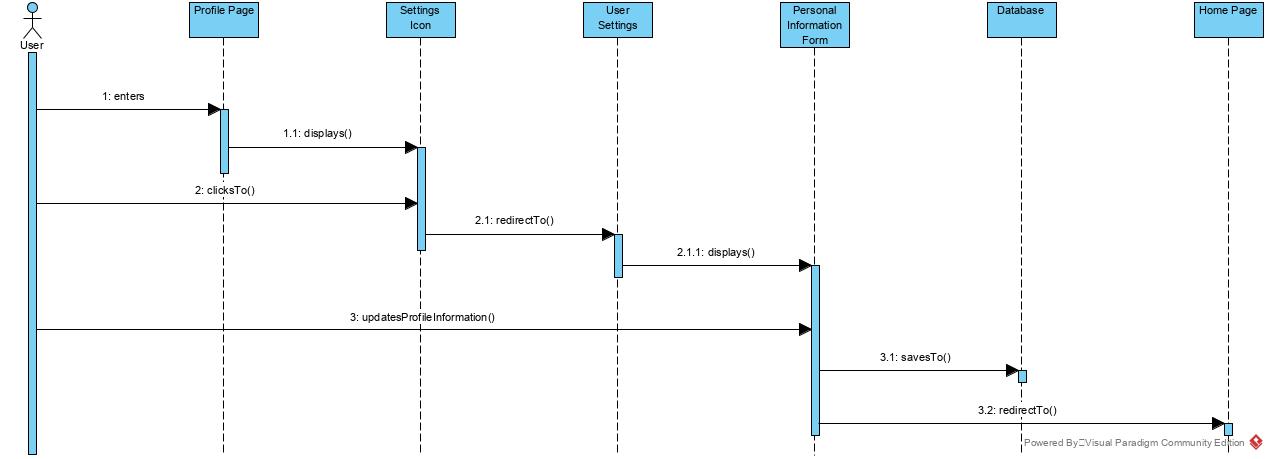
**LIKE COMMENT:**

****

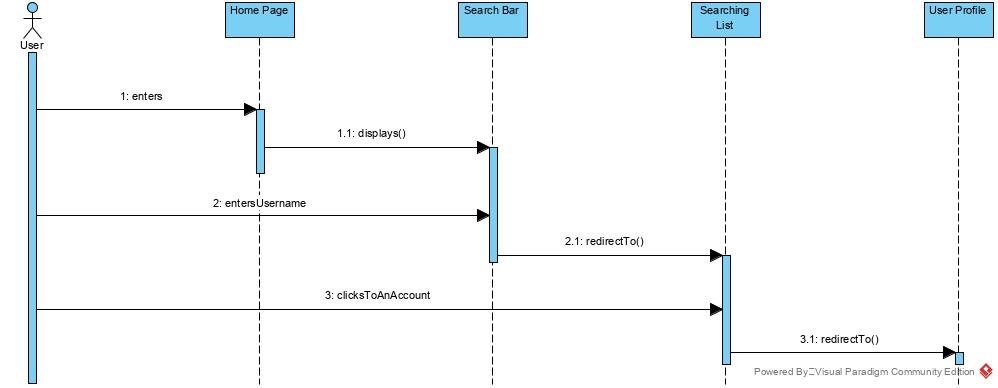
**DELETE COMMENT:**

****

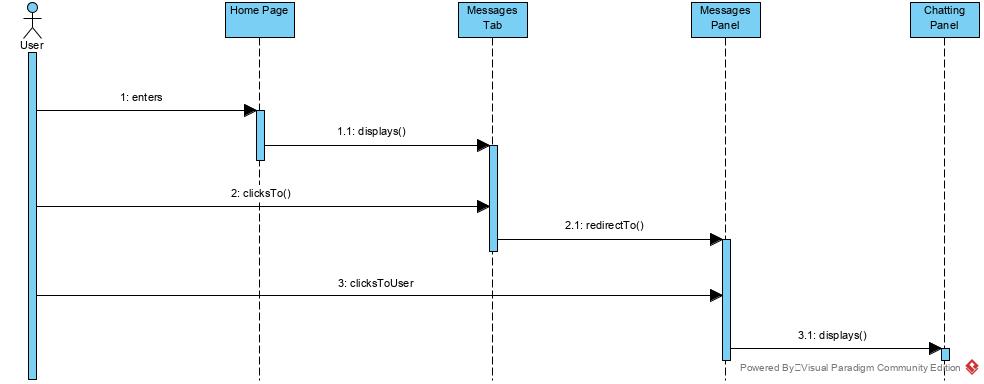
