



SOFTWARE ENGINEERING

CSC 4350 Spring 2018

Group Name: Renegon Bits

Jason Herrera, Brian Cabigon, Braxton McLean, Chris Scott, Nishant Sinha

Planning and Scheduling

Outline Plan:

For the Requirements and System Modeling of this product, we first must reformat our requirements to make them more specific and detailed for the use cases. Then, after the requirements have been reformatted, we will create our use cases based on the requirements with details to make the object modeling and test case identification easier. After all the use cases have been completed, we will create the test cases and object models simultaneously as neither one depends on the other. And finally, after the object modeling is finished, we will create the behavior models.

The most serious challenge we see is allowing users to identify animals and query the database for specific animals. A risk for this project that we have already encountered is having the idea fail. We had an idea before this and realized that we could not expand on ideas that already existed. We had to start from scratch and create a new idea which put us behind schedule and made us move more quickly to catch up. Another risk is that the scope of our project may be too large. We may have to scale back our project and start with only local shelters, small communities, and small amounts of data. Then, over time, the software will grow, and we could add support for larger areas, interact with more shelters and identify animals in more regions.

Scheduling:

Reporting and Monitoring:

Updates will be provided halfway through the duration of the task. Then, they more updates will be given based on status and completion of the task.

Tasks:

Task	Effort	Duration	Dependencies	People
Requirements (T1)	1 person-day	1 day	None	Jason
Use Cases (T2)	10 person-days	2 days	T1	Jason, Braxton, Brian, Chris, Nishant
Test Cases (T3)	4 person-days	1 day	T1, T2	Braxton, Brian, Chris, Nishant
Object Modeling (T4)	1 person-day	1 day	T2	Jason
Behavioral Modeling	1 person-day	1 day	T2	Jason

Requirements

Requirements List:

1. The system shall store user information including name, email, geolocation, username, and password.
2. The system shall store animal information including images, name, species, and features.
3. The system shall allow users to create a new account.
 - 3.1. If all fields are completed, and all entries are valid, account shall be created.
 - 3.2. Error message shall be displayed if any entries are invalid.
 - 3.3. Error message shall be displayed if any fields are empty.
4. The system shall validate login information of users.
 - 4.1. If correct username and password are entered, allow access to system.
 - 4.2. If username or password are incorrect, display error message.
 - 4.3. Allow up to 5 incorrect accesses within 5 minutes from one IP address before disallowing access from that IP address for 15 minutes.
5. The system shall display information queried from the database.
 - 5.1. If query is valid, and results are found, display results.
 - 5.2. If query is invalid, display error message.
 - 5.3. If no results found for query, display error message.
6. The system shall allow users to upload an image to the database with a response form attached with information about the animal.
 - 6.1. If image file is valid, and response form is filled, entry shall be created.
 - 6.2. Error message shall be displayed if any image file is invalid.
 - 6.3. Error message shall be displayed if response form is invalid.
7. The system shall allow users to post onto the Feed.
 - 7.1. If message is valid, post to Feed.
 - 7.2. If message is invalid, display error message.
8. The system shall allow users to share direct links to posts.
 - 8.1. If linked post exists, link is shared.
 - 8.2. If linked post does not exist, display error message.
9. The system shall allow recognized organizations to collect a datasheet of posts in their defined geolocation work area.
 - 9.1. If the user is a recognized organization, and the defined geolocation work area has posts, distribute datasheet.
 - 9.2. If user is not a recognized organization account, display error message.
 - 9.3. If geolocation work area has no posts, display error message.
10. The system shall allow recognized organization to post advertisements for animals in the shelter.
 - 10.1. If user is a recognized organization, post advertisement.
 - 10.2. If user is not a recognized organization, display error message.
11. The system shall allow administrators to delete posts made by other users.
 - 11.1. If user is an administrator and post exists, delete post.

- 11.2. If user is not an administrator, display error message.
- 11.3. If post does not exist display error message.
- 12. The system shall allow administrators to delete entries in the database.
 - 12.1. If user is an administrator and entry exists, delete entry.
 - 12.2. If user is not an administrator, display error message.
 - 12.3. If entry does not exist, display error message.
- 13. The system shall allow administrators to delete user accounts.
 - 13.1. If user is an administrator and account exists, delete account.
 - 13.2. If user is not an administrator, display error message.
 - 13.3. If account does not exist, display error message.

