

Table of Contents

I. Overview.....	3
II. Requirements:	3
1. Functional Requirements:	3
a. Manage User Account:	3
b. Send Messages:	3
c. Voice and video calling	4
d. Create & Manage Group	4
2. Nonfunctional Requirements:	4
a. Performance	4
b. Availability:.....	4
c. Security	4
d. Usability:	4
e. Compatibility:	5
f. Scalability:.....	5
g. Maintainability:	5
III. Functional modeling:	6
1. Use-Case Diagram	6
2. Use-Case Description	7
Use-Case 01 “Sign up”	7
Use-Case 02 “Sign in”.....	7
Use-Case 03 “Update account information”	8
Use-Case 04 “Send message”	10
3. Activity diagram.....	11
a. Sign up	11
b. Sign in	12
c. Update account information	13
d. Send message:	14
IV. Structural modeling.....	15
1. Robustness Diagram	15
a. Sign up	15
b. Sign in	16
c. Update account information	17
d. Send message:	18

2. 1st draft of Class diagram	19
V. Behavioral modeling	20
1. Sequence diagram	20
a. Update account information	20
b. Send message:	21
2. Communication diagram.....	22
a. Update account information	22
b. Send message:	23
3. State machine diagram (User Message)	24

I. Overview:

Messaging and social media platforms allow users to communicate and interact with each other using text, voice, and video messages. These platforms typically have a user-friendly interface that enables users to easily create an account, add friends or contacts, and send and receive messages.

In addition to messaging features, social media platforms like Zalo also include other features such as news feeds, event notifications, group chat, photo and video sharing, among others. These features are designed to enhance user engagement, increase user retention, and drive revenue from advertising and premium features.

Overall, messaging and social media platforms have become an integral part of modern communication and social interaction, providing a convenient and accessible way for people to stay connected and informed with their friends, family, and colleagues.

II. Requirements:

1. Functional Requirements:

a. Manage User Account:

- Users are able to **create a new account** by providing basic information, such as name, email and phone number.
- Users are able to **log in to their account** with a username and password.
- Users are able to **delete their accounts** if they want.
- **Store** user accounts information, such as profile picture, account settings in a centralized database.
- Users are able to **reset password** if they forget their login information.
- **Provide** a security mechanism, such as two-factor authentication, to prevent unauthorized people access to user accounts.
- Users are able to **update** their profile information, such as name, profile picture, status message.
- Users are able to **update** their contact information, such as phone number and email address.
- Users are able to **view** other user's profile and send them friend requests.
- **Store** updated profile information in centralized database.
- **Provide** a privacy settings functionality that allows users to control who can see their profile information.

b. Send Messages:

- Users (Sender) are able to **send** text messages to other users (Receiver).

- Users are able to **send** multimedia messages, such as photos, videos, audio recordings.
- Users are able to **create and send** group messages to multiple users.

- **Store** sent and received messages in centralized database.
- **Notify** users of new messages through push notifications.
- **Provide** a message search functionality for users to find old messages easily

c. Voice and video calling

- User can **make** a voice and video calls with other users on the platform.
- User can **make** a voice and video calls in a group.
- User can **switch** between audio and video modes.
- User are able to **set** such as volume and camera orientation.
- User are able to **hang up** the call.

a. Create & Manage Group:

- Users can **create** new groups and invite other users to join.
- **Store** group information, such as group name, member list in a centralized database.
- **Notify** users of new group activity, such as new messages or member addition/removal.
- **Provide** a group management functionality that allows group admins to add or remove member and control group settings.

2. Nonfunctional Requirements:

a. Performance:

- The system must respond to user requests within 1 second.
- The system must be able to handle a large number of users simultaneously without significant performance degradation.

b. Availability:

- The system must be available 24/7 with at least 99% uptime.
- The system must have a failover mechanism in place to ensure continuous service in case of system failure.

c. Security:

- The system must use strong encryption to protect user data.
- The system must have mechanisms in place to prevent unauthorized people access to user accounts.
- The system must implement proper authentication and authorization mechanisms.

a. Usability:

- The system must be easy to use and navigate for all users, regardless of technical expertise.
- The system must have clear and concise error messages to guide users when they encounter errors.
- The system must have a intuitive user interface.

e. Compatibility:

- The system must be compatible with a variety of devices and platforms, including mobile devices and various operating systems.
- The system must be able to integrate with other systems, such as payment gateways or third-party APIs.

f. Scalability:

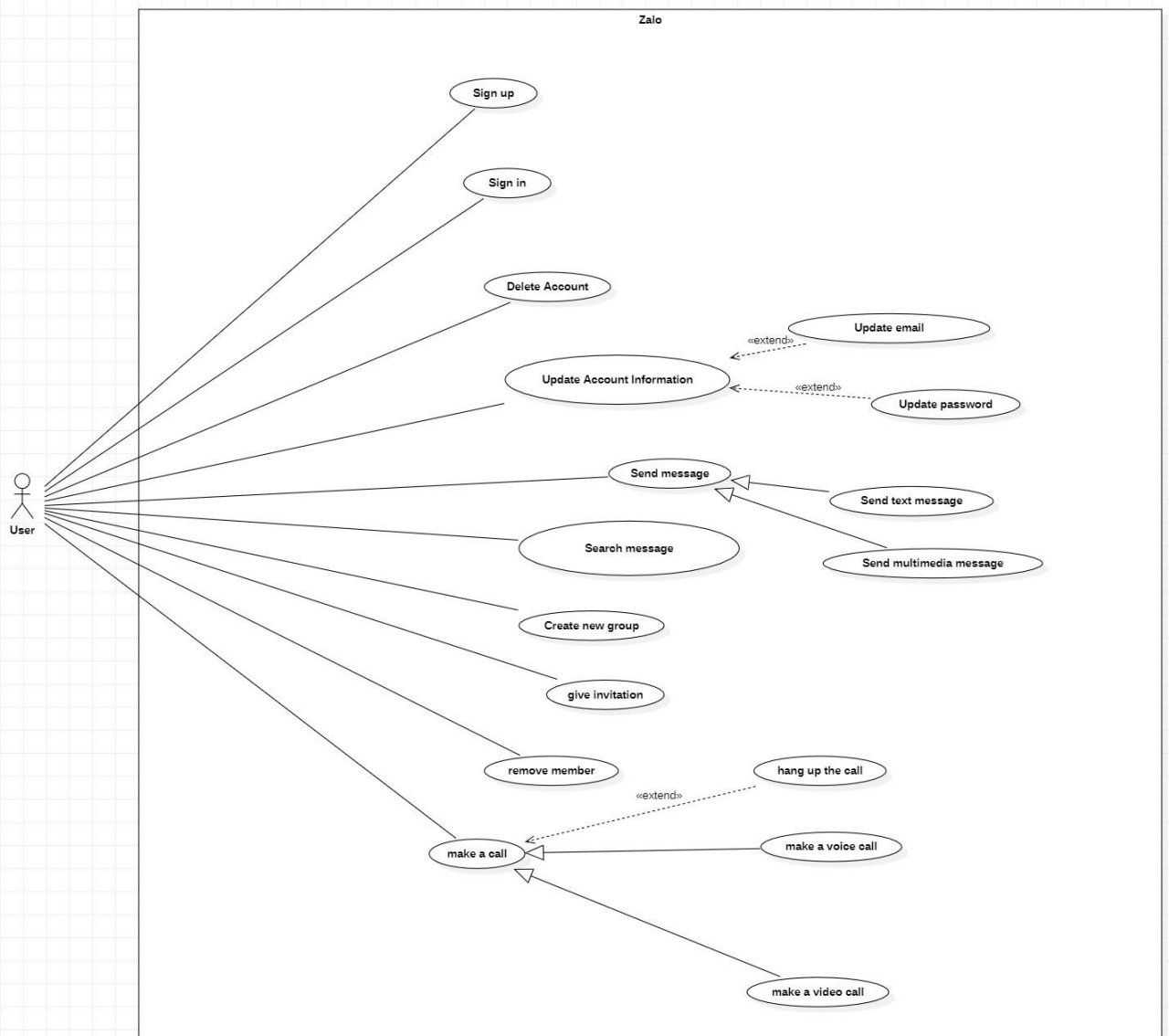
- The system must be able to handle a large number of users and data growth over time.
- The system architecture must be designed in a way that allows for easy scalability.

g. Maintainability:

- The system must be easy to maintain and update.
- The system must have clear and concise documentation for developers and administrators.

