SnapTweet

Mini Twitter

Software Design Report

For CSC 32200

Version 1.0

Mini Twitter	Version: 1.0
Software Design Report	Date: 11/21/23
Document Identifier: 10	

Date	Version	Description	Author
11/21	1.0	Completed the software design report for MiniTwitter	Group H

Mini Twitter	Version: 1.0
Software Design Report	Date: 11/21/23
Document Identifier: 10	

Table of Contents

Software Design Report

Introduction
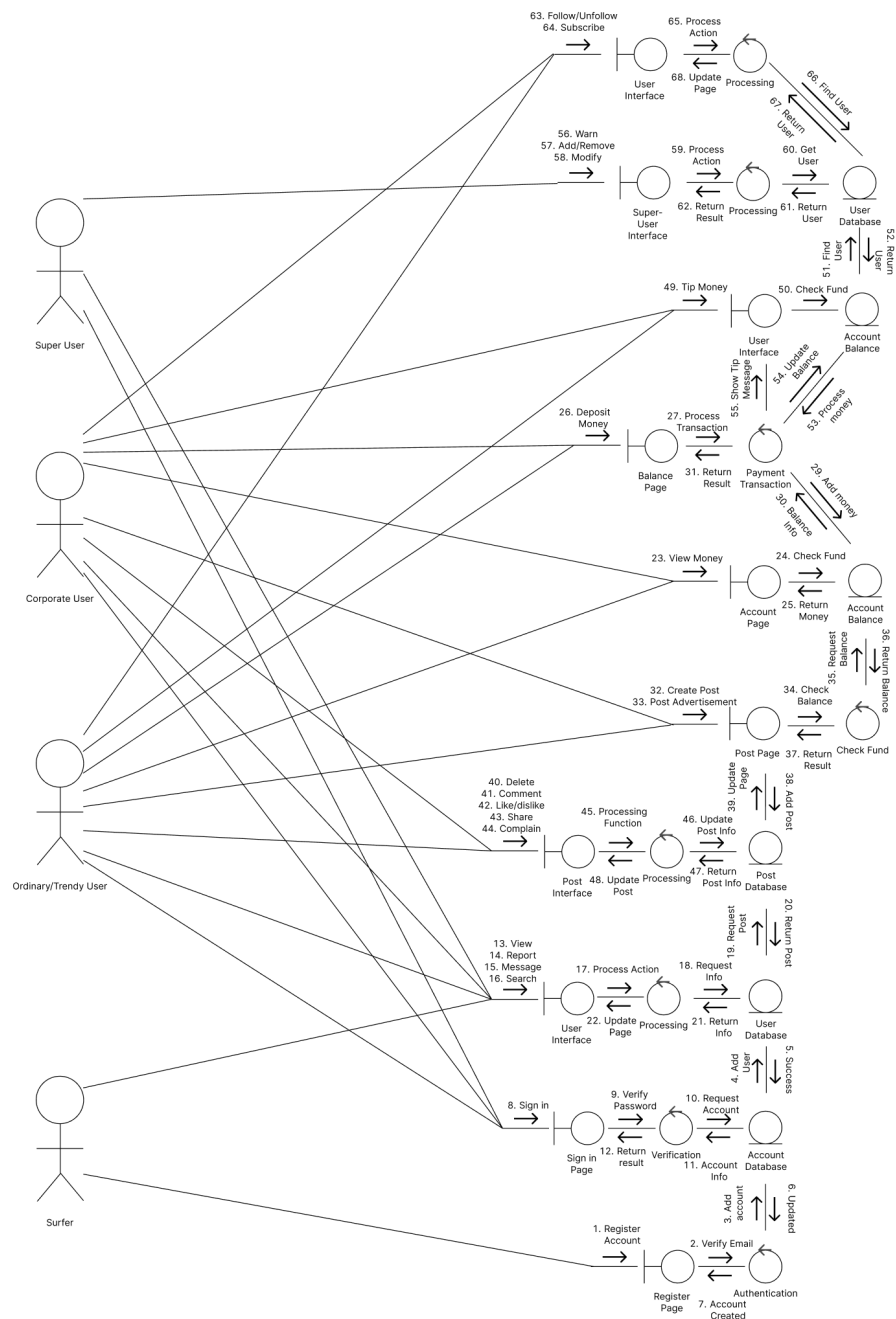
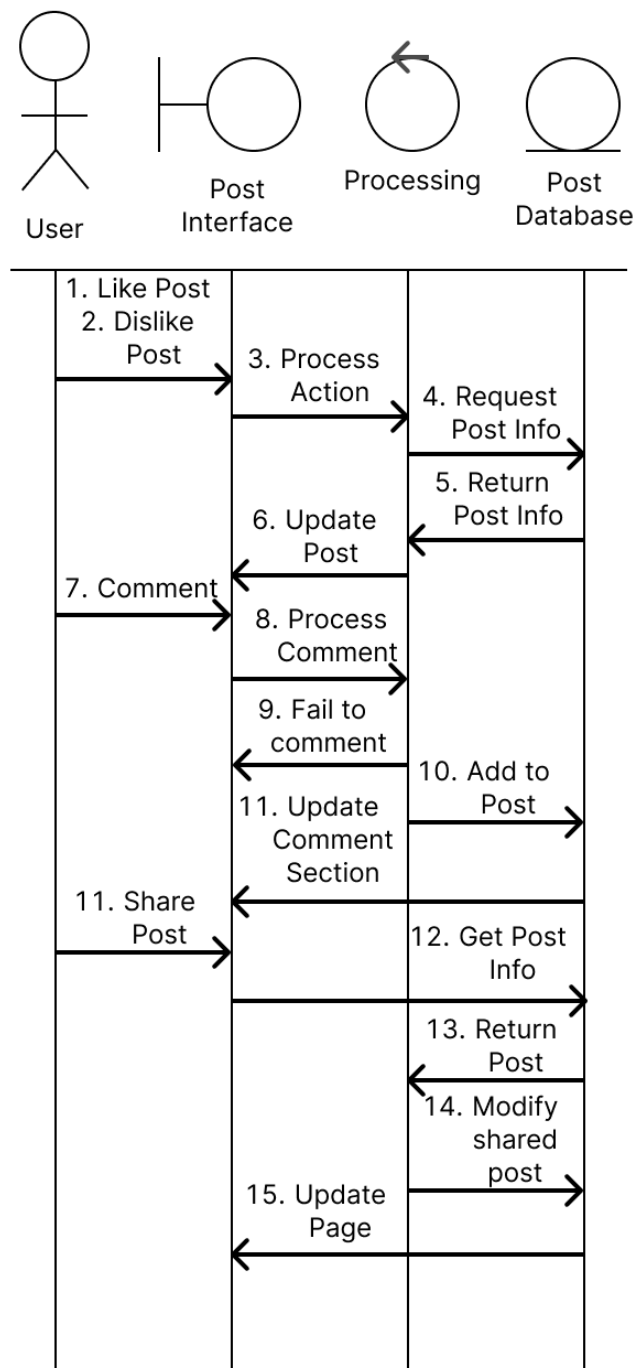| Mini Twitter	Version: 1.0
Software Design Report	Date: 11/21/23
Document Identifier: 10	

