M9: Final Product

Group Members:

- Houlin (Mike) He - 74423799
- Juntong Luo - 62465273
- Francisco Farinha - 73029621
- Sara Mohamed - 95590519

PART I. Final Project Report

1. Description, Requirements, Design, Code Review Results, and Test Plan and Results

1.1. Project Description

Whether a long-time resident, or a tourist, it can often be difficult to find interesting locations to visit or find hotspots of activity since most websites spit out the same top-ten list of famous or historical buildings. Our project challenges mobile users, especially tourists and travel enthusiasts, to visit a curated list of nearby points of interest where they receive trophies upon visiting each location.

These trophies are ranked based on popularity and user ratings. When nearby, the user can collect the trophy and take a picture, to act as a memento. The user can also view photos taken by other users of that location. People can zoom in the pictures taken by other users, by tapping on the picture. Users can like it by tapping the red heart, and unlike it by tapping on it again. Nine pictures appear under each trophy, and are chosen randomly by default. The user can tap on the “Sort” button to display the nine newest uploaded pictures, and tap again to display the highest liked pictures.

A friend system allows users to add friends based on the user's id or email. When a friend request is received, the user receives a notification. A friend’s leaderboard is included in the leaderboard page. It displays User's Friends and trophy counts sequentially. The user can also delete an existing friend.

Collected trophies are tallied up in a leaderboard displaying top 10 users with the highest score, encouraging friendly competition. A friend leaderboard shows the ranking including the user and the user's friends. The leaderboard is updated constantlyA private notification will also be sent if the user drops out from the leaderboard. A notification will be sent when the top position is swapped.

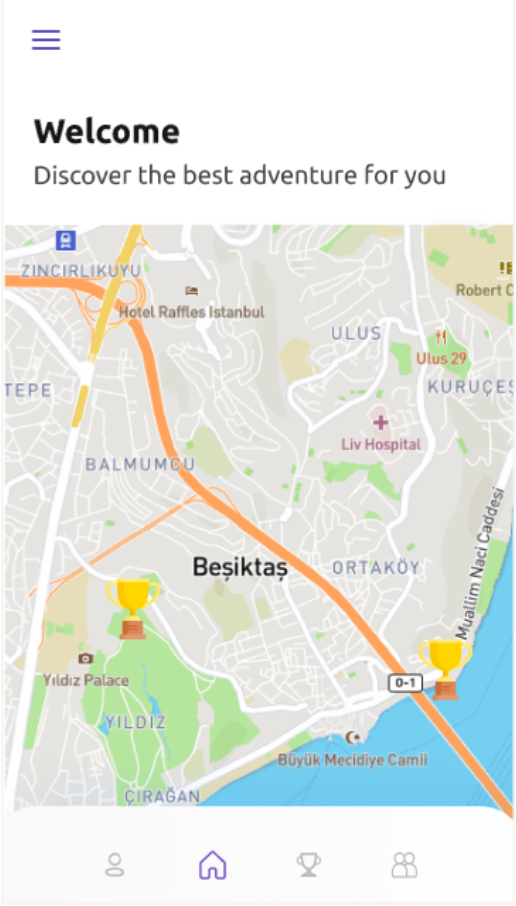
1.2. Non-trivial feature description

A call to the Places API returns a list of nearby locations ranked by prominence. Each of these is categorized into one of different types of location (park, cafe, church, etc). If the number of returned locations exceeds a certain threshold, we can use a history of the user's collected

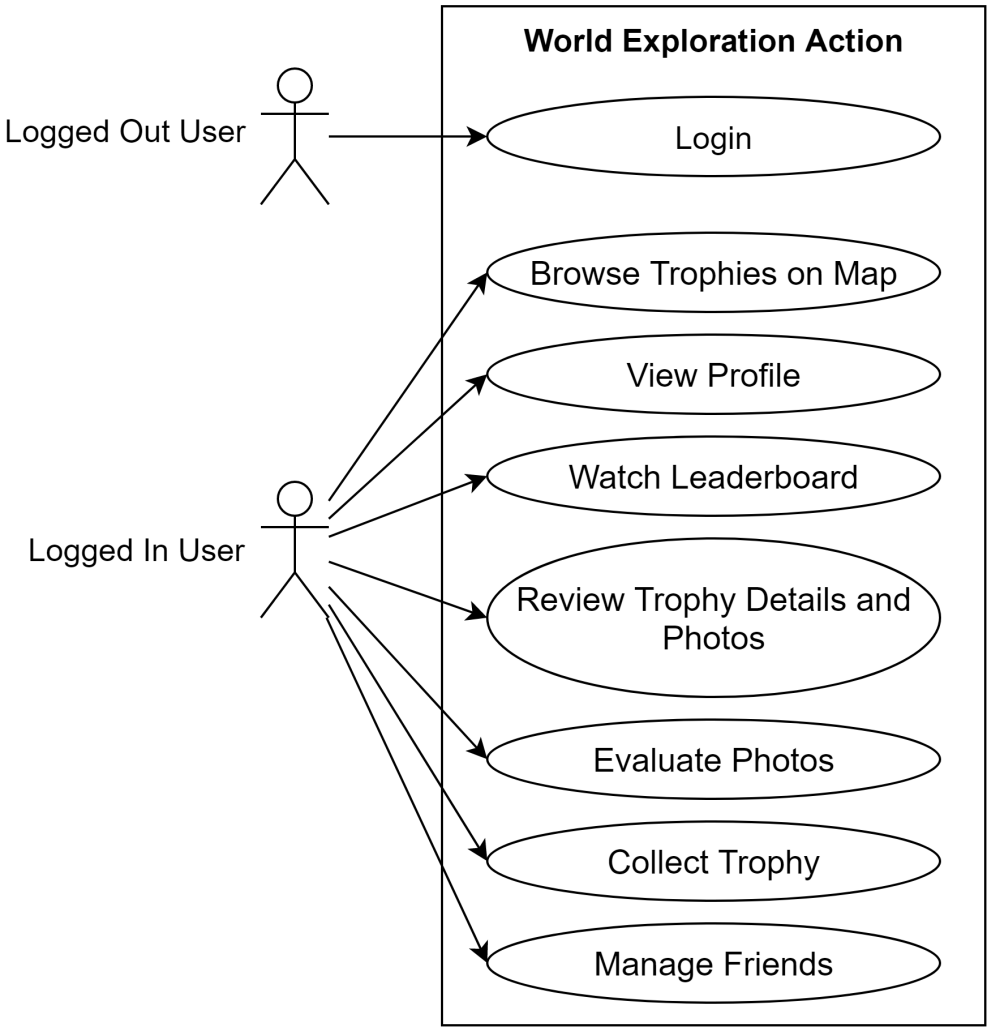
trophies to obtain a list of preferred types. We will filter the returned list of locations from Places API to only include those similar enough to the users preferences. We will use an Embeddings model, such as Word2Vec, to obtain vector encodings for each of the types, and use cosine similarity to determine how close a suggested location is to the user's preferences, and whether we will display it as a trophy to the user or not. Initially, we will use pretrained weights from the Common Crawl set, provided by Stanford as a starting point, but we will want to fine-tune the Embeddings layer with our own dataset, which we can obtain by scraping the WikiText corpus for sentences containing a set of 96 common tags from the Places API. Additionally, if performance is not deemed accurate enough, a more complex Transformer-based architecture will be employed. In this case, we will use BERT, and apply the same methodology. Starting from pre-trained weights, we can use BertForMaskedLM and train the language model on our dataset, after having frozen the base graph. The task being trained is Masked Language Modelling, which is an unsupervised task, so generating training data will be much more efficient than other supervised methods. In the end, the goal is to deploy the model in the backend, this way, it can grow to accommodate all the unique labels seen by every user without taking up space and performance on mobile.

1.3. Main Screen Mock-Up

The main screen will be a map pinned with different ranks of trophies (gold, silver and copper).



1.4. Use Case Diagram



1.5. Formal Use Case Specifications

Title: Login
Description: The actor logs in with their Google account
Actors: Logged Out User
Preconditions: The actor is not logged in and the Login page is displayed
Postconditions: The actor is logged in and the Map View is shown
Main Success Scenario:

1. The actor taps the Google Sign-In button
2. A Google Sign-In page pops up
3. The actor enters their email address or phone number
4. The actor clicks the Next button
5. The actor enters the password
6. The actor clicks the Next button
7. The Google Terms of Service page is shown
8. The user clicks the “I agree” button

9. The app displays the Map View

Extensions:

2a. The actor already has a Google account on the device
- 2a1. The “Choose an account” view pops up
- 2a2. If the actor clicks “Add another account”, continue on Step 2
- 2a3. If the actor clicks an existing account, continue on Step 9

4a. The entered account could not be found
- 4a1. The app stays on the Google Sign-In page
- 4a2. The app displays text indicating the Google account could not be found

6a. The actor entered a wrong password
- 6a1. The app stays on the Google Sign-In page
- 6a2. The app displays text indicating the password was incorrect

9a. Google Sign-In returns an error
- 9a1. The app displays an error message specifying the failure reason
- 9a2. The app stays on the Login page

Title: Browse Trophies on Map

Description: A map is shown of the area surrounding the current user location with trophies being displayed at various points of interest. Users can click on a trophy to go to that trophy's details page. This page can navigate users to other sub-pages.

Actors: Logged In User

Preconditions: User has logged in

Postconditions: Map view is displayed

Main Success Scenario:

1. User presses the home button
2. Map View of the location surrounding the user is displayed
3. Pins corresponding to trophies near the user are displayed

Extensions:

2a. Location information cannot be retrieved, GPS communication lost.
- 2a1. Display Error popup saying that a location could not be retrieved with two options: Retry and OK.
- 2a2. The User presses Retry to query location once more or OK to close the popup
- 2a3. The Map View zooms out to the largest zoom level, and shows no trophies.

3a. No Trophies exist near the user.
- 3a1. Display a popup message stating that no trophies could be located near the user with the option of pressing OK to close the popup
- 3a2. The Map view shows the location surrounding the user with no trophies displayed.

Title: View Profile

Description: A view that shows information about a user such as name, email, rank, pictures they have taken with the places listed, and the number of trophies they have earned.

Actors: Logged In User

Preconditions: User is logged in with the map view displayed.

Postconditions: Profile view is displayed

Main Success Scenario:

1. The actor taps the Profile button
2. The app displays the Profile View
3. The actor can see their name, email, rank, and the most updated trophy score. Additionally, there is a grid view of the user's most recent pictures.
4. The actor can optionally sign out using the sign-out button

Extensions:

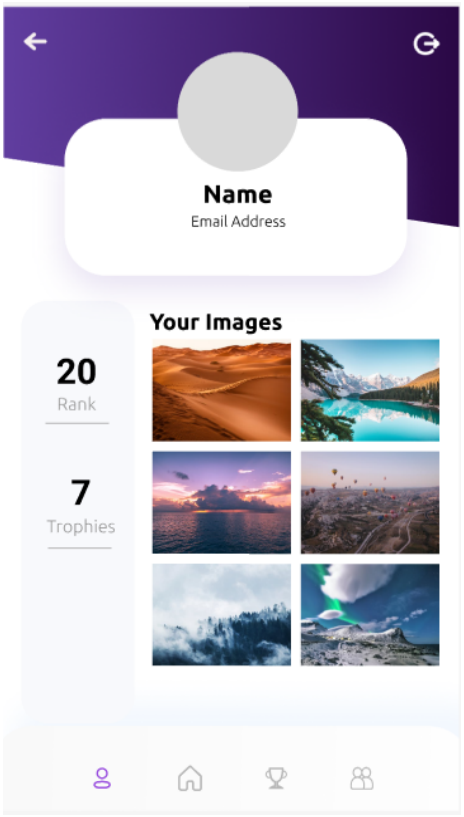
3a. Frontend is waiting for the backend reply

- 3a1. Display a loading text on rank, the places, the pictures, and the number of trophies while waiting for the backend response
- 3a2. The information is updated when a response is received

3b. User has not collected any trophies

- 3b1. In the rank and collected trophies fields, display a dash. In the grid view, display text stating that the user has not collected any trophies yet.

Sample for layout:



Title: Watch Leaderboard

Description: A screen that shows the usernames of the top 10 users (Global Leaderboard) or all the user friends list (Friends Leaderboard) along with their scores in descending order. The leaderboard supports real-time updates.

Actors: Logged In User

Preconditions: Map view is displayed

Postconditions: Leaderboard view is displayed

Main Success Scenario:

1. The actor taps the Leaderboard button
2. The actor chooses the Global Leaderboard button
3. User can see the most updated top 100 usernames and their scores
4. The actor chooses the Friends Leaderboard button
5. User can see the most updated friends list and their scores in descending order
6. The leaderboard is updated in real-time

Extensions:

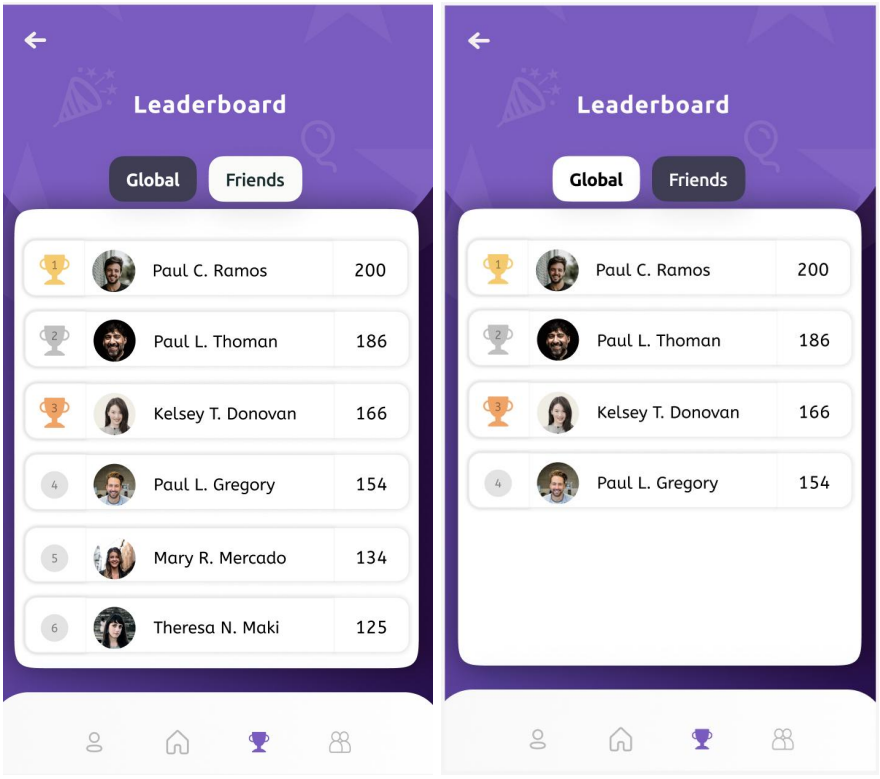
2a. Front-end still waiting for backend reply

- 2a1. Display loading text on the leaderboard while waiting for a backend reply.
- 2a2. When a message is received, update to show users.

2b. Server error, could not retrieve users to populate the leaderboard

- 2b1. Display Error message saying that connection to the server failed. With the option to Retry, to attempt a backend call again, or OK to dismiss the popup.
- 2b2. User presses OK to dismiss the popup and the leaderboard should display text saying Error.

Sample layout:



Title: Review Trophy Details and Photos

Description: The user taps on one of the trophies on the Map. At most 9 photos collected by other users in this location are displayed. The photos are randomly displayed. The user can sorts the photos by the likes number from high to low or by the time. Also, the number of users collected this trophy is displayed.

Actors: Logged In User

Preconditions: The Map is shown

Postconditions: Number of users who collected the trophy and at most 9 photos added by the users are displayed.

Main Success Scenario:

1. The user taps on a trophy on the map
2. The app opens a Trophy Details view, which displays the trophy title, at most 9 photos taken by other users and the number of users who have collected the trophy.
3. Optionally, the user can view a photo in full-screen and/or like the photo
4. Optionally, the user can view the top 9 photos sorted by time
    4.1. The user taps on the “Sort By”
    4.2. The sorting of photos changes to “Sort by Time” from new to old
    4.3. Photos display in time order
5. Optionally, the user can view the top 9 pictures sorted by the number of likes
    5.1. The user taps on the “Sort By”
    5.2. The sorting changes to “Sort by Like Number” from high to low
    5.3. Photos display in like number order
6. Optionally, the user can add a photo
    6.1. The user taps on “Collect trophy” button
    6.2. A button for adding a photo will appear on the right side of “Collect trophy” button

Extensions:

2a. No photos exist for this trophy.

- 2a1. A message should display where the photos would have been displayed. There are two options depending on the user's proximity to the trophy:
- Option 1: The user is at the location:
    - 1.1: A message saying "Take a picture now to be the first one! " will be displayed.
- Option 2: The user is not at the location:
    - 2.1: A message saying "Seems nobody else has come to this place yet.” will be displayed.

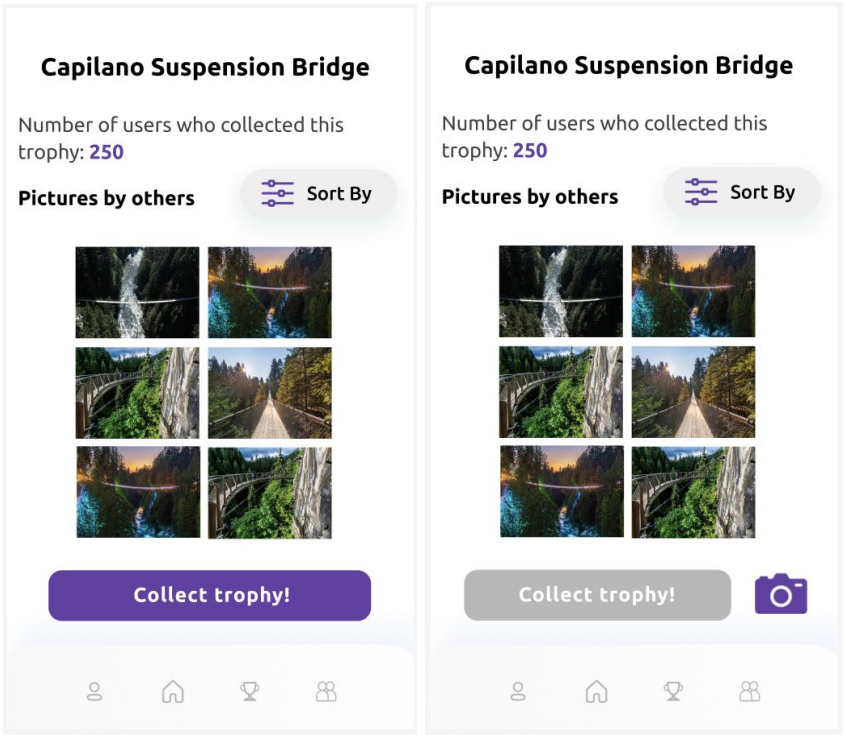
5.2a. Two pictures have the same uploaded time

- 5.2a1. The system compares the like numbers, the one that has the higher number of likes is considered to be “earlier”. If two also have the same number of likes, the one with a smaller ID is considered to be “earlier”.

