CSI 3471 Group Project Iteration 1

Project Vision

Our vision is to allow a user to optimize their happiness by providing him or her feedback regarding how their activities and relationships impact their mood. The user will have daily logs where they enter such information and in return he or she will receive a weekly report and suggestions on how to improve overall happiness. Moreover, we hope to allow the user to cultivate a thriving network of supportive friends by allowing users to share reports, moods, and activities. Moreover, we will allow users to post on each other's walls.

Team Members:

- Micah Dadson
- Jeremy Meadows
- Shuting (Iris) Chen
- Francis Ning
- Shivani Bobbala
- Joel Futagawa

- Website Link: http://francisning0203.wixsite.com/mysite-1
- Git link: https://github.com/jeremymeadows/Mentality
- Link to presentation: https://docs.google.com/presentation/d/1btB6YMMj-l8issRWKononz1wLBe9ovCciUoY3pSFLnY/edit#slide=id.p

Functional Requirements

-Register

User must be able to register and sign in.

System assignment a unique id to each user and save after user registered.

-Survey

A mood survey should display each time user registered.

Each survey should be saved and can be accessed for analyzing later.

-Report

User can be able to publish happiness scores.

User can be able to view the happiness scores.

User can enter mood data for a specific day.

System should save the mood data for user.

System should do statistic report to user weekly.

System should make predictions on mood fluctuation.

-Friends

User can add friends that they search for or are recommended.

User can add a record of time spent with a friend on a specific day.

User can edit the records of time spent with friends.

User can enter locations they spent their time at.

User can enter names of people that they had interactions.

User can be able to suggest activities by friends.

-Wall

User can be able to make a post to a friend’s wall.

User can be able to delete a post to a friend’s wall.

User can be able to write comments on post.

Nonfunctional Requirements

- System should comply with HIPAA requirements and protect the private information of users.
- Response time and net processing time should be within a reasonable limit in order for the user to use the app effectively and with ease.
- Application should be available via a web interface with a common and user-friendly user interface.
- System should be able to save user data reliably and return the data securely whenever prompted.
- Security - the system should be able to store user login information and authenticate users before allowing them access to their data.
- Portability - the system should not be tied to any specific OS, database, or cloud provider to ensure that users can access it on the technology of their choice.

Fully Dressed Use Cases

1. Register

ID: UC01
Primary Actor: Unregistered User
Stakeholders and Interests:
Unregistered User: Wants to register quickly with a guarantee that her data is protected.
Company: Wants to ensure that the database of registered users is appropriate and up to data to avoid customer negative feedback.
Pre conditions: The user has accessed the app.
Post conditions: The user is added to the database and may log in.
Main Success Scenario:

1. User enters a username
2. User enters a password
3. User retypes the password
4. User pushes submit
5. System checks that the username is not already in use
6. System checks that the two passwords are identical
7. User uploads a photo
8. User fills out mood history survey.
9. System registers the new user

Extensions:
5a. User name is already in use.

1. User is prompted to select another user name and password

5b. The two passwords are different.

1 User is requested to retype (twice) his/her password

7a. User skips this step.

Primary Actors	Unregistered User
Level	N/A
Complexity	N/A
Use Case Status	N/A
Implementation Status	N/A
Preconditions	N/A
Post-conditions	N/A
Author	N/A
Assumptions	N/A

1.1. Scenarios

1.1.1. Scenario

1.

1. Give weekly report

ID: UC17
Primary Actor: User
Stakeholders and Interests:
User: Wants to find out how his or her daily activities impact his or her mood.
Company: Wants the analysis and predictions to be accurate.
Pre conditions: The user has a registered account
Post conditions: The user receives an analyzed report that reveals the activities that make him or her the happiest with suggestions.
Main Success Scenario:
1. User enters information regarding his or her mood, activities, and relationships.
2. System stores information.
3. System generates a weekly report regarding mood patterns.
4. User chooses to make report public.
5. User receives comments from his or her friends.
6. System checks mood scores of friends to offer suggestions.
7. User rates the system suggestions.

Extensions:
3a. The user has not logged in the app within a seven day period
1. The user will not receive a weekly report
3b. Some logged activities are not associated with a mood
1.System ignores data
4a. User makes report private.
6a. User does not have any friends
1. System offers user positive affirmations

Primary Actors	System
Level	N/A
Complexity	N/A
Use Case Status	N/A
Implementation Status	N/A
Preconditions	N/A
Post-conditions	N/A
Author	N/A
Assumptions	N/A

1. Send Friend Request

ID: UC24

Primary Actor: User A

Stakeholders and Interests:

User A: Wants to create a network of friends.

Company: Wants to make sure that the data of private users is protected.

Pre conditions: User A has a public account and is logged in.

Post conditions: The user adds a friend to his or her network.

1. User A searches for User B.
2. User A views the profile of User B.
3. User A sends User B a friend request.
4. User B accepts the friend request from User A.
5. User B becomes a part of User A's network.
6. System recommends User A friends based on his or her network of friends.

Extensions:

2a. User B does not exist or User B is on private

1. User A searches for different user.

4a. User B denies User A's friend request.

1. User B is not added to User A's network

Level	N/A
Complexity	N/A
Use Case Status	N/A
Implementation Status	N/A
Preconditions	N/A
Post-conditions	N/A
Author	N/A
Assumptions	N/A

ID: Record Friend
Scope: Happiness Tracker System
Level: User Goals
Stakeholders and Interests
User – The person who wishes to add a friend to their record

Precondition: The user has already started the system

Postcondition: The system has a record of the friend the user wanted to add

Main Success Scenario

1. User selects a date to make changes to
2. User selects “add friend”
3. User inputs name of friend and amount of time spent with them
4. User selects “save changes”

Extensions

1.a The selected date is in the future
   1. A message reading “Date has not occurred. Cannot make changes” is returned

3.a The user is unable to provide a time
   1. The process proceeds as normal with the “time” value left empty

3.b The friend’s name is already recorded for that day
   1. No changes are made and a message reading “Friend already exists” is returned

ID: Posting on Another User's Wall
Scope: Happiness Tracker System
Level: User Goals
Stakeholders and Interests
User – The person who wishes to post on someone's wall
Recipient – The person whose wall is being posted on

Precondition: The user has already started the system

Postcondition: A post is made to another user's wall

Main Success Scenario

1. The user selects one of their friends
2. The user clicks "View Wall"
3. The user clicks "Make Post"
4. The user inputs the contents of the post and clicks confirm
5. The post is made and saved to the database

Extensions

2.a The system is unable to retrieve the recipient's wall
   1. A message reading "Retrieval error, cannot get wall" is returned

4.a The user provides no input
   1. A message reading "A post must have content" is returned and this step is repeated

5.a The system is unable to connect to the database
   1. A message reading "Connection Error, cannot connect to database"

ID: Editing a Friend
Scope: Happiness Tracker System
Level: User Goals
Stakeholders and Interests
User – The person who wishes to make changes to the friend they spent time with

Precondition: The user has already started the system

Postcondition: Changes are made to the friend on record

Main Success Scenario
1. The user selects a date to make changes to
2. The user selects the friend whose information needs changing
3. The user inputs the new information
4. The user selects “Save Changes”

Extensions
3.a The friend’s edited name is already recorded for that day
1. No changes are made and a message reading “Friend already exists” is returned

ID: Keep track of mood

Scope: Mentality

Level: User goal

Stakeholders and interests:

Customer

-user which uses the app

Counselor

-person which extracts customer's data from app

Precondition: Customer has account and is logged in

Postcondition: Entered information is saved

Main Success Scenario:

1. Customer wants to add information to app
2. Select a day that the information corresponds to
3. App will prompt user to answer series of questions
4. User may answer as many/few questions as they want
5. App will save answers

Extensions:

5a. No network connection

1. App will notify user that information couldn't be saved due to network error

ID: Enter locations visited

Scope: Mentality

Level: User goal

Stakeholders and interests:

Customer

-user which uses the app

Counselor

-person which extracts customer’s data from app

Precondition: Customer has account and is logged in

Postcondition: user receives notification

Main Success Scenario:

1. Customer wants to add location information to app
2. Select a day that the information corresponds to
3. App will prompt user to select places they spent their time on a map
4. If there isn't map data for a location, the user may enter custom information
5. App will save location

Extensions:

5a. No network connection

2. App will notify user that information couldn’t be saved due to network error

ID: Enter people interacted with

Scope: Mentality

Level: User goal

Stakeholders and interests:

Customer

-user which uses the app

Counselor

-person which extracts customer's data from app

Precondition: Customer has account and is logged in

Postcondition: user receives notification

Main Success Scenario:

6. Customer wants to add other people to app
7. Select a day that the information corresponds to
8. App will prompt user to name people that they interacted with
9. App will save data

Extensions:

5a. No network connection

3. App will notify user that information couldn't be saved due to network error

ID: Comment Post on Another User's Wall
Scope: Mentality
Level: User Goals

Stakeholders and Interests
User – The person who wants to write a comment on a post
Recipient – The person whose has posts on his wall

Precondition: The post is on Recipient's wall. User is registered.
Postcondition: The post is commented.

Main Success Scenario
1. The user selects one of their friends
2. The user clicks "View Wall"
3. The user clicks specific post which wants to be write comment
4. The user writes comment on a window
5. The user hit "submit"
6. The user confirmed his comment
5. The comment is displayed under the post

Extensions
2.a The system is unable to retrieve the recipient's wall
   3. A message reading "Retrieval error, cannot get wall" is returned
3.a The user was denied to comment the post
   4. A message reading "Sorry, you cannot comment this post." is returned
4.a The user chooses "cancel" on that confirmation message
   5. Posts are not commented successfully

ID: Delete Post on Another User's Wall
Scope: Mentality
Level: User Goals

Stakeholders and Interests
User – The person who wishes to post on someone's wall
Recipient – The person whose wall is being posted on

Precondition: The post is on Recipient's wall. User is registered.
Postcondition: The post is deleted.

Main Success Scenario
1. The user selects one of their friends
2. The user clicks "View Wall"
3. The user clicks specific post which wants to be deleted
4. The user confirmed the deletion by pop-up window
5. The post is erased from the wall

Extensions
2.a The system is unable to retrieve the recipient's wall
   3. A message reading "Retrieval error, cannot get wall" is returned
3.a The user was denied to delete the post
   4. A message reading "Sorry, you cannot delete this post." is returned
4.a The user choose "cancel" on that confirmation message
   5. Posts are not changed on the wall

ID: Suggest Activities

Scope: Mentality
Level: User Goals

Stakeholders and Interests
User – The person who is on his home page
Manager – Collect data and analyze correctly

Precondition: The person is on his home page.
Postcondition: The post is deleted.

Main Success Scenario
1. The user opens the application.
2. The user goes into his home page.
3. A block displays suggested activities.

Extensions
2.a The user is new and no data is collected and analyzed.
3. Activities will be randomly generated, then display.