Use Cases

Use Case 1

Summary: Login to system using account information.

Actors: Users and account database

Basic Course of Events:

1. User accesses the application.
2. The application prompts the user to log in to access the features of the application.
3. The user inputs their username and password into their respective fields.
4. The application connects to the database and determines if the account exists.
5. If the account exists, the application takes the account username and verifies that the input password matches the password in the database.
6. If the password and account match correctly to the database, the application logs the user in.

Alternative Paths: None. The user cannot access any other paths without logging in first.

Exception Paths: If the user's account does not exist, the application will prompt the user to enter a username and password to be added to the database. Once the user inputs a username and password, it is entered into the database and they can proceed with steps 2-6.

Trigger: For the user to interact with the application, they must first log in.

Assumptions: The user must already have an account or be willing to create an account with a username and password into the database.

Precondition: The user must have the application and an account.

Postcondition: The user is logged into their account and can access more features.

Use Case 2

Summary: Check geolocation of user

Actors: Users and application.

Basic Course of Events:

1. User logs into their account.
2. The application determines the user's location automatically.

3. The user's location is updated in the database.

Alternative Paths: If the system cannot determine a person's location automatically, the user can input a location manually.

Exception Paths:

Trigger: To determine what relevant posts to show based on geolocation, the system needs to know the user's geolocation. This system would accomplish this automatically.

Assumptions: The user has an account and is logged in

Precondition: The user must have the application and an account.

Postcondition: The system has the geolocation of the user to display a feed of posts that match their geolocation.

Use Case 3

Summary: The system shall be able to query the database.

Basic Course of Events:

1. The system sends a query to the database.
2. The database checks whether the query has correct permissions to perform the database operation.
3. If the query is well-formed and the query was sent with correct permissions, the database returns the queried information.
4. If the query is not well-formed or the query was not sent with correct permissions, the database returns an error.

Alternative Paths: None.

Exception Paths: If the database has too many queries to handle, the query may be rejected.

Precondition:

- The query was sent with correct access permissions for the database operation.
- The database contains information for the given query.

Postcondition: The query receives a response with information from the database.

Use Case 4

Summary: The system shall display information queried from the database.

Basic Course of Events:

1. The database is queried and correct information is returned.
2. The response from the database is displayed by the system.

Alternative Paths: None

Exception Paths: If incorrect information is returned by the database (wrong encoding), the information may not appear correctly.

Precondition:

- A well-formed query with correct permissions is sent to the database.
- The database returns the correct information.

Postcondition: The queried information is displayed on the system

Use Case 5

Summary: The system shall allow users to post onto the Feed.

Basic Course of Events:

1. A user with correct permissions posts a well-formed post to the Feed.
2. The post is sent to the database and the database returns a response to the query that it was correctly received
3. The post shows up on the Feed.

Alternative Paths: None.

Exception Paths:

- If the database does not correctly receive the post in step 2, it will not show up in the Feed.
- If the user's post is not well-formed, the system displays an error.

Precondition:

- The user is logged in and has permissions to post to the Feed.

Postcondition: The Feed now shows the user's post.

Use Case 6

Summary: Display posts made by other users in the Feed

Basic Course of Events:

1. Completion of use case *Check geolocation of user*.
2. User selects the option to view the Feed.
3. The system finds posts made by users in the same geolocation.
4. If there are posts made by users in the geolocation, those posts are displayed

Alternative Paths: At step 3, the user can opt to enter a geolocation manually instead and execution proceeds to step 4.

Exception Paths: If there are no posts in the geolocation, the system will display a message that says there were no posts found in place of step 4.

Precondition: The user must have a geolocation.

Postcondition: Posts have been displayed for the user.

Use Case 7

Summary: Share direct link viewable by others

Basic Course of Events:

1. Complete use case *Login to system using account information*.
2. Complete use case *Display posts made by other users in the Feed*.
3. User selects option to share a post made by another user.
4. The system generates a link for that post.
5. The system provides the link to the user.
6. User is prompted to copy the link.
7. The system returns the user to the Feed.

Alternative Paths: At steps 5 and 6, the user can choose to cancel this operation and go to step 7.
Exception Paths: If post does not exist, an error message is displayed and execution proceeds to step 7.

Precondition: Use case login to system using account information was successful and use case display posts made by other users in the Feed was successful.

Postcondition: Link is provided to user.