**UPDATE PROFILE INFORMATION:**

****

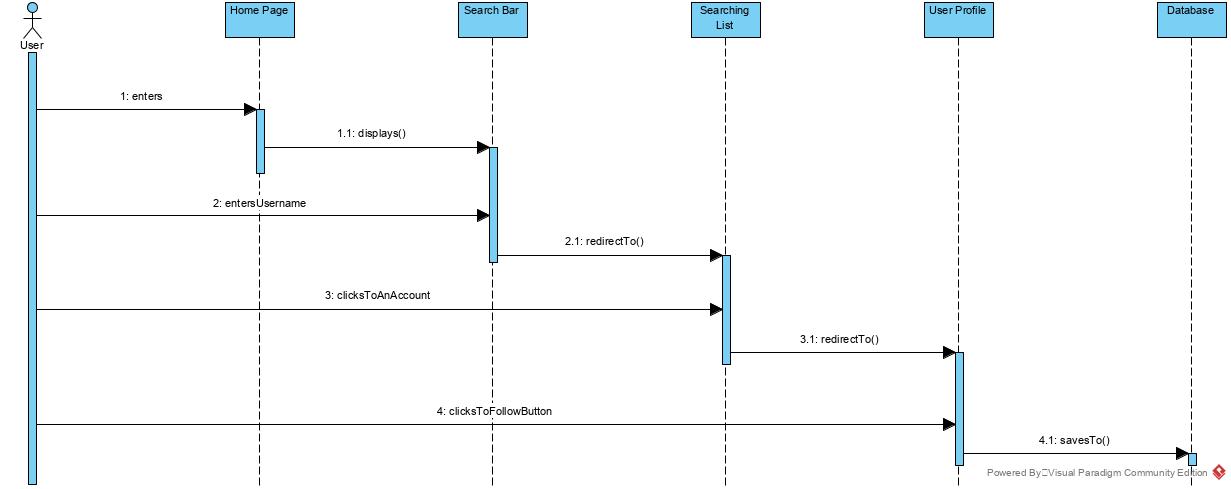
**SEARCH:**

****

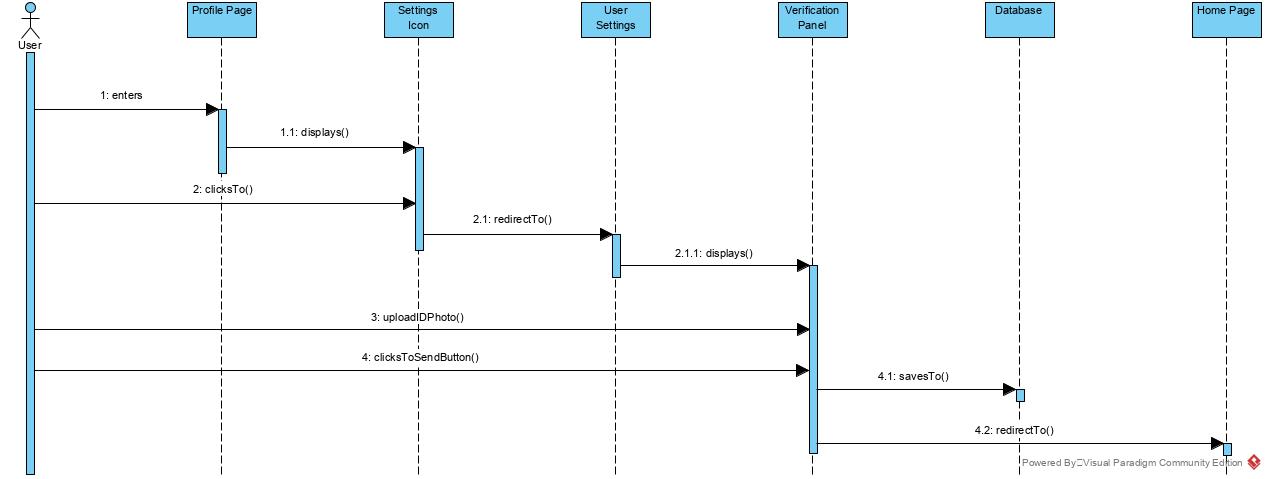