All Use Case

Sequence Class Diagram for Sign up, login, and logout of a system:
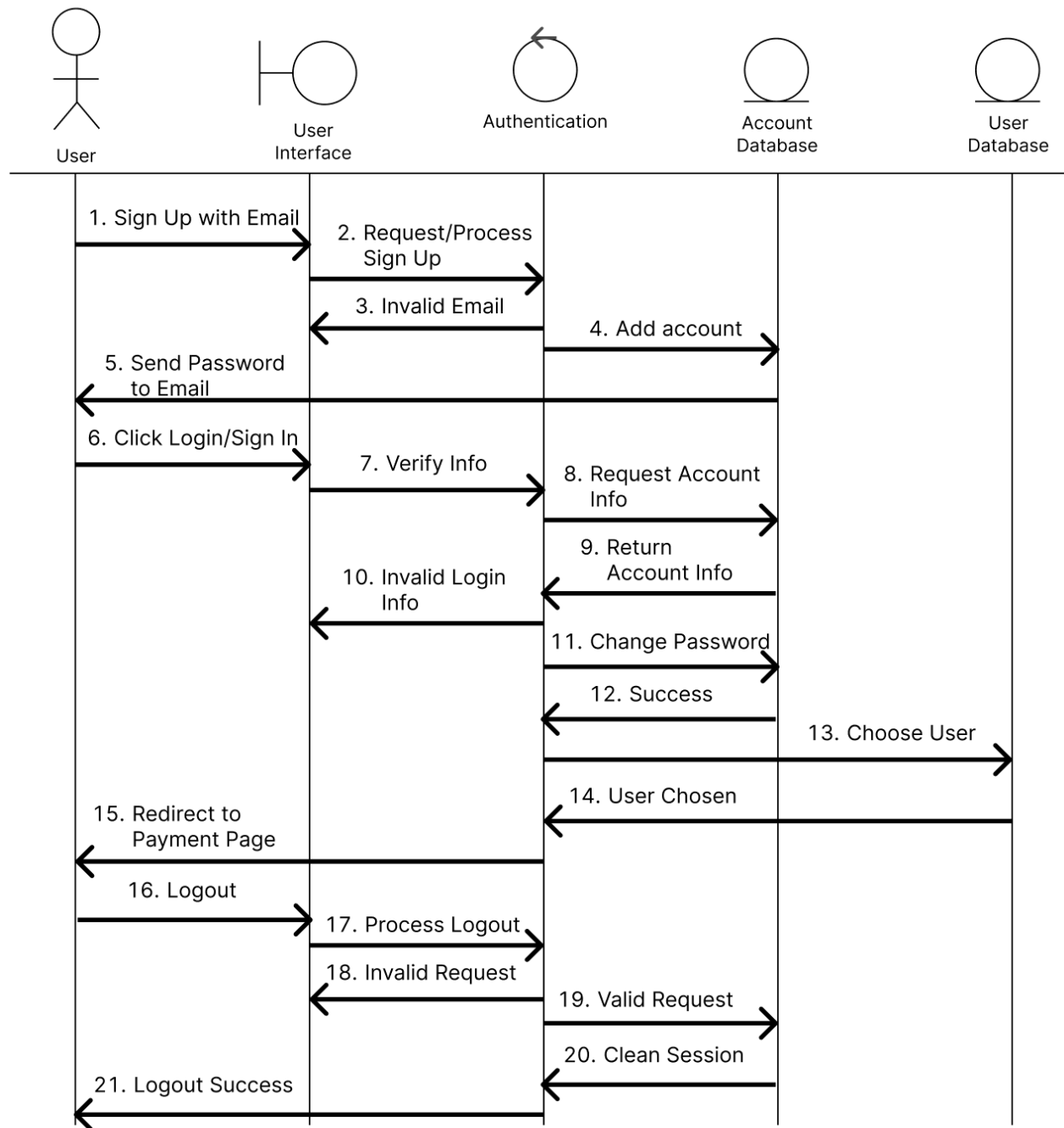
Mini Twitter	Version: 1.0
Software Design Report	Date: 11/21/23
Document Identifier: 10	

Sequence Class Diagram for like/dislike, comment, and share on a post:

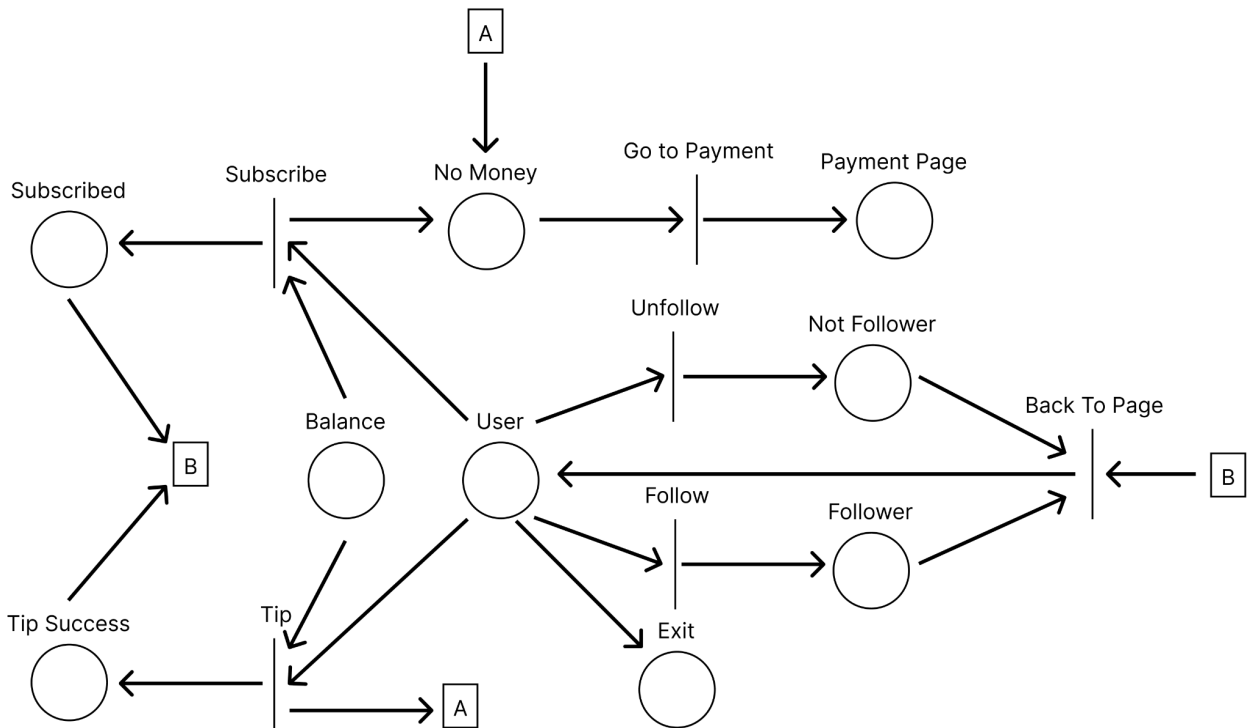| Mini Twitter	Version: 1.0
Software Design Report	Date: 11/21/23
Document Identifier: 10	

Sequence Class diagram for post, view balance, and deposit money:
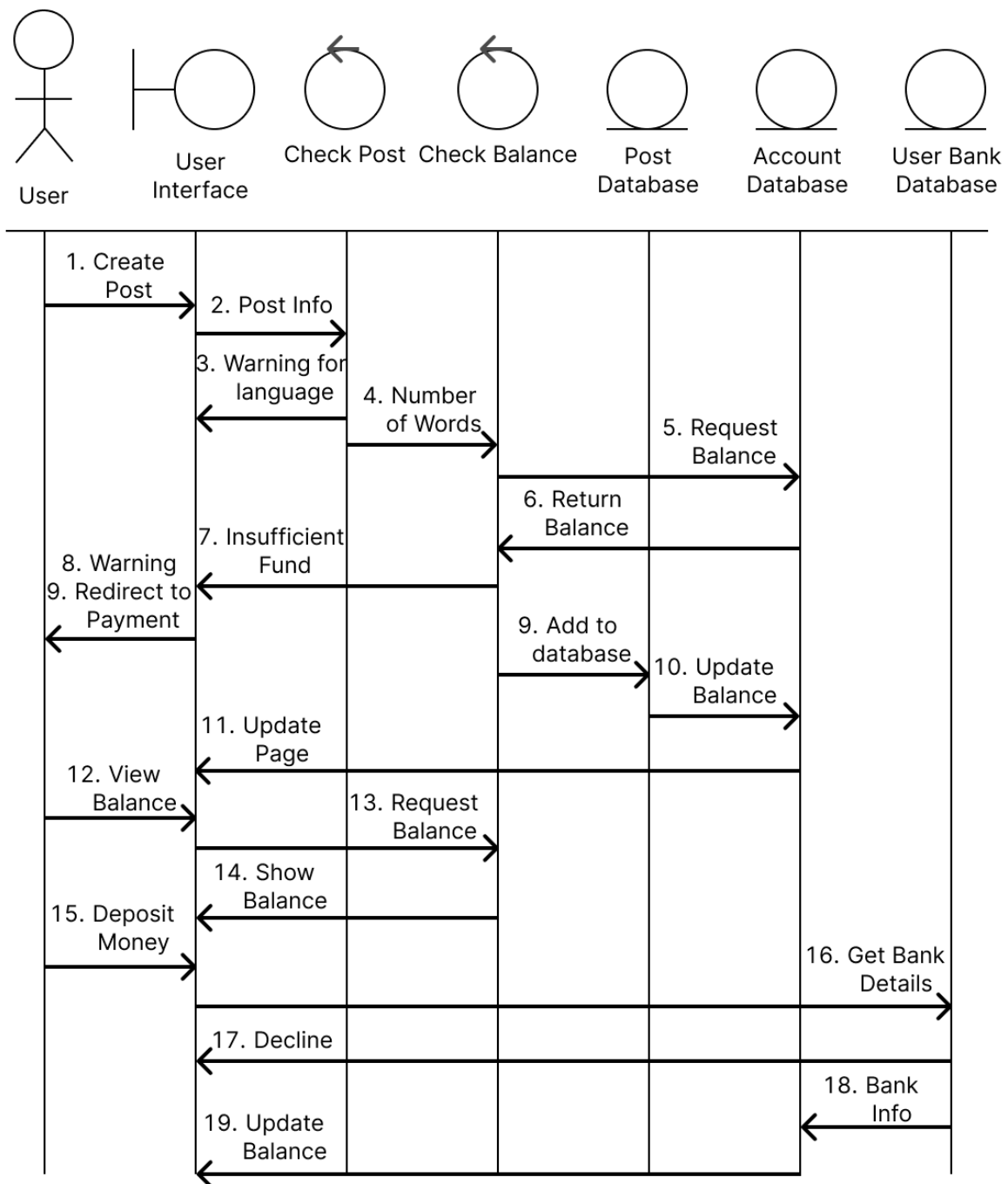
Petri-net for follow/unfollow, subscribe, and tip to user:

Mini Twitter	Version: 1.0
Software Design Report	Date: 11/21/23
Document Identifier: 10	