III. Functional modeling:

1. Use-Case Diagram:

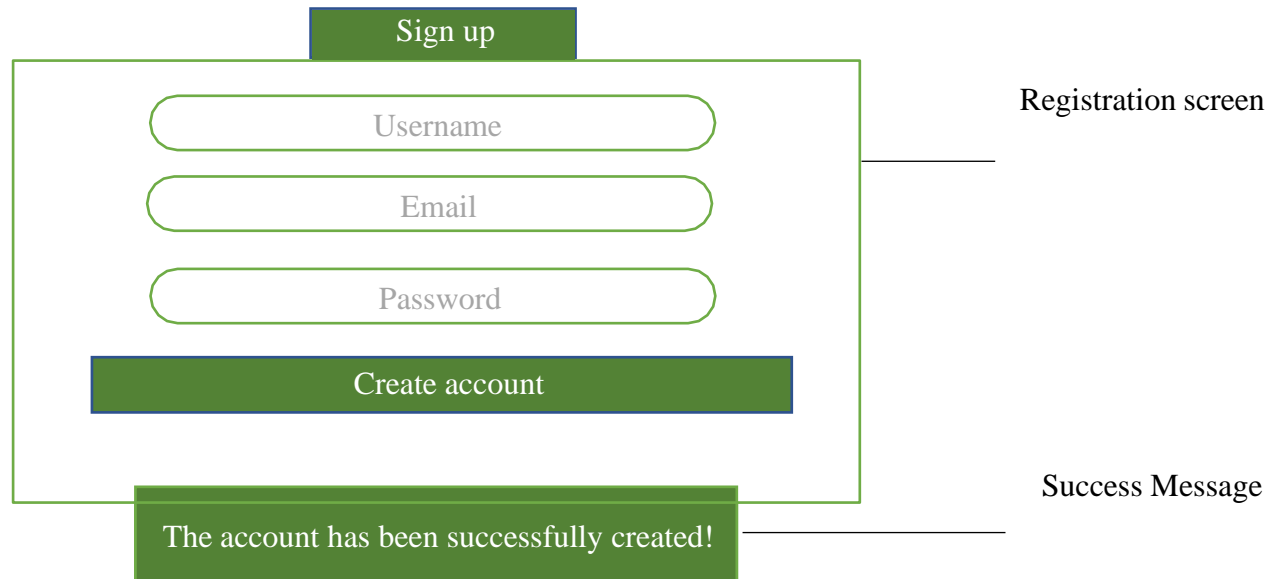


2. Use-Case Description:

Use-Case 01 "Sign up"

- Short description: The user creates a new account on the system by providing basic information. The information is stored in a centralized database.

- Visualization by storyboards:



- Basic scenarios:

- User clicks on the "Sign up" button on the system's homepage.
- System shows a Registration screen.
- User fills in the required information: Username isn't the same as an existing username in the system, Email isn't the same as an existing email in the system and Password isn't too short.
- User clicks on the "Create account" button to complete the registration process.
- The system validates the provided information and creates a new account for the user.
- The system stores account information in a centralized database.
- The system displays a Success Message and redirects the guest to the login page.

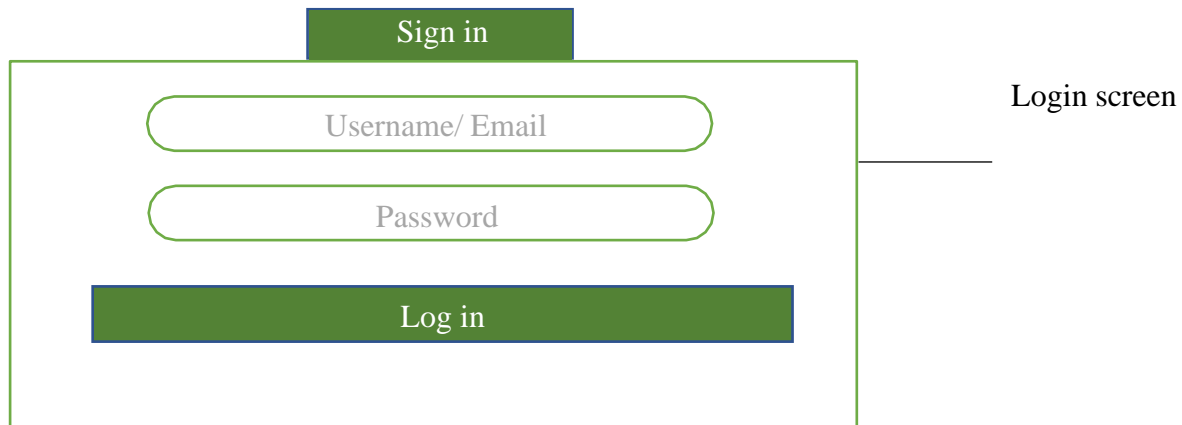
- Alternate scenarios:

- The user enters a username that already exists in the system: The system rejects the registration and responds with a message explaining why the registration was rejected.
- The user enters a email that already exists in the system: The system rejects the registration and responds with a message explaining why the registration was rejected.
- The user enters a password that is too short (<5 characters): The system rejects the registration and responds with a message explaining why the registration was rejected.

Use-Case 02 “Sign in”

- Short description: The user wants to sign in to access their accounts.

- Visualization by storyboards:



- Basic scenarios:

- The user clicks on the "Sign in" button on the system's homepage.
- The system displays a Login screen for the guest to enter their email and password.
- The user enters their account (username or email) and password, and then clicks on the "Log in" button.
- The system verifies the provided credentials and grants the user access to their account.
- The system displays the user's account dashboard.

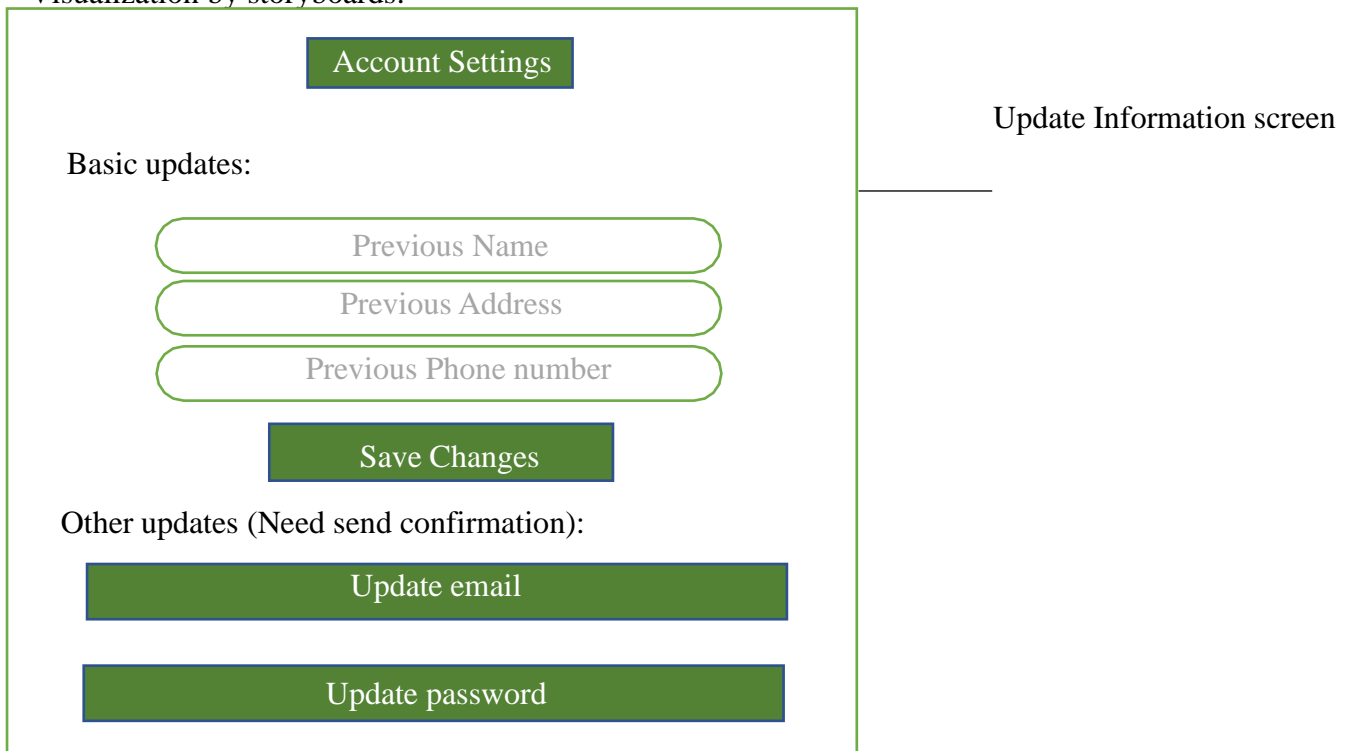
- Alternate scenarios:

- The user enters the incorrect username/ email and password: The system rejects the login and responds with a message explaining why the login was rejected.