Use Case 8

Summary: Collect datasheet of posts in defined geolocation work area

Basic Course of Events:

1. Complete use case *login to system using account information* with credentials of a recognized organization
2. Complete use case *check geolocation of user*
3. User selects option to collect datasheet of posts.
4. The system finds the posts made by users in the geolocation.
5. The system creates a file with the posts.
6. System provides a preview of the file.
7. User selects the download option.
8. User is returned from the datasheet collection.

Alternative Paths: At step 3, user can manually enter a geolocation and go to step 4. At step 6, the user can cancel this operation and go to step 8.

Exception Paths: If there are no posts in the geolocation, an error message is displayed and execution proceeds to step 8.

Precondition: Use case login to system using account information was successful and use case check geolocation of user is successful.

Postcondition: Datasheet is provided to the user.

Use Case 9

Summary: Post advertisements for animals in shelter in Feed.

Basic Course of Events:

1. Complete use case *Login to system using account information*.
2. Complete use case *Display posts made by other users in the Feed*.
3. User selects option to make a post (authorized as an organization in step 1).
4. The system provides a response prompt like use case 3
 - a. Information, image upload, additional details, personal blurb, etc.
5. Repeat set 4 as prompted until the user selects finish.
6. Query the user about how many hours/days they wish the ad to run.
7. Confirms selection, and uploads information to database.
8. The system returns the user to the Feed.

Alternative Paths: At steps 3 through 7, the user can choose to cancel this operation and go to step 2.

Exception Paths: If inaccurate/unacceptable date ranges are entered at step 6, they are prompted to enter the information again/differently.

Precondition: Use case login to system using account information was successful and use case display posts made by other users in the Feed was successful.

Postcondition: Ads will run against Feed spots upon approval.

Use Case 10

Summary: Toggle geolocation filters

Basic Course of Events:

1. Complete use case *Login to system using account information*.
2. Complete use case *Display posts made by other users in the Feed*.
3. User navigates to the Settings of the app.
4. The User toggles on/off geolocation filtering
5. The user may navigate away from the Settings.

Alternative Paths: None

Exception Paths: If the device's location services are disabled/unavailable, prompt the user to enable them via their phone's settings.

Precondition: Use case login to system using account information was successful and use case display posts made by other users in the Feed was successful.

Postcondition: After step 4, if the option is toggled on, the Feed contents will be more weighted to display nearby posts. If disabled, posts based on the initial state entry will be displayed.

Use Case 11

Summary: Delete posts made by other users.

Basic Course of Events:

1. Complete use case *Login to system using account information*.
2. Complete use case *Display posts made by other users in the Feed*.
3. User selects a post made by another user in the Feed.
4. The user selects the post options button
5. The user selects the option to delete post.
6. The system returns the user to the Feed.

Alternative Paths: At steps 3 - 5, the user can choose to cancel this operation and go to step 6.

Exception Paths: If user is not authorized as an administrator, Step 5 will return an error warning the user they are not authorized.

Precondition: Use case login to system using account information was successful and use case display posts made by other users in the Feed was successful.

Postcondition: The post is deleted and not viewable by future users.

Use Case 12

Summary: Remove Entries in Database

Basic Course of Events:

1. The system sends a query to the database
2. The database checks whether the client has correct permissions to perform the query.

3. If the query is sent with permissions and is a correct query, then the database entry that aligns with the requirements of the query will be removed and the client will receive an OK.
4. If the query is sent without permissions or is an erroneous query, then the client will receive an error and nothing in the database will be removed.

Alternative Paths: None

Exception Paths: If the database experiences excessive traffic, the query may be rejected.

Precondition:

- The query was sent with permissions to perform a database operation
- The database contains information that aligns with the requirements of the query.

Postcondition: The client receives an OK and the requested entry in the database will be removed.

Use Case 13

Summary: Remove Accounts in System

Basic Course of Events:

1. A user with the proper permissions will log in to the system and go to their settings where they can remove their account.
2. They will select a button in order to remove their account
3. The user will be prompted that their account will be non-recoverable if they proceed
4. If they proceed, query the database and remove their user account entry
5. If they do not proceed, return back to the settings screen.

Alternative Paths: None

Exception Paths: If the database experiences excessive traffic, the query may be rejected.

Precondition:

- The user is logged in with proper permissions to access their settings
- The query was sent with permissions to perform a database operation
- The database contains information that aligns with the requirements of the query.