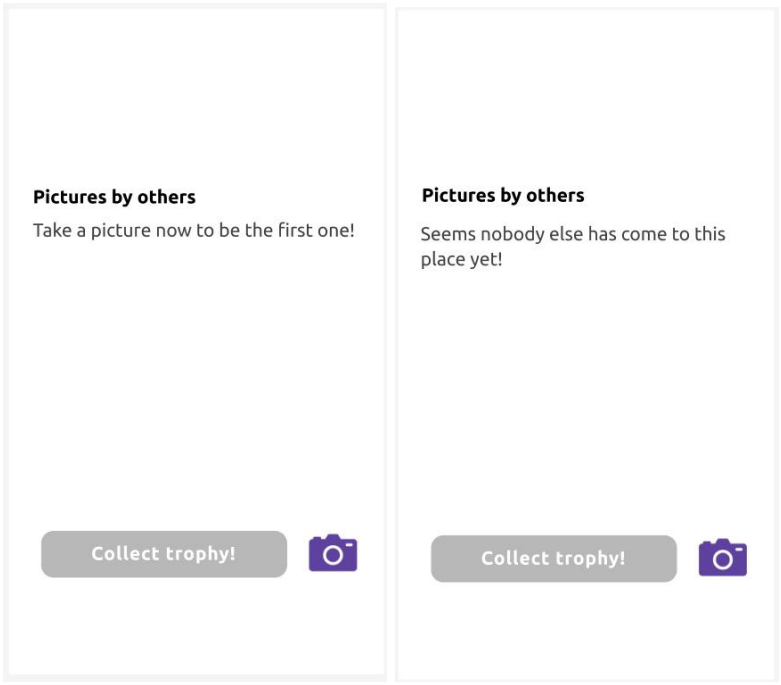
6.2a. Two pictures have the same number of like

- 6.2a1. The system compares the uploaded time, the one being uploaded earlier is considered to have a “higher” number of likes. If two also have the same number of likes, the one with a smaller ID is considered to be “higher”.

Sample for Layout:



Sample for layout (Extension 2b):



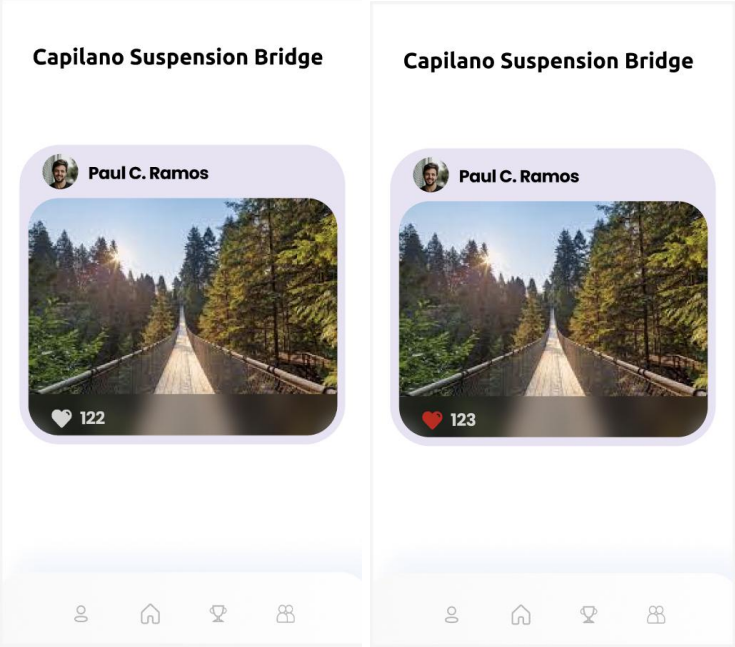
Title: Evaluate Photos

Description: The user taps on the photo. The photo is opened in a full view. The user can like the photo by clicking the heart icon.

Actors: Logged In User
Preconditions: Review trophy details is shown
Postconditions: The picture is opened in a full view and the number of likes of that photo is displayed on the bottom left corner of the photo.
Main Success Scenario:

1. The user taps on the photo
2. The picture is displayed in a full-screen view
3. The user can like the photo
   3.1. The user taps the like button
   3.2. The photo is liked by the user, and the number of likes of the photo is incremented
4. The user can unlike the photo
   3.1. The user taps the like button again (they should have liked it first)
   3.2. The photo is unliked by the user, and the number of likes of the photo is decremented

Sample for Layout:



Title: Collect Trophy
Description: The user taps on a trophy displayed in the Map view. The app displays the details of the trophy including photos taken by users. If the user is in the location of the trophy, the user can collect a trophy and optionally take a photo.
Actors: Logged In User
Preconditions: The Trophy Details view is displayed.
Postconditions: The Trophy Details view is displayed, and the trophy is collected. If the user chooses to take a photo, the user's photo is uploaded. If the user already has a photo for this trophy, the old photo is replaced.
Main Success Scenario:

1. The Trophy Details view is displayed.
2. Optionally, the actor can open a Google Maps navigation
    2.1. The user taps the Navigation button
    2.2. The app launches Google Maps to navigate to the trophy location
3. If the actor has not collected the trophy
    3.1. The actor taps the “Collect Trophy” button
    3.2. The app shows a loading icon with the text “Collecting…”, and waits for the backend response
    3.3. The app plays a trophy animation
    3.4. The "Collect Trophy" button is replaced with the "Take a Picture" button
4. Optionally, the actor can choose to take a picture and upload it
    4.1. The actor taps the “Take a Photo” button in the Trophy View
    4.2. The app opens the default camera app on Android
    4.3. The actor takes a photo
    4.4. The app shows a loading icon with the text “Uploading…” and uploads the photo to the backend
    4.5. The app displays a toast with the text “Uploaded!”
    4.6. Images in the Trophy Details view are updated to reflect the new photo taken by the actor

Extensions:

1a. The user is too far away from the trophy

- 1a1. The "Collect Trophy" or “Take a Photo” button is greyed out.
- 1a2. If the user still presses the button, display a toast message with the text “You are too far away. Keep it up!”

1b. Unable to obtain the user’s current location

- 1b1. Error popup saying that a location could not be retrieved with two options: Retry and OK
- 1b2. The User presses Retry to query location once more or OK to close the popup
- 1b3. Until the user's location can be determined, the "Collect Trophy" or “Take a Photo” button will be greyed out

3.2a. No response or error from the server

- 3.2a1. Display an error message stating that the connection to the server failed
- 3.2a2. The "Collect Trophy!" button is not replaced, in effect, the trophy was not collected
- 3.2a3. The user can attempt to collect again. The same error message will display if the issue persists

4.1a. The user has already taken a photo for this trophy

- 4.1a1. The app displays a dialog with the text “You’ve already taken a photo. Would you like to replace it?” and options: Yes and No
- 4.1a2. If the user clicks Yes, then the app continues Step 3.2. Otherwise, the app does nothing and stays on the Trophy Details view

4.3a. The user did not take any photo

- 4.3a1. Display a toast message with the text “No photo taken”

4.4a. No response or error from the server

- 4.4a1. Display a toast message with the text “Unable to upload the photo”

Title: Manage Friends
Description: Display a view that shows the user’s friends’ usernames and their trophy score and allow the user to manage their friends.
Actors: Logged In Users
Preconditions: Map view is displayed
Postconditions: Friend’s profile view is displayed
Main Success Scenario:

1. The actor taps the Friend Button
2. The app displays the Friend View
3. The user can see their Friend List with the user name on the left and their trophy score on the right. The user has the options: send a friend request, review a friend request from others, and view a friend’s profile
    - 3.1. Send Friend Request
        - 3.1.1. The user taps the “Search” from the top tap.
        - 3.1.2. The user types in another user’s User ID or username
        - 3.1.3. A list of users with the correct username or user ID is displayed
        - 3.1.4. The user clicks on the correct user
        - 3.1.5. The user clicks the “Submit” button
        - 3.1.6. A friend request is sent
    - 3.2. Review a friend request from others. The new friend request is shown on the top of the friend list. The user has two options: accept it or decline it
        - 3.2.1. Accept The Friend Request
            - 3.2.1.1. The user accepts the friend request
            - 3.2.1.2. The newly added friend appears on the friend list
        - 3.2.2. Decline The Friend Request
            - 3.2.2.1. The user declines the friend request
            - 3.2.2.2. The request disappears
    - 3.3. View a friend’s profile
        - 3.3.1. The actor taps on one friend
        - 3.3.2. The actor selects the View Profile option
        - 3.3.3. The profile of the friend is opened, as described in the View Profile use case
    - 3.4. Delete a current friend
        - 3.4.1. The actor taps on one friend
        - 3.4.2. The actor selects the Delete option
        - 3.4.3. The friend is deleted, and the list of friends is updated

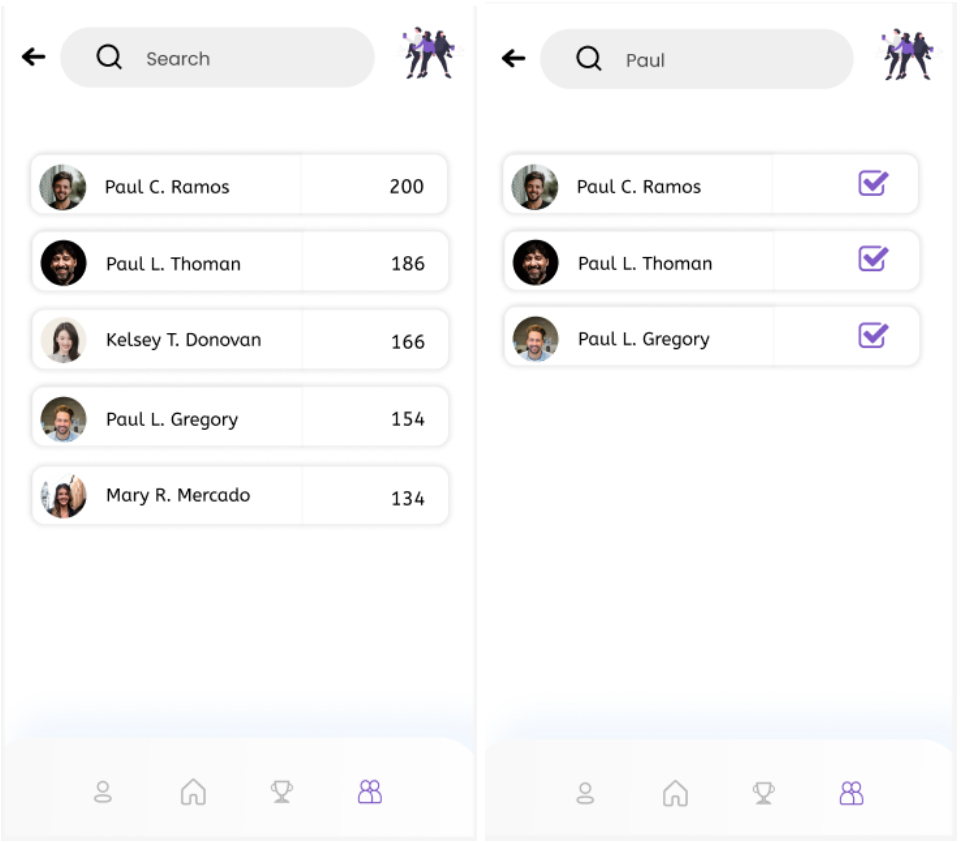
Extensions:

3a. No friend is added

- 3a1. A message “Seems you have no FRIENDS” will be displayed instead of the friend list

3.1.3a. No relevant user is found

- 3a1. A message “No User Found” is displayed instead of the user list



1.6. Non-Functional Requirements

- **Responsiveness**:
  - The app should respond and present complete information within 2 seconds after the user clicks any button. A smooth UI is important for the user experience and satisfaction. We plan to verify this by paying attention to the responsiveness when we use the app. Specifically, we can time how long it takes to change views and how long an average API request takes.
- **Usability**:
  - After logging in, the user should be able to perform any action with no more than 5 clicks (entering text will count as 1 click). This will help ensure the app is easy to use and the user can find all features quickly. We are going to verify this by counting the number of clicks required to perform every possible action.
- **Performance**:
  - The user should get a notification within 10 seconds after the top 1 user in the leaderboard is changed or the user gets in or drops out from the leaderboard. A quick notification helps users obtain the information they need to help them make decisions. We can verify this by setting only two users, and keep overtaking the top 1 position and see if the app can send notifications in 10 seconds.

- After users take and upload a picture, other users' trophy pictures are updated within 10 seconds so that the new picture can appear in the picture collection. The quick upload helps users share what they explore for the first time. We can verify this by uploading pictures, and seeing if the picture is updated at each user's end.
- After a friend request is sent, the receiver should get a notification and be able to see the request within 10 seconds. This is important for the user experience, especially for users who are friending someone they know in real life. We can verify this by letting a user send a friend request to another user, and measuring the delay between sending a request and receiving a notification.

1.7. Main Components

1.7.1 Modules

A. **Users**
   - **The Users module is responsible for storing and managing information about all users. This includes storing unique ID, preferred name, email, trophy score and friends, tracking Global and Friend leaderboards, and managing Friend requests.**

B. **Trophies**
   - **The Trophies module is responsible for storing all active Trophies and housing the ML model to generate curated lists of trophies depending on the User. Trophies themselves have metadata related to location, rating, relevant tags, number of collections, and a score determined by a combination of popularity and rating.**

C. **Photos**
   - The Photos module handles storage and retrieval of all user-taken photos, as well as relevant metadata such as date, location, relevant TrophyID and UserID.

D. **Message Manager**
   - The Message Manager module manages messages in a publisher/subscriber architecture.

1.7.2 Databases

A. **MongoDB**

The MongoDB database persists data in our application.

1.7.3 External Components

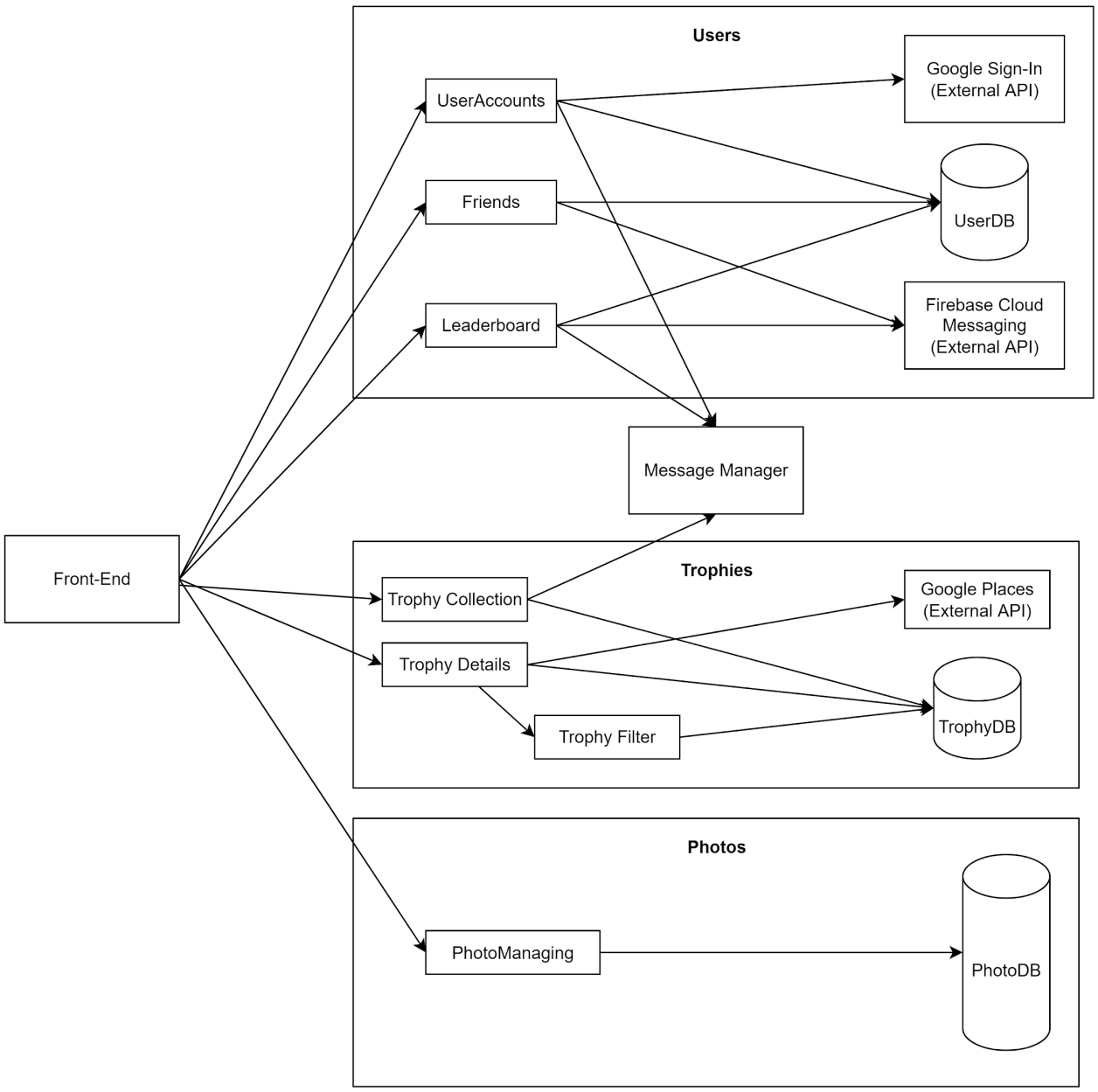
A. **Google Sign-In**
   - Used to sign in the users, and each user will have a space in the user database categorized by the user's Google ID

B. **Google Places API**
    - Used to retrieve the places of interests to generate trophies based on user location.
C. **Firebase Cloud Messaging**
    - The Firebase Cloud Messaging is for sending notifications to the user's Android device. Used especially for friend requests and rank change.

1.8. Main Sub-Modules

- **Users**
    - UserAccounts
        - Represents users, includes information such as UserID, email, preferred name, and ranking.
    - Friends
        - Represents the friends of one user and manages friend requests
    - Leaderboard
        - Manages the global leaderboard and friend leaderboards
    - UserDB
        - Stores all user information
- **Trophies**
    - Trophy Filter
        - Filter those trophies being displayed based on users
    - Trophy Detail
        - Stores details of all trophies
    - Trophy Collection
        - Responsible for collecting trophies
    - TrophyDB
        - Stores trophy information and other data useful for filtering trophies based on users' preferences
- **Photos**
    - PhotoManging
        - Manage information associated with each photo
        - Choose photos based on different criteria
        - Handles Storage and Retrieval of image files from PhotoDB
    - PhotoDB
        - Stores photos, number of likes, and the relationship between User IDs, Trophy IDs, and Picture IDs
- **Message Manager**
    - The Message Manager does not have any submodule.