Use-Case 03 “Update account information”

- Short description: The user updates their account information such as email, password. New information is stored in a centralized database.

- Visualization by storyboards:



The information has been successfully updated!

- Basic scenarios:

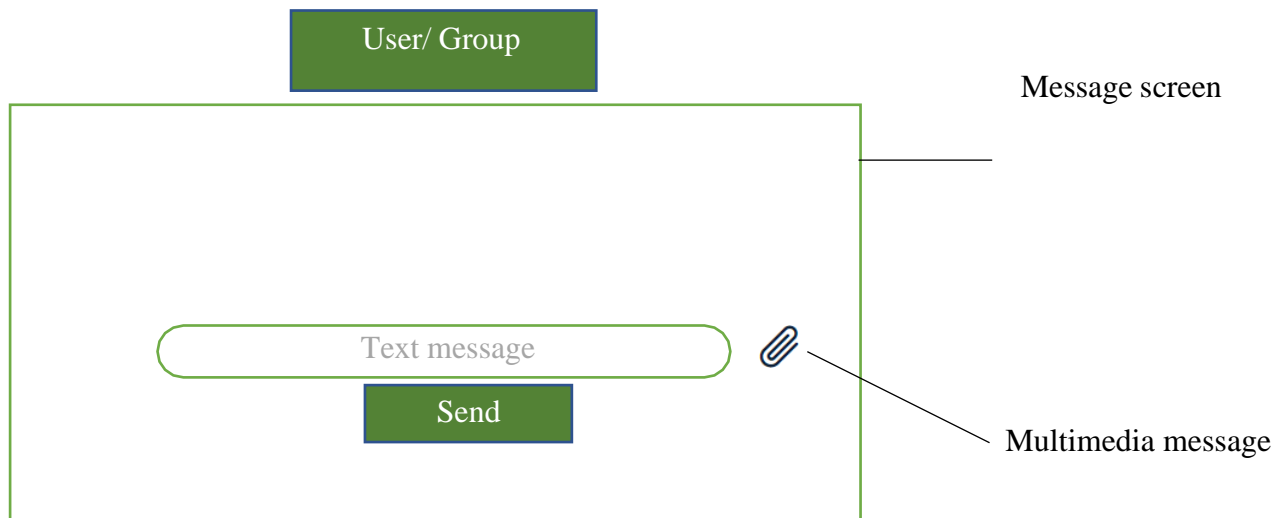
- The user clicks on the "Account Settings" button on the user's account dashboard.
- The system displays a Update Information screen (show Previous information retrieved from User Account) for the user to update basic information : Name isn't too long, Address and Phone Number is valid. Alternatively, the user can click on the "Update email" button or "Update password" button (which requires confirmation), and the system will redirects the user to another interface.
- The user changes the basic information and clicks on the "Save Changes" button to submit the form.
- The system verifies the provided information and stores them in User Account.
- The system displays the Confirmation Page.

- Alternate scenarios:

- User not logged in: The user is taken to the Login screen.
- The name is too long (>100 characters): The system rejects the update and responds with a message explaining why the update was rejected.
- The address is invalid (nonexistent address): The system rejects the update and responds with a message explaining why the update was rejected.
- The phone number is invalid (nonexistent phone number): The system rejects the update and responds with a message explaining why the update was rejected.

Use-Case 04 “Send message”

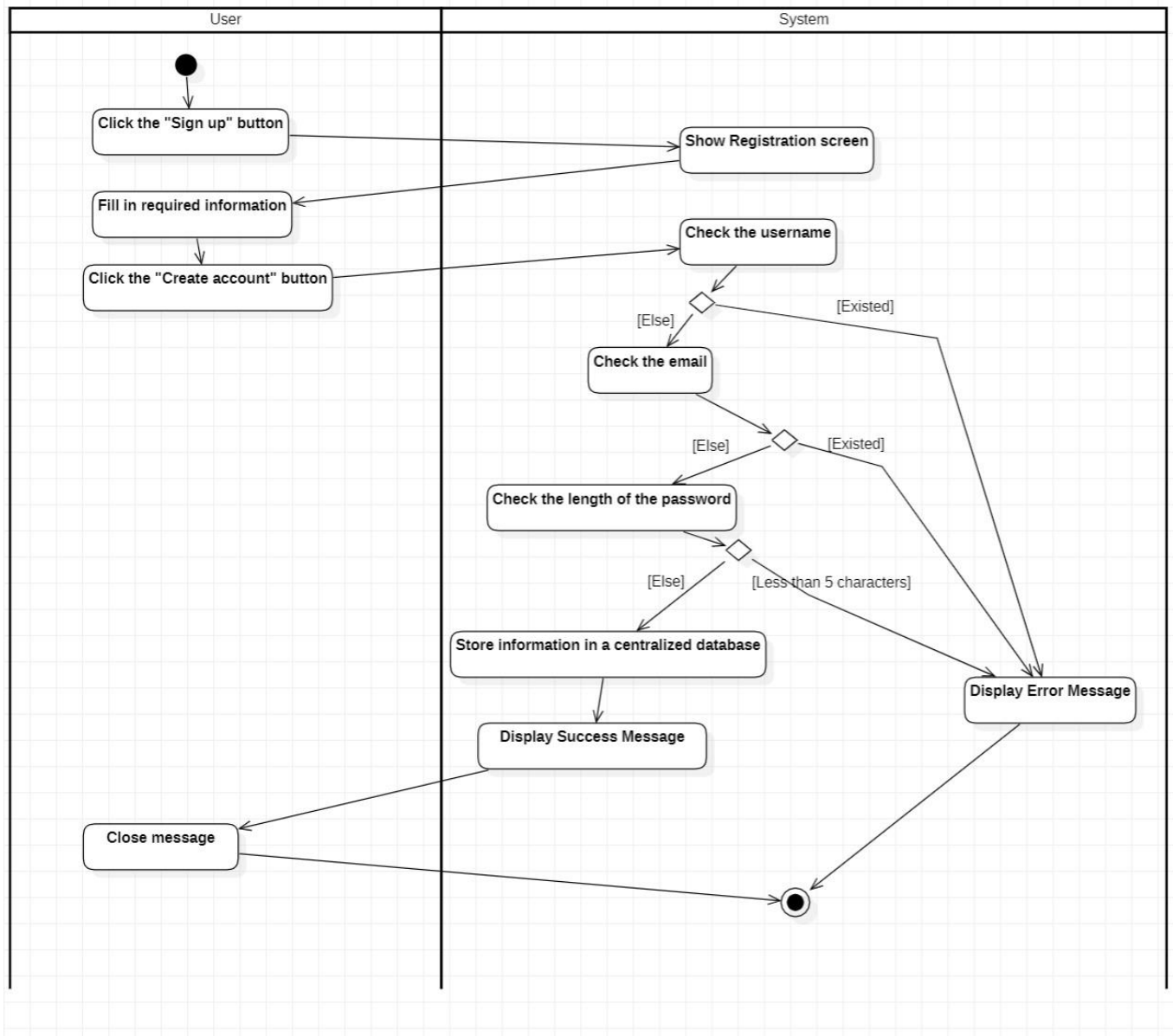
- Short description: The user send message on the messaging platform
- Visualization by storyboards:

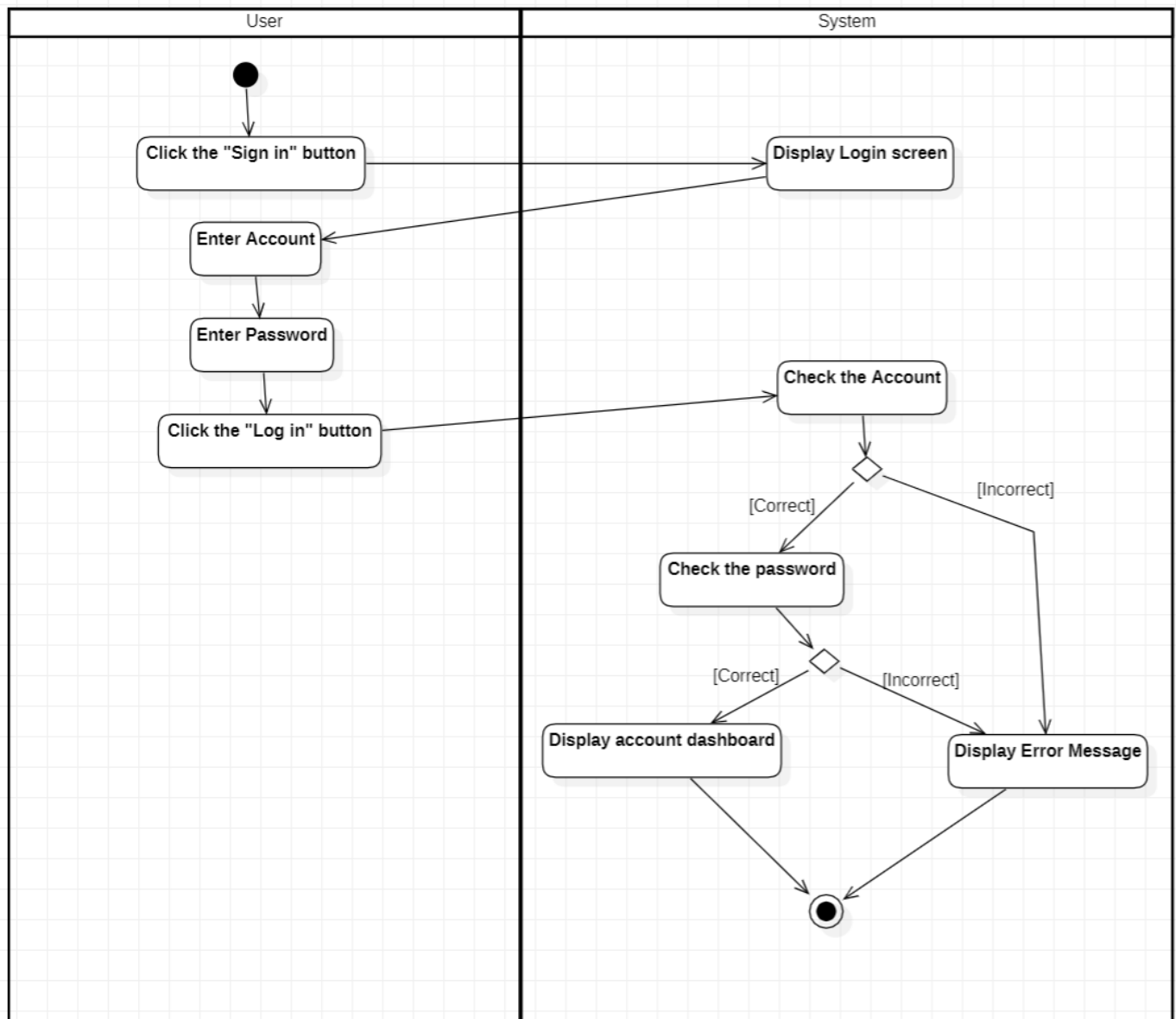


- Basic scenarios:
 - On the user's account dashboard, the user selects a receiver: another user or a group.
 - The system displays a Message screen.
 - The user enters the message content: text message isn't empty or long, multimedia message isn't too big.
 - The user clicks the "Send" button to send the message.
 - The system creates User's message and sends the message to the receiver.
- Alternate scenarios:
 - User not logged in: The user is taken to the Login screen.
 - The text message is empty: The system rejects sending and responds with a message explaining why sending was rejected.
 - The text message is too long (>500 characters): The system rejects sending and responds with a message explaining why sending was rejected.
 - The multimedia message is too big (>500 MB): The system rejects sending and responds with a message explaining why sending was rejected.

3. Activity diagram:

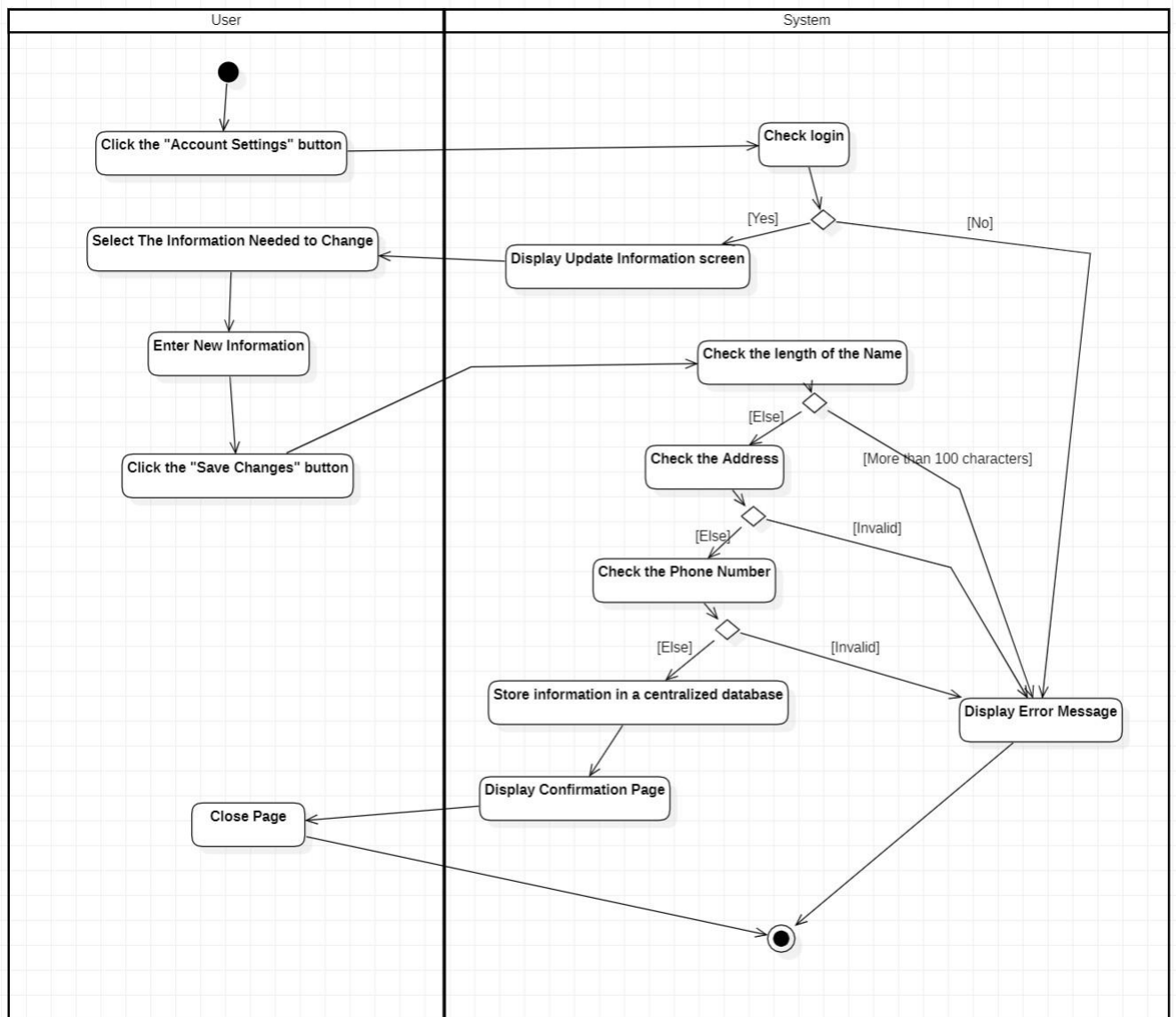
a. Sign up:



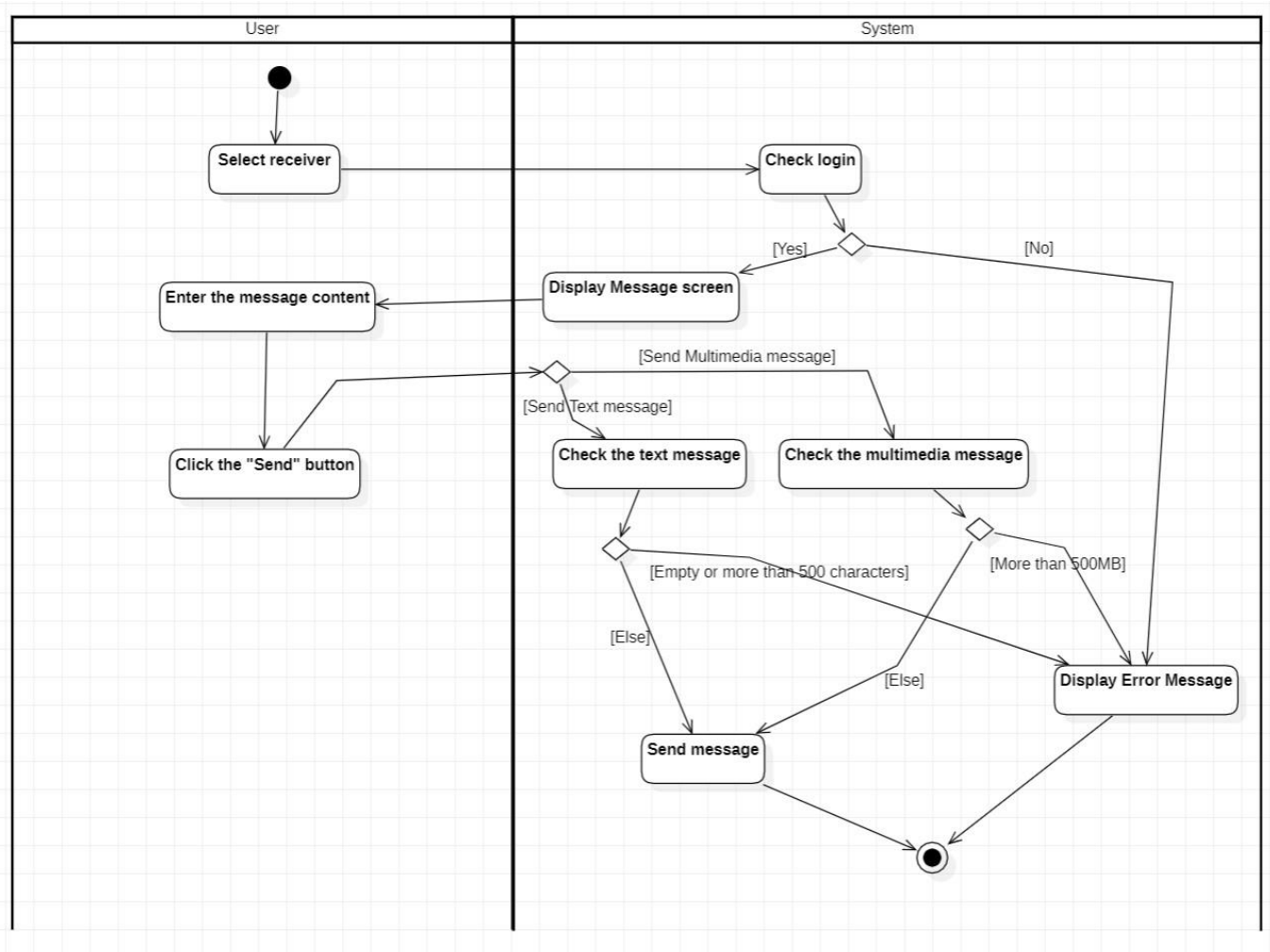


b. Sign in:

c. Update account information:



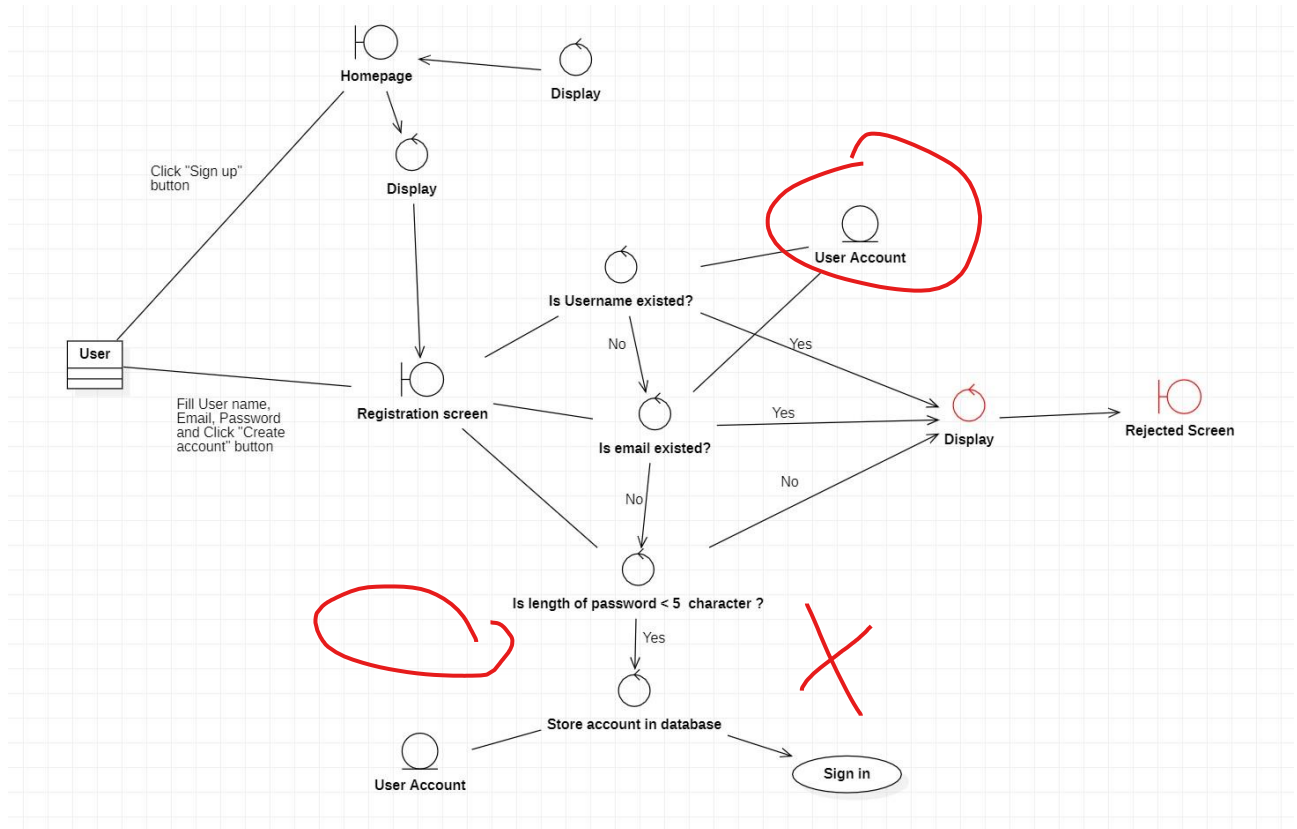
a. Send message:



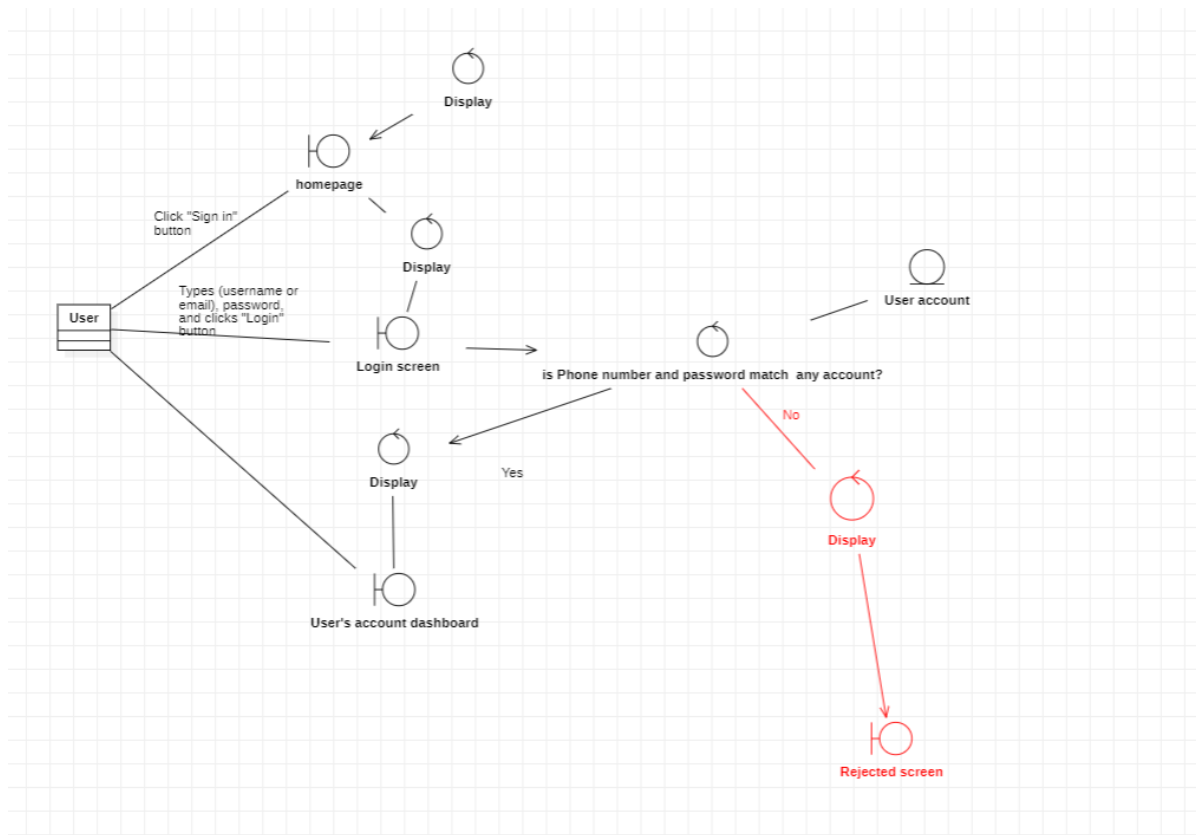
iv. Structural modeling:

1. Robustness Diagram:

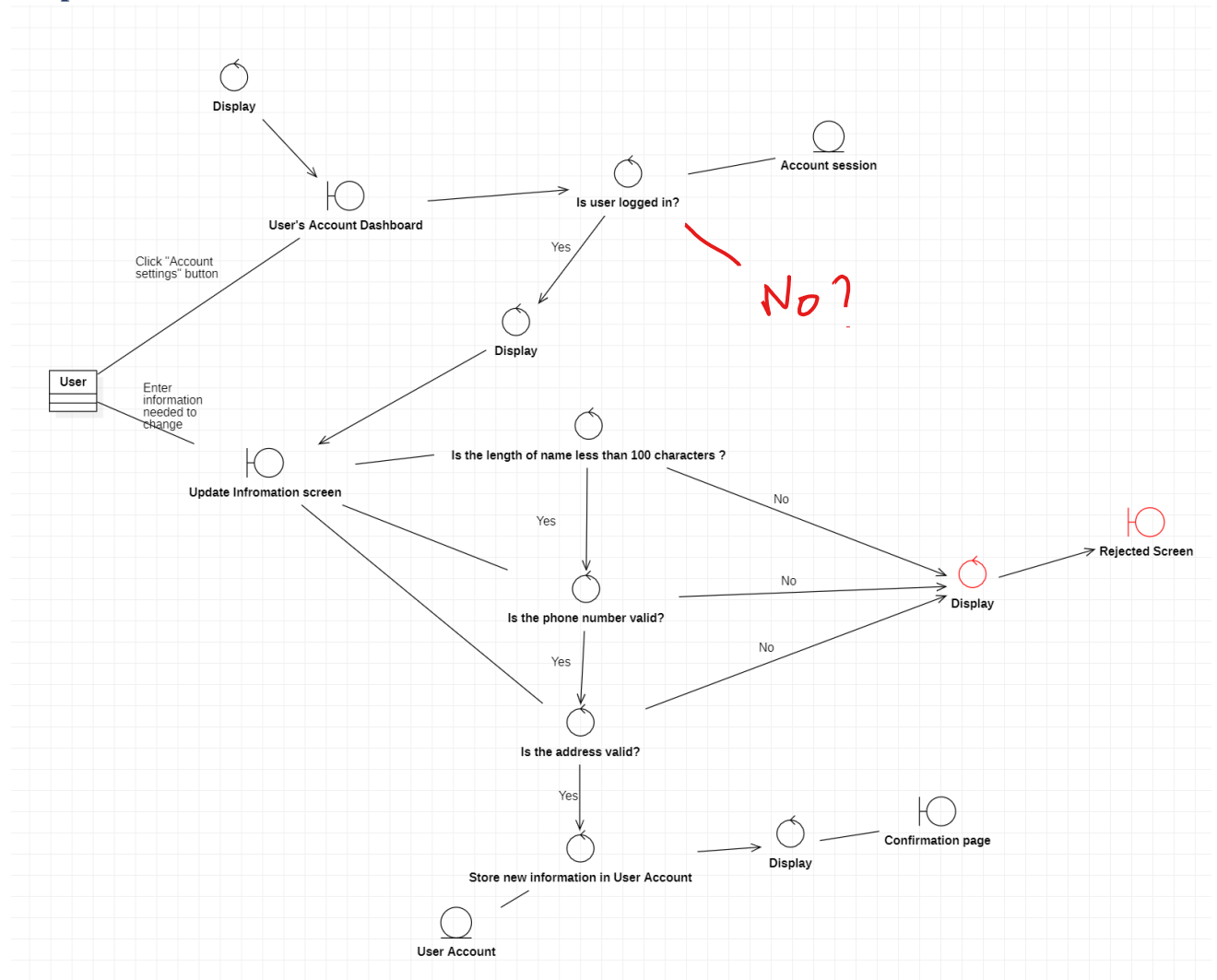
a. Sign up:

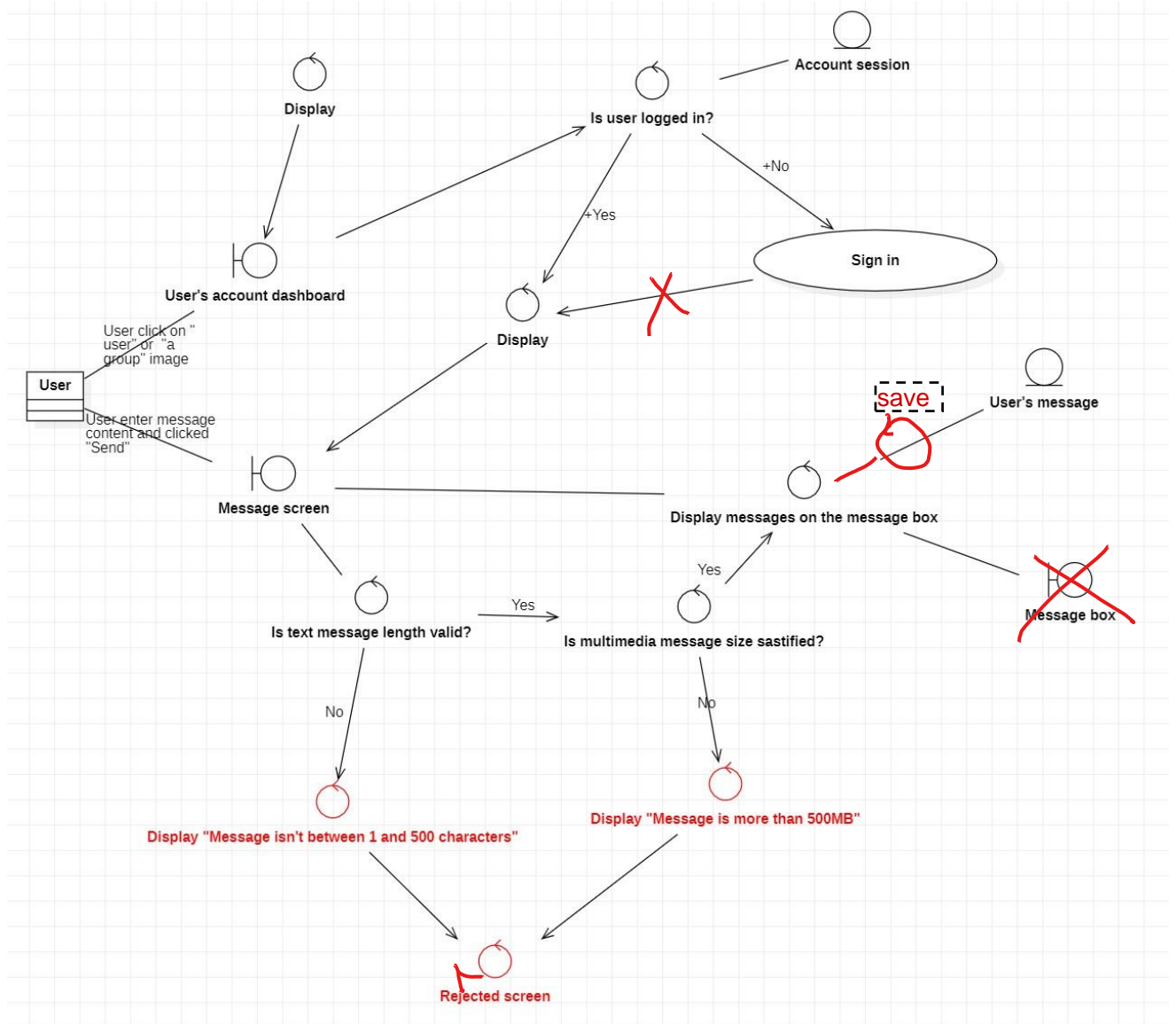


b. Sign in:



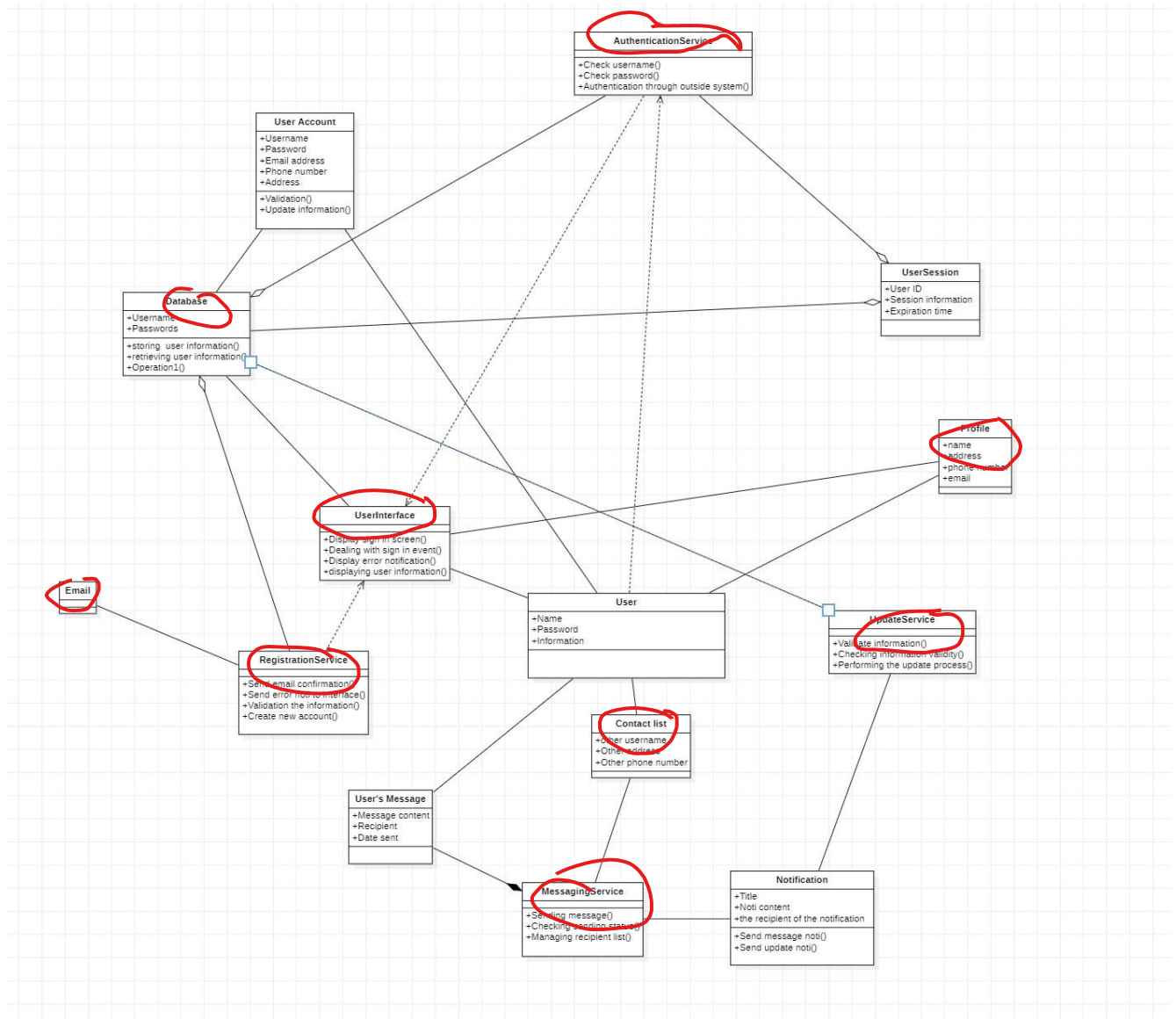
c. Update account information:





a. Send message:

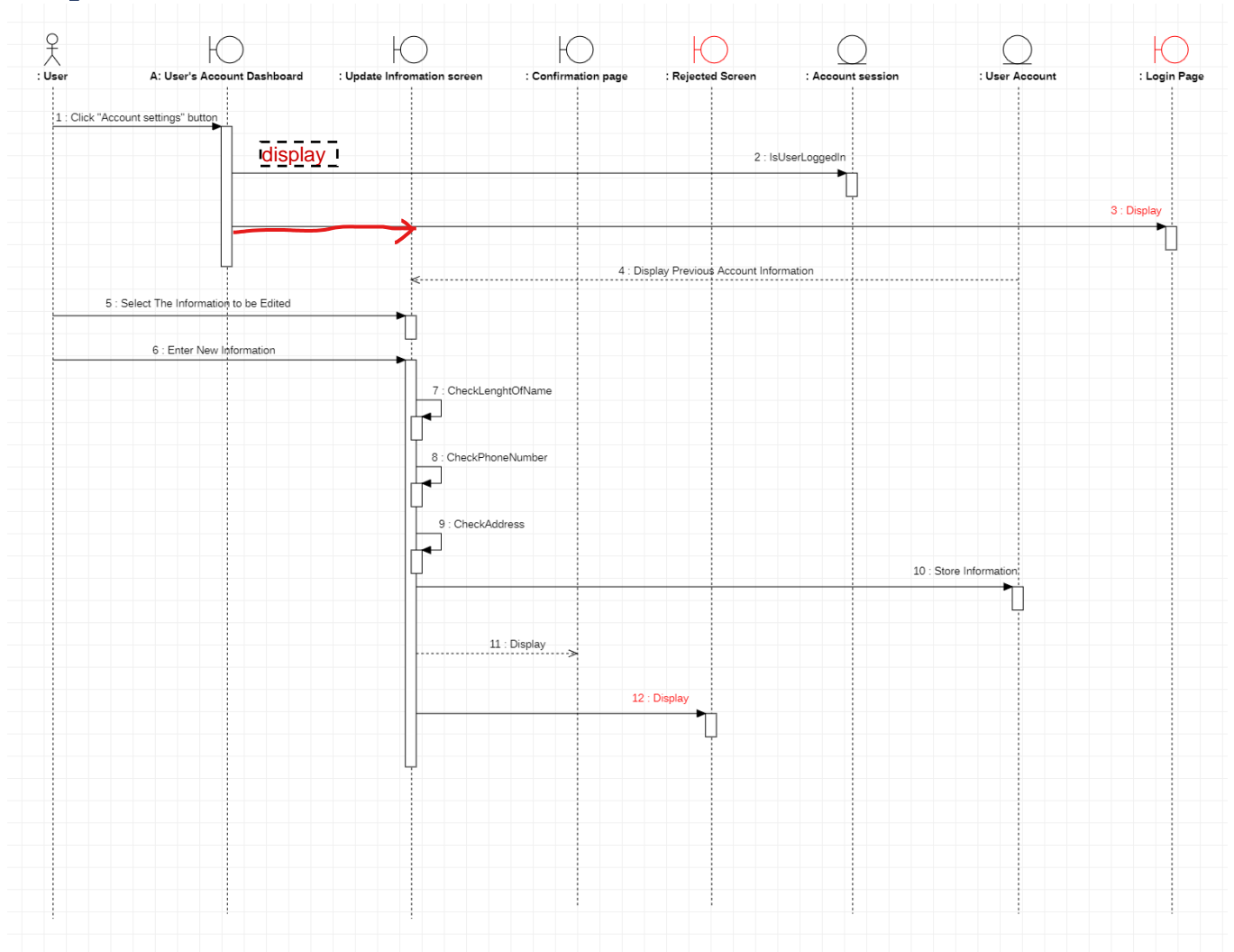
2. 1st draft of Class diagram:



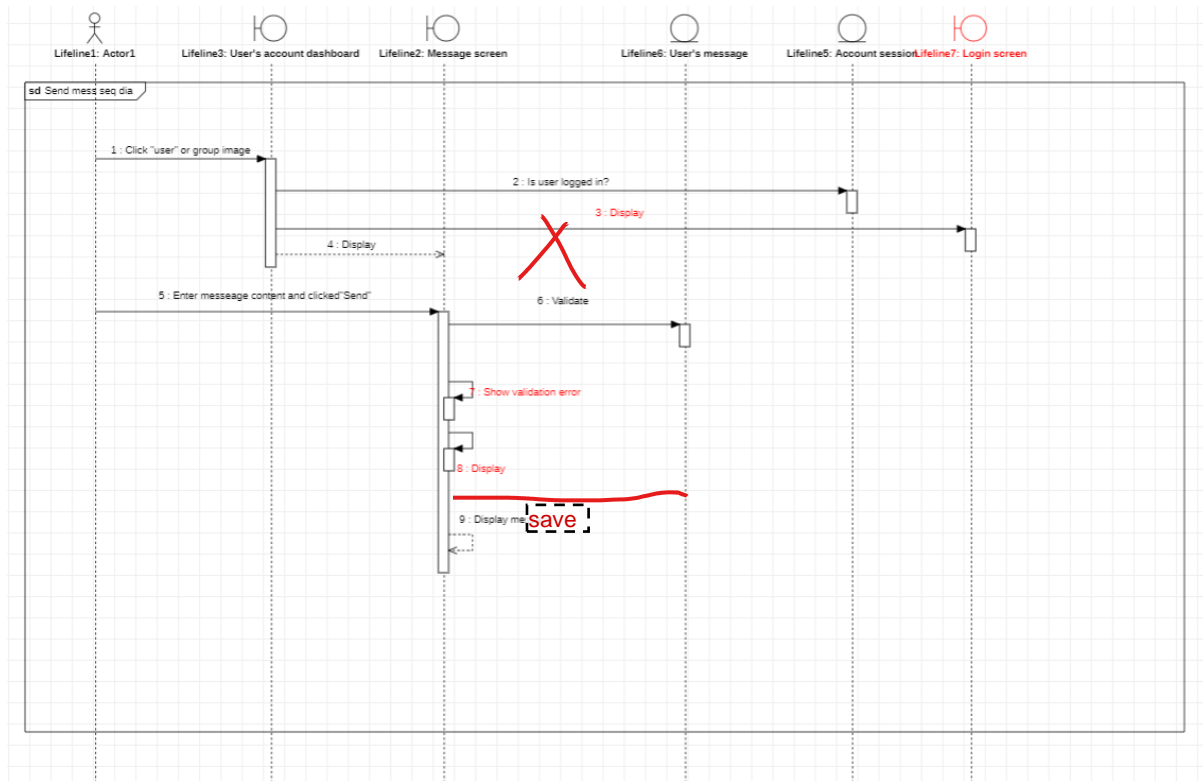
v. Behavioral modeling:

1. Sequence diagram:

a. Update account information:

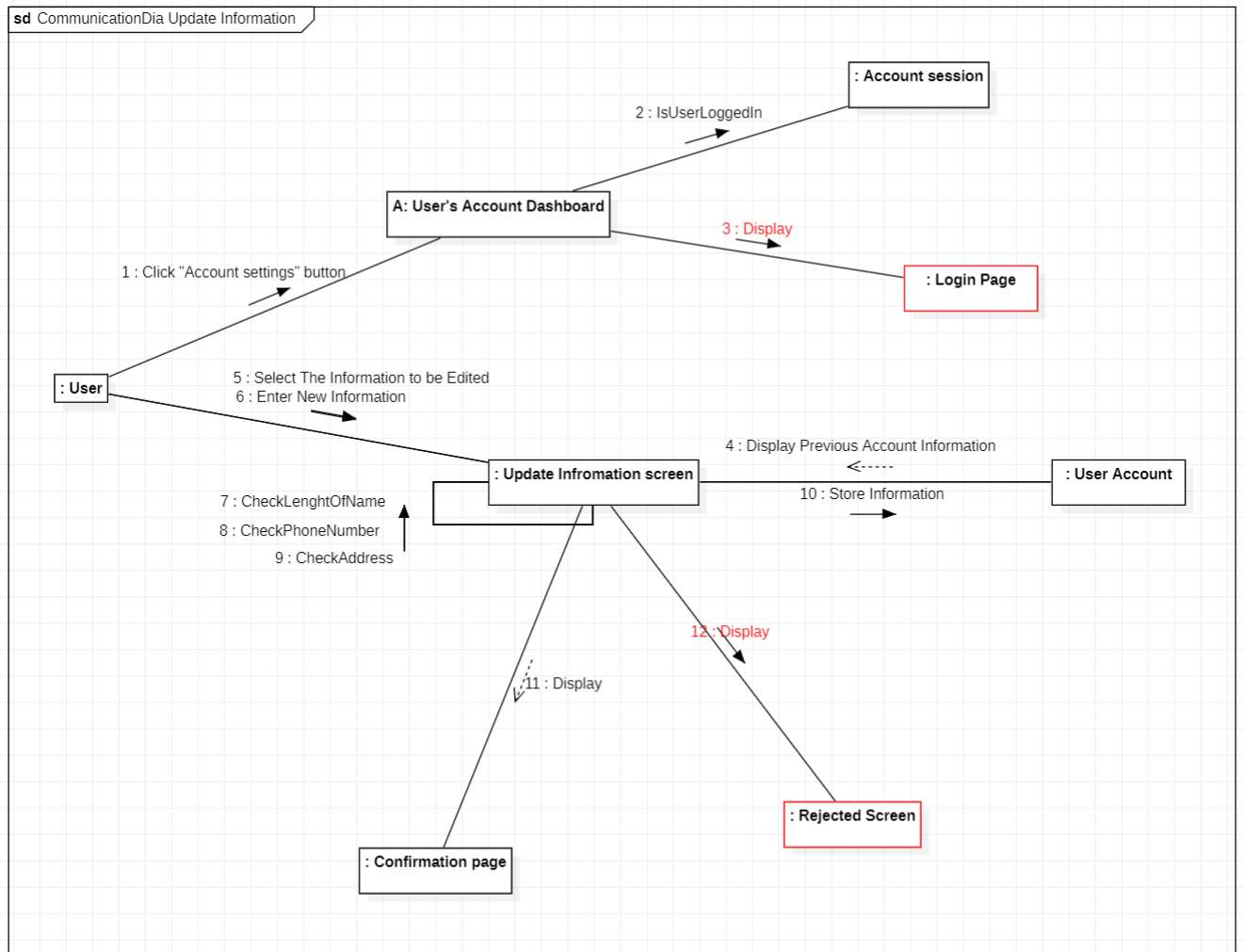


b. Send message:

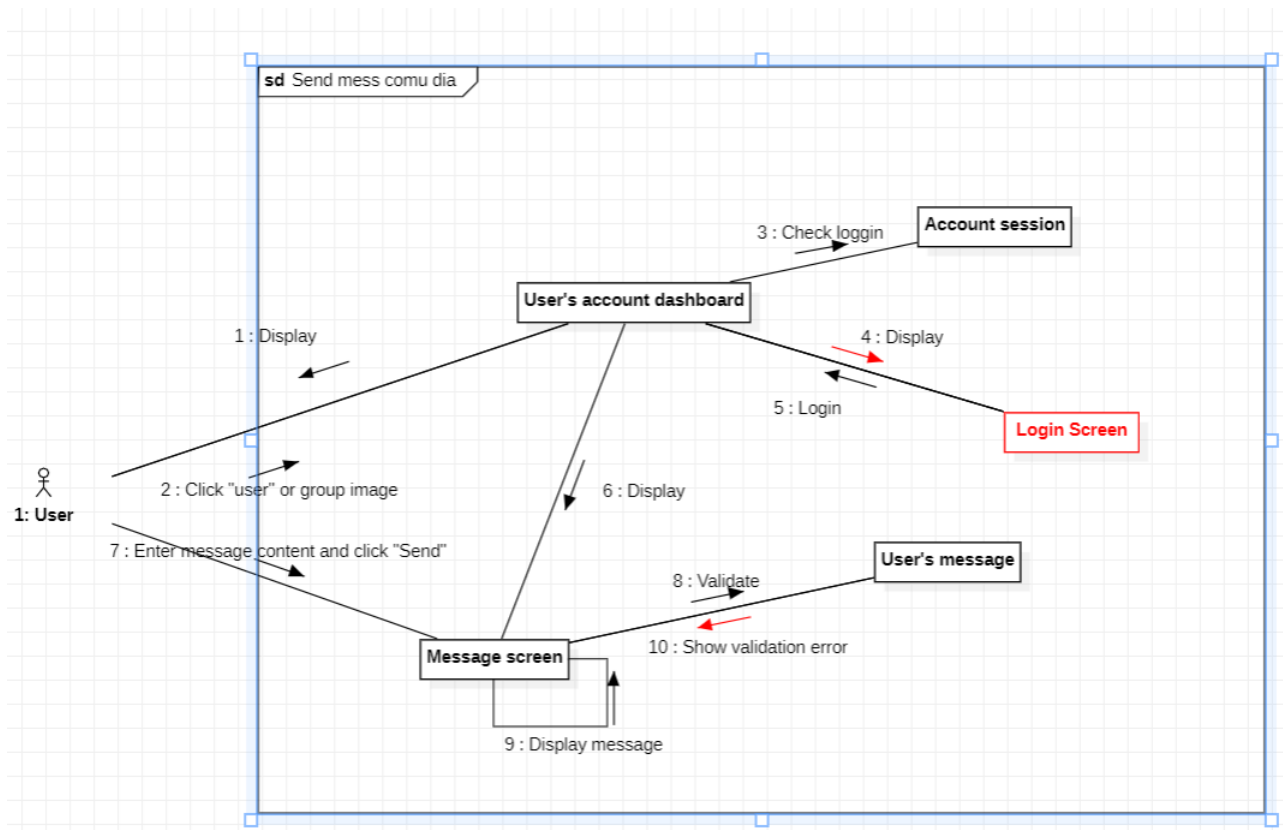


2. Communication diagram:

a. Update account information:



b. Send message:



3. State machine diagram (User Message):