**DIRECT MESSAGE:**

****

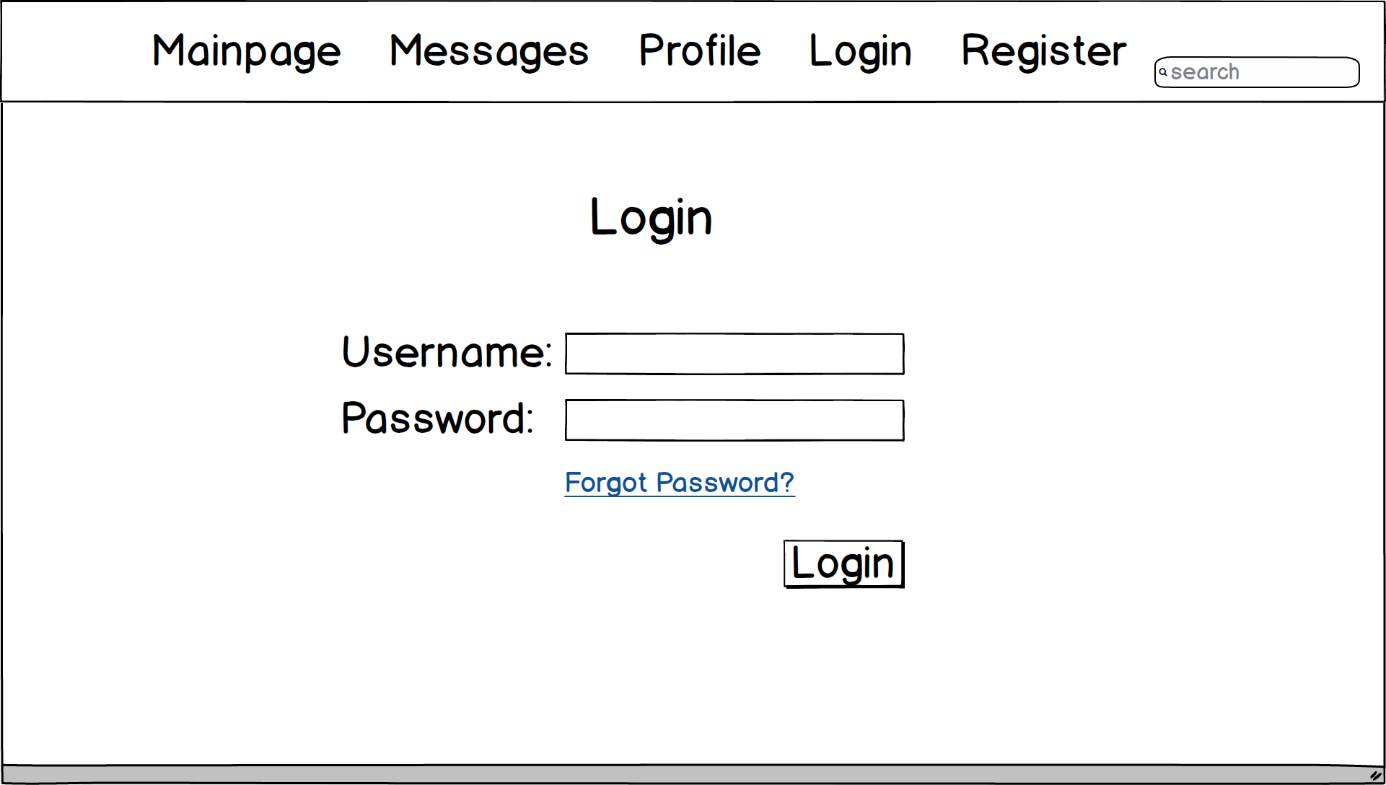
**FOLLOW USER:**

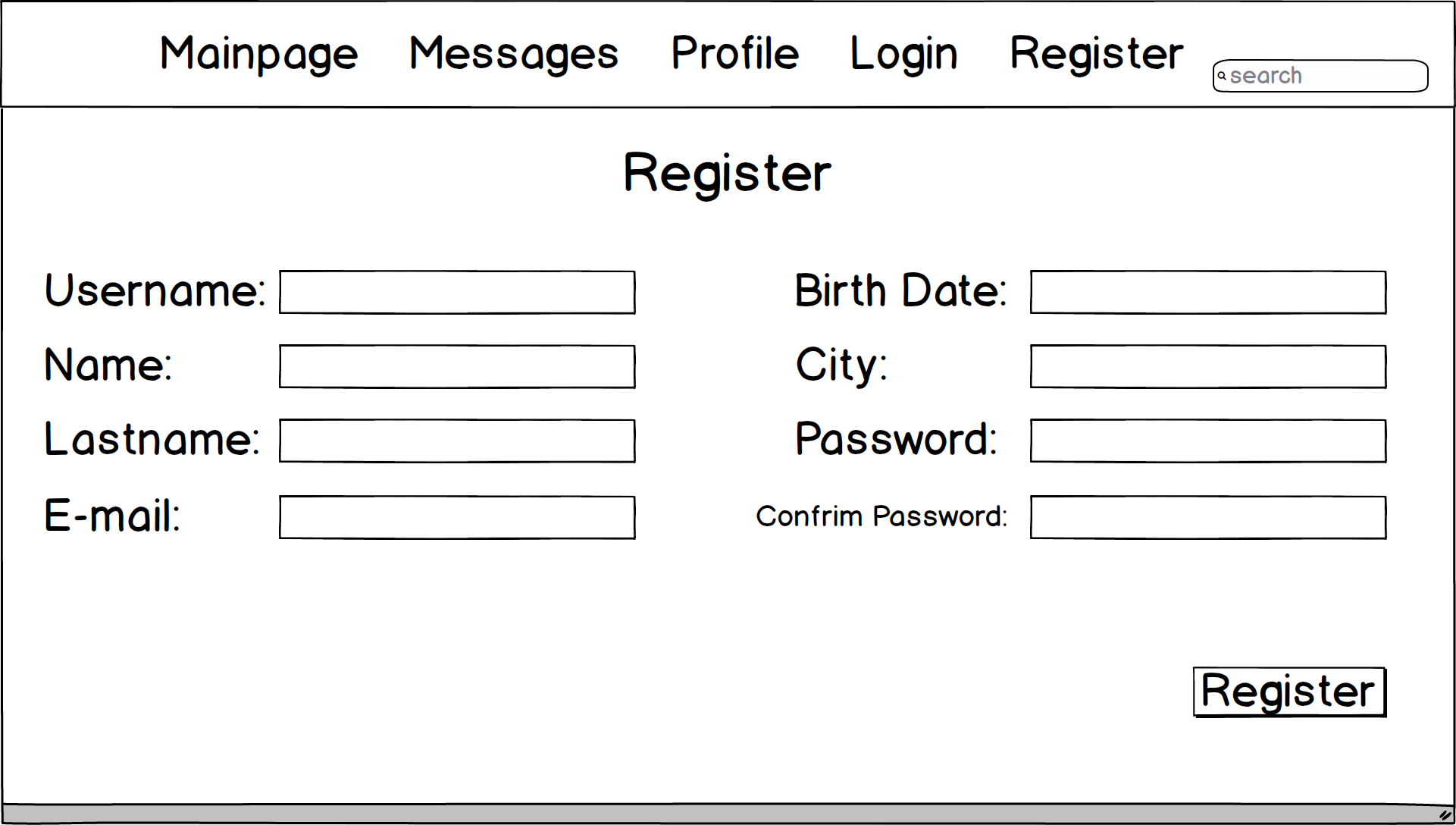
****

**VERIFICATION REQUEST:**