Petri-net for open, send, see, modify, search, and delete messages:
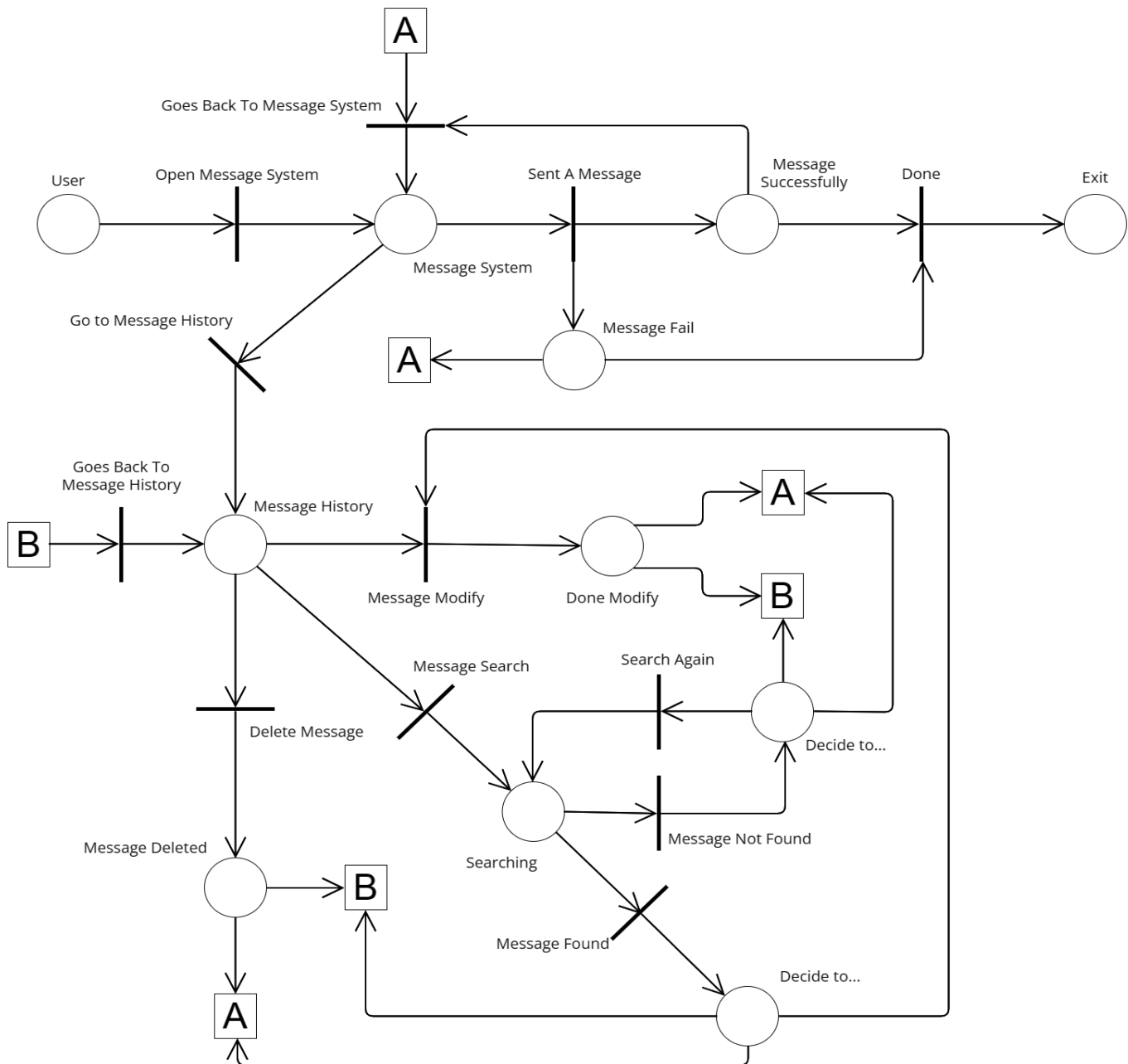
Mini Twitter	Version: 1.0
Software Design Report	Date: 11/21/23
Document Identifier: 10	

Petri-net for open, comment, view users, and posts on trending page and create messages, deleting message, and viewing message on trending page:
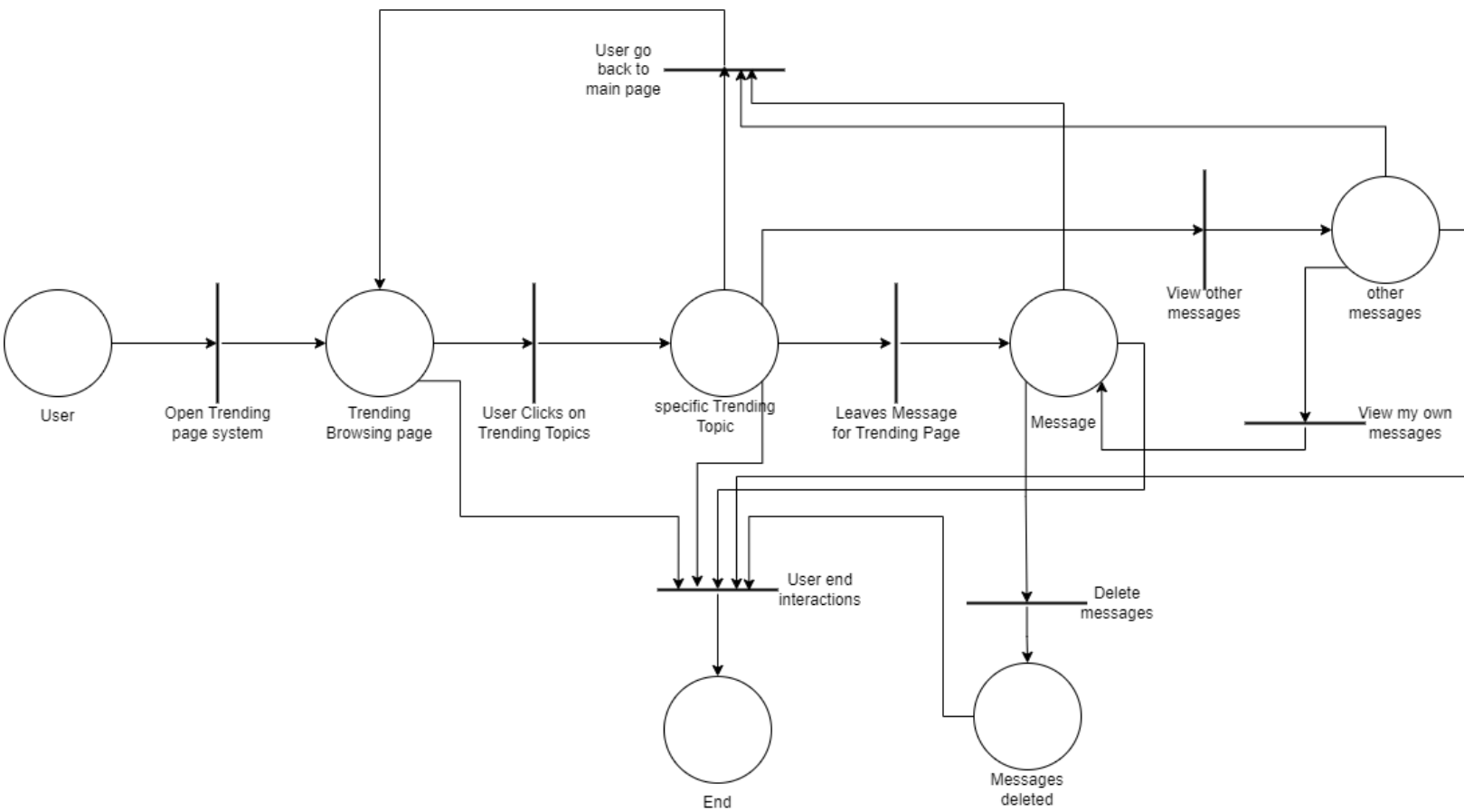
Mini Twitter	Version: 1.0
Software Design Report	Date: 11/21/23
Document Identifier: 10	