1.9. Sub-Modules Diagram



1.10. Sub-Module Interfaces

1.10.1 Users

- UserAccount

    - User loginWithGoogle(String idToken)
        - Sign in the user with the Token ID returned by Google Sign-In.
        - Param: tokenID - the ID token as returned by Google Sign-In
        - Return: the user's information, including User ID, name, email etc.
    - User getUserProfile(String userID)
        - Get user information such as email, preferred name, ranking, and friends from the database with user ID
        - Param: userID - the user's ID
        - Return: the user's basic profile
    - void signOut(String userId)
        - Sign out the current user
    - void uploadFcmToken(String userId, String fcmToken)
        - Upload the user's FCM Registration token, which is required for sending FCM Android notifications
    - List<User> searchUser(String query)
        - It extracts the users which have the same username or have the same user ID. If the user types in user ID, uppercase and lowercase matter; however, if it is username, the method does not distinguish uppercase and lowercase.
        - Param: query - either username or user ID
        - Return: A list of users that corresponds to the information provided
    - void onReceiveTrophyCollectedMessage(Message message)
        - Handle a new trophy collected event.
        - Param: message - the message containing the collector's User ID and the collected trophy's ID

- Friends
    - void sendRequest(String userID, String friendID)
        - It records the request
        - Param: userID - user ID
        - Param: friendID - friend's user ID
    - void declineUser(String userID, User friend)
        - It declines the user's friend request
        - Param: userID - user ID
        - Param: friend - the friend
    - void acceptUser(String userID, User friend)
        - It accepts the user's friend request
        - Param: userID - user ID
        - Param: friend - the friend
    - void deleteFriend(String userID, String friendID)
        - Unfriend a user
        - Param: userID - user ID
        - Param: friendID - ID of the friend to delete
    - List<User> retrieveFriends(String userID)

        - It retrieves all the user's friends
        - Param: userID - user ID
        - Return: user's friends as a list
    - List<User> getFriendRequests(String userID)
        - It returns all the friend requests of that user
        - Param: userID - user ID
        - Return: user's friend requests as a list
- Leaderboard
    - List<User> getGlobalLeaderboard()
        - Retrieve a list of users on the leaderboard.
        - Return: A list of user on the global leaderboard, sorted by their scores in non-increasing order
    - List<User> getFriendLeaderboard(String userId)
        - Retrieve a list of users on the leaderboard.
        - Return: The users friend and the user, sorted by their scores in non-increasing order
    - Time subscribeUpdate(String userId, String fcmRegistrationToken)
        - It subscribes the front-end device to the leaderboard, so the leaderboard module will send a notification when the leaderboard is updated. The subscription has an expiry time so the front-end should resubscribe before it expires.
        - Param: fcmRegistrationToken - the Firebase Cloud Messaging registration token which identifies the Android device
        - Return: The time when the subscription is about to expire.
    - void onReceiveUserScoreUpdatedMessage(Message message)
        - Handle a new user score updated event.
        - Param: message - the message containing the collector's User ID and the collected trophy's ID.
- UserDB
    - User findUser(String userID)
        - Retrieve user information such as email, preferred name, ranking, and friends with user ID from user database
        - Param: userID - User ID
        - Return: the user's information
    - void addUser(User newUser)
        - Create a space for the new added user in the database
        - Param: newUser
    - void addTrophytoUser(int trophyScore, String trophyID)
        - It adds the Trophy ID to the user in user database, representing the user has collected the trophy
        - Param: Trophy ID - the ID of the trophy
        - Param: trophyScore - the score the user earn when the trophy is collected
    - int computeUserRank(String userID)

        - It compare the user with all users in the database, and determines the rank of the user
        - Param: userID - the ID of user
    - List<User> findTopUsers(int num)
        - It returns top num of users, who are going to appear on the leaderboard.
        - Param: num - the top number of users appear on the leaderboard
        - Return: A list of users going to appear on the leaderboard
    - List<User> getFriends(String userID)
        - It finds the user's friends via user ID, and return it as a list of user
        - Param: userID - the ID of the user
        - Return: the list of user's friends
    - void sortByTrophyScore(List<User> users)
        - It sort the list of users by their trophy score from highest to lowest
        - Param: users - a list of users
    - List<User> searchUserDB(String token)
        - It extracts the users from the user database, which have the same username or have the same user ID. If the user types in user ID, uppercase and lowercase matter; however, if it is username, the method does not distinguish uppercase and lowercase.
        - Param: token - either username or user ID
        - Return: A list of user that corresponds to the information provided
    - String searchUserTokenDB(User friend)
        - It extracts the user token associate to the user from user database
        - Param: friend - the user being searched on
        - Return: the user token associates to the user
    - void incrementTrophyScore(String userID)
        - Increment the trophy score of a user
        - Param: userID - the ID of the user

1.10.2 Trophies

- TrophyDetail
    - TrophyTrophy getTrophyDetails(String TrophyID)
        - It retrieves trophy information includes trophy title, trophy location, and the number of people collected the trophy with the Trophy ID
        - Param: TrophyID - the ID of the trophy
        - Return: the trophy information displayed on the screen except for the photos in a list
    - Trophy[] getTrophiesUser(String UserID, double lat, double lon)
        - Retrieve a list of uncollected trophies for a given user, to be displayed on the map. This will also generate trophies to show the user a constant number of uncollected trophies. Generated trophies are based on location and filtered to match the user's preferred tags.
        - Param: UserID - the ID that identifies the user. One of the collections in TrophyDB is indexed by UserID

        - Param: lat, lon - location object encoding latitude and longitude coordinates of the User.
        - Return: List of trophies based on location and user tags.
- TrophyFilter
    - List<TrophyTrophy> filterTrophies(String userID, List<TrophyTrophy> trophies)
        - Retrieve user tags and pass through Embeddings model to keep Trophies that match the user's history.
        - Param: userID - the ID of the user
        - Param: trophies - list of Trophy objects parsed from Places API
        - Return: List of Filtered Trophy objects based on user history.
    - List<TrophyTrophy> mlFilter(List<TrophyTrophy> trophies, List<String> userTags)
        - Param: trophies - List of Trophy Objects parsed from Places API
        - Param: userTags- The user's historic tags
        - Return: list of Trophies with tags similar to user tags.
- TrophyCollection
    - void collectTrophy(String userID, String trophyID)
        - Process the event of the user collecting a trophy
        - Param: userID - the ID of the user
        - Param: trophyID - the ID of the trophy
- TrophyDB
    - int getTrophyScore(String TrophyId)
        - Returns the trophy score searched by the trophy id
        - Param: TrophyId - the trophy ID
        - Returns: the score associates to the trophy
    - List<String> getTrophyText(String TrophyID)
        - It retrieves trophy information includes trophy title, trophy location, and the number of people collected the trophy with trophyID from the trophy database
        - Param: TrophyID - the ID of the trophy
        - Return: the trophy information displayed on the screen except for the photos in a list
    - void addUserToTrophy(String userID, String trophyID)
        - It add the userID to the trophy, which is searched up by trophyID. It represents that the user has already collected the trophy
        - Param: userID - the ID of the user
        - Param: TrophyID - the ID of the trophy
    - void removeUncollectedTrophy(String userID,String trophyID)
        - It removes the trophy from the user's uncollected trophies. The uncollect trophies list only include those trophies near the user
        - Param: userID - the ID of the user
        - Param: TrophyID - the ID of the trophy
    - void addCollectedTrophy(String userID,String trophyID)
        - It adds the trophy to the user's collected trophies list

        - Param: userID - the ID of the user
        - Param: TrophyID - the ID of the trophy
    - void storeTrophies(String userID, Trophy[] trophies)
        - Param: userID - the ID of the user
        - Param: trophies - List of Trophy objects to store in TrophyDB for User.
    - String[] getUserTags(String userID)
        - Queries the TrophyDB to retrieve the list of historic user tags.
        - Param: userID - the ID of the user.
        - Return: List of user historic trophy tags

1.10.3 Photos

- PhotoManaging
    - List<String> getPhotoIDsByTrophyID(String trophyID, Order order)
        - It retrieves the photo IDs, and the rules of retrieving them are determined by the parameter order
        - Param: TrophyID - the ID of the trophy
        - Param: order - the order of sorting of pictures, can be “random”, “time”, and “like”
        - Return: A list of photo ID sorted by order
    - List<String> getPhotoIDsByUserID(String userID)
        - Retrieve IDs of photos taken by the user
        - Param: userID - the ID of user
        - Return: A list of photo IDs
    - void userLikePhoto(String userID, String picID)
        - It either likes or unlike the picture, and it will be determined by other sub-modules. If the user did not like the picture before, the user will like the pic; and vice versa.
        - Param: userID - the ID of user
        - Param: picID - the picture ID which the user tap the “like” button on
        - Return: void
    - void uploadPhoto(String userID, String trophyID, Image photo)
        - Upload the picture taken by the user, which will grant the picture with the information it needs; for example, the picture ID
    - Image getImage(String picID)
        - It retrieve the image based on the photo ID
        - Param: picID - the photo ID
        - Return: the image associates with the photo ID
- PhotoDB
    - List<String> getRandom(int limit)
        - It retrieves the photo IDs up to the limit number from the database, and all photos are retrieved randomly. If the number of picture are less than limited number, all pictures’ ID will be returned
        - Param: limit - limit number of item to be returned
        - Return: A list of photo ID

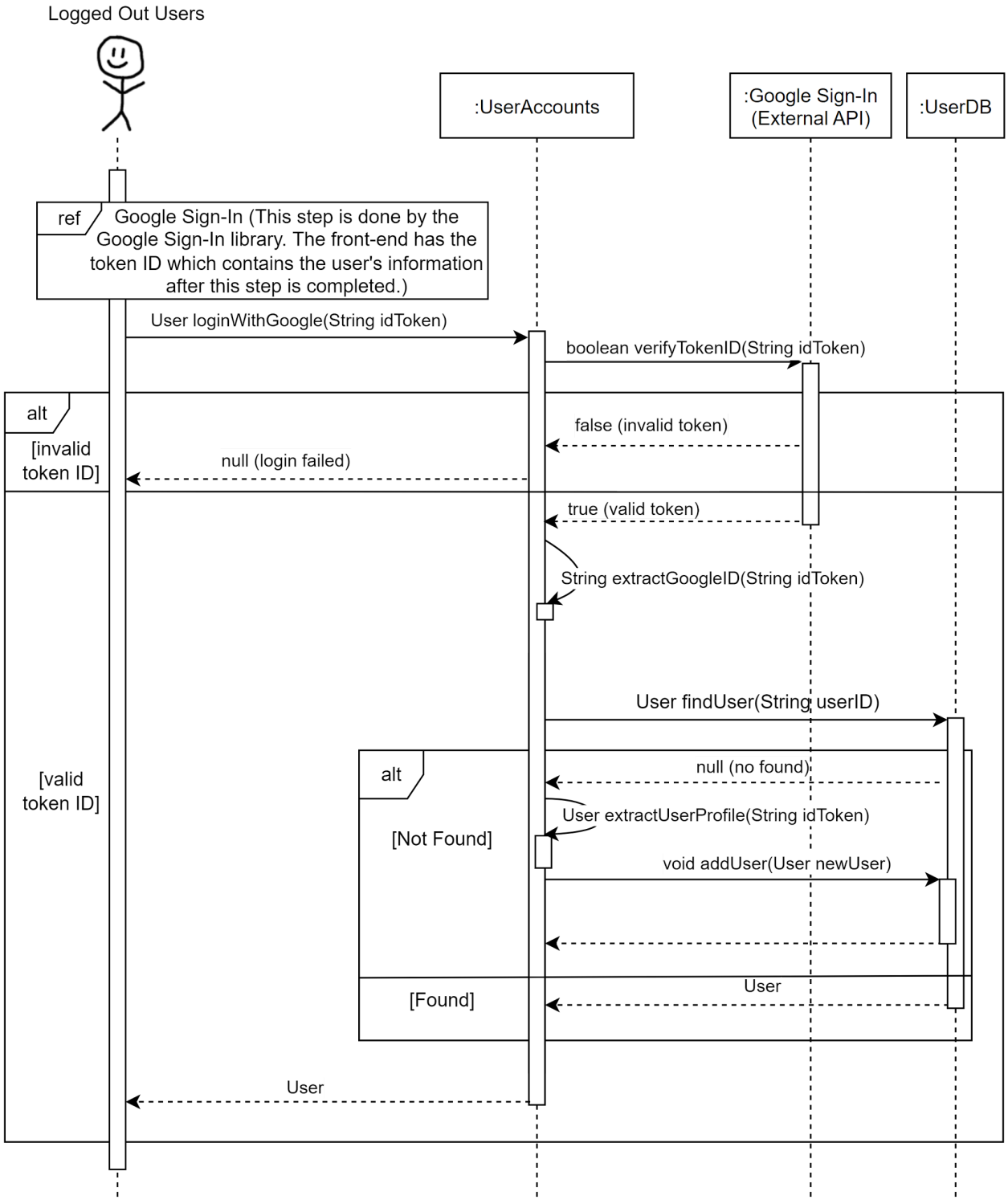
- List<String> getSortedByTime(int limit)
  - It retrieves the photo IDs up to the limit number from the database, and all photos chosen have the latest uploaded time. If the number of picture are less than limited number, all pictures' ID will be returned
  - Param: limit - limit number of item to be returned
  - Return: A list of photo ID
- List<String> getSortedByLike(int limit)
  - It retrieves the photo IDs up to the limit number from the database, and all photos chosen have the highest numbers of likes. If the number of picture are less than limited number, all pictures' ID will be returned
  - Param: limit - limit number of item to be returned
  - Return: A list of photo ID
- void userLikePhoto(userID, picID)
  - It represents that the user likes the picture, and increments the total number of likes of that picture by one in the database. It also records who liked the picture in the database.
  - Param: userID - the User ID
  - Param: picID - the photo ID
- void userUnikePhoto(userID, picID)
  - It represents that the user unlikes the picture, and decrements the total number of likes of that picture by one in the database. It also records who unliked the picture in the database.
  - Param: userID - the User ID
  - Param: picID - the photo ID
- void pictureUpload(String userID, String tophyID, String picID, Image photo)
  - It stores the picture into photo database with the necessary information; for examples, the user uploaded the photo, the trophy it belongs to and its unique photo ID
- void replace(String userID, String tophyID, String picID, Image photo)
  - It replaces the picture, which belongs to the same trophy and is taken by the same use, with the new picture uploaded
- Image findImage(String picID)
  - It finds the image based on the photo ID in the photo database
  - Param: picID - the photo ID
  - Return: the image associates with the photo ID

1.10.4 Message Manager

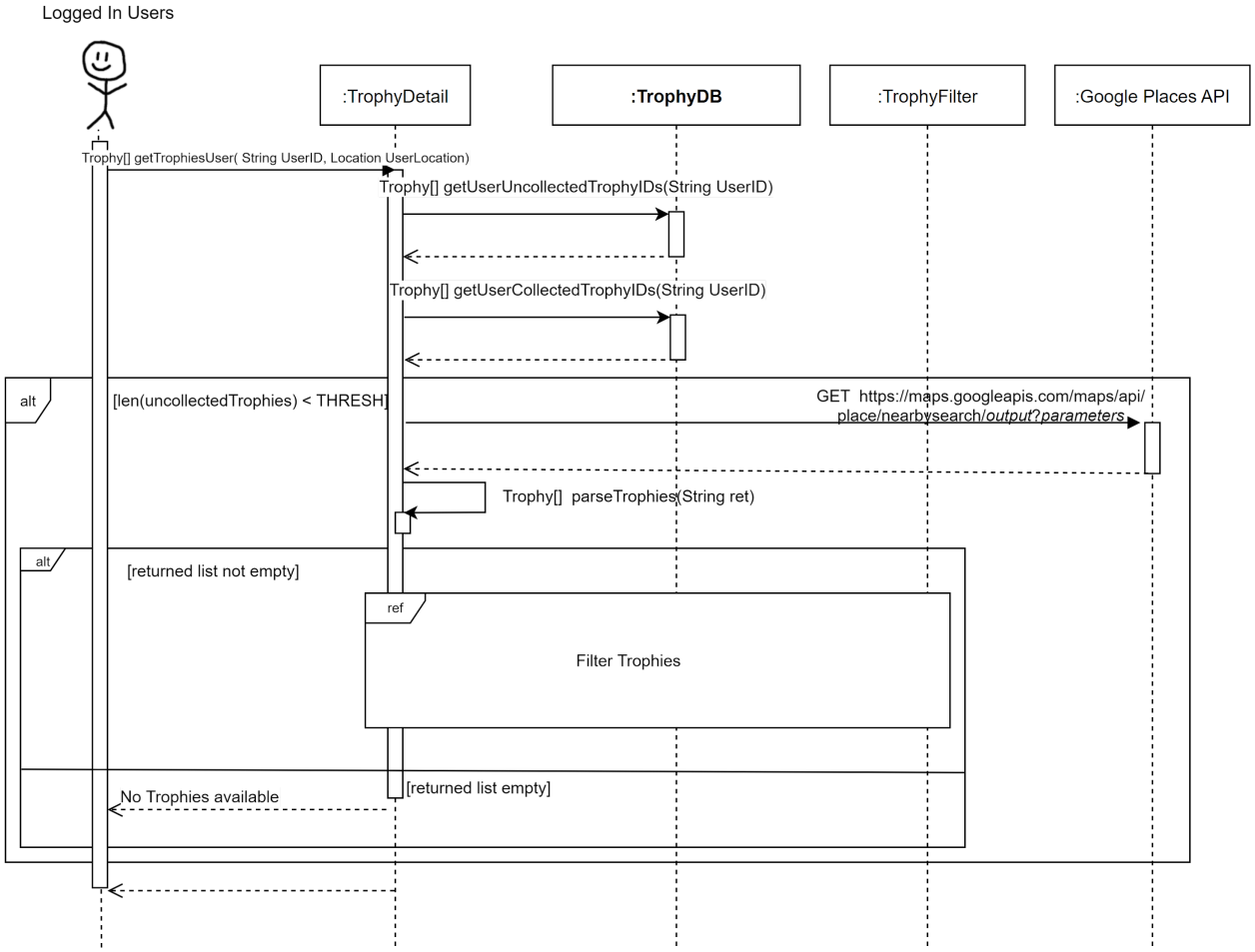
- Message Manager
    - void publishNewMessage(Message message)
        - Publish a new message. Subscribers of the same type will receive the message. (Each message has a type)
        - Param: message - the message to publish
    - void notifySubscribers(Message message)
        - Send the message to all subscribers of the same type. (Each message has a type)
        - Param: message - the new message that will be sent to the subscribers