ID/Use Case Name: Create new journal entry
Scope: Happiness Tracker System
Level: User goal
Stakeholders and Interests:

User -person who is interested in creating a journal entry
Counselor - person who will use user's data extracted from the app

Precondition: user is authenticated and logged into his/her account
Postcondition: journal entry saved
Main success scenario:

1. User wants to fill out their daily journal entry
2. User will click on journaling part of dashboard
3. System will display all previous entries made by user and the option to create a new one
4. User will click on the option to create a new journal entry
5. System will open a new screen in which user can enter information
6. User will write some text into the textbox that is displayed and fill in the other fields that are presented by the system
7. User will hit save journal entry and system will save the journal entry along with the current date and time

Extensions:

7a. No network connection

1. System will save entry offline until the user is connected to the internet again. Once user is online again, system will save entry to the central database

7b. User wants to edit something in the entry

1. User will select edit option that exists for every previous journal entry

7c. User wants to save the journal entry with a different date/time

1. User will have the option to manually set the date and time for each journal entry

ID/Use Case Name: Enter sleep information
Scope: Happiness Tracker System
Level: User goal
Stakeholders and Interests:

User - person who is interested in entering their sleep information
Counselor - person who will use user's data extracted from the app

Precondition: user is authenticated and logged into his/her account
Postcondition: sleep information saved
Main success scenario:

1. User is interested in entering his/her sleep information
2. User will select option that allows him/her to enter sleep info
3. System will prompt user to select the day/night they want to enter info for
4. System will prompt user to enter the number of hours that the user slept
5. System will ask the user about his/her quality of sleep that night
6. System will ask user if he/she dreamt last night
7. User will select save information and system will save the data

Extensions:

6a. User says that they did dream that night

1. System will prompt user to make a new entry in his/her dream journal

7a. No network connection

1. System will save information offline until the user is connected to the internet again. Once user is online again, system will save information to the central database

ID/Use Case Name: Enter weather information
Scope: Happiness Tracker System
Level: User goal
Stakeholders and Interests:

User - person who is interested in entering the weather information
Counselor - person who will use user's data extracted from the app

Precondition: user is authenticated and logged into his/her account
Postcondition: sleep information saved
Main success scenario:

1. User is interested in entering weather information
2. User will select option that allows him/her to enter weather info
3. System will prompt user to select the date they want to enter info for
4. System will prompt user to enter the average temperature that day
5. System will prompt user to enter weather conditions that day
6. User will select save information and system will save the data

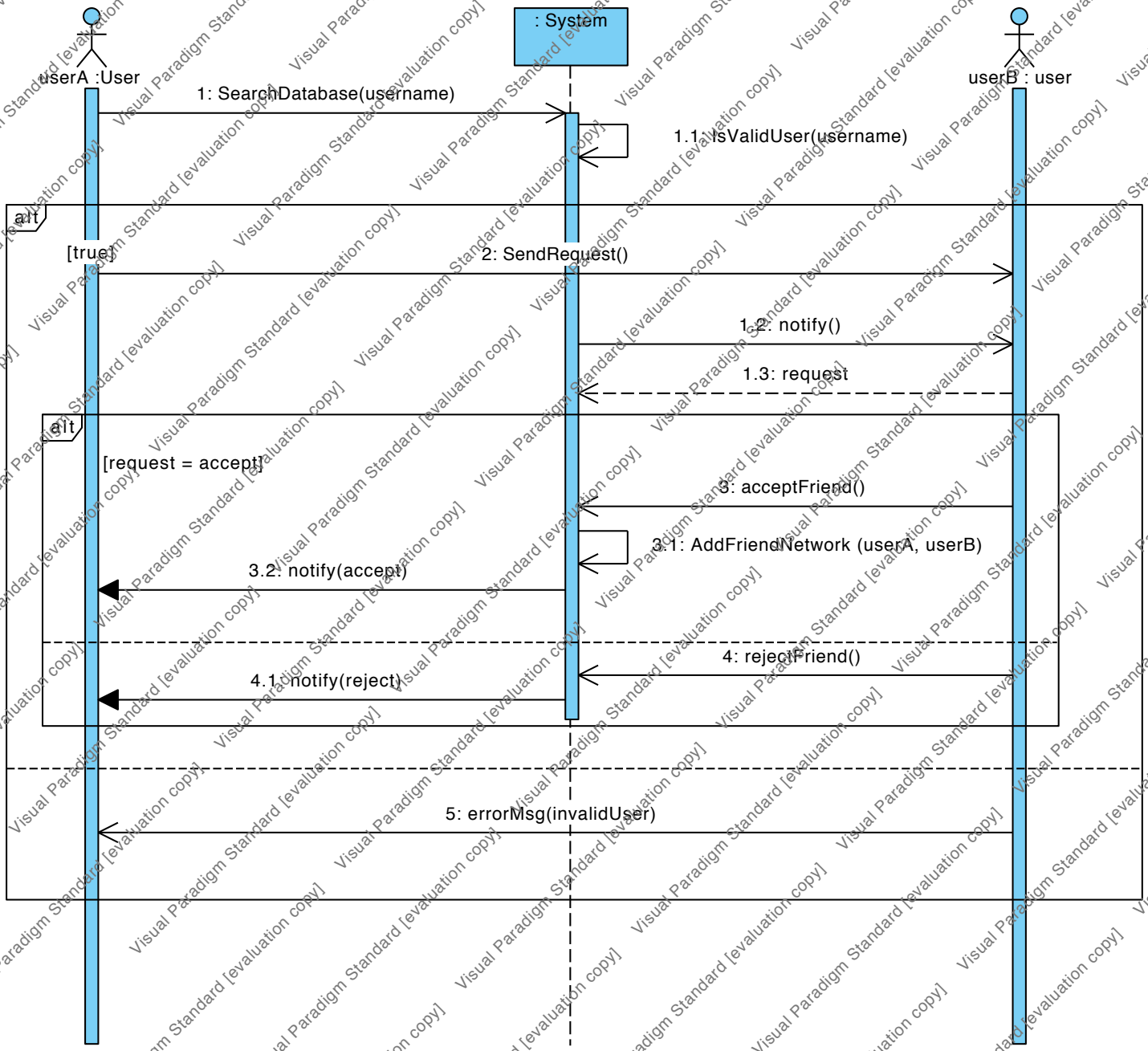
Extensions:
7a. No network connection

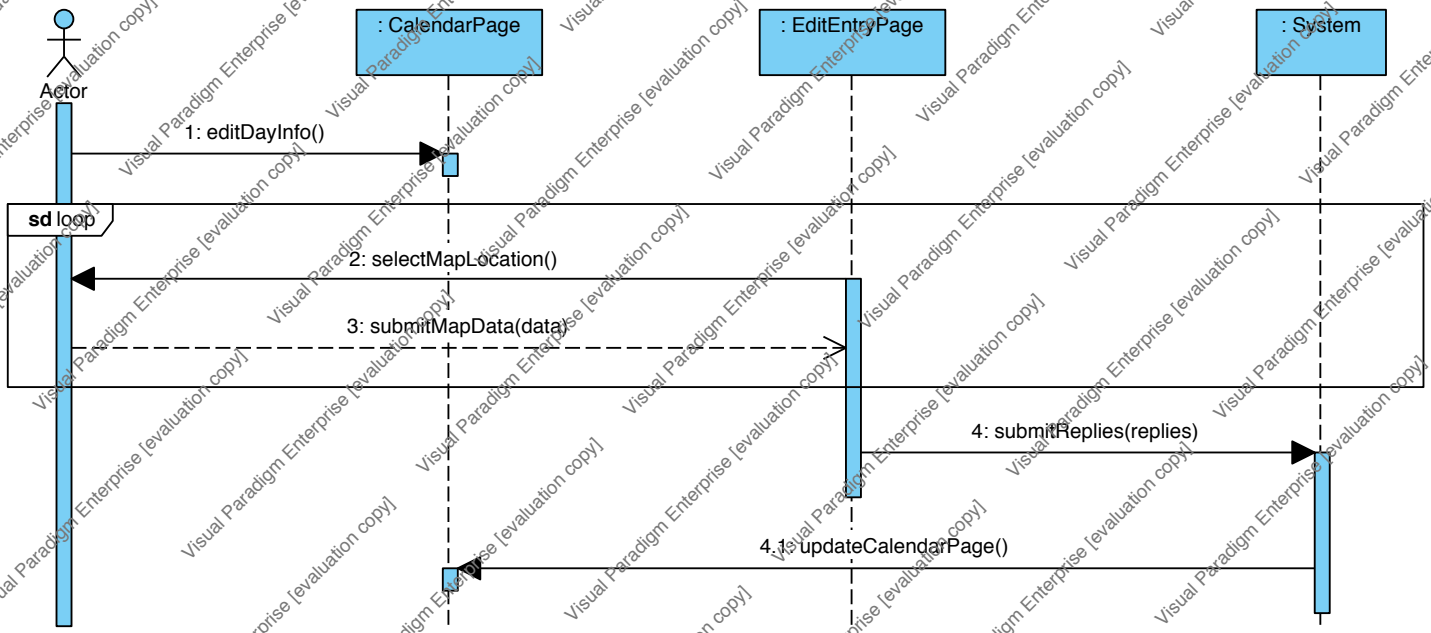
1. System will save information offline until the user is connected to the internet again. Once user is online again, system will save information to the central database