Petri-net for complains, warnings, and reports on users:
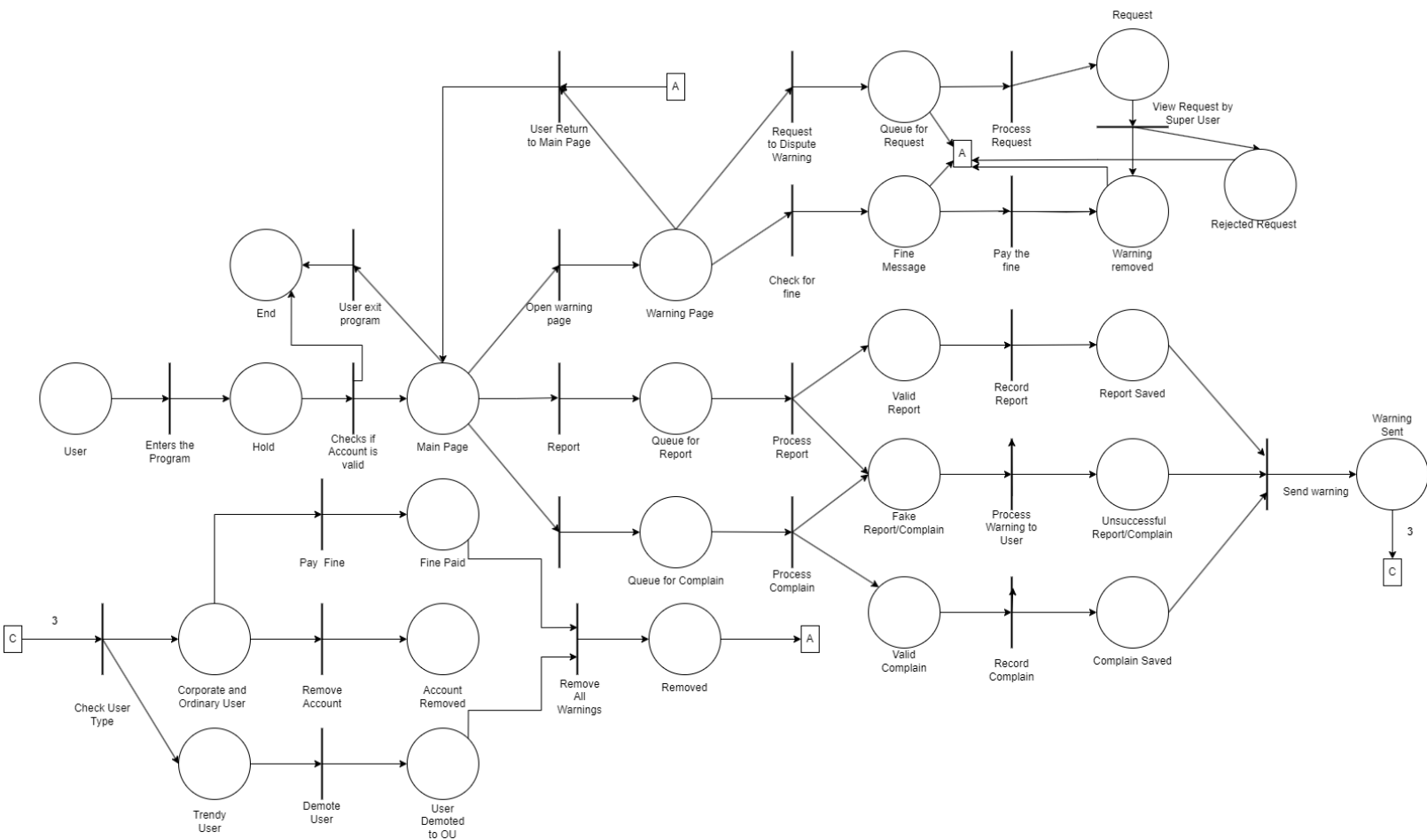
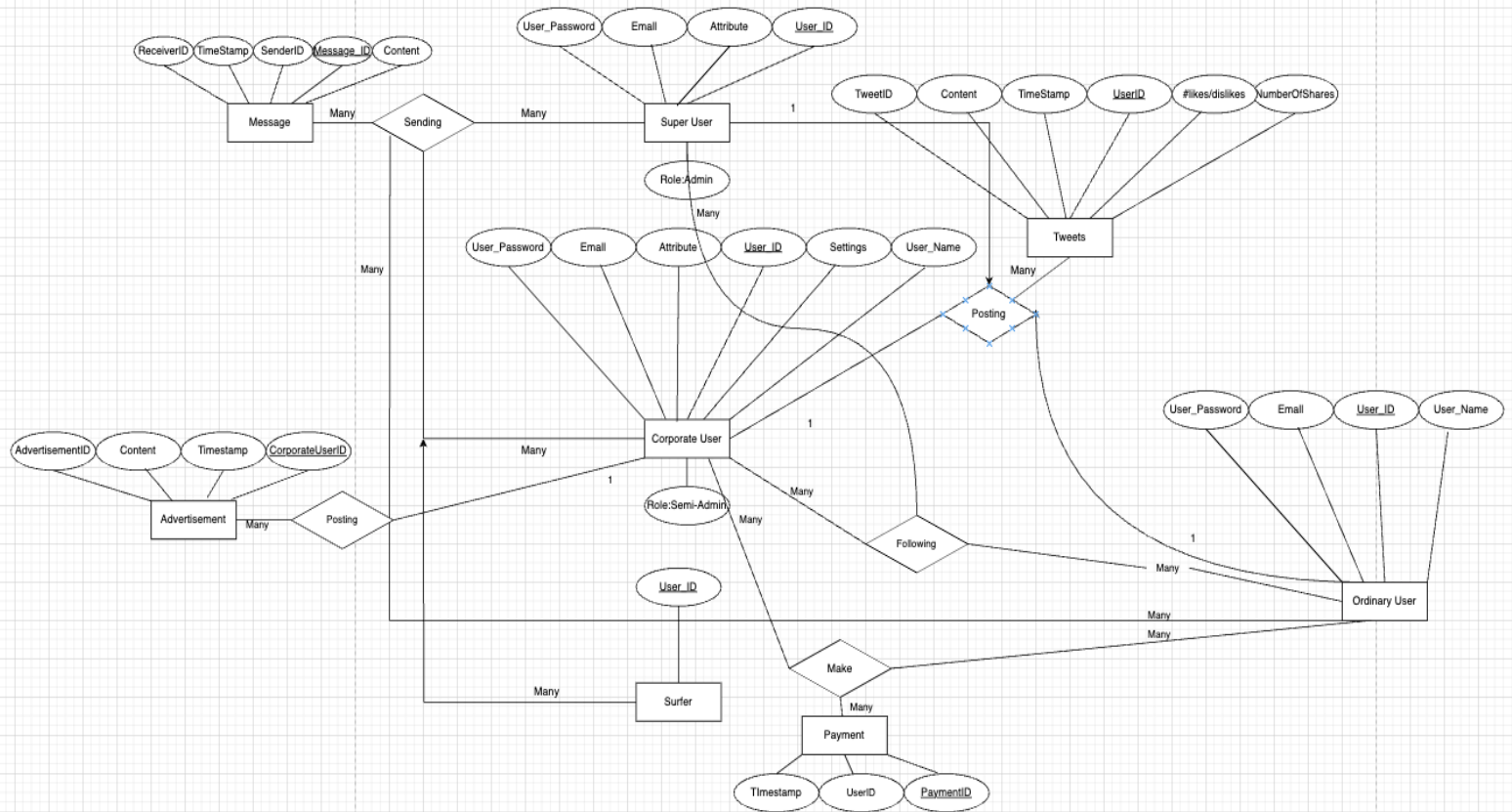| Mini Twitter	Version: 1.0
Software Design Report	Date: 11/21/23
Document Identifier: 10	