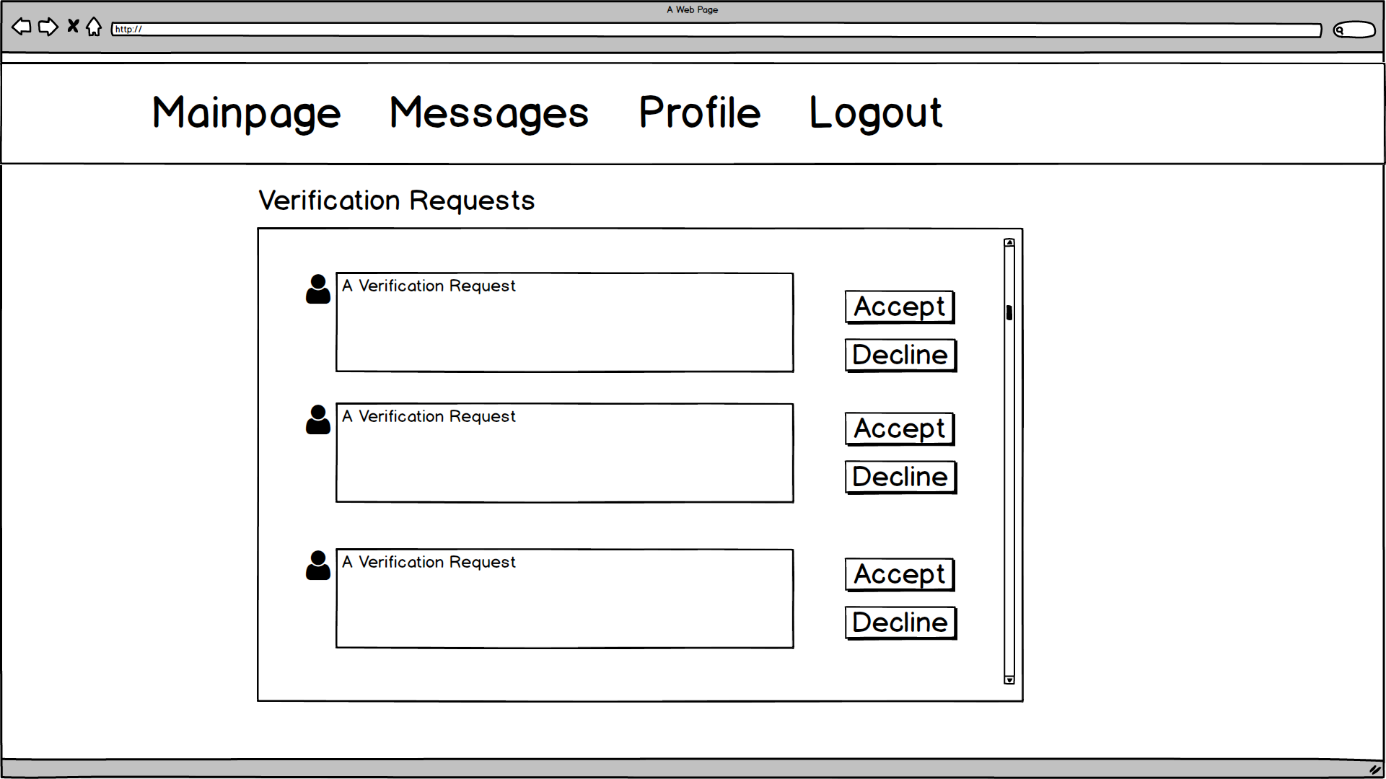
****

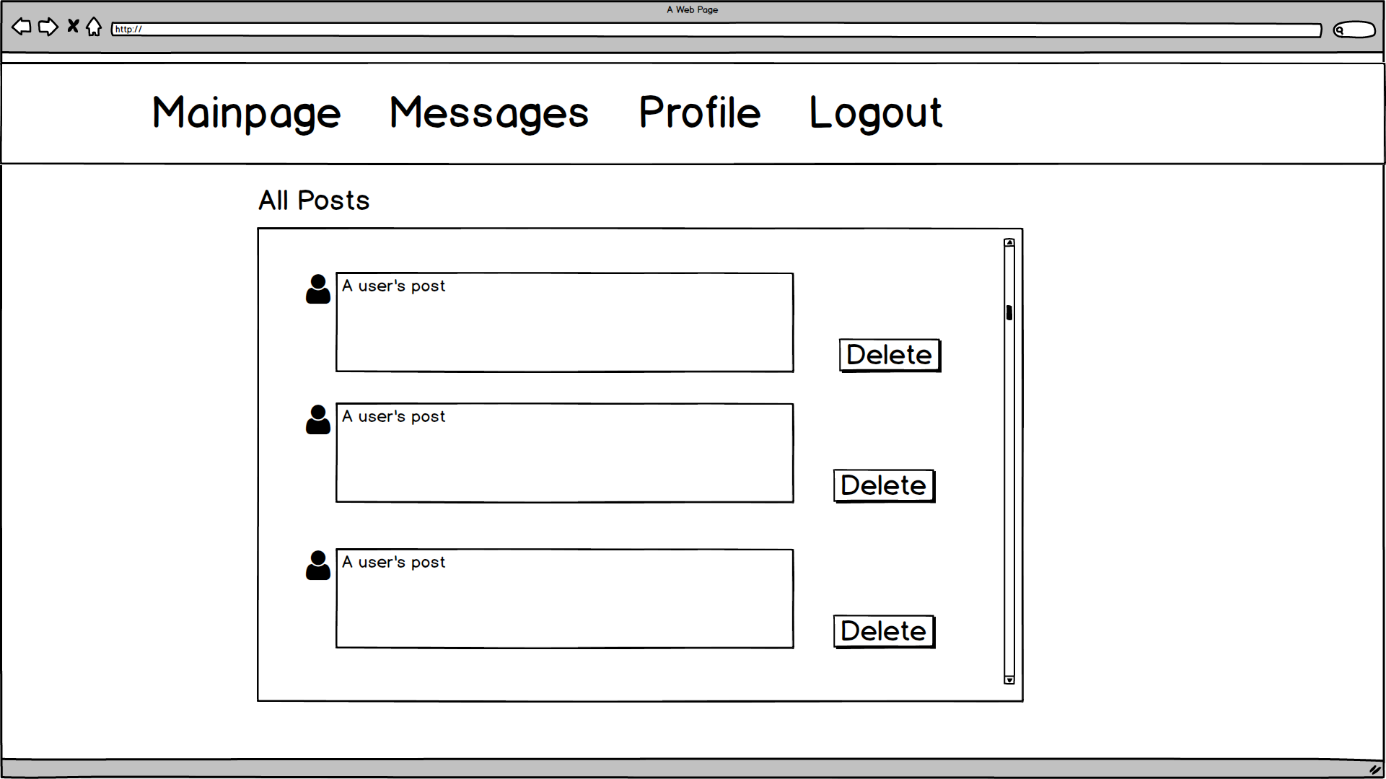
* + 1. **User interface—navigational paths and screen mock-ups**