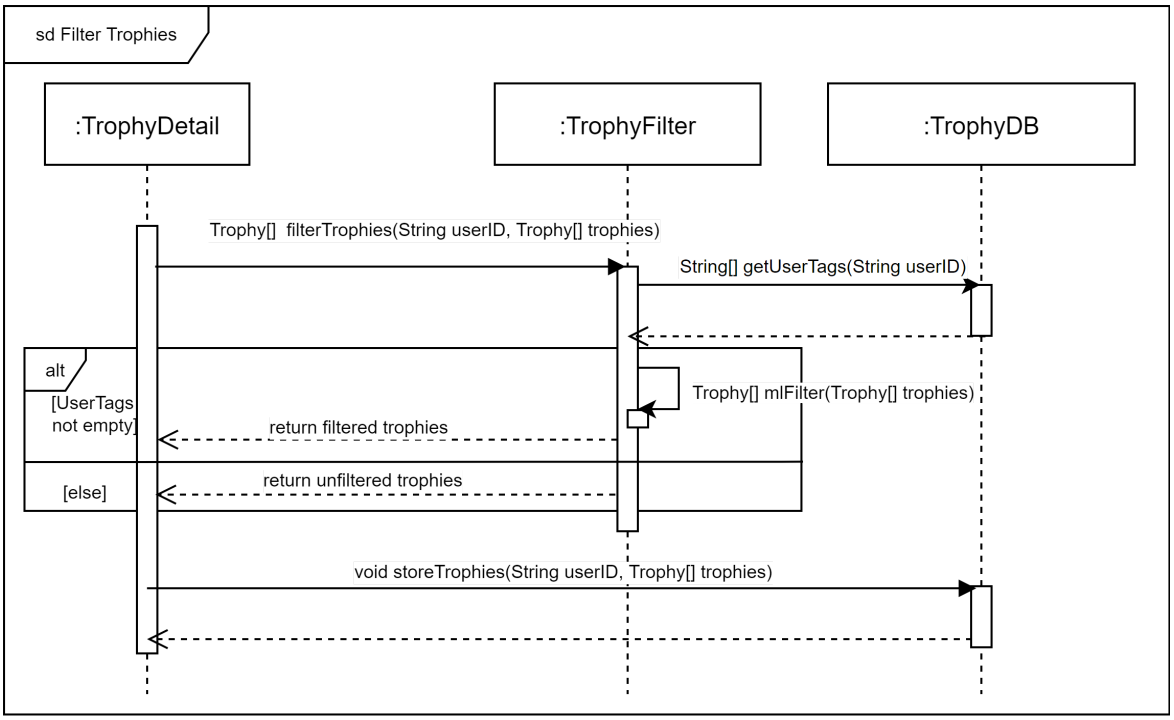
1.11. Sequence Diagrams

1.11.1 Login

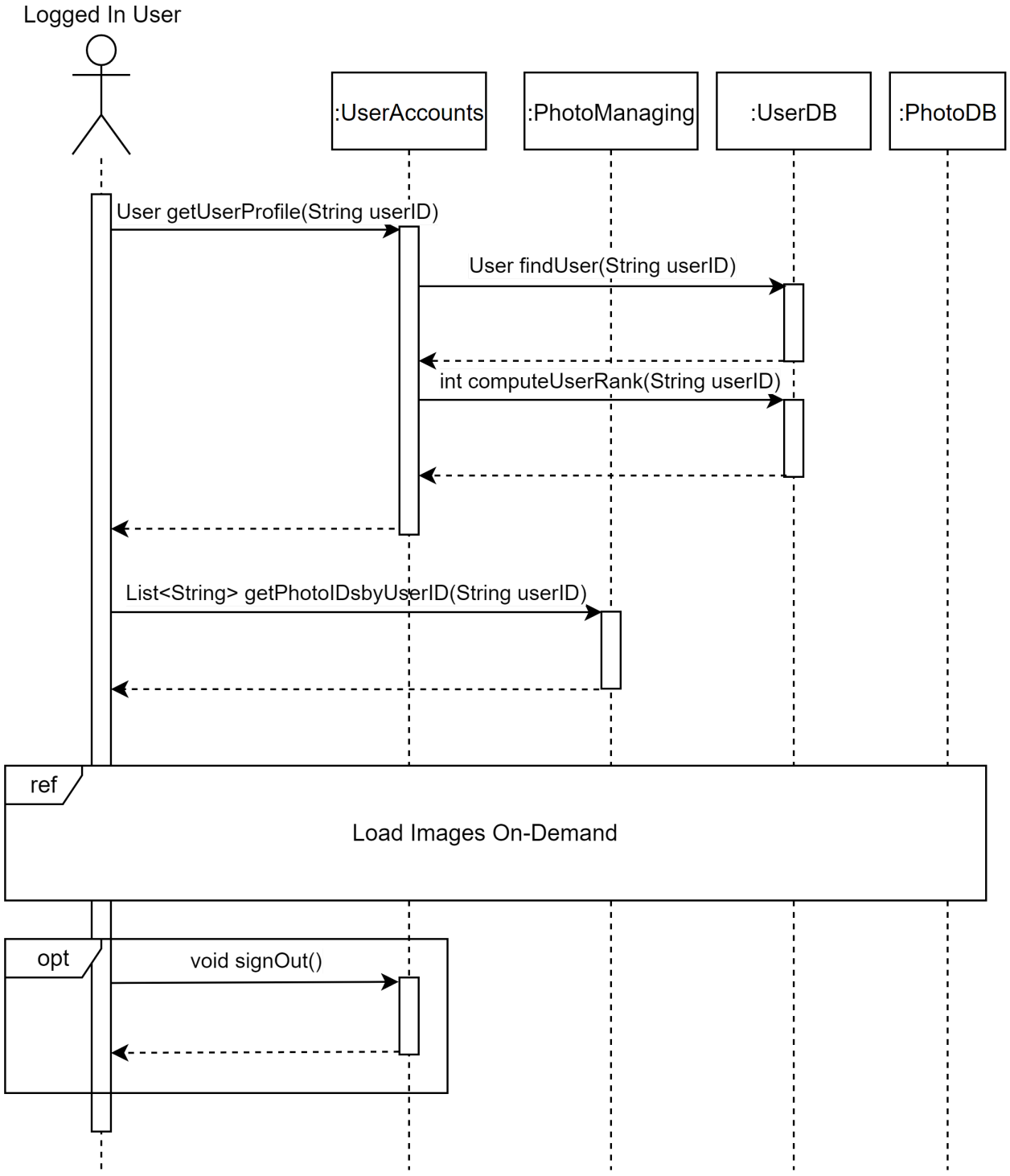


1.11.2 Browse Trophies on Map

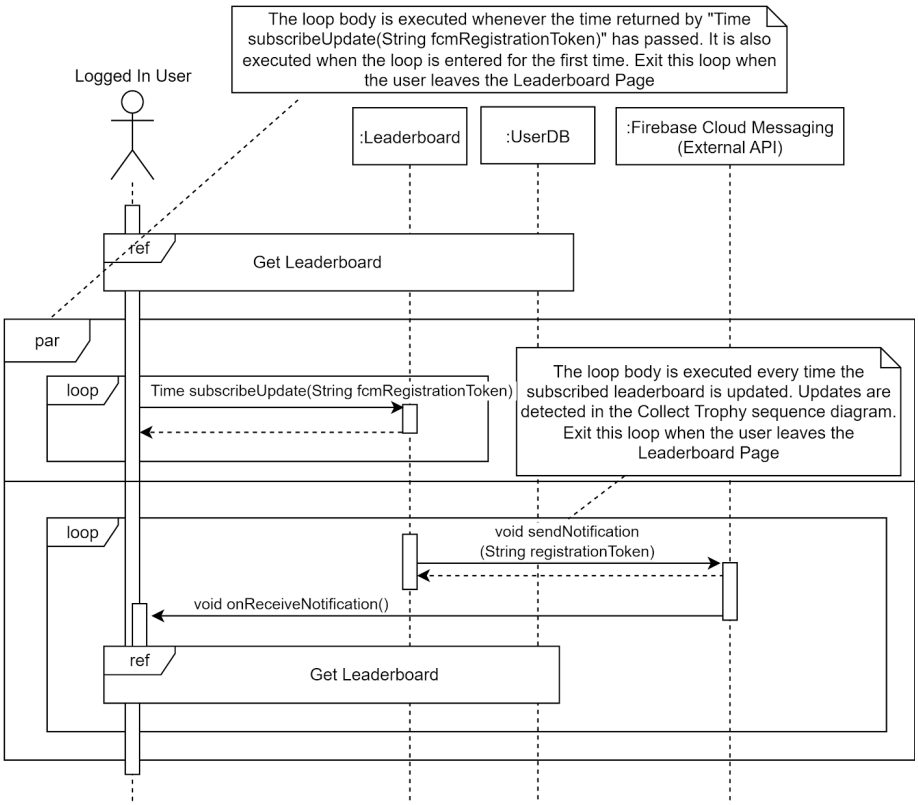


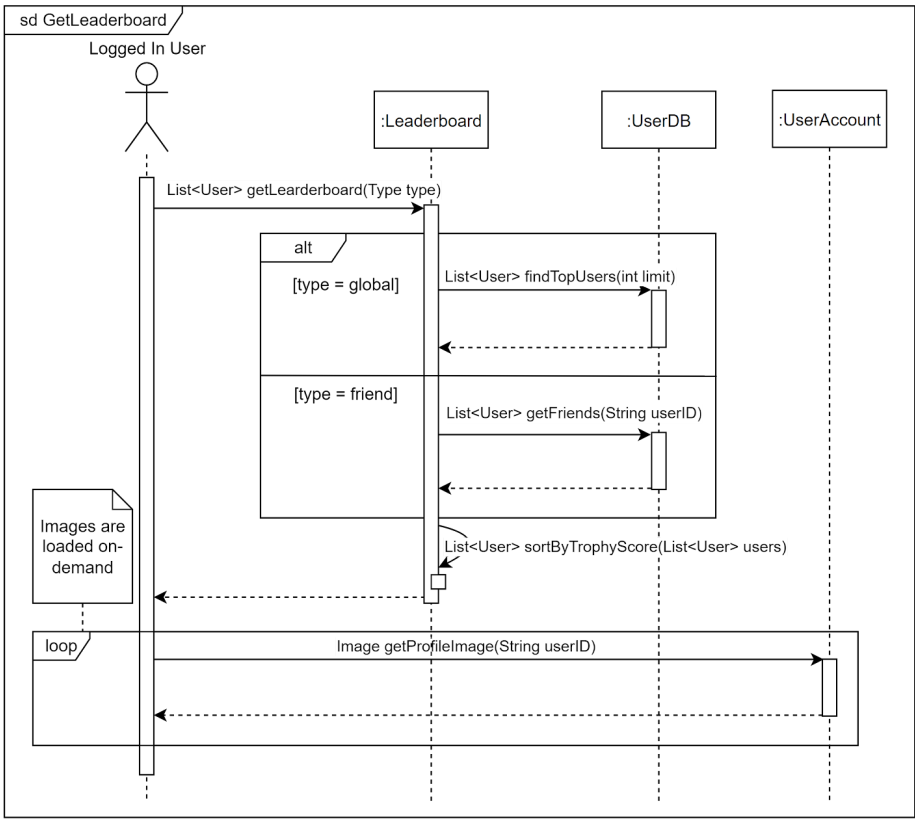

sd Filter Trophies
:TrophyDetail
:TrophyFilter
:TrophyDB
Trophy[] filterTrophies(String userID, Trophy[] trophies)
String[] getUserTags(String userID)
alt
[UserTags not empty]
Trophy[] mlFilter(Trophy[] trophies)
return filtered trophies
[else]
return unfiltered trophies
void storeTrophies(String userID, Trophy[] trophies)