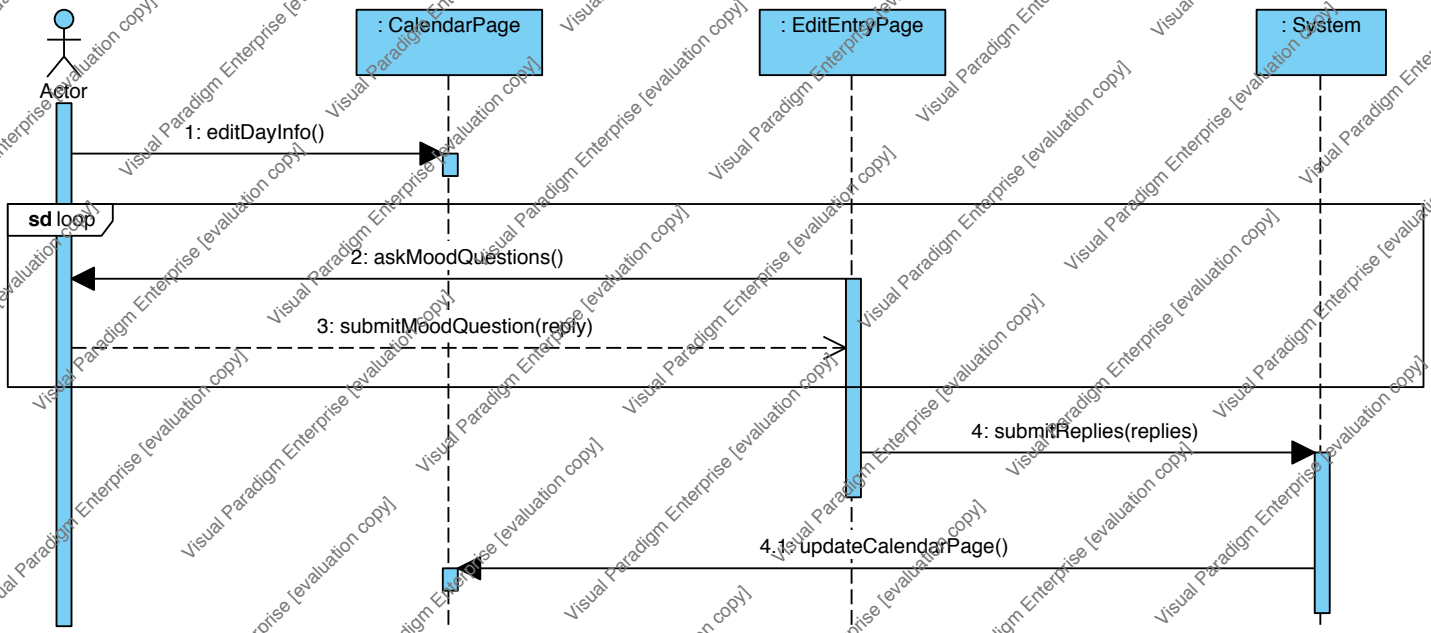
Traceability Matrix

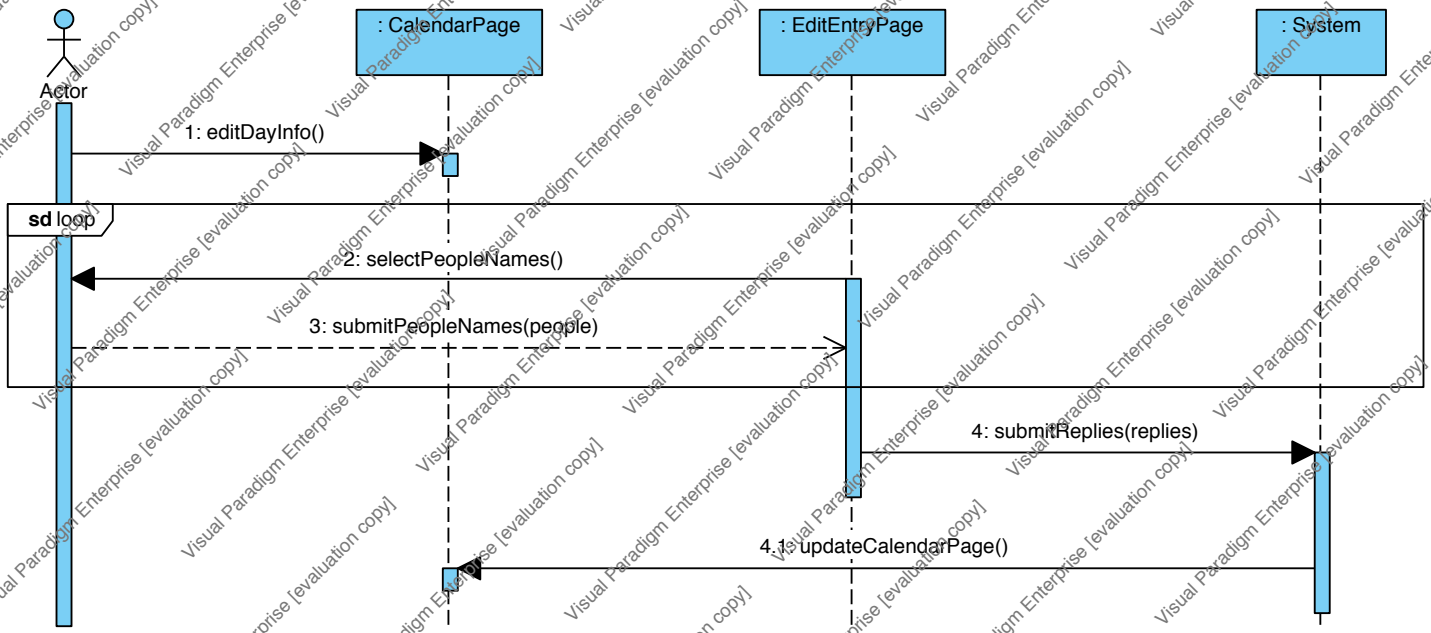
Traceability Matrix	Register	Sign In	Track Mood	Use Journal	Enter Activities	Track Company	Track Places Visited	Associate Dates	Track Sleep	Access Mood Stats	Receive Weekly Reports	Post to Walls	Publish Happiness Score
Add Friend													
Register User	x	x	x										
Weekly Report			x								x		x
Location Tracking							x	x					
Tracking Mood			x					x					
Record Friend						x		x					
Edit Friend						x		x					
Post to Wall												x	
Delete Post													
Post Comment												x	

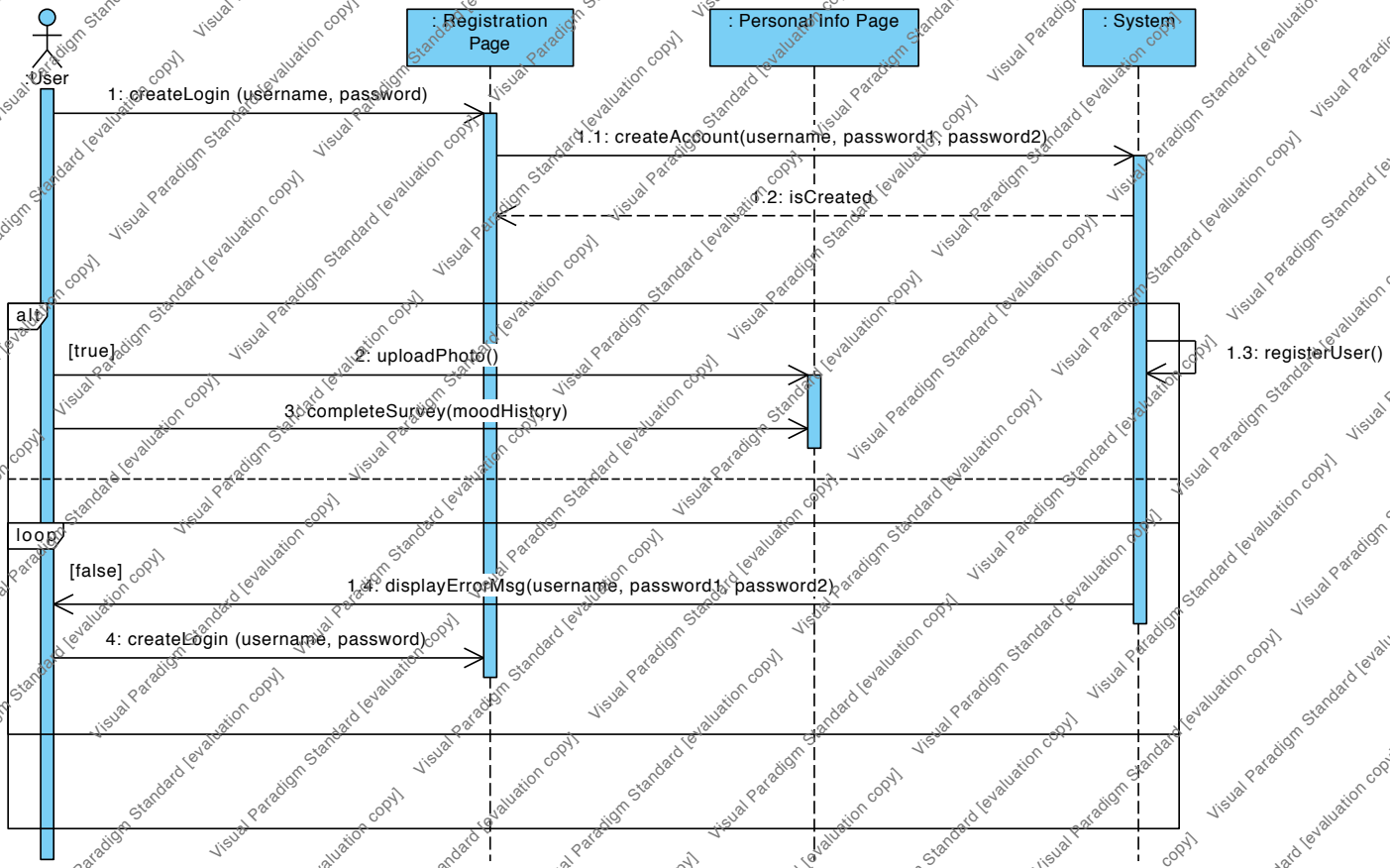
System Sequence Diagrams

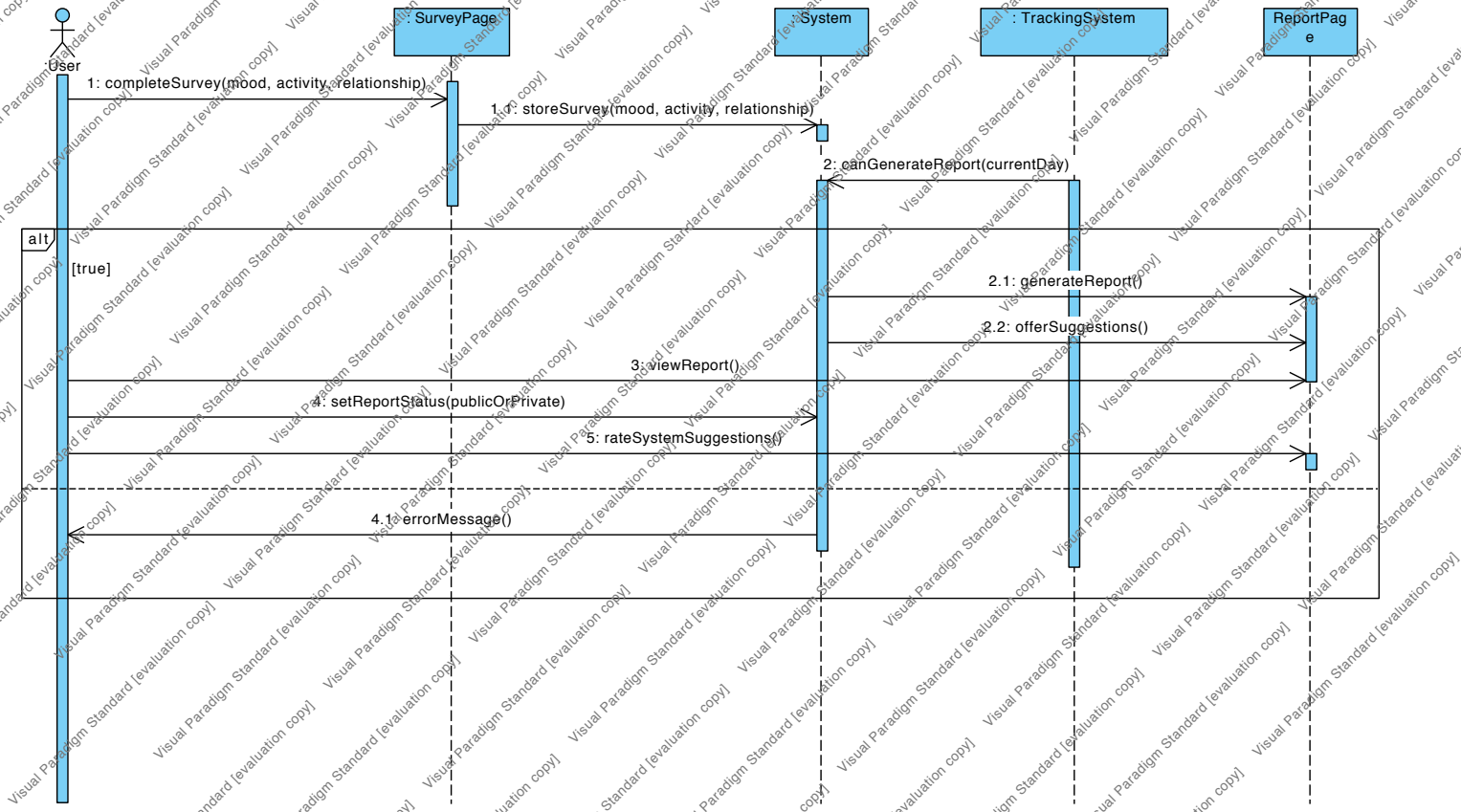

: System
userA :User
userB : user
1: SearchDatabase(username)
1.1: isValidUser(username)
alt
[true]
2: SendRequest()
1.2: notify()
1.3: request
alt
[request = accept]
3: acceptFriend()
3.1: AddFriendNetwork (userA, userB)
3.2: notify(accept)
4: rejectFriend()
4.1: notify(reject)
5: errorMsg(invalidUser)