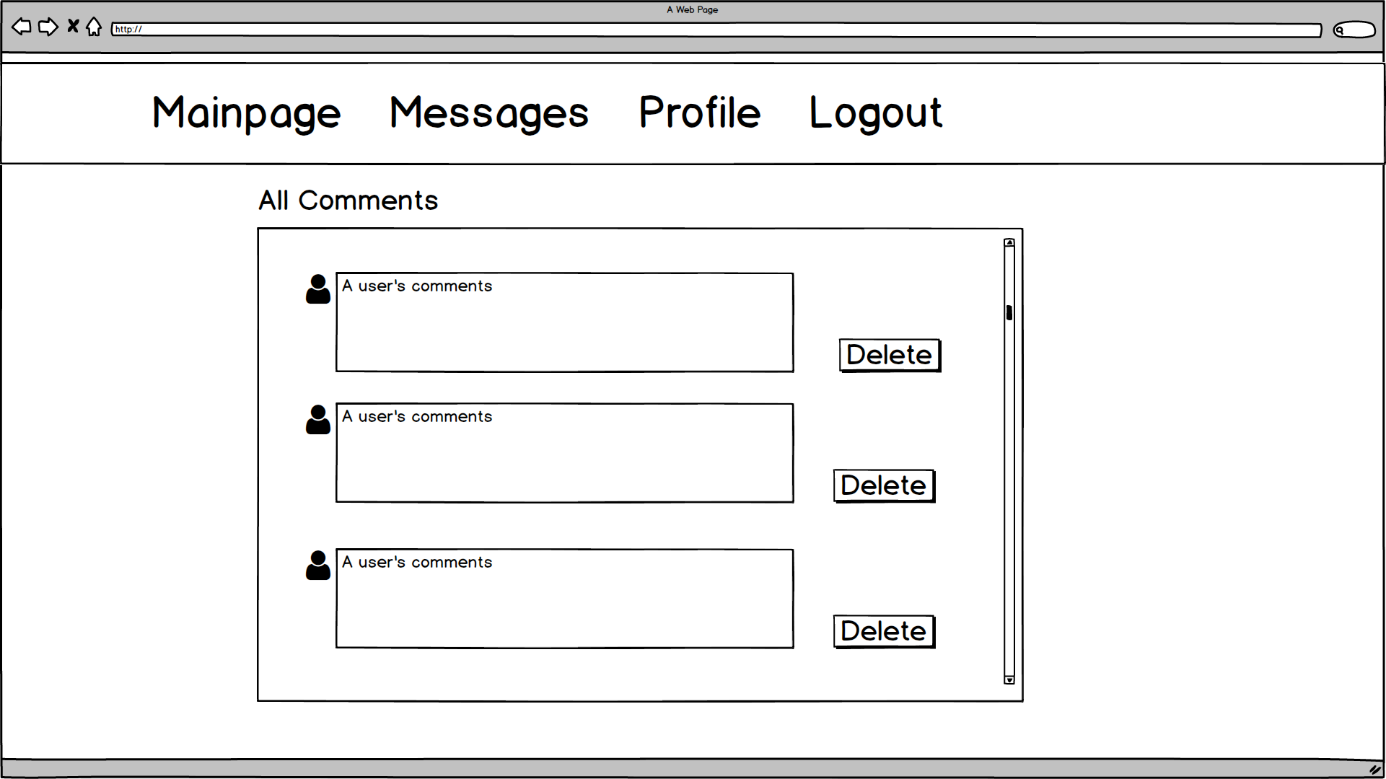
****

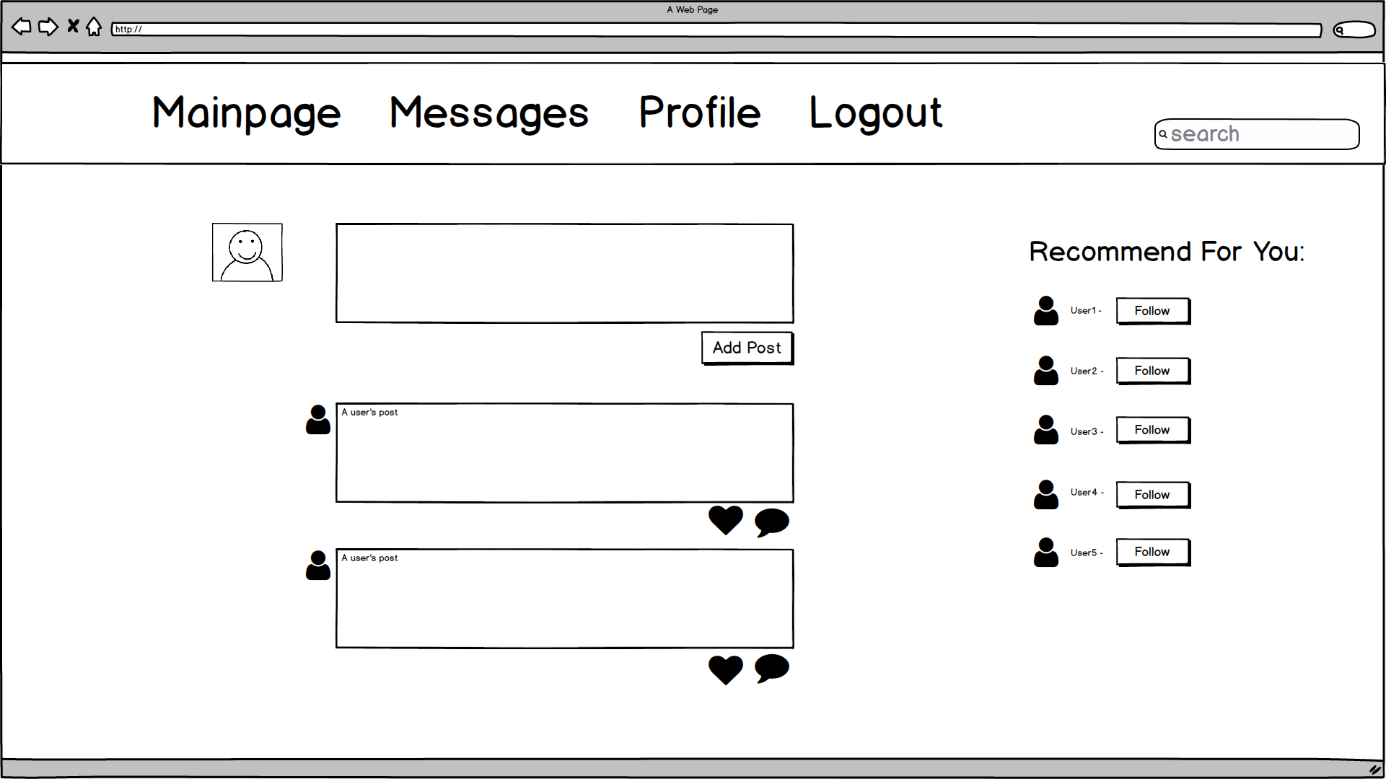
****

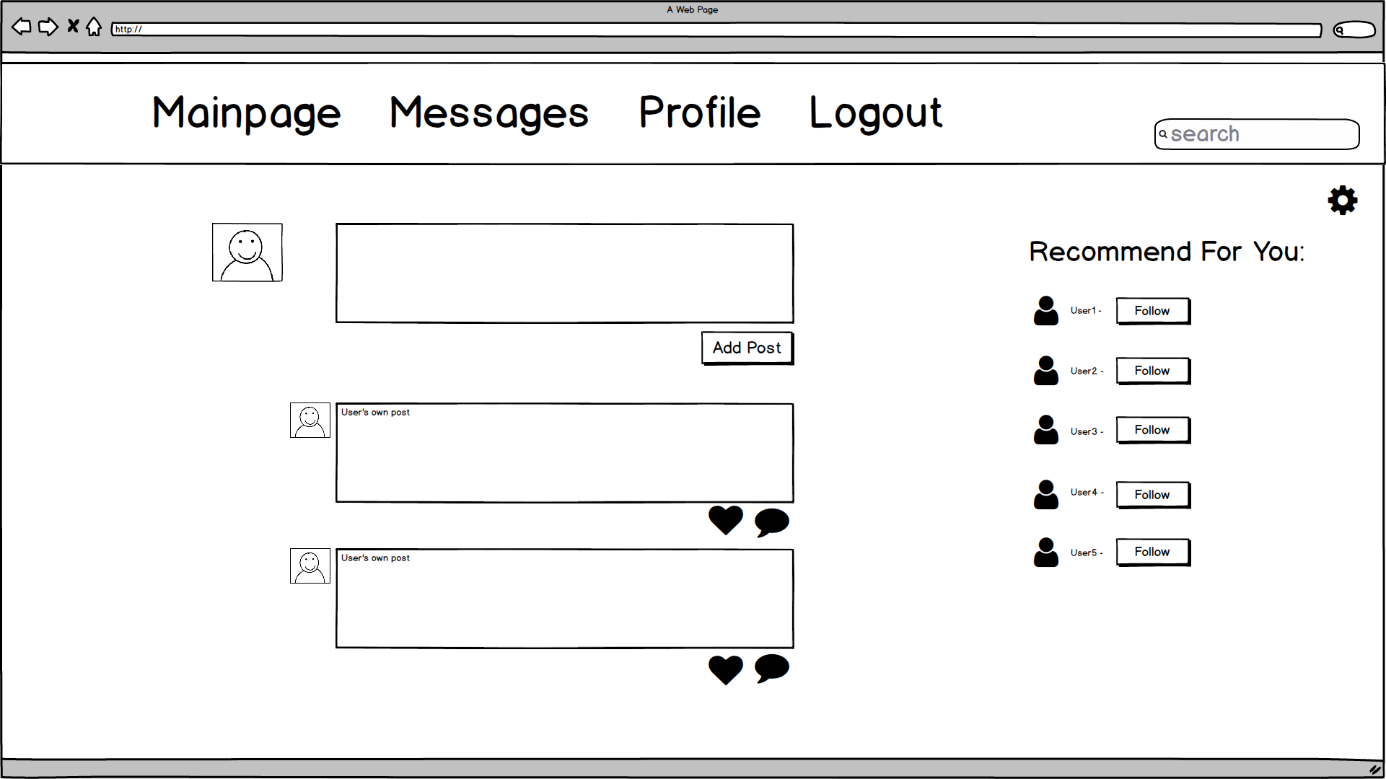
****

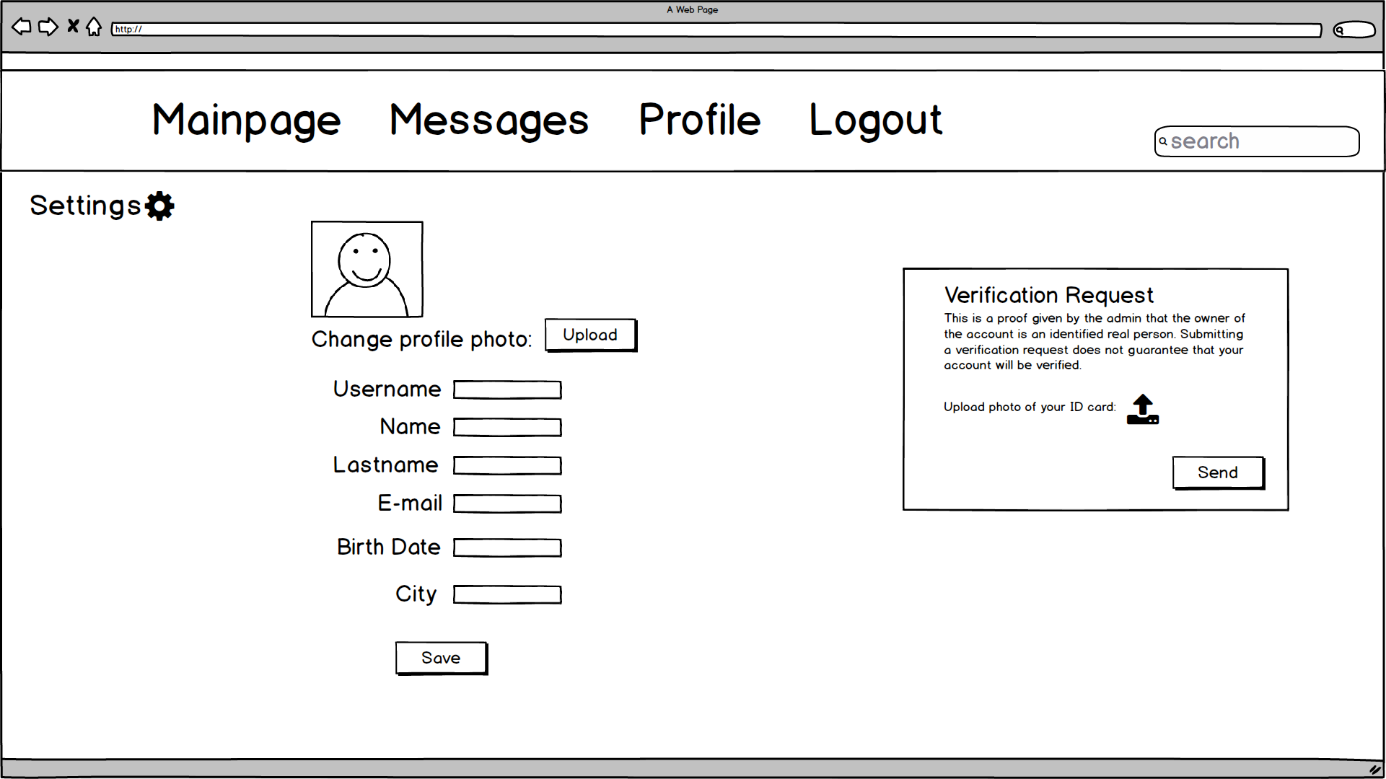
****

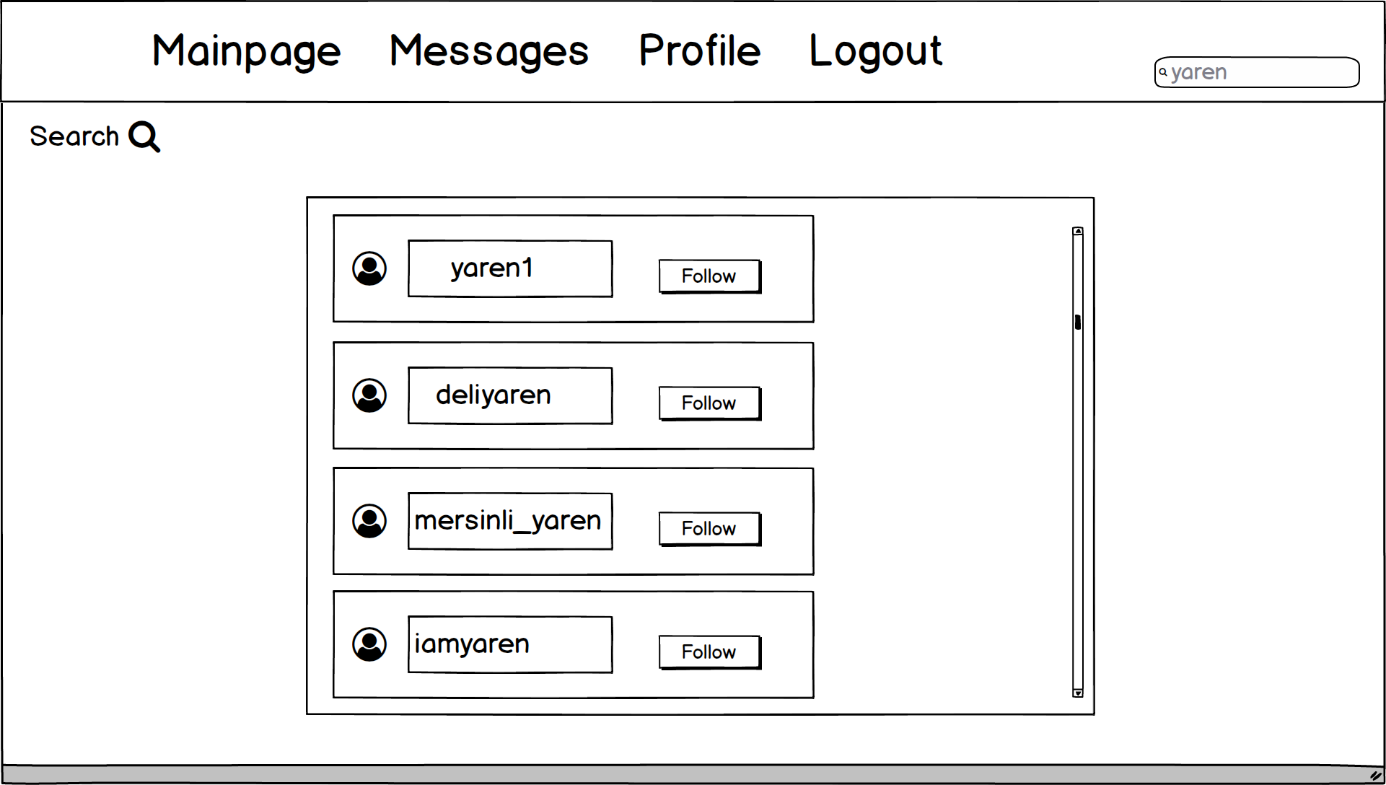