1.11.3 View Profile

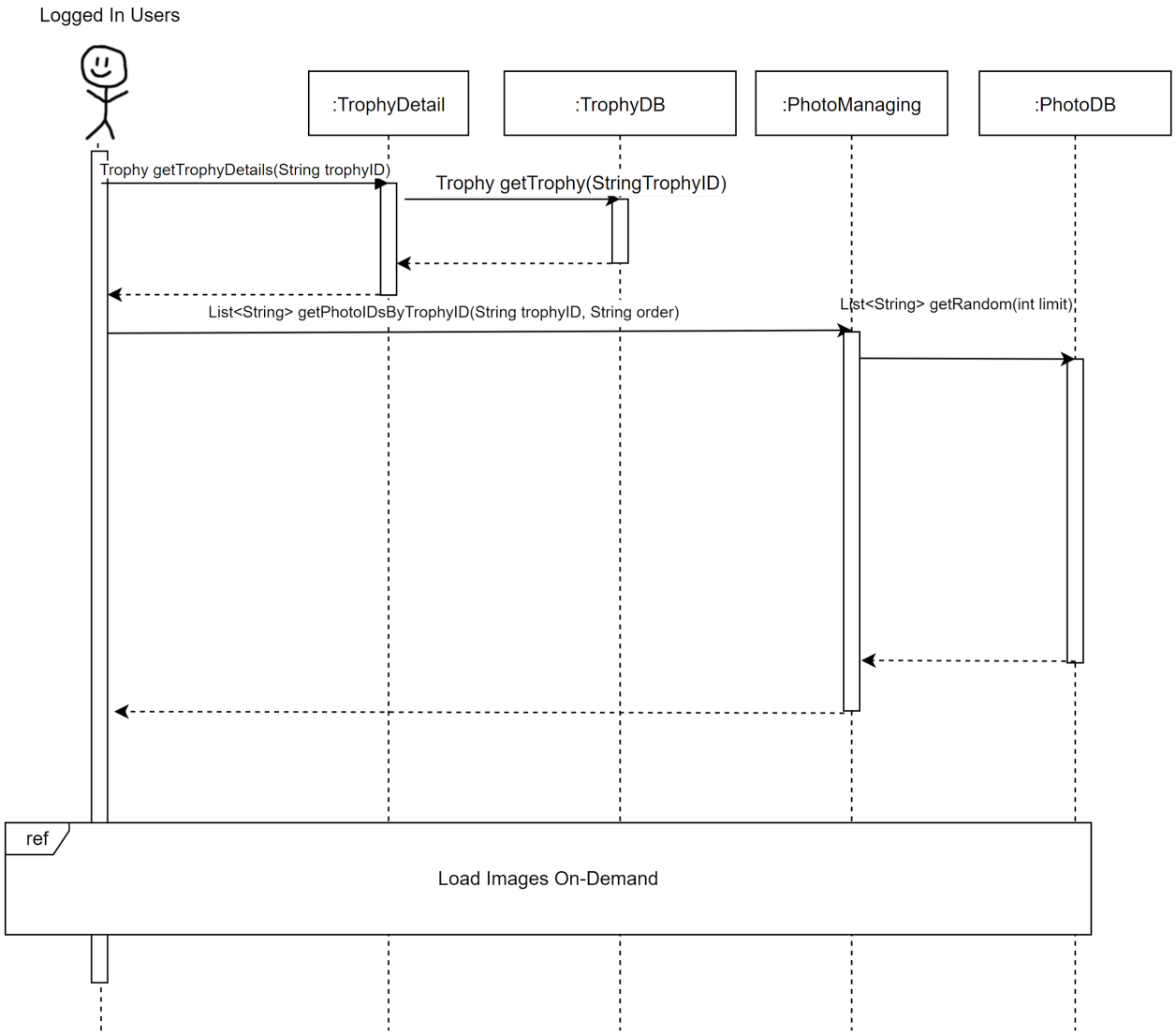


1.11.4 Watch Leaderboard

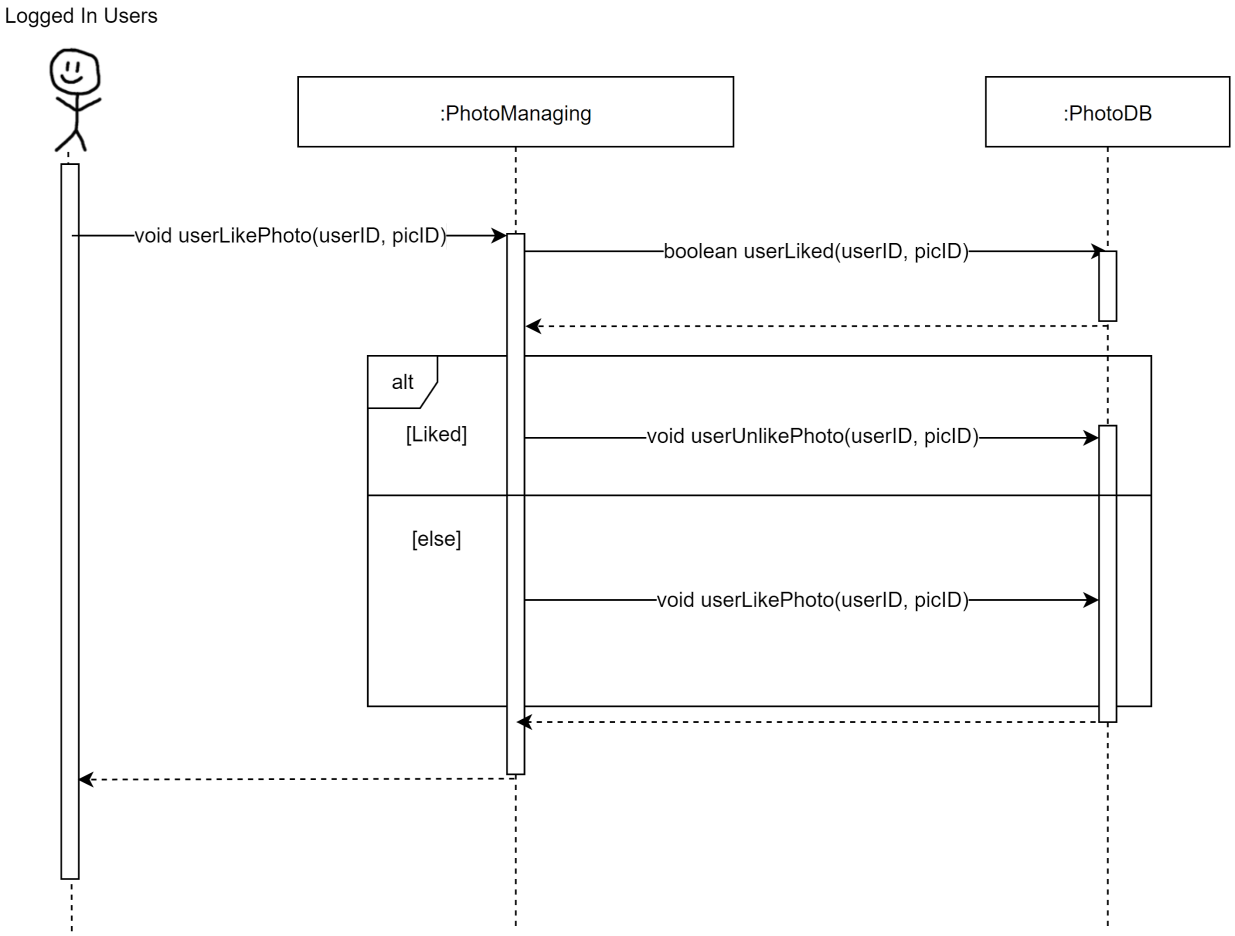




1.11.5 Review Trophy Details and Photos

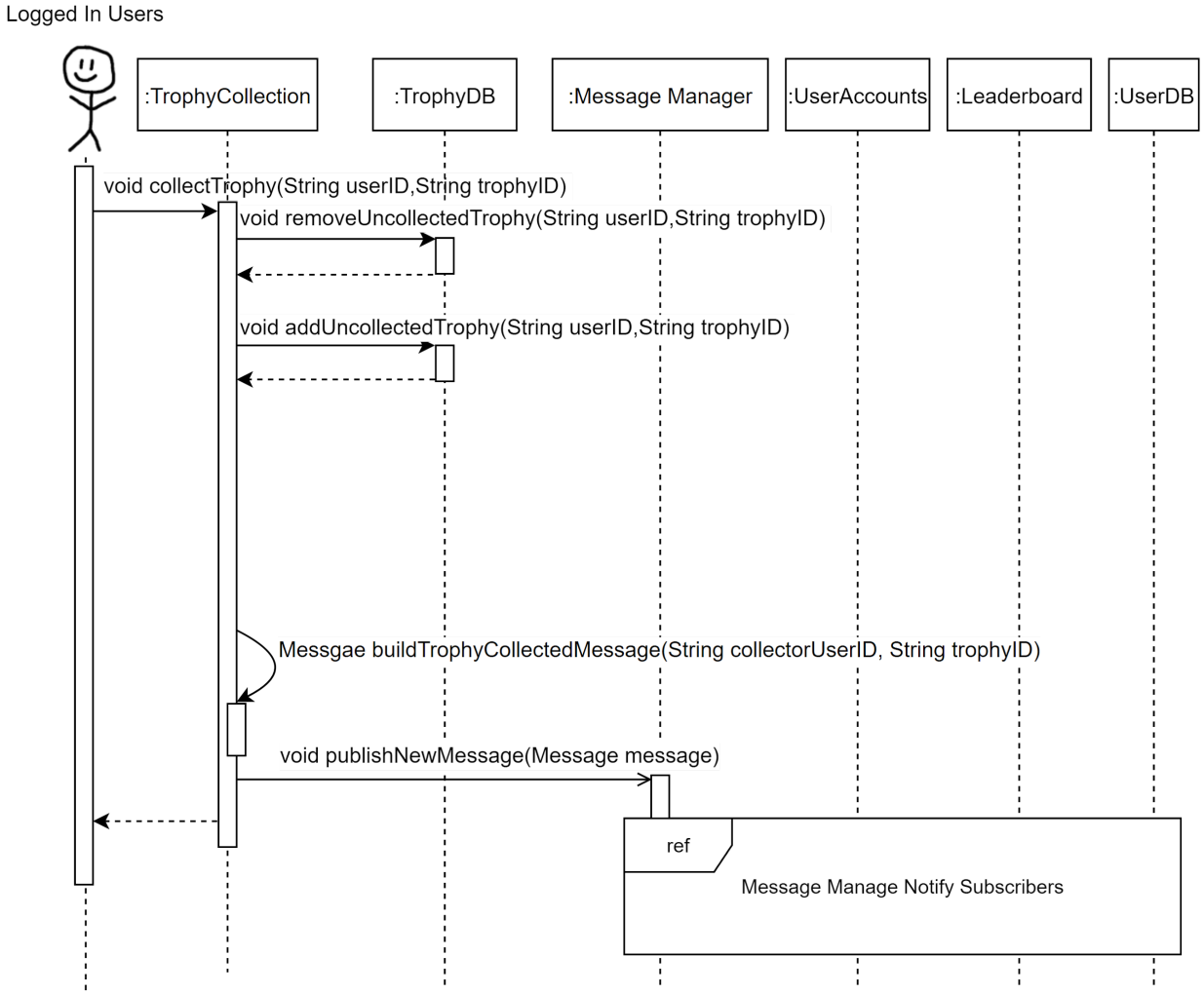


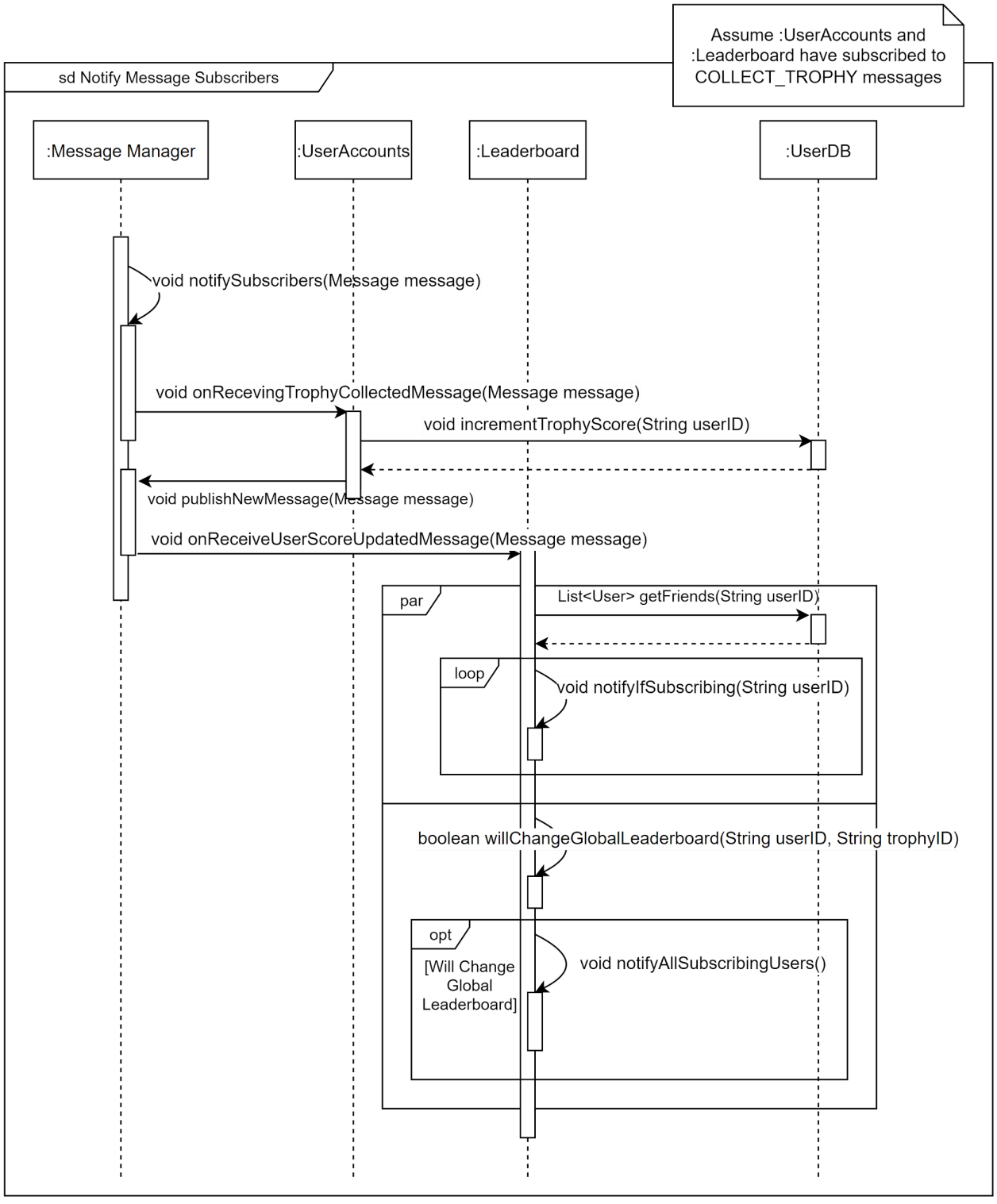
1.11.6 Evaluate Photos



1.11.7 Collect Trophy

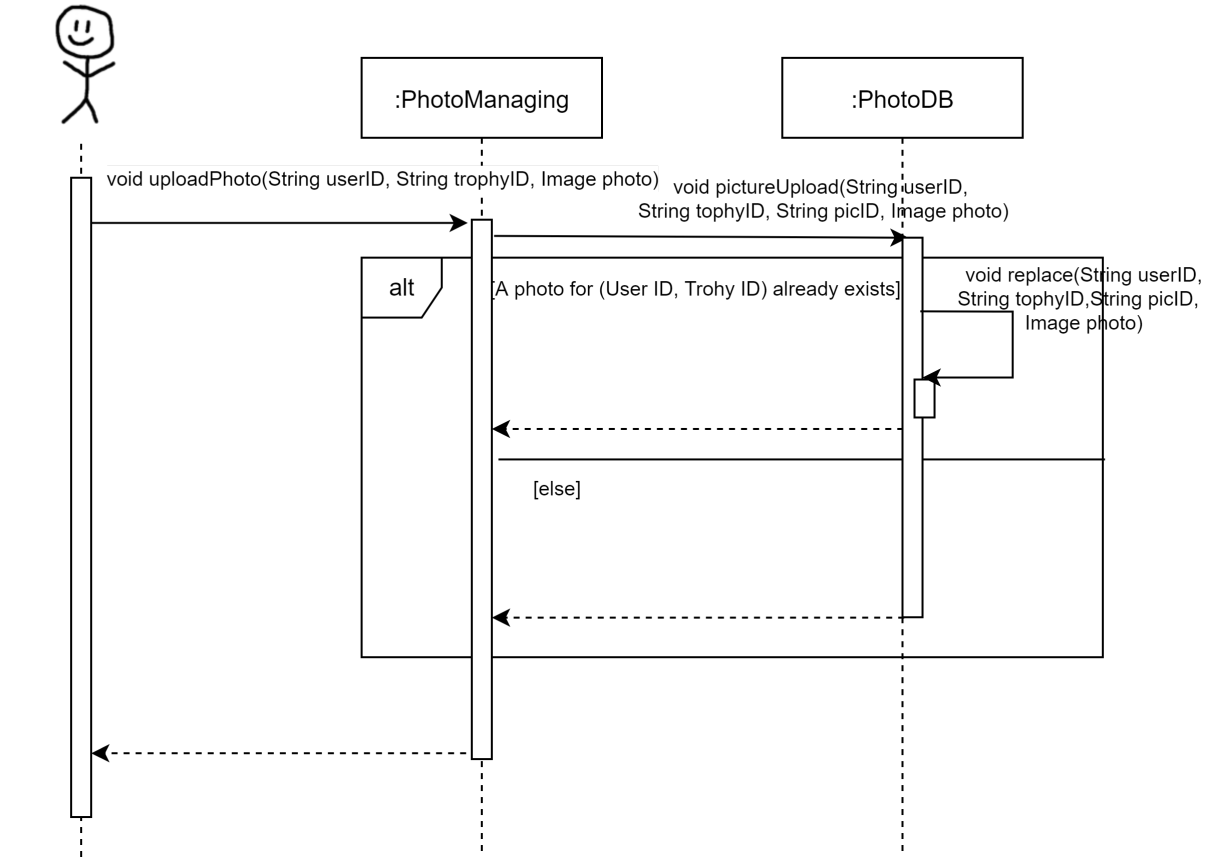
1.11.7.1 Collect Trophy




sd Notify Message Subscribers
Assume :UserAccounts and :Leaderboard have subscribed to COLLECT_TROPHY messages
:Message Manager
:UserAccounts
:Leaderboard
:UserDB
void notifySubscribers(Message message)
void onRecevingTrophyCollectedMessage(Message message)
void incrementTrophyScore(String userID)
void publishNewMessage(Message message)
void onReceiveUserScoreUpdatedMessage(Message message)
par
List<User> getFriends(String userID)
loop
void notifyIfSubscribing(String userID)
boolean willChangeGlobalLeaderboard(String userID, String trophyID)
opt
[Will Change Global Leaderboard]
void notifyAllSubscribingUsers()