E-R diagram

Detailed Design

1. **Account Creation and signin/signout**
   Function: Register(userName, userEmail)
   　　　　Set registrationStatus to "Registration Failed: Invalid Input"
   　　　　if (validateInput(userName, userEmail))
   　　　　　　if (isUniqueUser(userName, userEmail))
   　　　　　　　　createUserAccount(userName, userEmail)
   　　　　　　　　sendConfirmationEmail(userEmail)
   　　　　　　　　changePassword()
   　　　　　　　　redirectPaymentPage()
   　　　　　　　　Set registrationStatus to "Registration Successful"
   　　　　　　Else
   　　　　　　　　set registrationStatus to "Registration Failed: User Already Exists"
   　　　　Else
   　　　　　　set registrationStatus to "Registration Failed: Invalid Input"
   　　　　Return registrationStatus

   Function: validateInput(userName, userEmail)
   　　　　Return true if input is valid/in the right form, false otherwise

   Function: isUniqueUser(userName, userEmail)
   　　　　Return true if the username and email is unique, false otherwise

   Function: createUserAccount(userName, userEmail)
   　　　　Include error handling for any issues in account creation
   　　　　Set the user information into the account database
   　　　　Return none

   Function: sendConfirmationEmail(userEmail)
   　　　　Generate random password
   　　　　Send a email with the temporary password generated
   　　　　Return none

   Function: changePassword()
   　　　　Enter new password
   　　　　Set password to the new password
   　　　　Return none
   Function redirectPaymentPage()

```
            Change UI to the payment page
            Return None


    Function: Signin(userEmail, userPassword)
            Set signinStatus to "Sign in failed: Invalid Email or Password"
            status = False
            if(checkUserEmail(userEmail) and checkUserPassword(userPassword) and
status)
                    Set signinStatus to "Sign in Complete"
                    setUserStatus(userEmail)
            Return signinStatus


    Function: setUserStatus(userEmail)
            Obtain user from user database
            Set status to logged in
            Return none


    Function: checkUserEmail(userEmail)
            Check with user database
            Return true if success else false


    Function: checkUserPassword(userPassword)
            Check with user database
            Return true if success else false


    Function Signout()
            Clear current session
            print("User signed out successfully")
            Return none


2.  Payment System
    Function: addFunds(userEmail, amount)
            bankInfo = retrieveBankInfo(userEmail)
            if bankInfo:
                    balance = bankInfo.get('balance')
                    if balance > amount:
                            updateBalance(userEmail, balance - amount)
                            status = takeMoney(amount)
                            print(f"Transaction Status: {status}")
```

```
                else:
                        print("Insufficient balance to add funds.")
        else:
                print("Failed to retrieve bank information.")


Function: retrieveBankInfo(userEmail)
        Retrieve bank information through an API
        Balance = userEmail.getBalance()
        return balance


Function: updateBalance(userEmail, newBalance)
        userBalance = newBalance
        Return none


Function: giveWarning(message)
        Add warning to list
        print(f"Warning: {message}")
        Return none


Function: takeMoney(amount)
        print(f"Taking {amount} from the user's account")
        Take money using an API
        return True if success
```

3. **Post Tweets/Post Advertisements**

```
Function: postTweets(post, user)
        words = countwords(post)
        amount = calculateMoney(words, user)
        takemoney(amount)
        add post to database


Function: countwords(post)
        Word = 1
        Video = 15
        Image = 10
        Return sum of words, videos, images


Function : calcuateMoney(words, user)
        If user is CU
```

```
                    Return $1 * words
            Else
                    If words > 20
                            Return $0.1 * (words - 20)
                    Else
                            Return 0


    Function: postAdvertisement(ad, user)
            If user is CU
                    Add ad to database
                    Update Page
            else
                    GiveWarning(message)
                    takeMoney(10)
            Return true if success, else false


    Function: ClickAd(ad)
            Add one to total click
            Ad's user pay SU $0.1
            Return none


    Function: ClickPost(post)
            Enlarge post
            Increment click count
            Return none
```