****

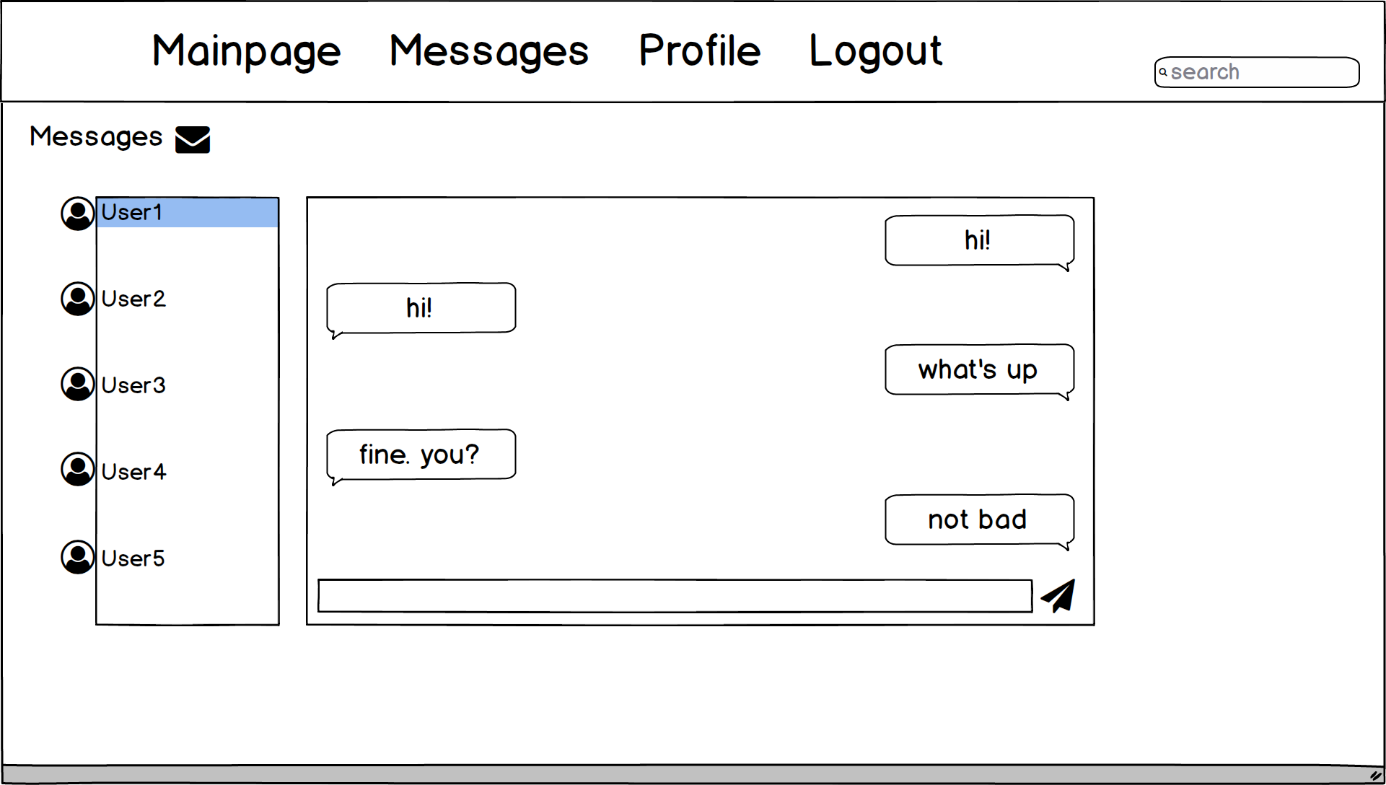
****

****

****

****

****

****

1. **Glossary**

To establish a clear terminology, developers identify the **participating objects** for each use case. Developers should identify, name, and describe them unambiguously and collate them into a glossary.

1. **References**

This subsection should:

* Provide a complete list of all documents referenced elsewhere in the RAD, or in a separate, specified document.
* Identify each document by title, report number - if applicable - date, and publishing organization.
* Specify the sources from which the references can be obtained.

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

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