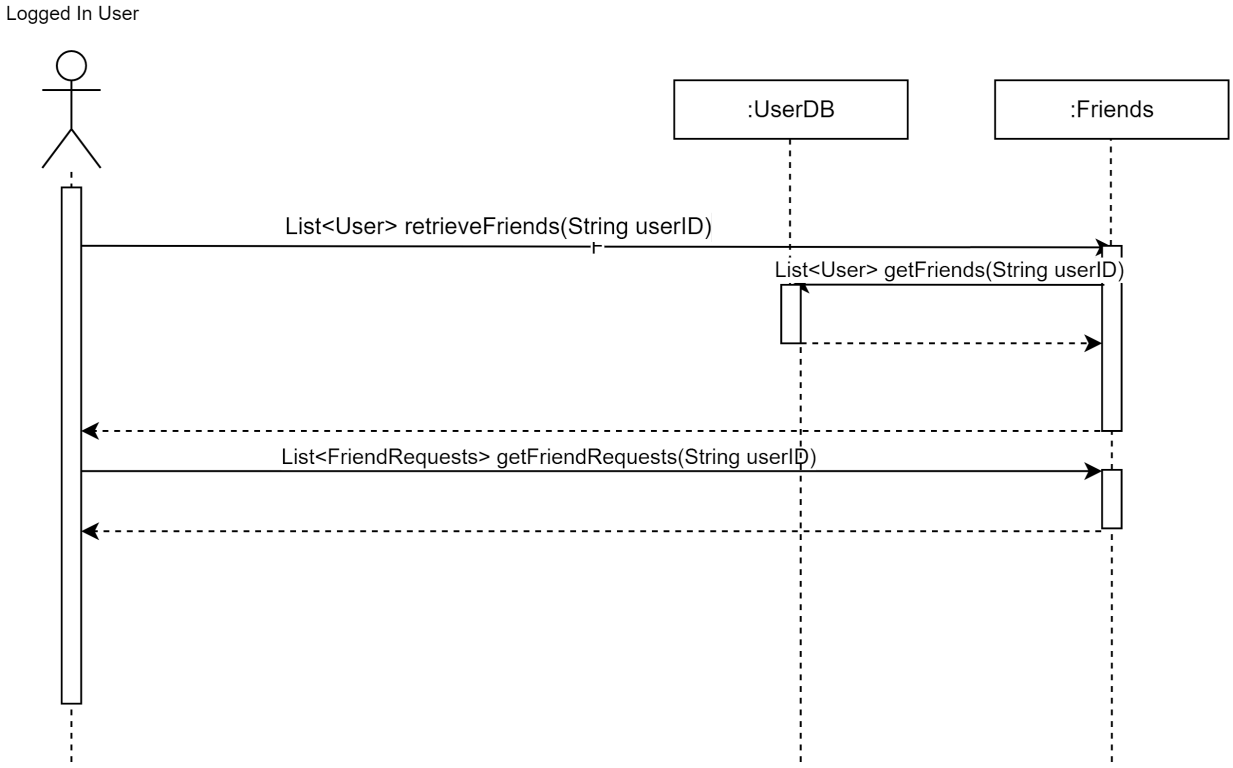
1.10.7.2 Take a Picture

Logged In Users

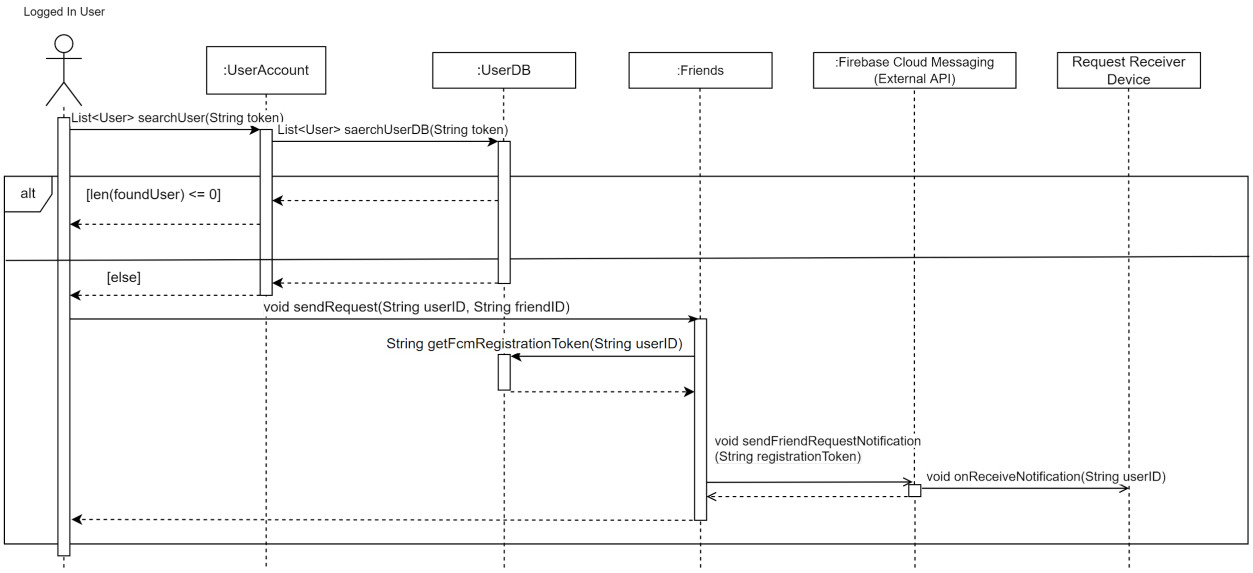


1.10.8 Manage Friends

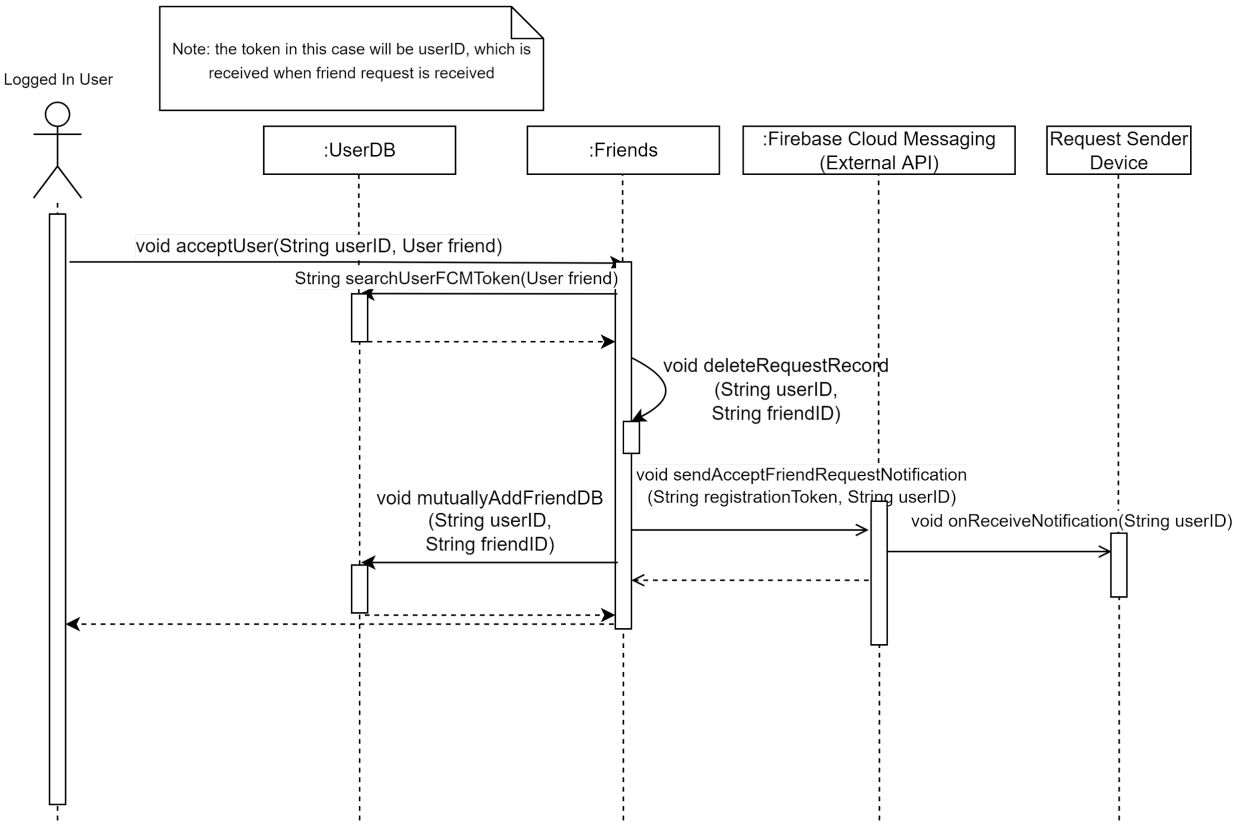
1.10.8.1 View Friends



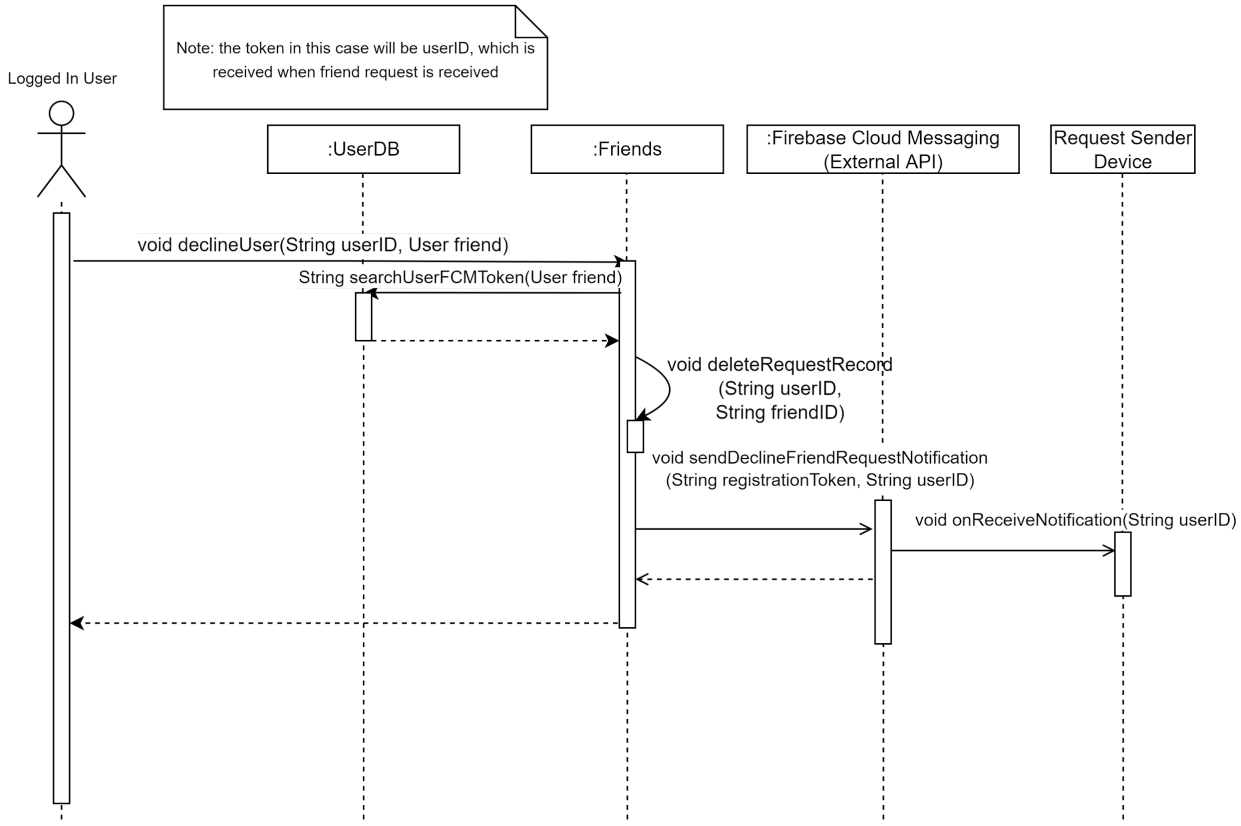
1.10.8.2 Send Friend Request



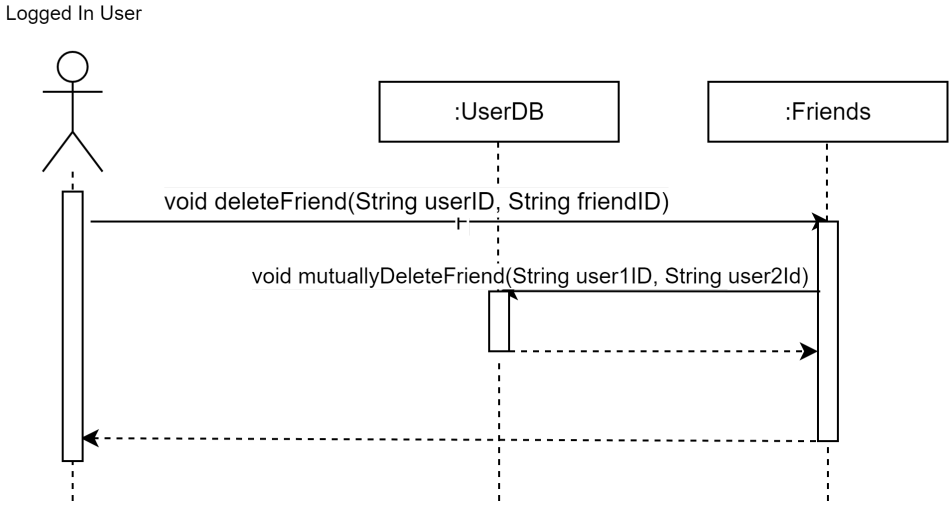
1.10.8.3 Accept Friend Request



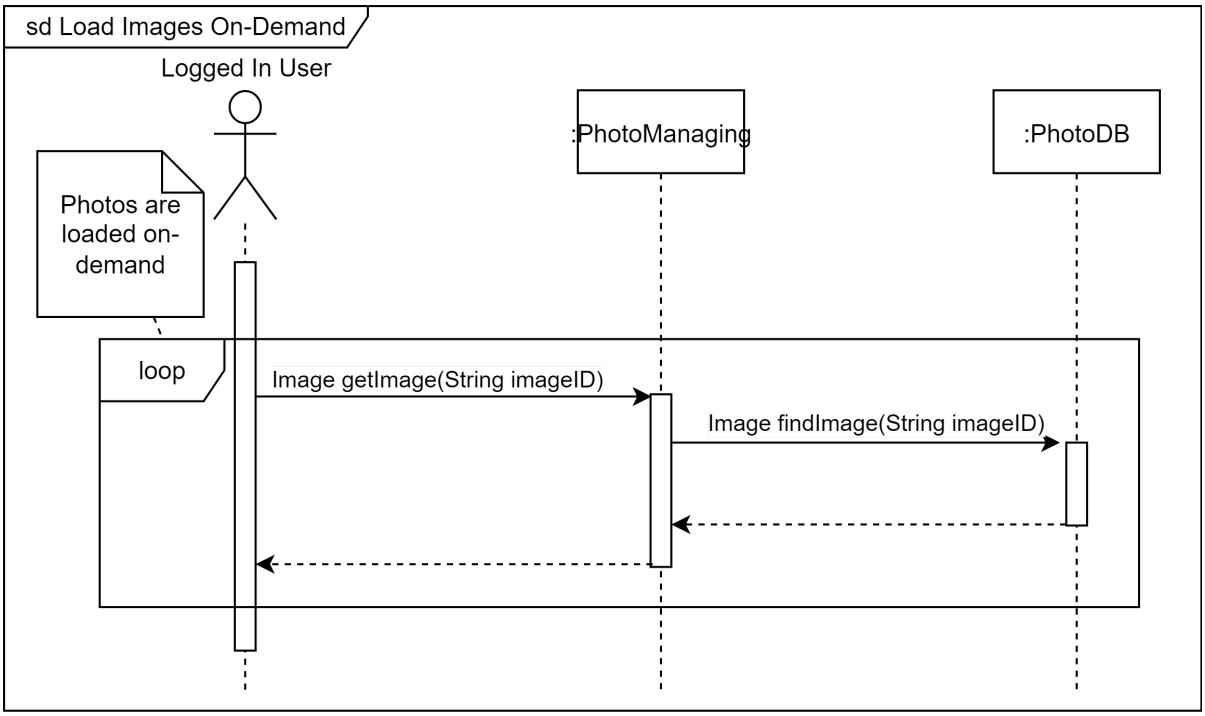
1.10.8.4 Decline Friend Request



1.10.8.4 Delete Friend



1.10.9 Common Sequence Diagrams



1.12. “Beyond the Minimal Scope” Functionality

The non-trivial functionality will be implemented as a function in the backend used to generate a list of curated locations. The input to the function is a list of Strings corresponding to historic tags for previous locations the user has encountered and a list of Strings corresponding to the unique tags found in the list of candidate locations returned by Places API. The main logic loop is to tokenize each of the tags into vectors and obtain their corresponding encodings via a call to the EmbeddingLayer. Then, cosine similarity is computed between each user and candidate tags. If the similarity is larger than a certain threshold, the proposed Trophy is added to a set. This is to avoid duplicated Trophies.

```
 private Trophy[] mlFilter(self, String[] UserTags, Trophy[]
candidateTrophies){

  filteredTrophySet = new HashSet<Trophy>();
  tokenizedUserTags = self.tokenize(UserTags);
  tokenizedCandidateTags = self.tokenize(candidateTrophies);

  encodedUserTags = self.EmbeddingLayer(tokenizedUserTags);
  encodedCandidateTags = self.EmbeddingLayer(tokenizedCandidateTags);

  for (int i=0, i < encodedCandidateTags.length(), i++){
```

```
     for (int j=0, j<i, j++){
        if (i!=j && cosineSimilarity(encodedUserTags[j],
encodedCandidateTags[i]) > self.THRESHOLD){
          filteredTrophySet.add(candidateTrophies[i]);
        }
     }
  }
  return List.copyOf(filteredTrophySet);
}
```

The non-trivial part of this is the creation of the EmbeddingLayer. This will require a data collection and training process. The data collection process involves a few steps

1. Create a list of relevant Tags based on Places API.
2. Download WikiText Corpus.
3. Filter Corpus for sentences including any of the relevant Tags.
4. For each of the retrieved sentences, create a copy with one of the words removed (not a stopword, which are words like the, a, etc). This is to generate fine-tuning data. The loss function is determined based on what type of embedding is used, Word2Vec, GloVe, etc.
5. Download GloVe or Word2Vec Common Crawl set pre-trained embeddings. This is our starting point.
6. Load EmbeddingLayer model with pre-trained weights. In addition to the weights, we will have access to the vocab associated with the Common Crawl set.

Then, for training, the procedure is simple. With the pre-trained weight loaded, we can simply use Tensorflow's implementation of the high level Keras library to run model.fit(). The corresponding loss function is minimized, improving the relevance of our embeddings.

1.13. Frameworks

A. MongoDB
   - MongoDB allows image storage by creating schemas with Mongoose. It is also a flexible database, which has a change-friendly design.
   - Alternative: NoSQL

B. Azure Cloud Provider
   - Free account is provided by UBC. It allows users to build, deploy, and manage applications easily and quickly.
   - Alternative: Google Cloud Server, Amazon Cloud Server, IBM Cloud Server

C. Notification Framework
   a. Firebase Cloud Messaging

- It allows users to send messages to specific topics using topic message; in other words, it can send notifications to specific users by subscribing them to an unique topic. Most importantly, it is free.
- Alternative: Backendless, Pubnub

D. Node.js
- It is used for the event-based server, and back-end API. It is designed to be cross-platform, and it is easy to use.

1.14. Code Review Results

1.14.1. Automatic Code Review

- Codacy Repository Dashbhttps://app.codacy.com/gh/CPEN-321-World-Exploration-Action/CPEN-321-World_Explo ration_Action/dashboardoard:
- Commit SHA: 5a96716ea66dc38a385c2b1e4fd527d9b6ee85d9
  - There are 0 issues left

Issues breakdown

All clear!

When issues come up, you'll be able to see here the issues breakdown per category.

1.14.2. Manual Code Review

Our team analyzed the front and backend of our peer group, and outlined a list of issues. We then shared these with the Peer Group and they selected two of these.

1.14.3.1. Two Major Issues Chosen By Peer Group

1. Security Issue: Google Places API Key is Uploaded to Repository
    - Private Keys should not be publicly available in the code base. Especially not as a raw String.
    - frontend/app/src/main/java/com/example/shopeer/EditSearchActivity.java

```
public class EditSearchActivity extends AppCompatActivity {
    private static final String TAG = "editSearchActivity";
    private static final String searchUrl = "http://20.230.148.126:8080/match/searches?email=";
    int SERVER_TIMEOUT_MS = 1000; // num ms wait for server
    private static final String API_KEY = "AIzaSyCbVC78h8NLcGxWfe6poNdqjQ-BvoNOB4A"; // Sally's API key for Google Places
```

2. Efficiency Issues In Backend
    a. Currently the backend sends a query for each email. A more efficient way is to combine them into one query:
    user_collection.find({ email : { $in : email_array } });

```
241  async function get_object_array_from_email_array(email_array) {
242      // console.log(email_array)
243      var array = []
244      for (let i = 0; i < email_array.length; i++) {
245          var return_cursor = await user_collection.findOne({ email: email_array[i] })
246          if (!return_cursor) {
247              throw "Error: Invalid email"
248          }
249          // console.log(return_cursor)
250          array.push(return_cursor)
251      }
252      // console.log(array)
253      return array
254  }
```

    b. Similarly: $pull can deal with different field simultaneously to reduce the time of searching matched element in the database. For example:

```
await user_collection.updateOne({ email: profile_email }, { $pull: { invites:
target_peer_email } })
await user_collection.updateOne({ email: profile_email }, { $pull: {
received_invites: target_peer_email } })
```

Can be updated to

```
await user_collection.updateOne({ email: profile_email }, { $pull: { invites:
target_peer_email, received_invites: target_peer_email } })
```

1.14.3.2. Other Issues Detected:

- Design Problem - Everything is under controllers.

```
84  user_profile_router.post("/registration", async (req, res) => {
85      var profile = req.query
86      try {
87          profile_email = req.query.email
88          var find_cursor = await user_collection.findOne({ email: profile_email })
89          if (find_cursor) {
90              res.status(200).send("User already exists")
91          } else {
92              var user_object = create_user_object(profile)
93              var result_debug = await user_collection.insertOne(user_object)
94              if (!result_debug) {
95                  res.status(400).json({response: "Failed to register user."})
96                  return
97              }
98              res.status(200).send(user_object)
99          }
100
101     } catch (err) {
102         console.log(err)
103         res.status(400).send(err)
104     }
105 })
106
107 function create_user_object(body) {
108     var user_object = {
109         name: body.name,
110         email: body.email,
111         description: body.description,
112         photo: body.photo,
113         // FCM_token: body.FCM_token,
114         searches: [],
115         peers: [],
116         invites: [],
117         received_invites: [],
118         blocked: []
119     }
120     return user_object
121 }
```

- Security Problem - Private Key is uploaded to repository
  In the peers.js file

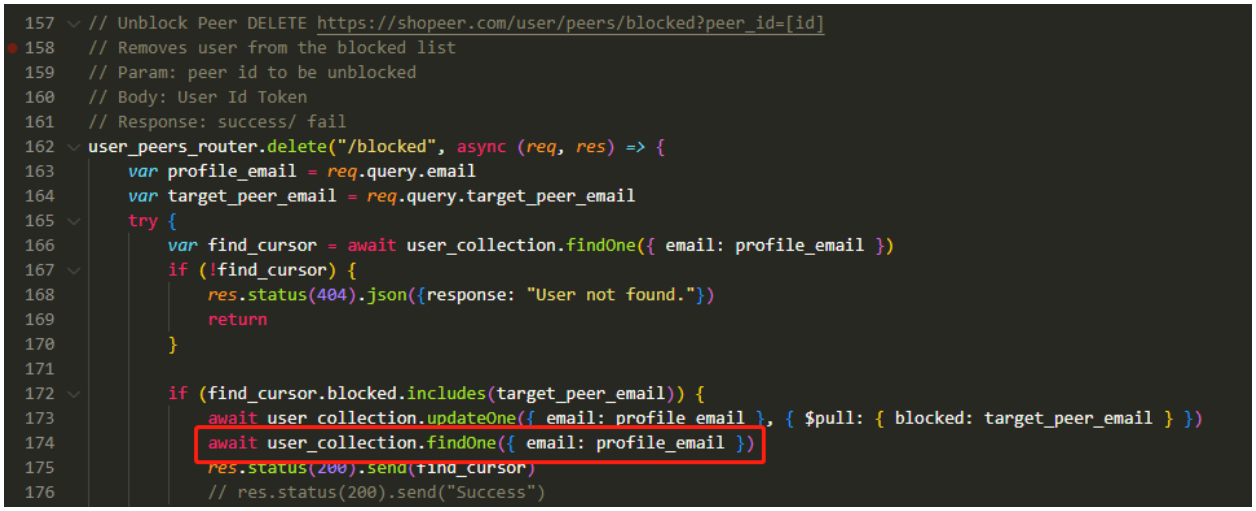
cpen-shopeer-firebase-adminsdk-i0ot8-ffcaa4fefb.json

```
1   {
2     "type": "service_account",
3     "project_id": "cpen-shopeer",
4     "private_key_id": "ffcaa4fefbd3114c2abddc2eaca5abdb364488b4",
5     "private_key": "-----BEGIN PRIVATE KEY-----\nMIIEuwIBADANBgkqhkiG9w0BAQEFAASCBKUwgg
6     "client_email": "firebase-adminsdk-i0ot8@cpen-shopeer.iam.gserviceaccount.com",
7     "client_id": "102887084242891488810",
8     "auth_uri": "https://accounts.google.com/o/oauth2/auth",
9     "token_uri": "https://oauth2.googleapis.com/token",
10    "auth_provider_x509_cert_url": "https://www.googleapis.com/oauth2/v1/certs",
11    "client_x509_cert_url": "https://www.googleapis.com/robot/v1/metadata/x509/firebase
12  }
13
```

- Design Problem - When user is blocked, he might need to be pulled from received_invites field as well