Postcondition: The client's account and information will be removed from the database and the user will be led back to the home page.

Use Case 14

Summary: Delete advertisements made by organization

Basic Course of Events:

1. A user with the proper permissions will log in to the system and go to an interface that will allow for the removal of posts that they deem as non-genuine advertisement posts.
2. They will select a post that they deem as a non-genuine advertisement post
3. They will select an option to delete the said post.
4. They will be prompted a confirmation for deleting the post
5. If they proceed, the post's database entry will be removed
6. If they do not proceed, the post's database entry will not be removed and the user will be returned to the options page.

Alternative Paths: None

Exception Paths: If the database experiences excessive traffic, the query may be rejected.

Precondition:

- The user is logged in with proper permissions to access their settings
- The query was sent with permissions to perform a database operation
- The database contains information that aligns with the requirements of the query

Postcondition: The post will be removed from the database and the user will be led back to the list of posts.

Test Cases

Requirement ID: 4

Requirement Description: The system should be able to query the database.

Rationale: To ensure that the database is atomic and will always either return a correct response or an error.

Inputs:

- A well-formed query.
- An ill-formed query.

Outputs:

- The correct information from the query.
- An error.

Persistent Changes:

- If the query to the database adds a state or updates a state, the database must then contain that information.

Related Requirements and Use Cases: 5, 6

Test Cases:

- Send both well-formed and ill-formed queries to the database and then check the database to ensure any persistent changes remain.

Creator: Nishant Sinha

Requirement ID: 5

Requirement Description: The system shall display information queried from the database.

Rationale: To ensure the queried information from the database is visible.

Inputs:

- A search term. E.g. dogs, cats, spotted.

Outputs:

- A list of posts relevant to the search terms.

Persistent Changes: None.

Related Requirements and Use Cases: 4

Test Cases:

- Different search terms.
- Cats, dogs, spotted, golden retriever.

Creator: Nishant Sinha

Requirement ID: 6

Requirement Description: The system shall allow users to post on the Feed.

Rationale: So that users can engage with posts and view other user's content

Inputs: None. The user should be able to view the feed as soon as they are logged in.

Outputs: All posts in a scrollable container.

Persistent Changes: None.

Related Requirements and Use Cases: 5

Test Cases:

- Viewing the system not logged in.
- Viewing the system as a user properly logged in.

Creator: Nishant Sinha

Requirement ID: 7

Written by: Braxton McLean

Requirement Description: Post to Feed as user

Rationale: This is the source of the majority of the application's user content, to allow users to post sightings/information about lost animals.

Inputs:

1. The user navigates to the home/main page of the application
2. The user presses a "make post" button
3. A form requesting additional information will appear, including photo, description of animal, location, and time (required), as well as a text description
4. A confirmation notice of post submission

Outputs: The application will send the post data to the backend where it will be processed and added to the database.

Persistent Changes: The post data will be added to the database for querying and appearing in Feeds.

Related Requirements and Use Cases: Requirements 4, 5, 7 - Uses 1, 3, 6.

Requirement ID: 8

Requirement Description: Share direct link to posts viewable by others

Rationale: This requirement was to allow for users to share posts with other people in order to spread information of lost animals that they may be familiar with.

Inputs:

The user navigates to a post that exists in the database

They will be able to copy a link that leads directly to the post so that the post may be viewable for others

Outputs: The output will be a link that they can give to others to allow them to view the post.

Persistent Changes: None

Related Requirements and Use Cases: Requirement 4, 8

Test Cases: 5

Requirement ID: 9

Requirement Description: Allow recognized organizations to collect a datasheet of posts in their defined geolocation work area

Rationale: This requirement was to allow organizations to get a log of the animals that have been seen in their immediate area and they can work with people to see who the animal belongs to as well as who they can return it to if they find it among their animals.

Inputs:

The organization user navigates to the Feed

The organization user will click a button

The organization user will then download a datasheet of posts written for the area close to the organization

Outputs: The output will be a datasheet of posts in their defined geolocation work area

Persistent Changes: Persistent Changes

Related Requirements and Use Cases: Requirement 4, 5, 9

Test Cases: 5

Requirement ID: 10

Written by: Braxton McLean

Requirement Description: Shelters can post advertisements for animals in shelters to be displayed to other users.