4. Like/dislike/share/comment Post
```
    Function: Like(post)
            If post.getILikedStatus() == true
                    removeLike(post)
            else
                    addLike(post)


    Function: RemoveLike(post)
            Decrement total like
            Update post


    Function: Addlike(post)
            Increment total like
```

Update post

Function: Dislike(post)
If post.getDislikedStatus() == true
removeDislike(post)
else
addDislike(post)

Function: RemoveDislike(post)
Decrement total dislike
Update post

Function: AddDislike(post)
Increment total dislike
Update post

Function: SharePost(post, user)
postTweets(post, user)
addComment(comment, post)

Function: addComment(post)
Enter comment
Add comment to post
Return True if success, else false

5. **View/send/delete/edit Messages**
Function: viewMessages(user, otherUser)
Open message of party one and party two
displayMessages()
Return none

Function: displayMessages()
Open message UI
Return none

Function: sendMessages(user, other user)
Enter message
Send message to other user
Update UI

Return true if message is delivered, else false

Function: editMessage(message)
Message becomes edible
Complete edit
Return true if edit is complete, else false

Function: deleteMessage()
Message = messages.getmessage()
Delete messages[message]

6. **Get/View User Profile**
Function: ViewProfile(user)
Show profile screen
Return None

Function: getUserProfile(userID)
if (isUserExist(userID))
userProfile = getUser(userID)
else
set userProfile to "User Profile Not Found"
return userProfile

Function: isUserExist(UserID)
Return True if userID in database, else false

Function: getUser(userID)
Return User[userID]

7. **Add/resolve/remove warning**
Function: addWarning(userID)
userID.warnings += 1

Function: removeWarning(userID)
userID.warnings -= 1

Function: payTheFine(userID)
Fine = userID.warnings*FineConstant
If retrieveBankInfo(userEmail) == True

```
                    While userID.warnings > 0
                            removeWarning(userID)
                            Print ("Success")
            Else
                    Print ("Insufficient balance")
            Print "Failed to remove warnings")


Function: checkWarnings(userID)
        If userID.warnings > 3
                PunishUser(userID)
        Else
                Return 0


Function: removeAccount(userID)
        Remove userID
        Print("Account removed")


Function: PunishUser(userID)
        If userID.type == OrdinaryUser
                removeAccount(userID)
                Return Success
        Else
                userID.type = OrdinaryUser
                While userID.warnings > 0
                        removeWarning(userID)
                Return Success


Function: resolveWarning(userID, decision)
        If Decision == true
                removeWarning(userID)
        Else
                Return false


Function: reportComplainHandler(userID, targetUserID, reportResult, complainResult)
        If reportResult or complainResult == true
                addWarning(targetUserID)
        Else
                addWarning(userID)
```

8. Follow/Subscribe/Tip User

Function: FollowUser(user, userToFollow)

    if userExist(userToFollow):

        Open Following UI

        Add userToFollow into the Followed list

        print("You Are Now Following")

    else

        print("User Not Found")

        Return False

Function: Subscribe(user, subscribing, balance)

    Open Subscription UI

    Status = "Failed to subscribe

    balance = user.bankInfo.get('balance')

    if balance > \$10:

        update(balance - 10)

        Subscribing.subscriber += 1

        Status = "Subscribed"

    Return Status

Function: tipUser(user, userToTip, balance)

    Open Tip UI

    ShowBalanceInTotal(balance)

    if balance > \$0.01

        ProceedToTip(user, userToTip)

        Update balance(balance - userToTip)

    else

        print("Balance Insufficient")

9. Complain/report Post or User

Function: reportPostOrUser(postOrUser)

    get(postOrUser)

    clickOn(reportButton)

    selectIssue(issueType)

    Enter details

    submitReport()

    Return True if success, else false

Function: DisputeWarning(warning)

clickDispute(disputeButton)
Provide Information
submitDispute()
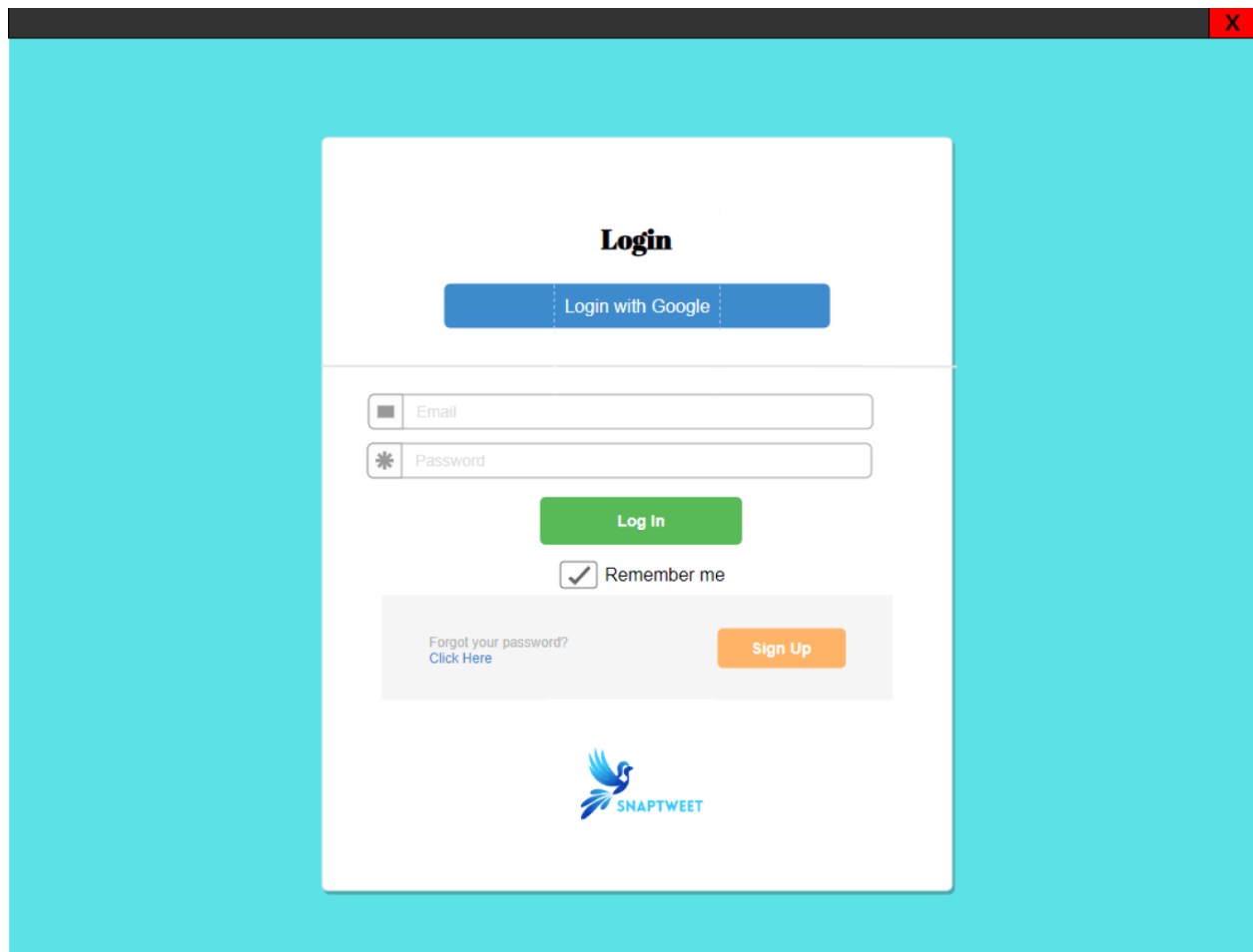Return True if submitted, else false