In the peers.js file

- Meaningless Statement - findOne statement is useless in this case
  In the peers.js document



- Security Issue - No User authentication for any room methods. Meaning knowledge of a user's email allows anyone to pull all chat room ids for that user, and then subsequently retrieve all chat history and delete the rooms.

```
/*************** Room submodule *************** */

/**
 * Get Chatrooms GET https://shopeer.com/chat/room/all?email=[email]
 * Gets an array of all chat rooms that a user is present in
 * Body: {"email": <user email>}

 * Response: array of room_objects

 */
// curl -X "GET" -H "Content-Type: application/json" -d '{"email": "[redacted]@gmail.com" }' localhost:8081/chat/room/all
router.get("/all", async (req, res) => {
    try {
        var roomsCursor = coll.find({
            "peerslist":{"$in":[req.query.email]}
        })
        roomArr = []
        await roomsCursor.forEach(getRooms = (room) => {
            // roomArr.push(room._id)
            roomArr.push(room)
        })

    res.status(200).send(roomArr)
    } catch (err) {
        console.log(err)
        res.status(400).send(err)
    }
})
```

Missing Authentication Middleware

```
/**
 * Remove Chatroom DELETE https://shopeer.com/chat/room?room_id=[room_id]
 * Deletes the chatroom and its history from the Room Collection
 * Response:success / fail
 */
// curl -X "DELETE" -H "Content-Type: application/json" -d '' localhost:8081/chat/room?room_id=62c4ac94ee79eff89f8ac0bc
router.delete("/", async (req, res) => {
    try {
        var doc = await coll.deleteOne({
            _id: ObjectId(req.query.room_id)
        })
        if (doc.deletedCount == 1) {
            res.status(200).send("\nRoom deleted\n")
        } else {
            res.status(200).send("\nroom does not exist.\n")
        }

    } catch (err) {
        console.log(err)
        res.status(400).send(err)
    }
})
```

Only room_id is needed to delete room

- Efficiency Issue - textViewMessage and messageTime TextViews could be identified only one time outside the classes.

```
    // Helper classes
    public class SenderViewHolder extends RecyclerView.ViewHolder {
        TextView textViewMessage;
        TextView messageTime;

        public SenderViewHolder(@NonNull View itemView) {
            super(itemView);
            textViewMessage = itemView.findViewById(R.id.senderMessage);
            messageTime = itemView.findViewById(R.id.messageTimeS);
        }
    }

    public class ReceivedViewHolder extends RecyclerView.ViewHolder {
        TextView textViewMessage;
        TextView messageTime;

        public ReceivedViewHolder(@NonNull View itemView) {
            super(itemView);
            textViewMessage = itemView.findViewById(R.id.receivedMessage);
            messageTime = itemView.findViewById(R.id.messageTimeR);
        }
    }
}
```

- Incorrect status code - https://github.com/Shopeer/Shopeer/blob/dev/backend/chat/room.js

```
77  // curl -X "PUT" -H "Content-Type: application/json" -d '{"email": "hello@gmail.com" }' localhost:80
78  router.put("/", async (req, res) => {
79      try {
80          var doc = await coll.updateOne(
81              {_id: ObjectId(req.query.room_id)},
82              {$addToSet: {"peerslist": req.body.email}}
83          )
84          if (doc.matchedCount == 0) {
85              res.status(200).send("\nCould not find this room.\n")
86          } else if (doc.modifiedCount == 0) {
87              res.status(200).send("\n" + req.body.email + " is already a member of this room\n")
88          } else {
89              res.status(200).send("\n" + req.body.email + " added to room\n")
90          }
91      } catch (err) {
92          console.log(err)
93          res.status(400).send(err)
94      }
95  })
```

- Efficiency - bubble sort is slow, consider using the default sort function: https://stackoverflow.com/a/7157638
  https://github.com/Shopeer/Shopeer/blob/dev/backend/match/suggestions_algo.js#L48

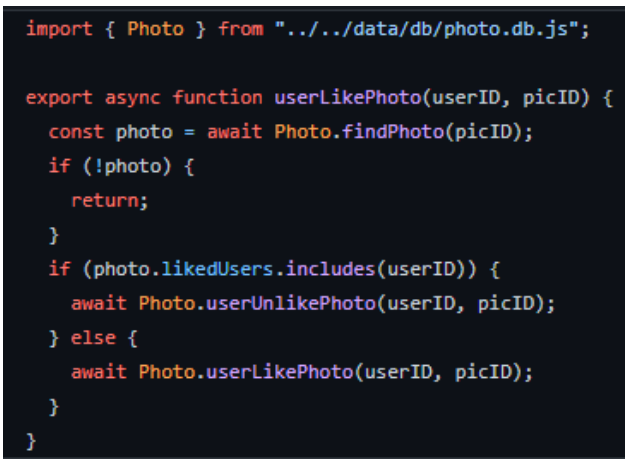
```
47
48          // Bubble sort
49          for (let i = 0; i < match_list.length-1; i++) {
50              console.log(match_list[i][1])
51              console.log(match_list[i+1][1])
52              if (match_list[i][1] < match_list[i+1][1]) {
53                  temp = match_list[i+1]
54                  match_list[i+1] = match_list[i]
55                  match_list[i] = temp
56              }
57          }
58
```

1.14.3.3. Two Major Issues In Our Code

1. No Error Checking for Certain Database functions
    - Unclear error return
    - Example: trophydetails.js lines 93-108

```
93  export async function getTrophyDetails(ids){
94      return await TrophyTrophy.find({trophy_id:{$in: ids}})
95  }
96
97  // Dev function
98  export async function getAllTrophies(){
99      return await TrophyTrophy.find({})
100 }
101
102 export async function getAllTrophiesUsers(){
103     return await TrophyUser.find({})
104 }
105
106 export async function createTrophy(req){
107     return await TrophyTrophy.create(req.body)
108 }
```

2. Lazy Class Code Smell
    - Photos module has 3 submodules that seem like they could be merged into one

photomanaging.js	

```
import { Photo } from "../../data/db/photo.db.js";

export async function userLikePhoto(userID, picID) {
  const photo = await Photo.findPhoto(picID);
  if (!photo) {
    return;
  }
  if (photo.likedUsers.includes(userID)) {
    await Photo.userUnlikePhoto(userID, picID);
  } else {
    await Photo.userLikePhoto(userID, picID);
  }
}
```

photosorting.js

```
import { Photo } from "../../data/db/photo.db.js";
import { BadRequestError } from "../../utils/errors.js";

export async function getPhotoIDsByUserID(userId) {
  return await Photo.getPhotosByUser(userId);
}

export async function getPhotoIDsByTrophyID(trophyID, order, userId) {
  let photos;

  if (order == "random") {
    photos = await Photo.getRandom(trophyID, 9);
  } else if (order == "time") {
    photos = await Photo.getSortedByTime(trophyID, 9);
  } else if (order == "like") {
    photos = await Photo.getSortedByLike(trophyID, 9);
  } else {
    throw new BadRequestError(`Invalid order: ${order}`);
  }

  photos = setUserLiked(photos, userId);

  return photos;
}

function setUserLiked(photos, userId) {
  return photos.map(({ likedUsers, ...attrs }) => ({
    ...attrs,
    user_liked: likedUsers.includes(userId),
  }));
}
```

photostoring.js

```
import { Photo } from "../../data/db/photo.db.js";

export async function uploadPhoto(userId, trophyId, photoId) {
  await Photo.addOrReplacePhoto(photoId, trophyId, userId);
}
```

1.14.3.4. Issue Fixed

1. No Error Checking When Calling Certain Database Functions
    - We addressed this issue by adding code to handle potential errors. A few examples of how we fixed the issue:
        - backend/src/services/trophies/trophycollection.js

```
8 -   TrophyUser.removeUncollectedTrophy(userId, trophyId);
9
```

```
9  +   const success = await TrophyUser.removeUncollectedTrophy(userId, trophyId);
10 +   if (!success) {
11 +     throw new BadRequestError("Invalid userId or trophyId");
12 +   }
13
```

- backend/src/services/users/friends.js

```
7    export async function retrieveFriends(userId) {
8  -   return await User.getFriends(userId);
```

```
8    export async function retrieveFriends(userId) {
9  +   try {
10 +     return await User.getFriends(userId);
11 +   } catch (err) {
12 +     throw new NotFoundError("Cannot find the user");
13 +   }
14   }
```

```
11 export async function getFriendRequests(userId) {
12     const requestors = friendRequests.get(userId);
```

```
16 export async function getFriendRequests(userId) {
17 +   const user = await User.findUser(senderId);
18 +   if (!user) {
19 +     throw new NotFoundError("Cannot find the user");
20 +   }
21 +
```

```
35   export async function deleteFriend(userId, friendId) {
36 -   await User.deleteFriend(userId, friendId);
37 -   await User.deleteFriend(friendId, userId);
```

```
53    export async function deleteFriend(userId, friendId) {
54  +   const result = await User.mutuallyDeleteFriend(userId, friendId);
55  +   if (!result) {
56  +     throw new BadRequestError("They are not friends.");
57  +   }
58    }
```

- backend/src/services/users/useraccounts.js

```
14    export async function getUserProfile(userId) {
15      const userDocument = await User.findUser(userId);
16      if (!userDocument) {
17  -     return null;
18      }
```

```
14    export async function getUserProfile(userId) {
15      const userDocument = await User.findUser(userId);
16      if (!userDocument) {
17  +     throw new NotFoundError("Could not find the user");
18      }
```

- We have a default error handler that catches exceptions thrown.
  - backend/src/app.js

```
40  /* error handler */
41  app.use((err, req, res, next) => {
42    console.log(err);
43    res.status(err.status || 500).json({
44      status: err.status,
45      message: err.message,
46    });
47  });
```

2. Lazy Class Code Smell Fix
   - The issue is that the Photos module has three submodules that could be merged into one. These are photosorting.js, photomanaging.js, and photostoring.js.
     In photosorting.js file

```
import { Photo } from "../../data/db/photo.db.js";
import { BadRequestError } from "../../utils/errors.js";

export async function getPhotoIDsByUserID(userId) {
  return await Photo.getPhotosByUser(userId);
}

export async function getPhotoIDsByTrophyID(trophyID, order, userId) {
  let photos;

  if (order == "random") {
    photos = await Photo.getRandom(trophyID, 9);
  } else if (order == "time") {
    photos = await Photo.getSortedByTime(trophyID, 9);
  } else if (order == "like") {
    photos = await Photo.getSortedByLike(trophyID, 9);
  } else {
    throw new BadRequestError(`Invalid order: ${order}`);
  }

  photos = setUserLiked(photos, userId);

  return photos;
}

function setUserLiked(photos, userId) {
  return photos.map(({ likedUsers, ...attrs }) => ({
    ...attrs,
    user_liked: likedUsers.includes(userId),
  }));
}

```

Photomanaging.js

```
import { Photo } from "../../data/db/photo.db.js";

export async function userLikePhoto(userID, picID) {
  const photo = await Photo.findPhoto(picID);
  if (!photo) {
    return;
  }
  if (photo.likedUsers.includes(userID)) {
    await Photo.userUnlikePhoto(userID, picID);
  } else {
    await Photo.userLikePhoto(userID, picID);
  }
}

```

photostoring.js

```
import { Photo } from "../../data/db/photo.db.js";

export async function uploadPhoto(userId, trophyId, photoId) {
  await Photo.addOrReplacePhoto(photoId, trophyId, userId);
}

```

- To fix this, we merged all of these files into one photomanaging.js file, and updated the photos.controllers.js file to only import this file.

https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/bugfix/peer_review/backend/src/services/photos/photomanaging.js

```
import { Photo } from "../../data/db/photo.db.js";
import { BadRequestError } from "../../utils/errors.js";

/* Managing */
export async function userLikePhoto(userID, picID) {
  const photo = await Photo.findPhoto(picID);
  if (!photo) {
    return;
  }
  if (photo.likedUsers.includes(userID)) {
    await Photo.userUnlikePhoto(userID, picID);
  } else {
    await Photo.userLikePhoto(userID, picID);
  }
}

/* Storing */
export async function uploadPhoto(userId, trophyId, photoId) {
  await Photo.addOrReplacePhoto(photoId, trophyId, userId);
}

/* Sorting */
export async function getPhotoIDsByUserID(userId) {
  return await Photo.getPhotosByUser(userId);
}

export async function getPhotoIDsByTrophyID(trophyID, order, userId) {
  let photos;

  if (order == "random") {
    photos = await Photo.getRandom(trophyID, 9);
  } else if (order == "time") {
    photos = await Photo.getSortedByTime(trophyID, 9);
  } else if (order == "like") {
    photos = await Photo.getSortedByLike(trophyID, 9);
  } else {
    throw new BadRequestError(`Invalid order: ${order}`);
  }

  photos = setUserLiked(photos, userId);

  return photos;
}

function setUserLiked(photos, userId) {
  return photos.map(({ likedUsers, ...attrs }) => ({
    ...attrs,
    user_liked: likedUsers.includes(userId),
  }));
}
```

1.15. Test Plan

1.15.1. Backend Module Tests Design and Locations

- **Users**
    - Friends
        - List<User> retrieveFriends(String userId)

Scenario	Input	Expected Result	Location in Git
No user found	[assume: Mike is not one of the users] “Mike”	Throws an Error	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/e2d6cf472ad8af67ace43884b7efe8de090e51bf/backend/src/services/users/__tests__/friends.test.js#L29
User found, and the user has friends	[assume: Mike2 is a user, and there are friends in the user’s friend-list field] “Mike2”	Return: String array of user IDs	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/e2d6cf472ad8af67ace43884b7efe8de090e51bf/backend/src/services/users/__tests__/friends.test.js#L35
User found, and the user has no friend	[assume: Mike3 is a user, and there is no friend in the user’s friend-list field] “Mike3”	Return: []	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/e2d6cf472ad8af67ace43884b7efe8de090e51bf/backend/src/services/users/__tests__/friends.test.js#L41
“userid” field is invalid	null	Throws an Input Error	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/e2d6cf472ad8af67ace43884b7efe8de090e51bf/bac

			kend/src/services/users/__tests__/friends.test.js#L46

- List<User> getFriendRequests(String userID)

Scenario	Input	Expected Result	Location in Git
No friend request associates to the user ID is found	[assume: Mike has no friend request] “Mike”	Return: []	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/e2d6cf472ad8af67ace43884b7efe8de090e51bf/backend/src/services/users/__tests__/friends.test.js#L54
Friend request associates to the user ID is found, and the requestors of the friend requests are found in database	[assume: Mike2 has friends request] “Mike2”	Return: List<User>	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/e2d6cf472ad8af67ace43884b7efe8de090e51bf/backend/src/services/users/__tests__/friends.test.js#L58
“userid” field is null	null	Throws an Input Error	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/e2d6cf472ad8af67ace43884b7efe8de090e51bf/backend/src/services/users/__tests__/friends.test.js#L64

- void sendRequest(String senderId, String targetId)

Scenario	Input	Expected Result	Location in Git
“senderId” and “targetId” are found in the userDB, and “senderId” has sent a request to “targetId” before; “targetID” user has a fcm token	[assume: Sender has sent a request to Target before] “Sender", “Target”	Throws a BadRequestError and the target user does not receive a notification.	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/e2d6cf472ad8af67ace43884b7efe8de090e51bf/bac

			kend/src/services/users/__tests__/friends.test.js#L70
“senderId” and “targetId” are found in the userDB	[assume: Sender has not sent a request to Target before] “Sender2", “Target”	The request is stored in the map The "New Friend Request" friend request is received by “Target”	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/e2d6cf472ad8af67ace43884b7efe8de090e51bf/backend/src/services/users/__tests__/friends.test.js#L78
The sender and the target are already friends	[assume: The sender and the target are already friends] “Sender", “Target”	Throws BadRequestError, no FCM message is sent	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/e2d6cf472ad8af67ace43884b7efe8de090e51bf/backend/src/services/users/__tests__/friends.test.js#L78
“senderId” is not found in the userDB	[assume: Sender3 is not found in userDB] “Sender3”, “Target”	Throws NotFoundError, no FCM message is sent to the target user	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/e2d6cf472ad8af67ace43884b7efe8de090e51bf/backend/src/services/users/__tests__/friends.test.js#L89
“senderId” and “targetId” are found in the userDB, and “targetID” user does not have a fcm token	[assume: TargetNoToken has no fcm token] “Sender”, “TargetNoToken”	The request is stored in the map No FCM message is sentt	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/e2d6cf472ad8af67ace43884b7efe8de090e51bf/backend/src/services/users/__tests__/friends.test.js#L94
“targetId” is not found in userDB	[assume: TargetNotInDB is not	Throws NotFoundError	https://github.com/CPEN-321-World-Explor

	in userDB] “SenderNotSent”, “TargetNotInDB”		ation-Action/CPEN-321-World_Exploration_Action/blob/e2d6cf472ad8af67ace43884b7efe8de090e51bf/backend/src/services/users/__tests__/friends.test.js#L100
“senderId” and “targetId” are the same	“Test_user_1”, “Test_user_1”	Throws a BadRequestError	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/e2d6cf472ad8af67ace43884b7efe8de090e51bf/backend/src/services/users/__tests__/friends.test.js#L105
“senderId” and “targetId” are both null	null, null	An null entry is set in friendRequests Map with value of null being pushed in "User null does not have an fcm_token" is displayed. No "New Friend Request" friend request is sent	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/e2d6cf472ad8af67ace43884b7efe8de090e51bf/backend/src/services/users/__tests__/friends.test.js#L109

- void deleteFriend(String userId, String friendId)

Scenario	Input	Expected Result	Location in Git
Successfully deleted friends mutually	“User”, “Target”	“User” and “Target” (the ID) are pull from each other’s friends field	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/e2d6cf472ad8af67ace43884b7efe8de090e51bf/backend/src/services/users/__tests__/friends.test.js#L115
Two users are mutually not friends	[assume: UserStranger and TargetStranger are not friends]	Throws a BadRequestError	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration

	“UserStranger”, “TargetStranger”		_Action/blob/e2d6cf472ad8af67ace43884b7efe8de090e51bf/backend/src/services/users/__tests__/friends.test.js#L121
Either or both user’s id is/are not found in userDB	[assume: UserNotInDB is not found in userDB] “UserNotInDB”, “Target”	Throws a BadRequestErrorNothing	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/e2d6cf472ad8af67ace43884b7efe8de090e51bf/backend/src/services/users/__tests__/friends.test.js#L125
Either or both user’s id is/are null (illegal ID)	“User”, null	Throws an Error	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/e2d6cf472ad8af67ace43884b7efe8de090e51bf/backend/src/services/users/__tests__/friends.test.js#L129

- void acceptUser(String userId, String friendId)

Scenario	Input	Expected Result	Location in Git
Successfully accepted the friend request, the sender has FCM token	“User”, “Friend”	“User” and “Friend” add each other to their friend fields "Accepted Friend Request" is sent to “Friend” with message of User’s name plus “ accepted your friend request”	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/e2d6cf472ad8af67ace43884b7efe8de090e51bf/backend/src/services/users/__tests__/friends.test.js#L135
No such request	[assume: friendRequests has not entry of UserNotInMap] ”UserNotInMap”, “Friend”	Throws a BadRequestError	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/e2d6cf472ad8af67ace43884b