Rationale: This is the source of the majority of the application's user content, to allow users to post sightings/information about lost animals.

Inputs:

1. The user navigates to the home/main page of the application
2. The user, authorized as a shelter presses the "make post/advertisement" button
3. A form requesting additional information will appear, including photo, description of animal, shelter name and locations well as a text description.
4. Repeat steps 3 until the user selects "finish"
5. Prompt for a duration with a specified maximum and minimum.
6. A confirmation notice of post submission

Outputs: The application will send the post data to the backend where it will be processed and added to the database as advertisements to be displayed to users.

Persistent Changes: The advertisement data will be added to the database for querying and appearing in Feeds.

Related Requirements and Use Cases: Requirements 4, 5, 10 - Uses 1, 2, 3, 7, 10.

Requirement ID: 11

Written by: Braxton McLean

Requirement Description: Administrators may delete posts made by other users.

Rationale: Administrators will be tasked with moderation of submissions, or review of submissions, as well as active user status of viewing the Feed. They may need to delete posts they find violate the purpose of the application.

Inputs:

1. The user navigates to the home/main page of the application and views the Feed.
2. The user, authorized as an administrator navigates to a particular post
3. The user selects the "Extra Options" menu of the post and selects "delete"
4. A confirmation notice of post deletion.

Outputs: The application will send a request to the backend where it will be processed and post removed from the database.

Persistent Changes: Completion of these steps will remove the entry from the Feed and database.

Related Requirements and Use Cases: Requirements 4, 5, 11 - Uses 1, 2, 3, 7, 11.

Requirement ID: 12

Requirement Description: Allow administrators to delete entries in the database

Rationale: This requirement would allow administrators to remove posts from the database that are either malicious and/or irrelevant to the purposes of the website

Inputs:

- The administrator user is logged in

- The administrator user will go to a post

- The admin user will then click the post and select the delete option and be prompted to verify his action

- If they proceed, the post will be deleted and the admin user will be brought back to the feed.

- If they do not proceed, then the user will be led back to the post screen

Outputs: The post will be deleted

Persistent Changes: The post and its information will be removed from the database and the database will have one less post.

Related Requirements and Use Cases: Requirement 4, 5, 12

Test Cases: 5

Requirement ID: 13

Requirement Description: Allow administrators to delete user accounts

Rationale: The requirement would allow administrators to remove users from the product if they infringe on any legal/copyright laws, are disorderly, or are misusing/abusing the service.

Inputs:

- The administrator user is logged in

- The administrator user will go to a user's information

- The admin user will then click the user and select the "Close Account" option and be prompted to verify his action

- If they proceed, the user account will be removed and the admin user will be brought back to the feed.

- If they do not proceed, then the user will be led back to the user's profile page

Outputs: The user account will be deactivated

Persistent Changes: The user account and its information will be removed from the database and the database will have one less post

Related Requirements and Use Cases: Requirement 4, 5, 12

Test Cases: 5

Object Modeling

Objects:

- Ordinary Clients
- Recognized Organizations
- Administrators
- Animals
- Posts
- Advertisements

Associations:

1. Ordinary Clients create Posts.
2. Recognized Organizations create Posts.
3. Administrators create Posts.
4. Animals are described by Ordinary Clients.
5. Animals are described by Recognized Organizations.
6. Animals are described by Administrators.
7. Administrators verify Posts.
8. Administrators verify Animals
9. Administrators verify Ordinary Clients.
10. Administrators verify Recognized Organizations.
11. Advertisements are created by Recognized Organizations.

Multiplicity:

1. Ordinary Clients: 1, Posts: 1..*
2. Recognized Organizations: 1, Posts: 1..*
3. Administrators: 1, Posts: 1..*
4. Animals: 1..*, Ordinary Clients: 1
5. Animals 1..*, Recognized Organizations: 1
6. Animals 1..*, Administrators: 1
7. Administrators: 1..*, Posts: 1..*
8. Administrators: 1..*, Animals: 1..*
9. Administrators: 1..*, Ordinary Clients: 1..*
10. Administrators: 1..*, Recognized Organizations: 1..*
11. Advertisements: 1..*, Recognized Organizations: 1

Attributes:

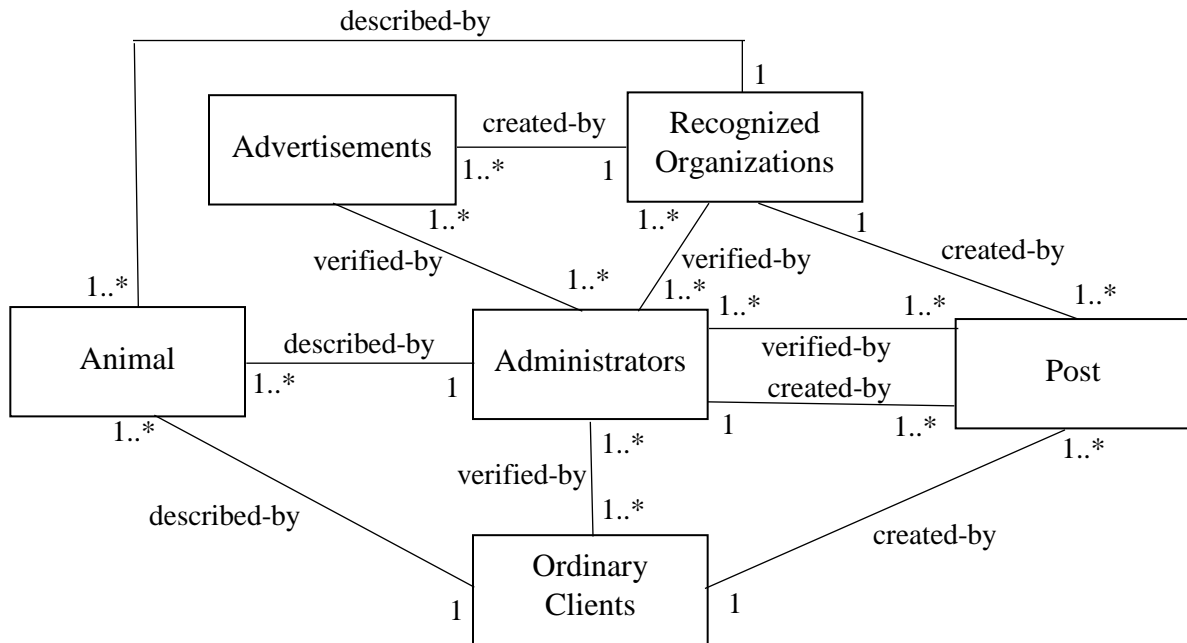
- Ordinary Clients
 - o Name
 - o Email
 - o Username
 - o Password
 - o Geolocation

- Posts made
 - Time
 - Date
- Recognized Organizations
 - Name
 - Email
 - Username
 - Password
 - Geolocation
 - Posts made
 - Advertisements made
 - Associated Animals
 - Time
 - Date
- Administrators
 - Name
 - Email
 - Username
 - Password
 - Geolocation
 - Posts made
 - Time
 - Date
- Animals
 - Name
 - Physical features
 - Geolocation
 - Associated users
 - Time
 - Date
- Posts
 - Post creator
 - Geolocation
 - Associated animals
 - Time
 - Date
- Advertisements
 - Advertisement creator
 - Geolocation
 - Associated animals
 - Time
 - Date

Operations:

- Ordinary Clients:
 - new()
 - checkGeolocation()
 - addAnimal()
 - createPost()
 - sharePost()
 - findAnimal()
 - displayPosts()
- Recognized Organizations:
 - new()
 - checkGeolocation()
 - addAnimal()
 - createPost()
 - sharePost()
 - findAnimal()
 - displayPosts()
 - createAdvertisement()
 - collectDatasheet()
- Administrators:
 - new()
 - checkGeolocation()
 - addAnimal()
 - createPost()
 - sharePost()
 - findAnimal()
 - displayPosts()
 - removeUser()
 - removeAnimal()
 - removePost()
 - removeAdvertisement()
- Animals:
 - new()
 - addAssociation()
- Posts:
 - new()
 - createLink()
 - createDatasheet()
- Advertisements:
 - new()
 - createLink()

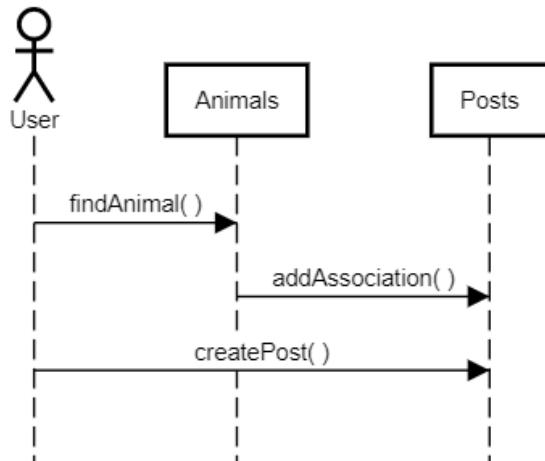
Class Diagram:



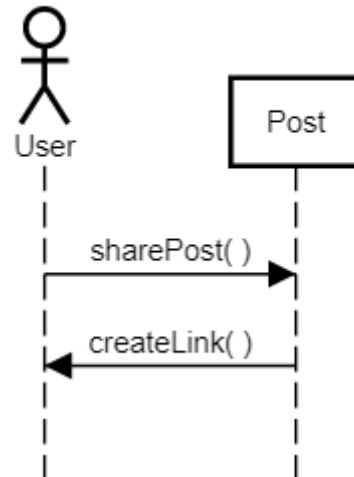
Behavioral Modeling

Sequence Diagrams:

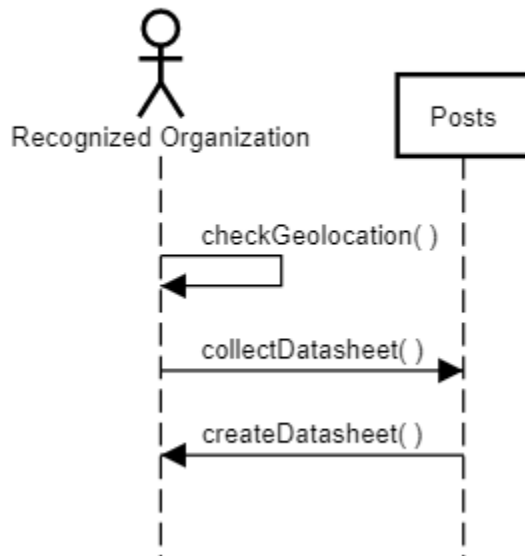
User Creating Post



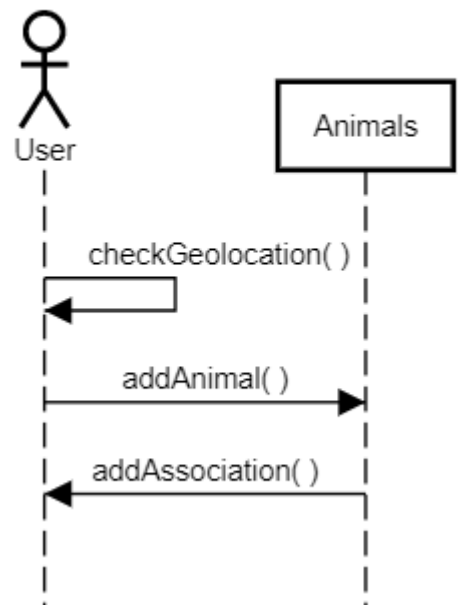
User Sharing Link



Recognized Organization Collecting Data



User Adding Animal



Appendix

Project Description Video:

https://www.youtube.com/channel/UCm0MVS3hisUg1tg7LhFCrtQ?view_as=subscriber

GitHub:

Logs

commit 7ecf23bb154f2afc298ec3db8dfe43e65df1297c
Author: JasonHerrera <jasonrh4513@bellsouth.net>
Date: Sat Feb 24 23:32:21 2018 -0500

Add files via upload

Slack Logs Feb 24

commit 60544d7ca17b33198491f33d1a25e069f20bc6cc
Author: JasonHerrera <jasonrh4513@bellsouth.net>
Date: Sat Feb 10 23:44:20 2018 -0500

Add files via upload

commit 4371794d5273a39767dbec56253d5a22364acc86
Author: JasonHerrera <jasonrh4513@bellsouth.net>
Date: Sat Feb 10 23:33:35 2018 -0500

Add files via upload

commit 26cc746f8bb1d92425366e94cb59eb171c7cbe3b
Author: JasonHerrera <jasonrh4513@bellsouth.net>
Date: Sat Feb 10 23:30:19 2018 -0500

Slack:

Logs

<https://github.com/JasonHerrera/Software-Engineering-Project/blob/master/Renegon%20Bits%20Slack%20export%20Feb%2024%202018.zip>