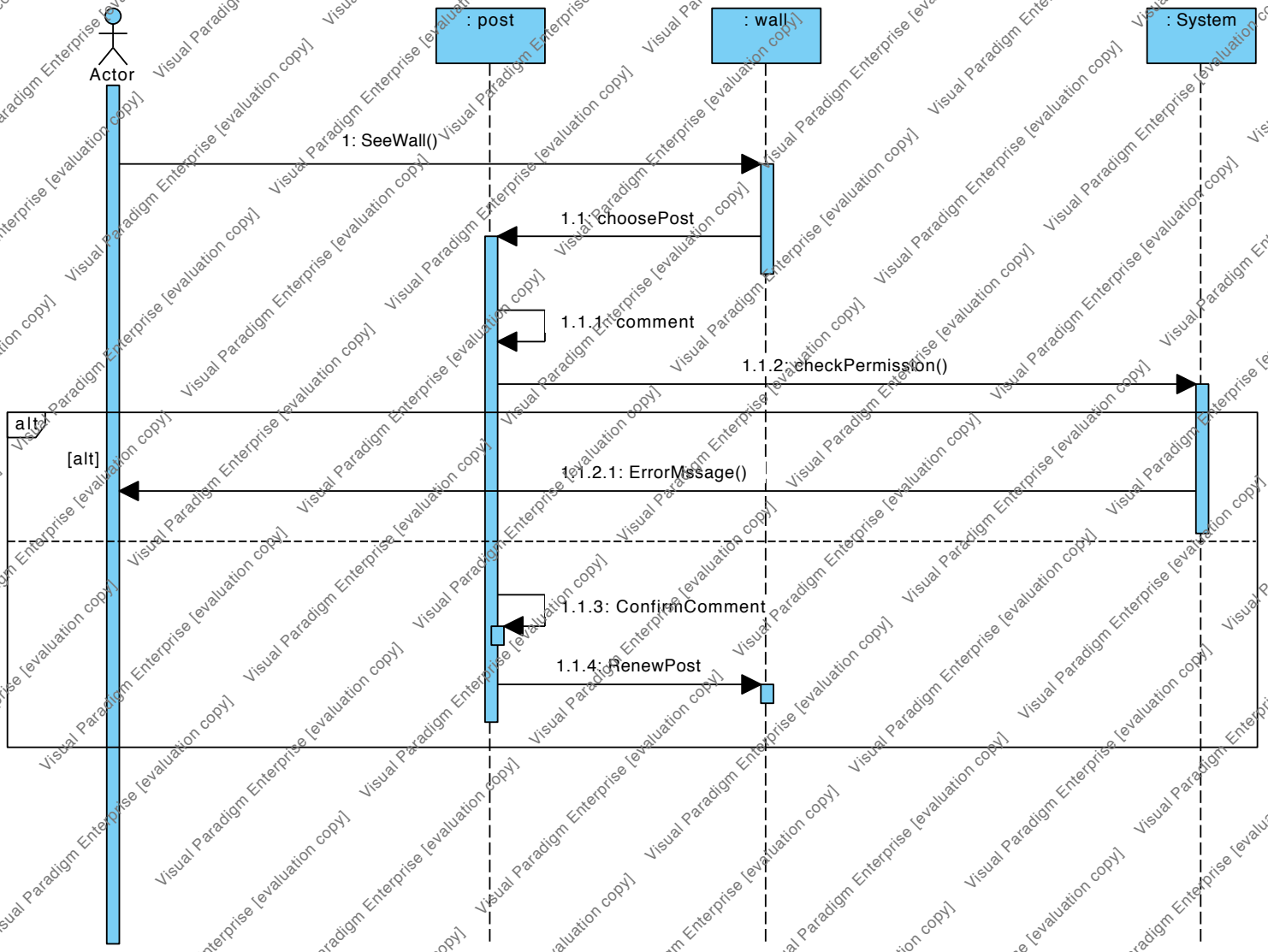

Actor
: CalendarPage
: EditEntryPage
: System
1: editDayInfo()
sd loop
2: selectMapLocation()
3: submitMapData(data)
4: submitReplies(replies)
4.1: updateCalendarPage()

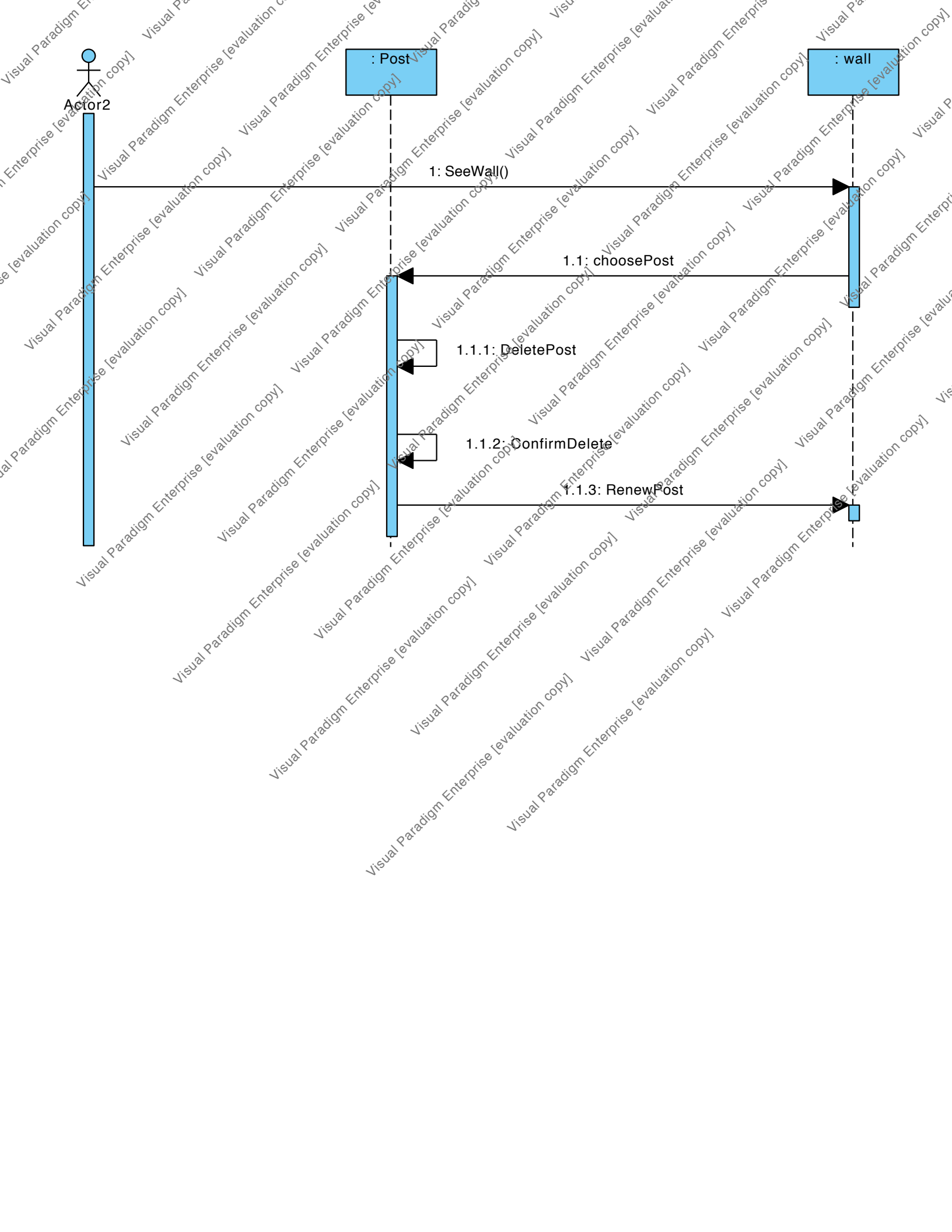

Actor
: CalendarPage
: EditEntryPage
: System
1: editDayInfo()
sd loop
2: askMoodQuestions()
3: submitMoodQuestion(reply)
4: submitReplies(replies)
4.1: updateCalendarPage()