10. Toggle Dark Mode
Function: toggleDarkMode(isDarkModeOn: boolean)
if (isDarkModeOn)
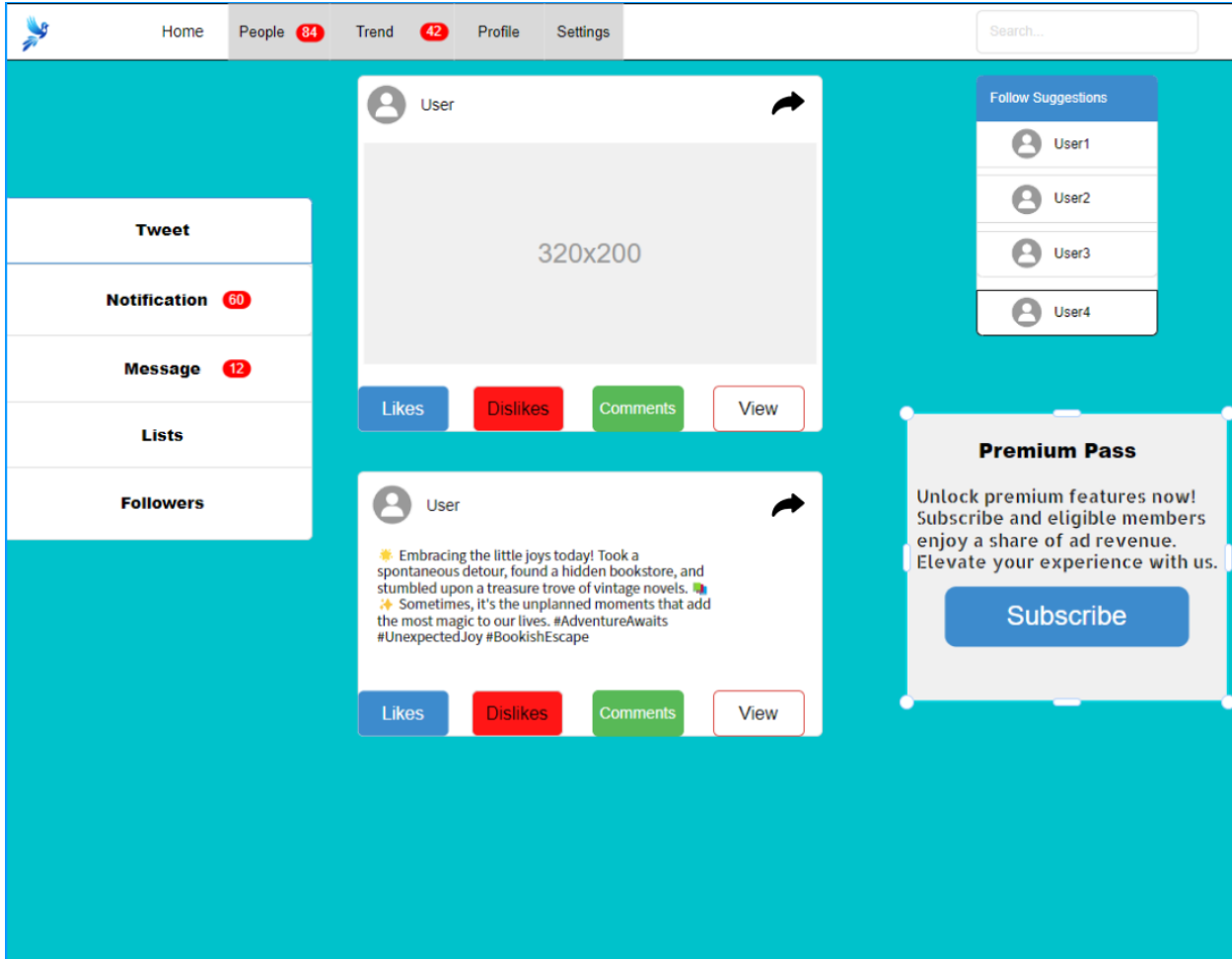   setTheme('light')
else
    setTheme('dark')
 Return none

Mini Twitter	Version: 1.0
Software Design Report	Date: 11/21/23
Document Identifier: 10	

System Screens

Mini Twitter	Version: 1.0
Software Design Report	Date: 11/21/23
Document Identifier: 10	

Mini Twitter	Version: 1.0
Software Design Report	Date: 11/21/23
Document Identifier: 10	

Mini Twitter	Version: 1.0
Software Design Report	Date: 11/21/23
Document Identifier: 10	

Mini Twitter	Version: 1.0
Software Design Report	Date: 11/21/23
Document Identifier: 10	

Memos

Group Meeting: November 18th, 2023, 3PM - 5PM, All Attended
Finished Work:
- Most part of Use Case Diagrams
- Overall Collaboration diagrams
- E-R diagram
- Some of Detailed Design

November 20th, 2023
- Modification with the diagrams
- Added System Screens

November 21th, 2023
- Some modification with the diagram
- Added more parts in Detailed Design
- Complete Detailed Design

Possible Concerns: None

Mini Twitter	Version: 1.0
Software Design Report	Date: 11/21/23
Document Identifier: 10	

Address

https://github.com/WeCanPretend0903/Mini_Twitter