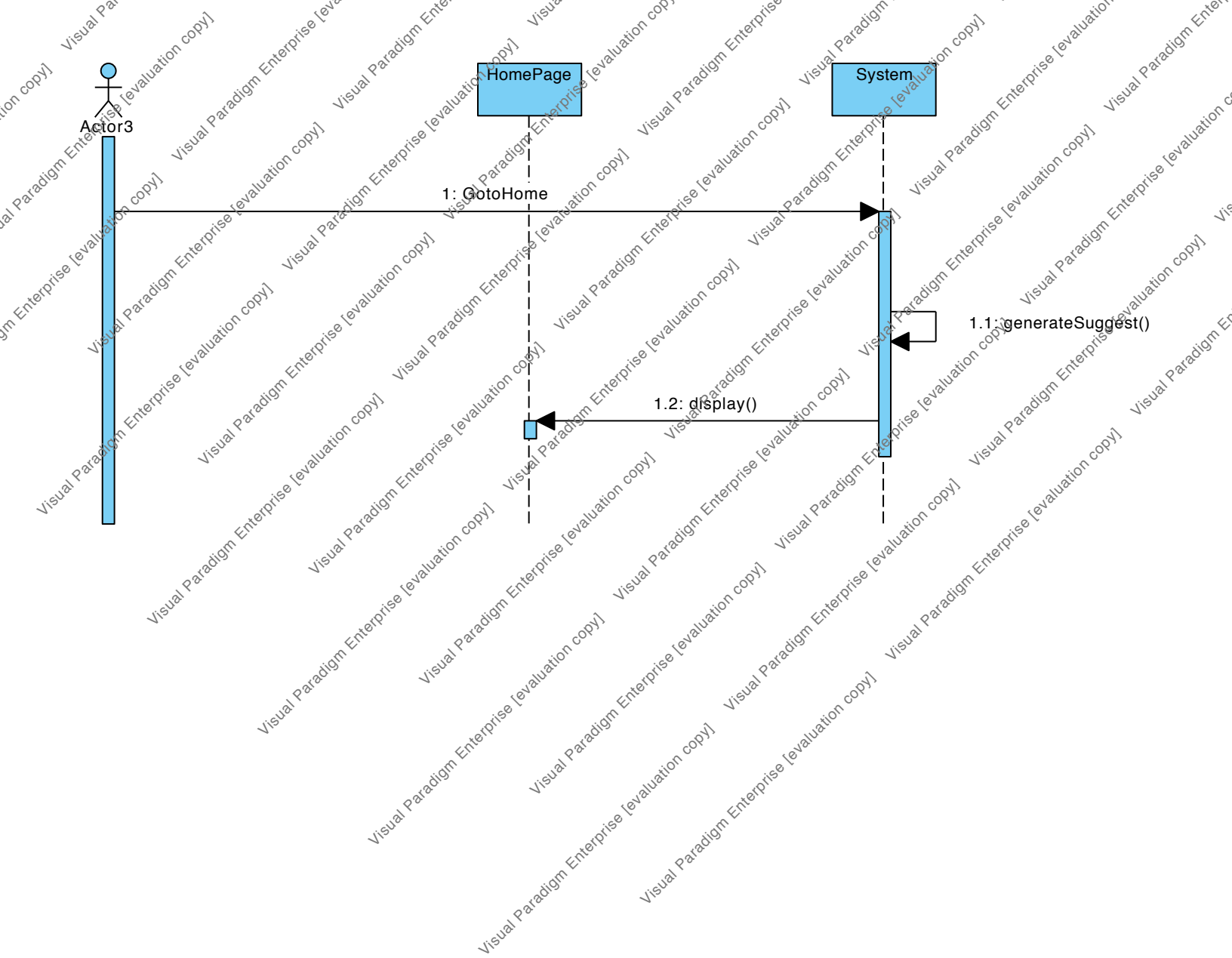


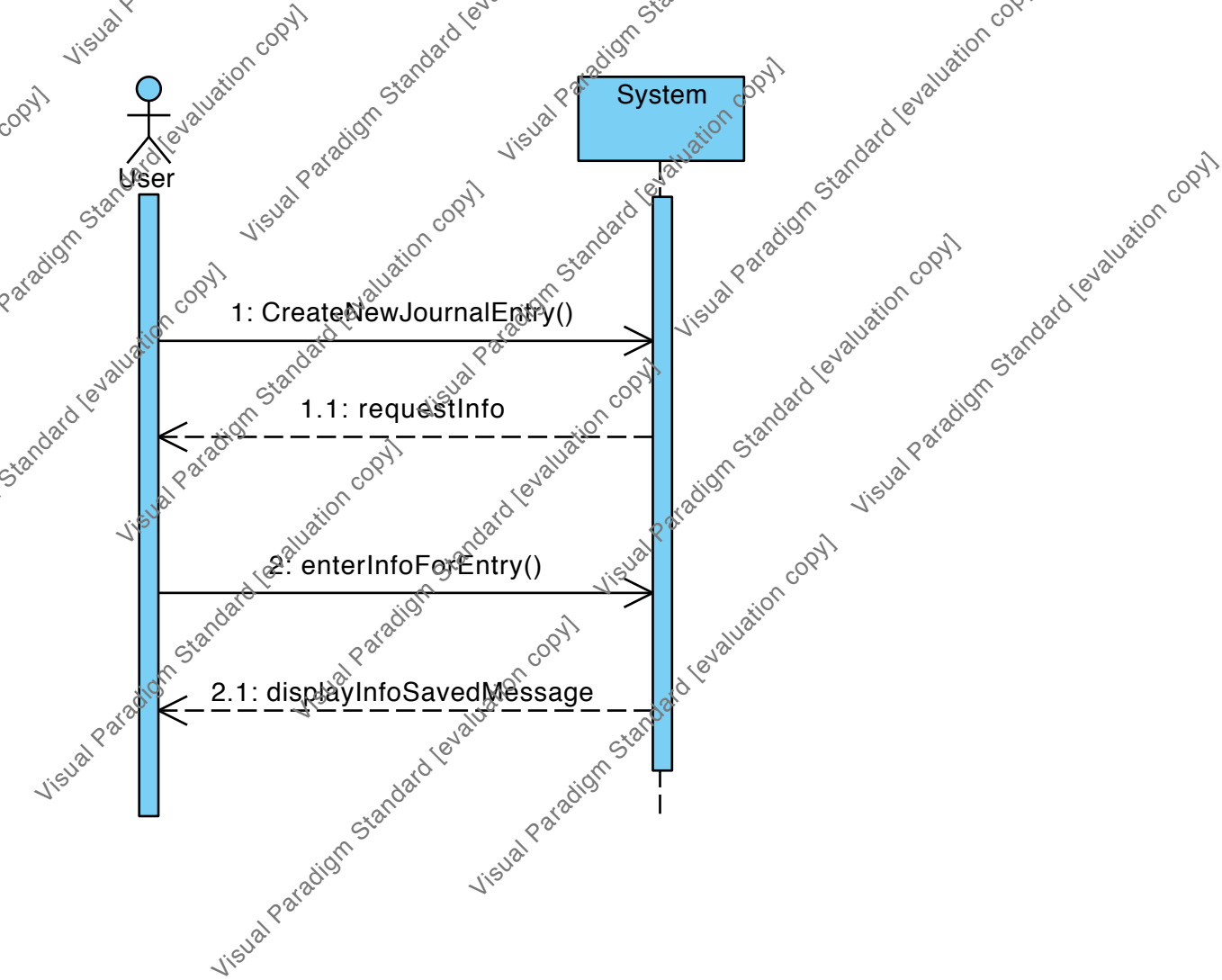

: User
: Registration Page
: Personal Info Page
: System
1: createLogin (username, password)
1.1: createAccount(username, password1, password2)
1.2: isCreated
alt
[true]
2: uploadPhoto()
1.3: registerUser()
3: completeSurvey(moodHistory)
loop
[false]
1.4: displayErrorMsg(username, password1, password2)
4: createLogin (username, password)

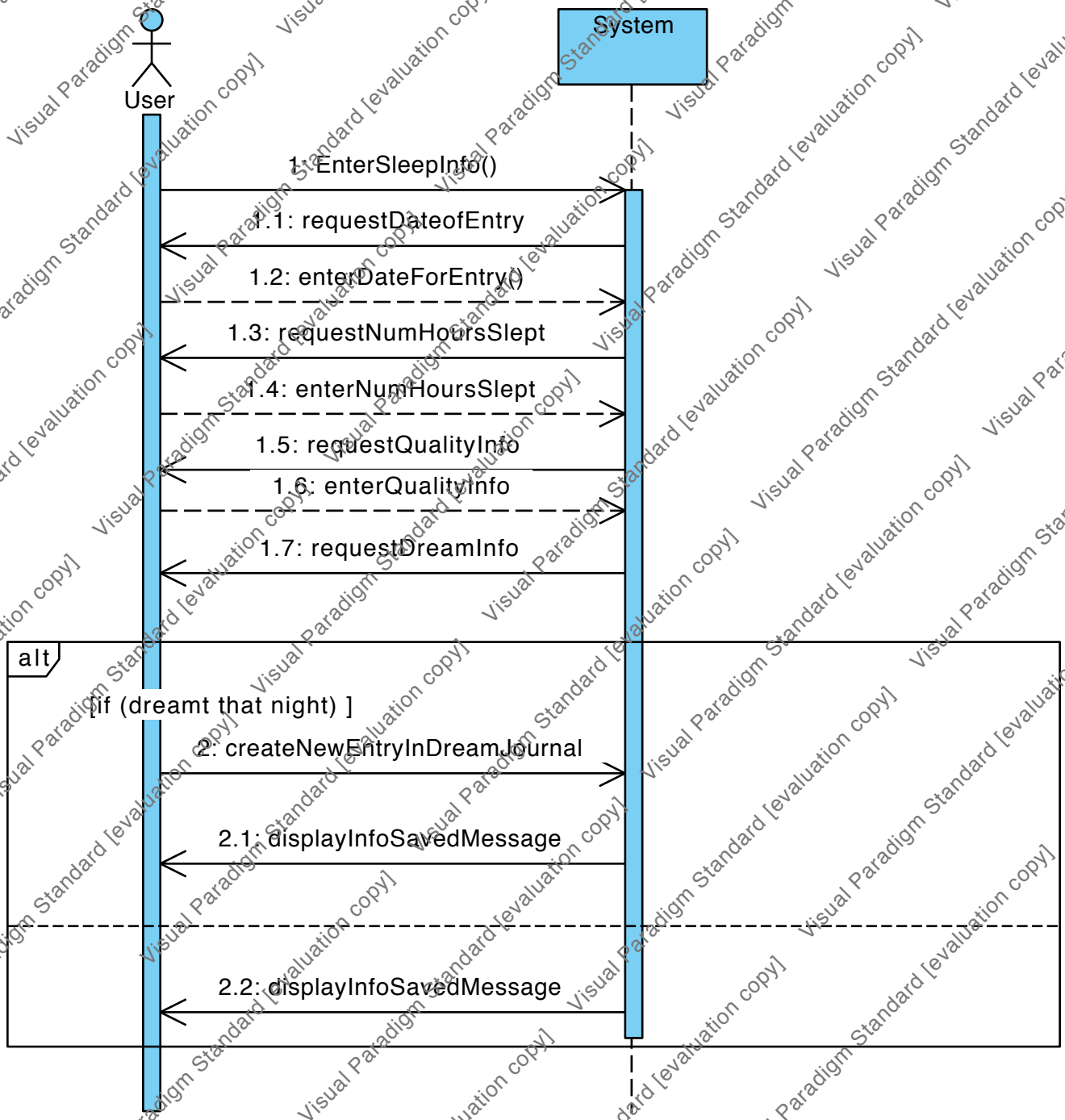

User
: SurveyPage
: System
: TrackingSystem
ReportPage
1: completeSurvey(mood, activity, relationship)
1.1: storeSurvey(mood, activity, relationship)
2: canGenerateReport(currentDay)
alt
[true]
2.1: generateReport()
2.2: offerSuggestions()
3: viewReport()
4: setReportStatus(publicOrPrivate)
5: rateSystemSuggestions()
4.1: errorMessage()

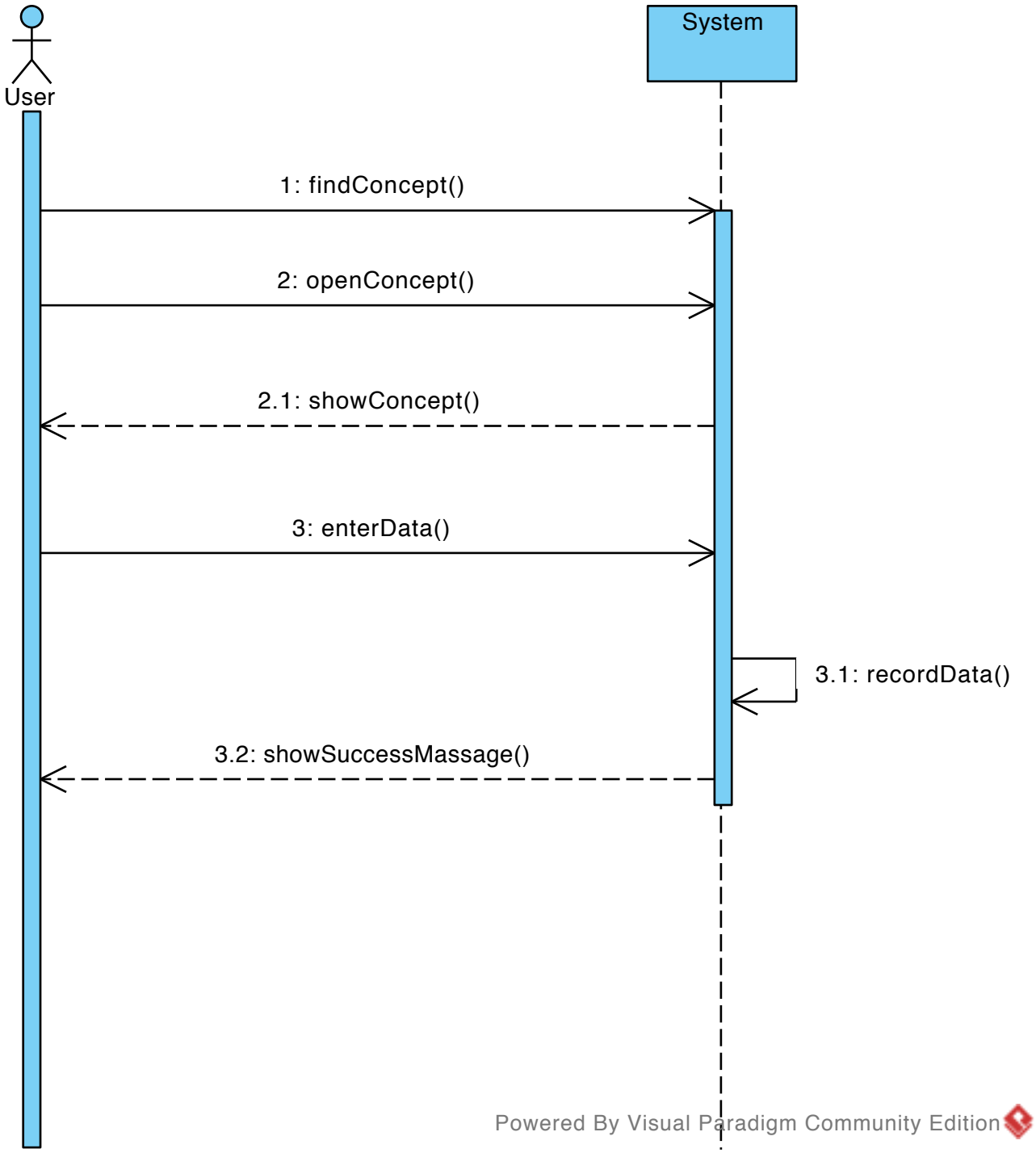

Actor
: post
: wall
: System
1: SeeWall()
1.1: choosePost
1.1.1: comment
1.1.2: checkPermission()
alt
[alt]
1.1.2.1: ErrorMessage()
1.1.3: ConfirmComment
1.1.4: RenewPost

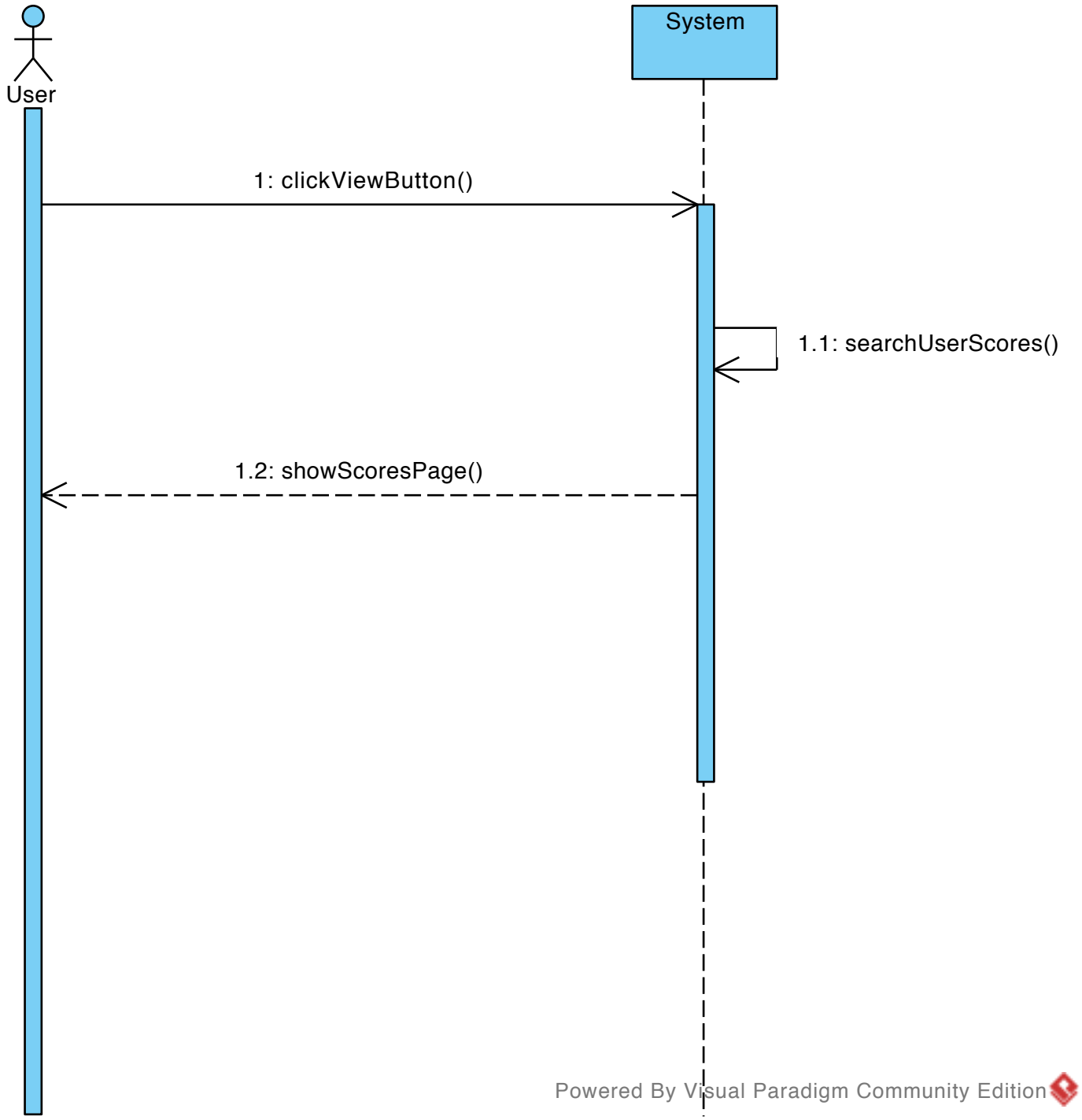

Actor2
: Post
: wall
1: SeeWall()
1.1: choosePost
1.1.1: DeletePost
1.1.2: ConfirmDelete
1.1.3: RenewPost

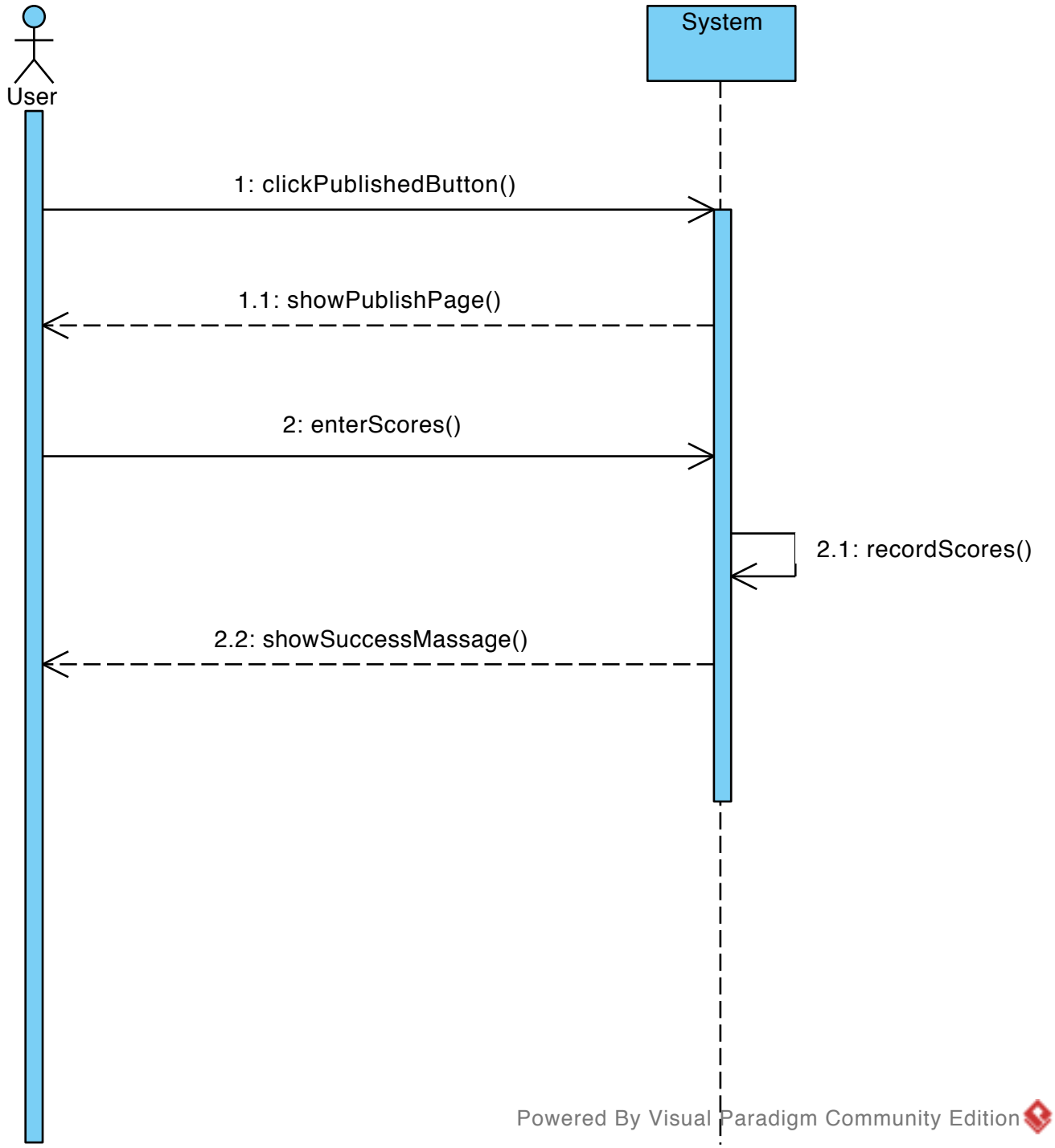

Actor3
HomePage
System
1: GotoHome
1.1: generateSuggest()
1.2: display()

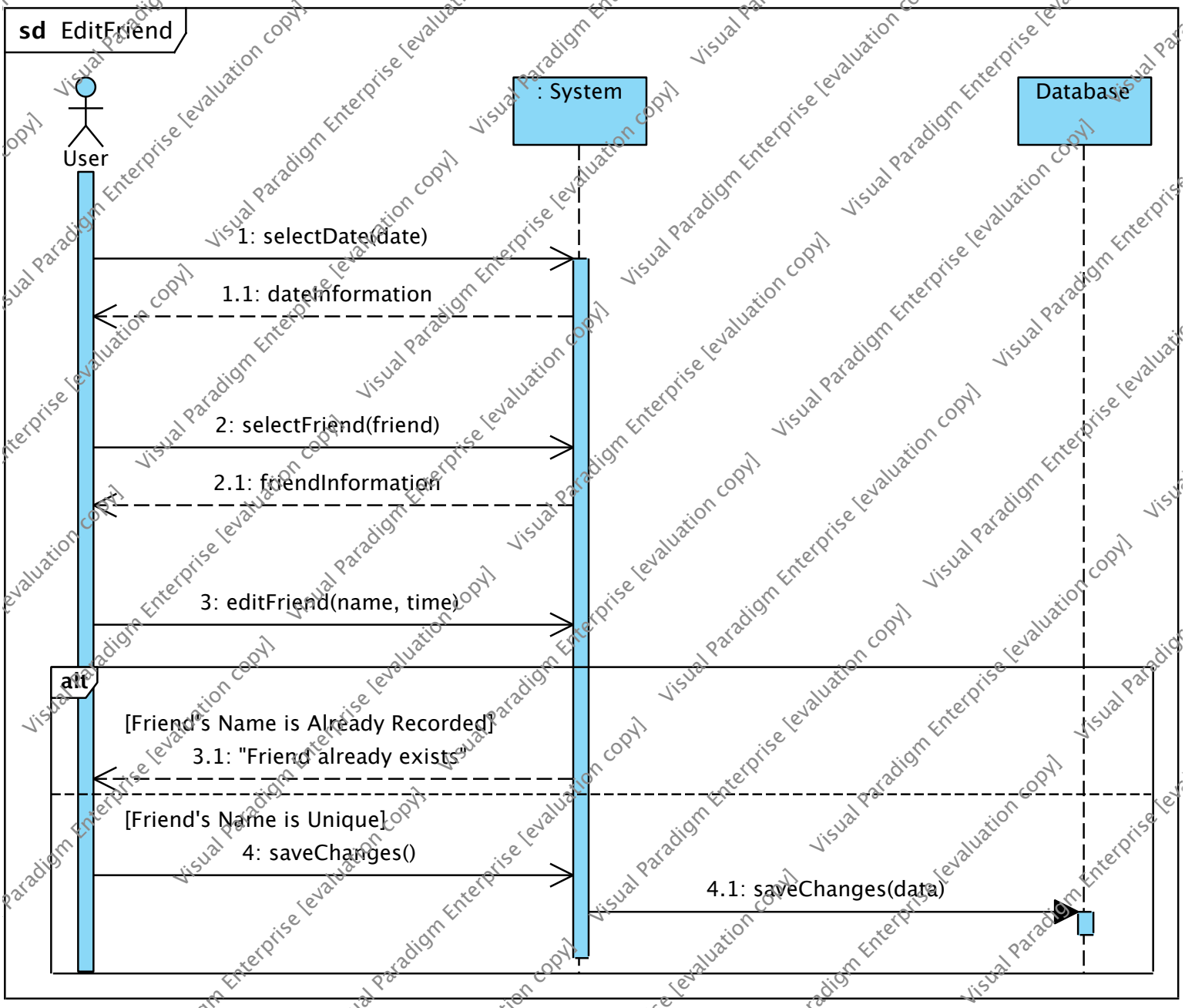

User
System
1: CreateNewJournalEntry()
1.1: requestInfo
2: enterInfoForEntry()
2.1: displayInfoSavedMessage

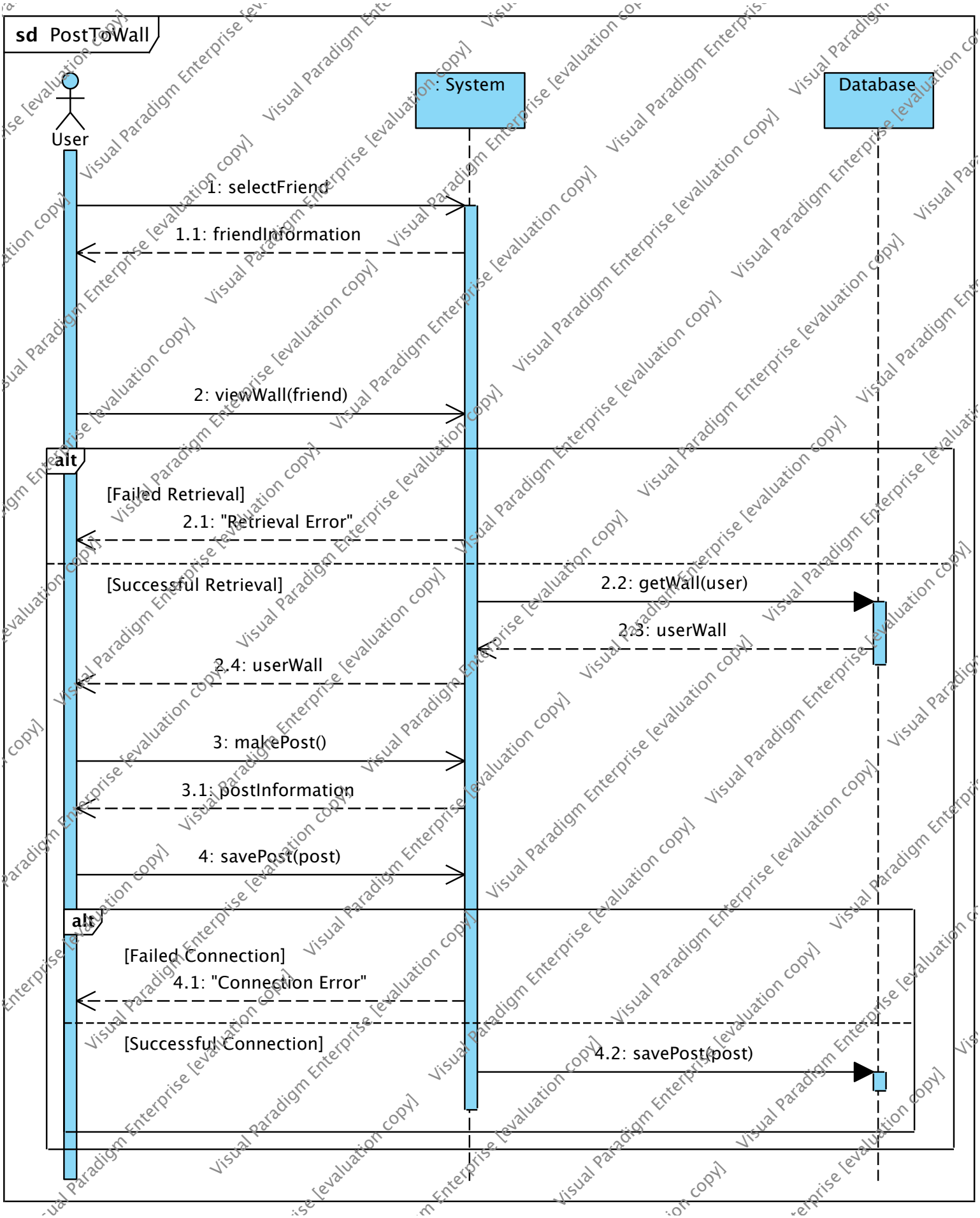

User
System
1: EnterSleepInfo()
1.1: requestDateofEntry
1.2: enterDateForEntry()
1.3: requestNumHoursSlept
1.4: enterNumHoursSlept
1.5: requestQualityInfo
1.6: enterQualityInfo
1.7: requestDreamInfo
alt
[if (dreamt that night)]
2: createNewEntryInDreamJournal
2.1: displayInfoSavedMessage
2.2: displayInfoSavedMessage

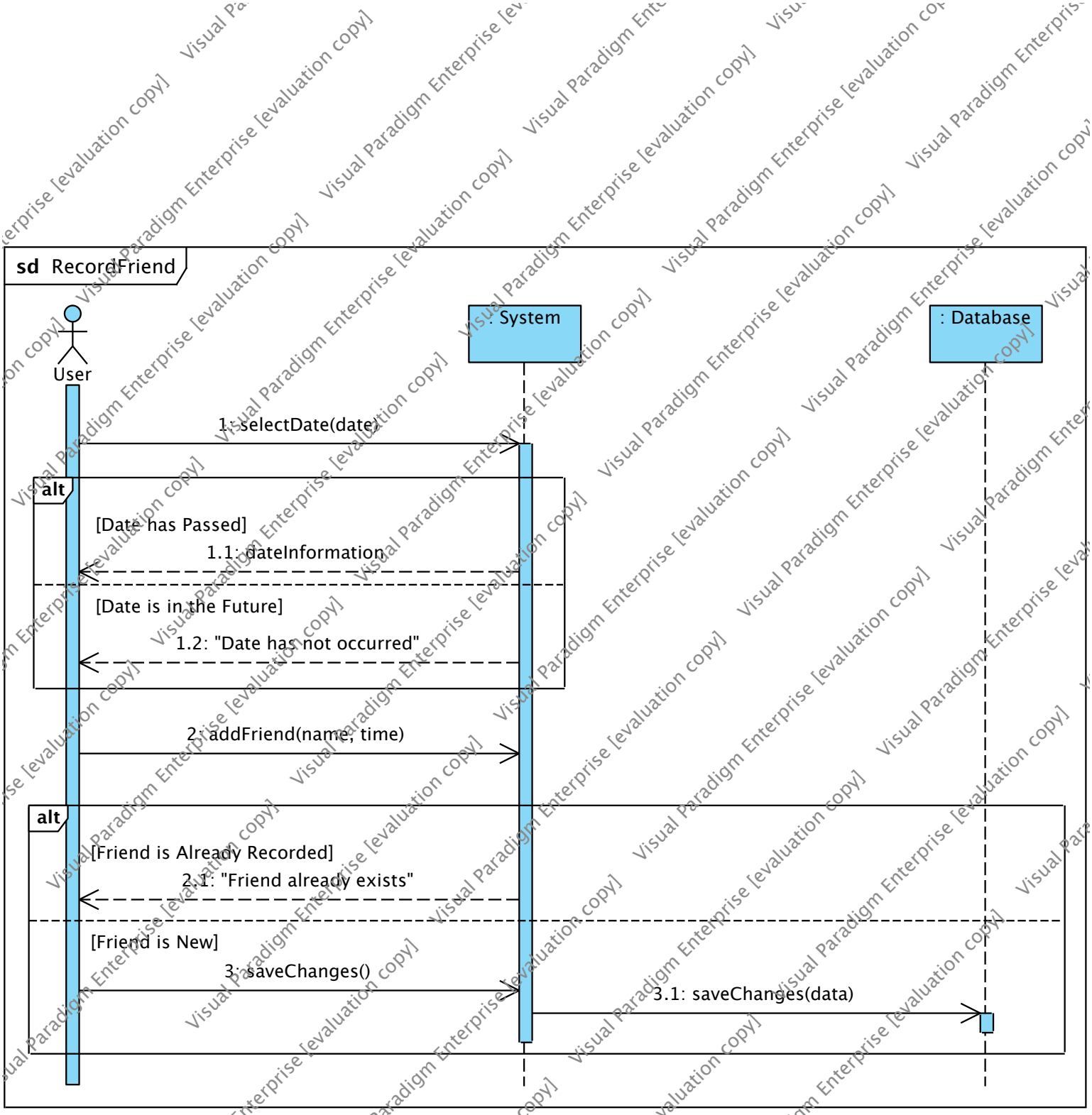

User
System
1: findConcept()
2: openConcept()
2.1: showConcept()
3: enterData()
3.1: recordData()
3.2: showSuccessMassage()
Powered By Visual Paradigm Community Edition

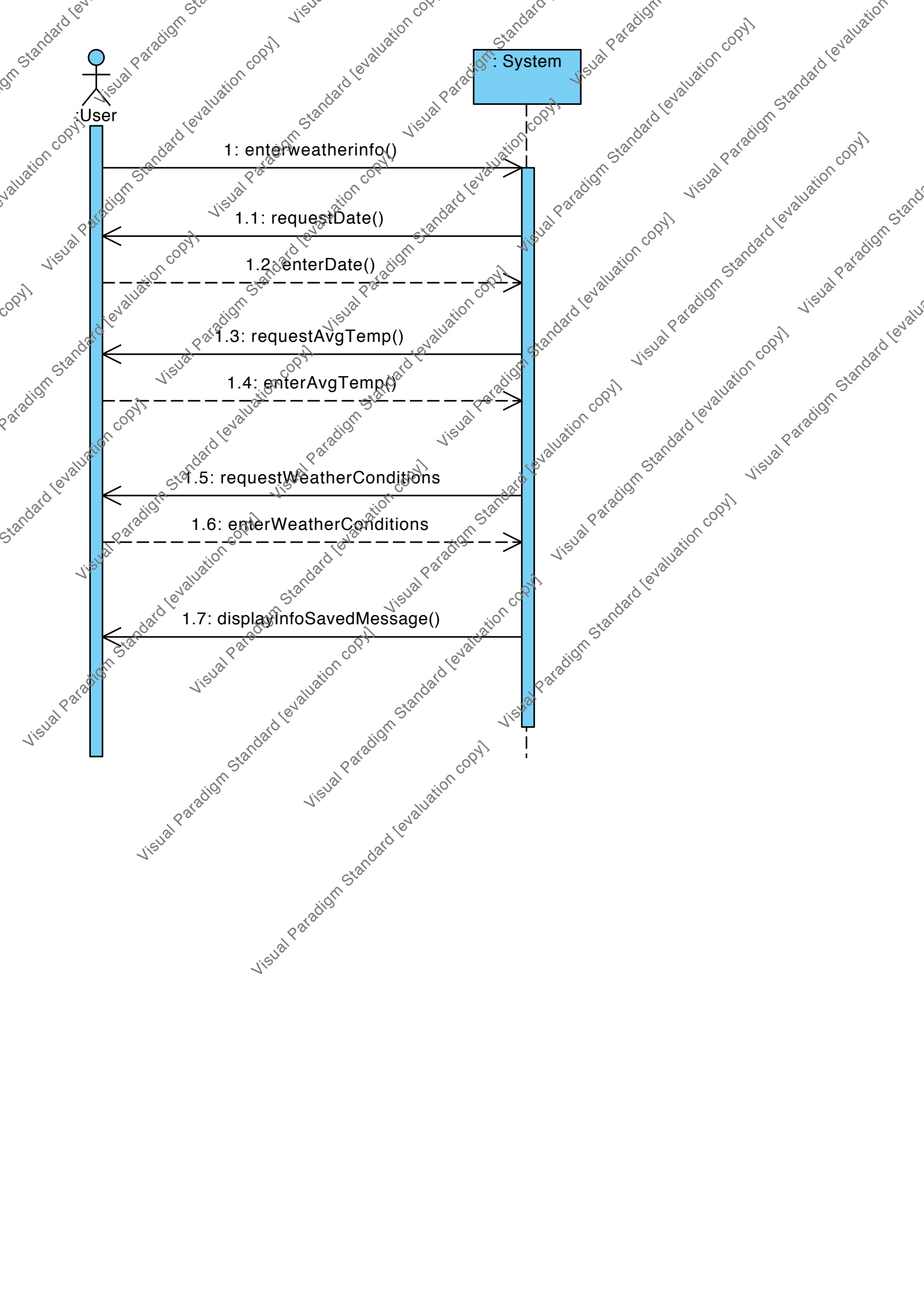

User
System
1: clickViewButton()
1.1: searchUserScores()
1.2: showScoresPage()
Powered By Visual Paradigm Community Edition


User
System
1: clickPublishedButton()
1.1: showPublishPage()
2: enterScores()
2.1: recordScores()
2.2: showSuccessMassage()
Powered By Visual Paradigm Community Edition


sd EditFriend
User
: System
Database
1: selectDate(date)
1.1: dateInformation
2: selectFriend(friend)
2.1: friendInformation
3: editFriend(name, time)
alt
[Friend's Name is Already Recorded]
3.1: "Friend already exists"
[Friend's Name is Unique]
4: saveChanges()
4.1: saveChanges(data)

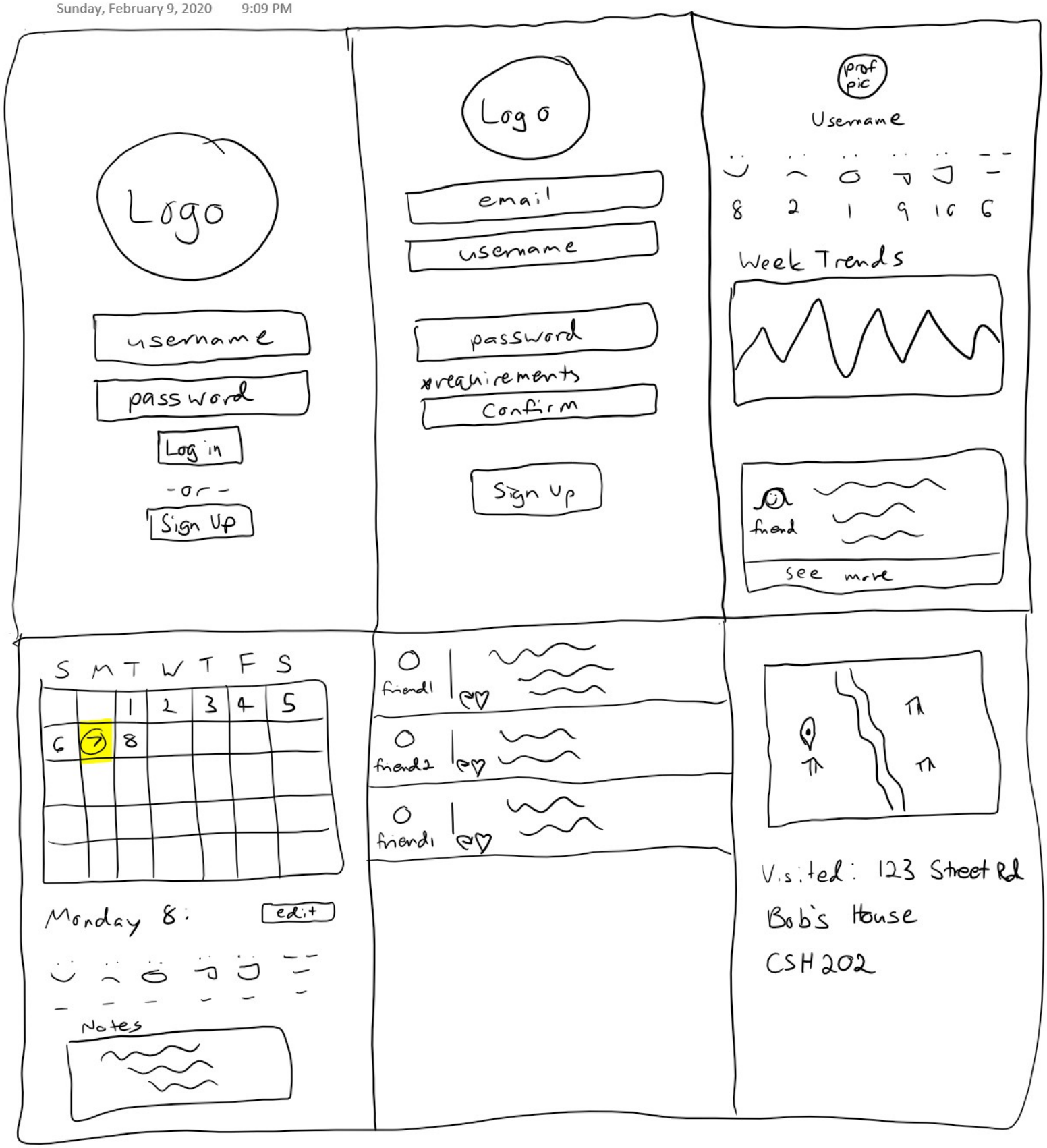

sd PostToWall
User
: System
Database
1: selectFriend
1.1: friendInformation
2: viewWall(friend)
alt
[Failed Retrieval]
2.1: "Retrieval Error"
[Successful Retrieval]
2.2: getWall(user)
2.3: userWall
2.4: userWall
3: makePost()
3.1: postInformation
4: savePost(post)
alt
[Failed Connection]
4.1: "Connection Error"
[Successful Connection]
4.2: savePost(post)


sd RecordFriend
User
: System
: Database
1: selectDate(date)
alt
[Date has Passed]
1.1: dateInformation
[Date is in the Future]
1.2: "Date has not occurred"
2: addFriend(name, time)
alt
[Friend is Already Recorded]
2.1: "Friend already exists"
[Friend is New]
3: saveChanges()
3.1: saveChanges(data)


:User
: System
1: enterweatherinfo()
1.1: requestDate()
1.2: enterDate()
1.3: requestAvgTemp()
1.4: enterAvgTemp()
1.5: requestWeatherConditions
1.6: enterWeatherConditions
1.7: displayInfoSavedMessage()

User Interface Wireframes

Sunday, February 9, 2020 9:09 PM



Teamwork Plan

Teamwork Plan for the: “Happiness Tracker”:

Due Date for everyone’s individual components: 02/06/2020

******As you complete your use cases, please make a note so no duplicates are made. Also please upload to github******

This is all the work for iteration one divided among the group members:

Micah Dadson: Vision, Domain Model, 3 Use Cases, System Sequence Diagrams

Jeremy Meadows: Github, Wireframes, 3 Use Cases, System Sequence Diagrams

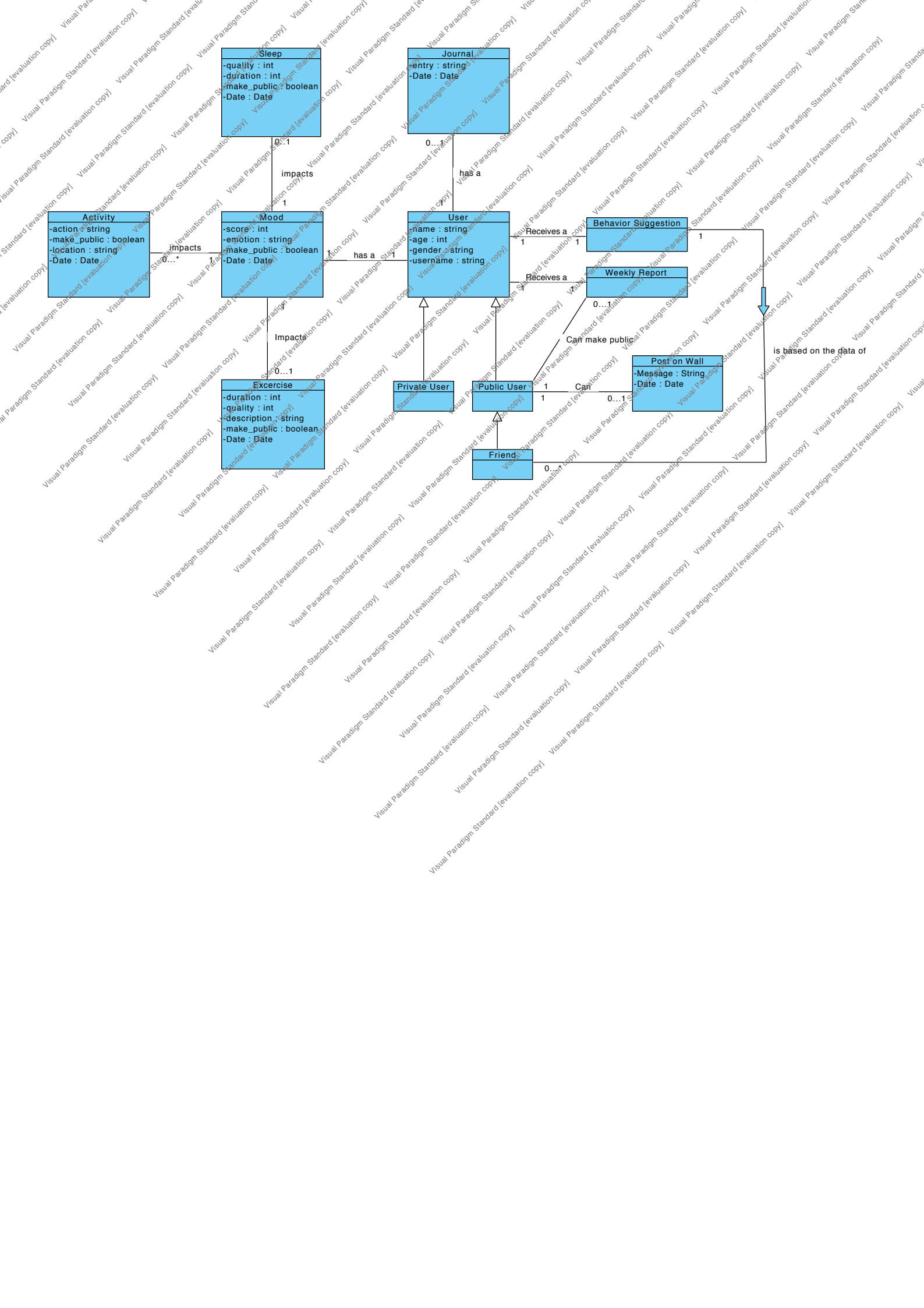
Shivani Bobbala: Nonfunctional Requirements, 3 Use Cases, System Sequence Diagrams

Iris: Functional Requirements, 3 Use Cases, System Sequence Diagrams

Francis: Website, 3 Use Cases, System Sequence Diagrams

Joel Futagawa: 3 Use Cases, System Sequence Diagrams, **Traceability Matrix**

Domain Model


Sleep
-quality : int
-duration : int
-make_public : boolean
-Date : Date
Journal
-entry : string
-Date : Date
Activity
-action : string
-make_public : boolean
-location : string
-Date : Date
Mood
-score : int
-emotion : string
-make_public : boolean
-Date : Date
User
-name : string
-age : int
-gender : string
-username : string
Behavior Suggestion
Weekly Report
Excercise
-duration : int
-quality : int
-description : string
-make_public : boolean
-Date : Date
Private User
Public User
Post on Wall
-Message : String
-Date : Date
Friend
0...1
impacts
1
has a
0...*
Receives a
Can make public
Can
is based on the data of
Impacts

Suggested Points

Redistribution

Points Distribution

Micah Dadson: Vision, Domain Model, 3 Use Cases, 3 System Sequence Diagrams, PowerPoint

COMPLETE

5/5

Joel Futagawa: 3 Use Cases, 3 System Sequence Diagrams, Traceability Matrix

COMPLETE

5/5

Jeremy Meadows: Github, Wireframes, 3 Use Cases, 3 System Sequence Diagrams

COMPLETE

5/5

Iris: Functional Requirements, 3 Use Cases, 3 System Sequence Diagrams

COMPLETE

5/5

Shivani Bobbala: Nonfunctional Requirements, 3 Use Cases, System Sequence Diagrams

COMPLETE

5/5

Francis- Website, 3 Use Cases, System Sequence Diagrams

COMPLETE

5/5

TimeSheet

	January 31	02/01/2020	02/02/2020	02/03/2020	02/04/2020	02/05/2020	02/06/2020	02/07/2020	02/08/2020	02/09/2020	02/10/2020
Fransis											
Iris										1 hours	2 hours
Jeremy							4 hours				
Joel					4 hours						
Micah	2 hours		2 hours					30 minutes		30 minutes	1.5 hours
Shivani										half hour	2 hours