			7efe8de090e51bf/backend/src/services/users/__tests__/friends.test.js#L143
Successfully accepted the friend request, the sender does not have FCM token	[assume: FriendNoToken is not found in userDB] “User”, “FriendNoToken”	“User” and “Friend” add each other to their friend fields No FCM message is sent.	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/e2d6cf472ad8af67ace43884b7efe8de090e51bf/backend/src/services/users/__tests__/friends.test.js#L147
Invalid userId	null, “Friend”	Throws TypeError	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/e2d6cf472ad8af67ace43884b7efe8de090e51bf/backend/src/services/users/__tests__/friends.test.js#L156
Invalid friendId	“User”, null	Throws TypeError	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/e2d6cf472ad8af67ace43884b7efe8de090e51bf/backend/src/services/users/__tests__/friends.test.js#L160

- void declineUser(String userId, String friendId)

Scenario	Input	Expected Result	Location in Git
Successfully declined the friend request	“User”, “Friend”	The friendRequests removes “Friend” from the entry indexed by “User” "Declined Friend Request" is sent to “Friend” with message of User’s	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/e2d6cf472ad8af67ace43884b7efe8de090e51bf/backend/src/services/users/__tests__/friends.t

		name plus ” declined your friend request”	est.js#L166
No such entry found in friendRequests	[assume: friendRequests has not entry of UserNotInMap] ”UserNotInMap”, “Friend”	Throws an Error	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/e2d6cf472ad8af67ace43884b7efe8de090e51bf/backend/src/services/users/__tests__/friends.test.js#L176
friendId has no fcm_token	[assume: FriendNoToken is not found in userDB] “User”, “FriendNoToken”	No "Declined Friend Request" is sent. "User FriendNotInDB does not have an fcm_token" is shown in terminal	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/e2d6cf472ad8af67ace43884b7efe8de090e51bf/backend/src/services/users/__tests__/friends.test.js#L180
Invalid userId	null, “Friend”	Throws TypeErrorInput Error	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/e2d6cf472ad8af67ace43884b7efe8de090e51bf/backend/src/services/users/__tests__/friends.test.js#L189
Invalid friendId	“User”, null	Throws TypeErrorInput Error	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/e2d6cf472ad8af67ace43884b7efe8de090e51bf/backend/src/services/users/__tests__/friends.test.js#L193

- Leaderboard
  - void onReceiveUserScoreUpdatedMessage(Message message)

Scenario	Input	Expected Result	Location in Git

Successfully handled score updated message, global leaderboard is updated, the top 1 user is changed, and some user dropped out from the leaderboard	{
	userId: “User”,
	}
		A multicasting of leaderboard update is sent to all subscribers
		
		A message with the topic “new_champion” with “User”’s name and score is sent to all subscribers of the topic “new_champion”
		
		Messages sent to each user who dropped out from the leaderboard to notify their drop
			https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/e2d6cf472ad8af67ace43884b7efe8de090e51bf/backend/src/services/users/__tests__/leaderboard.test.js#L33

Global leaderboard updated, but nobody drops out from the leaderboard and the champion does not change	{
	userId: “User”,
	}
		A multicasting of leaderboard update is sent to all subscribers
		No other fcm message is sent
			https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/e2d6cf472ad8af67ace43884b7efe8de090e51bf/backend/src/services/users/__tests__/leaderboard.test.js#L58

Global leader board is not updated, and the user has friends who are subscribing	{
	userId: “UserNoGlobal”,
	}
		An FCM message is sent to the user's friends who are subscribing	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/e2d6cf472ad8af67ace43884b7efe8de090e51bf/backend/src/services/users/__tests__/leaderboard.test.js#L82

Global leader board is not updated, userId has no friend	[assume: “UserNoFriend” has no friend]
	{
	userId: “UserNoFriend”,
	}
		No FCM message sent	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/e2d6cf472ad8af67ace43884b7efe8de090e51bf/backend/src/services/users/__tests__/leaderbo

			ard.test.js#L105

userId is not in the userDB	[assume: “UserNotInDB” cannot be found in userDB]
	{
	userId: “UserNotInDB”,
	}
		Throws an Error	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/e2d6cf472ad8af67ace43884b7efe8de090e51bf/backend/src/services/users/__tests__/leaderboard.test.js#L123

Global leader board is not updated, userId has no friend who subscribes	[assume: “UserNoSubscriber” has no friend who subscribes]
	{
	userId: “UserNoSubscriber”,
	}
		No FCM message sent	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/e2d6cf472ad8af67ace43884b7efe8de090e51bf/backend/src/services/users/__tests__/leaderboard.test.js#L127

no userId field in the message	{
	
	}
		Throws anError	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/e2d6cf472ad8af67ace43884b7efe8de090e51bf/backend/src/services/users/__tests__/leaderboard.test.js#L146

- List<User> getGlobalLeaderboard()

Scenario	Input	Expected Result	Location in Git
Successfully executed, number of users exceeds or equals the number of users need to be in the leaderboard		Return: An array of users sorted by their score in descending order. The number of items in array is limited by the number of users need to be in the leaderboard	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/e2d6cf472ad8af67ace43884b7efe8de090e51bf/backend/src/services/users/__tests__/leaderboard.test.js#L236

Successfully executed, number of users is less than the number of users need to be in the leaderboard	[redacted]	Return: An array of all users sorted by their score in descending order.	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/e2d6cf472ad8af67ace43884b7efe8de090e51bf/backend/src/services/users/__tests__/leaderboard.test.js#L253

- List<User> getFriendLeaderboard(String userId)

Scenario	Input	Expected Result	Location in Git
Successfully get the friend leaderboard of the userId	“User”	Return: An array of User’s friend (include User himself) sorted by their score in descending order	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/e2d6cf472ad8af67ace43884b7efe8de090e51bf/backend/src/services/users/__tests__/leaderboard.test.js#L185
The userId has no friend	“UserNoFriend”	Return: An array contains only the User’s score	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/e2d6cf472ad8af67ace43884b7efe8de090e51bf/backend/src/services/users/__tests__/leaderboard.test.js#L203
userId is not in userDB	[assume: “UserNotIn” is not in the userDB] “UserNotIn”	Throws Not In DB Error	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/e2d6cf472ad8af67ace43884b7efe8de090e51bf/backend/src/services/users/__tests__/leaderboard.test.js#L221
Invalid userId	null	Throws Input Error	https://github.com/CPEN-321-World-Exploration-Action/CPEN-3

			21-World_Exploration_Action/blob/e2d6cf472ad8af67ace43884b7efe8de090e51bf/backend/src/services/users/__tests__/leaderboard.test.js#L227

- **Time subscribeUpdate(String userId, String fcmRegistrationToken)**

Scenario	Input	Expected Result	Location in Git
Successfully set user as subscriber	“User”, “fcmToken”	Return: An expire time, which is calculated by the time call the function plus validDuration “User” is added to “subscribers”, along with his userId, the “fcmToken” and the expire time	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/e2d6cf472ad8af67ace43884b7efe8de090e51bf/backend/src/services/users/__tests__/leaderboard.test.js#L154
validDuration is null	“User”, “fcmToken”	Return: An expire time, which is the time call the function “User” is added to “subscribers”, along with his userId, the “fcmToken” and the expire time	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/e2d6cf472ad8af67ace43884b7efe8de090e51bf/backend/src/services/users/__tests__/leaderboard.test.js#L166
Either or both userId or/and fcmRegistrationToken is/are invalid input(s)	null, undefined	Throws Input Error	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/e2d6cf472ad8af67ace43884b7efe8de090e51bf/backend/src/services/users/__tests__/leaderboard.test.js#L174

- UsersAccounts
  - void onReceiveTrophyCollectedMessage(Message message)

Scenario	Input	Expected Result	Location in Git

Successfully received trophy collected message	{
	userId: “User”,
	trophyScore: “Score”,
	}
		“User”’s score in userDB is incremented by the amount of “Score”
		
		Called leaderboard.onReceiveUserScoreUpdated Message with the “User”’s user ID being contained in the message field of userId
			https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/e2d6cf472ad8af67ace43884b7efe8de090e51bf/backend/src/services/users/__tests__/useraccounts.test.js#L33

userId is invalid	{
	userId: null,
	trophyScore: “Score”,
	}
		Nothing changes in userDB
		
		Throws Input Error
			https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/e2d6cf472ad8af67ace43884b7efe8de090e51bf/backend/src/services/users/__tests__/useraccounts.test.js#L51

userId is not found in userDB	[assume: “UserNotIn” is not in the userDB]
	{
	userId: “UserNotIn”,
	trophyScore: “Score”,
	}
		Nothing changes in userDB
		
		Throws Not In DB Error
			https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/e2d6cf472ad8af67ace43884b7efe8de090e51bf/backend/src/services/users/__tests__/useraccounts.test.js#L65

trophyScore is null	{
	userId: “User”,
	trophyScore: null,
	}
		Nothing changes in userDB
		
		Throws Input Error
			https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/e2d6cf472ad8af67ace43884b7efe8de090e51bf/backend/src/services/users/__tests__/useraccounts.test.js#L74

No userId field in	{	Nothing changes in	https://github.com/CP

input	trophyScore: “Score”, }	userDB Throws Input Error	EN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/e2d6cf472ad8af67ace43884b7efe8de090e51bf/backend/src/services/users/__tests__/useraccounts.test.js#L88

trophyScore is undefined or missing	{
	userId: “User”,
	}
		Nothing changes in userDB
		
		Throws Input Error
			https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/e2d6cf472ad8af67ace43884b7efe8de090e51bf/backend/src/services/users/__tests__/useraccounts.test.js#L94

Both userId and trophyScore fields are missing	{ }	Nothing changes in userDB Throws Input Error	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/e2d6cf472ad8af67ace43884b7efe8de090e51bf/backend/src/services/users/__tests__/useraccounts.test.js#L107

- **User getUserProfile(String userID)**

Scenario	Input	Expected Result	Location in Git
Successfully get userId’s profile	“User”	Return: The user stored in userDB associates to “User”; and a rank field is added with the rank of “User” among all users	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/e2d6cf472ad8af67ace43884b7efe8de090e51bf/backend/src/services/users/__tests__/useraccounts.test.js#L117
userId is not in userDB	[assume: UserNotInDB is not in userDB]	Throws Not In DB Error	https://github.com/CPEN-321-World-Exploration-Action/CPEN-3

	“UserNotInDB”		21-World_Exploration_Action/blob/e2d6cf472ad8af67ace43884b7efe8de090e51bf/backend/src/services/users/__tests__/useraccounts.test.js#L133
userId is invalid	null	Throws Input Error	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/e2d6cf472ad8af67ace43884b7efe8de090e51bf/backend/src/services/users/__tests__/useraccounts.test.js#L145

- void uploadFcmToken(String userId, String fcmToken)

Scenario	Input	Expected Result	Location in Git
Successfully updated	“User”, “Token”	“User”’s fcm_token field in userDB is updated to “Token”	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/e2d6cf472ad8af67ace43884b7efe8de090e51bf/backend/src/services/users/__tests__/useraccounts.test.js#L147
userId is not found in userDB	[assume: UserNotInDB is an ID that is not in userDB] “UserNotInDB”, “Token”	Nothing happens Throws Not In DB Error	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/e2d6cf472ad8af67ace43884b7efe8de090e51bf/backend/src/services/users/__tests__/useraccounts.test.js#L160
userId is invalid	null, “Token”	Nothing happens Throws Input Error	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/e2d6cf472ad8af67ace43884b

			7efe8de090e51bf/backend/src/services/users/__tests__/useraccounts.test.js#L167
fcmToken is invalid	“User”, null	Nothing happens Throws Input Error	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/e2d6cf472ad8af67ace43884b7efe8de090e51bf/backend/src/services/users/__tests__/useraccounts.test.js#L174

- User loginWithGoogle(String idToken)

Scenario	Input	Expected Result	Location in Git
Successfully login user with Google	[assume: idToken is a valid token helps user to login with Google] “idToken”	Return: The user stored in userDB associates to “idToken”; and a rank field is added with the rank of the user among all users	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/e2d6cf472ad8af67ace43884b7efe8de090e51bf/backend/src/services/users/__tests__/useraccounts.test.js#L185
idToken is invalid	[assume: idTokenInvalid is invalid and cannot be verified] “idTokenInvalid”	Throws a BadRequestError	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/e2d6cf472ad8af67ace43884b7efe8de090e51bf/backend/src/services/users/__tests__/useraccounts.test.js#L209
The user associates to the idToken is not found in the userDB	[assume: the userId associates to idTokenNotInDB cannot be found in userDB] “idTokenNotInDB”	Return: A new user created with the field user_id being specified to the userId associates to idTokenNotInDB	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/e2d6cf472ad8af67ace43884b7efe8de090e51bf/backend/src/services/users/__tests__/useracc

			ounts.test.js#L213

- List<User> searchUser(String query)

Scenario	Input	Expected Result	Location in Git
Successfully searched the user, query matches user_id field in userDB	“UserID”	Return: An array of matched User who has the userId “UserID”	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/e2d6cf472ad8af67ace43884b7efe8de090e51bf/backend/src/services/users/__tests__/useraccounts.test.js#L227
Successfully searched the user, query matches email field in userDB	“Email”	Return: An array of matched User who has the email of “Email”	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/e2d6cf472ad8af67ace43884b7efe8de090e51bf/backend/src/services/users/__tests__/useraccounts.test.js#L238
Successfully searched the user, query matches name field in userDB	“Name”	Return: An array of matched User who has the name “Name”	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/e2d6cf472ad8af67ace43884b7efe8de090e51bf/backend/src/services/users/__tests__/useraccounts.test.js#L258
No matched user found in userDB in any field	“UserNotFound”	Return: []	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/e2d6cf472ad8af67ace43884b7efe8de090e51bf/backend/src/services/users/__tests__/useraccounts.test.js#L279
The query input is	null	Throws Input Error	https://github.com/CP

invalid, either null or undefined			EN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/e2d6cf472ad8af67ace43884b7efe8de090e51bf/backend/src/services/users/__tests__/useraccounts.test.js#L295

- void signOut(String userId)

Scenario	Input	Expected Result	Location in Git
Successfully logged the user out	“User”	A signout message along with “User” userId is displayed in terminal	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/e2d6cf472ad8af67ace43884b7efe8de090e51bf/backend/src/services/users/__tests__/useraccounts.test.js#L302
userId input is invalid, either null or undefined	null	Throws Input Error	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/e2d6cf472ad8af67ace43884b7efe8de090e51bf/backend/src/services/users/__tests__/useraccounts.test.js#L306

- **Trophies**
  - Trophycollection
    - void collectTrophy(String userID, String trophyID)

Scenario	Input	Expected Result	Location in Git
UserId successfully collected the trophy	“User”, “Trophy”	“Trophy” is removed from “User”’s field of uncollectedTrophies, and added to collectedTrophies “User” is added to “Trophy”’s field of list_of_collectors, and	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/8ccc424514c0f8a6ade1a18e21e9a4757e7daac5/backend/src/services/trophies/__tests__/troph

		“Trophy”’s number_of_collectors is added by 1 userAccounts.onReceiveTrophyCollectedMessage is called with userId, trophyId, and trophyScore field being filled in	ycollection.test.js#L31
UserId collected the trophyId, however userId has collected the trophyId before	[assume: TrophyCollected is already collected by User] “User”, “TrophyCollected”	The database is not updated and the interface returns a duplication error BadRequestError	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/8ccc424514c0f8a6ade1a18e21e9a4757e7daac5/backend/src/services/trophies/__tests__/trophycollection.test.js#L65
Invalid userId field	null, “Trophy”	The database is not updated and the interface returns an Input Error	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/8ccc424514c0f8a6ade1a18e21e9a4757e7daac5/backend/src/services/trophies/__tests__/trophycollection.test.js#L97
Invalid trophyId field	“User”, null	The database is not updated and the interface returns an Input Error	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/8ccc424514c0f8a6ade1a18e21e9a4757e7daac5/backend/src/services/trophies/__tests__/trophycollection.test.js#L121
Both userId and trophyId are invalid	null, null	Databases are not updated and throws an Input Error	https://github.com/CPEN-321-World-Exploration-Action/CPEN-3

			21-World_Exploration_Action/blob/8ccc424514c0f8a6ade1a18e21e9a4757e7daac5/backend/src/services/trophies/__tests__/trophycollection.test.js#L136

- Trophydetails
  - Trophy[] getTrophiesUser(String userId, double lat, double lon)

Scenario	Input	Expected Result	Location in Git
There are trophies in userId's MIN_DISTANCE_METERS, the number of userId's uncollected trophies is less than MAX_TROPHIES, and there are trophies around the userId's location of lat and lon. The User has no collected trophy	"User", 123, 123	Return: the updated list of uncollected trophy based on the lat and lon (said to be userId's location) New trophies are generated for "User" to collect so that the number of original uncollected trophies reach MAX_TROPHIES. Then, the "User"'s uncollectedTrophyIDs field is updated.	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/e2d6cf472ad8af67ace43884b7efe8de090e51bf/backend/src/services/trophies/__tests__/trophydetails.test.js#L367
There are trophies in userId's MIN_DISTANCE_METERS, the number of userId's uncollected trophies is less than MAX_TROPHIES, and there are trophies around the userId's location of lat and lon. The User has collected trophies	[assume: UserTrophy has collected trophies] "UserTrophy", 123, 123	Return: the updated list of uncollected trophy based on the lat and lon (said to be userId's location). This list is also combined with trophies included in collectedTrophies New trophies are generated for "User" to collect so that the number of original uncollected trophies reach MAX_TROPHIES.	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/e2d6cf472ad8af67ace43884b7efe8de090e51bf/backend/src/services/trophies/__tests__/trophydetails.test.js#L425

		Then, the “User”’s uncollectedTrophyIDs field is updated.	
There is no trophy in userId’s MIN_DISTANCE_METERS, the number of userId’s uncollected trophies is less than MAX_TROPHIES, and there are trophies around the userId’s location of lat and lon. The User has collected trophies	[assume: UserTooFar has no trophy in MIN_DISTANCE_METERS of lat and lon] “UserTooFar”, 123, 123	Return: the new list of uncollected trophy is created based on the lat and lon (said to be userId’s location). This list is also combined with trophies included in collectedTrophies New MAX_TROPHIES numbers of trophies are generated for “User” to collect. Then, the “User”’s uncollectedTrophyIDs field is updated.	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/e2d6cf472ad8af67ace43884b7efe8de090e51bf/backend/src/services/trophies/__tests__/trophydetails.test.js#L469
There are trophies in userId’s MIN_DISTANCE_METERS, the number of userId’s, the number of userId’s uncollected trophies is more than MAX_TROPHIES. The User has collected trophies	[assume: “UserMany” has uncollected trophies in MIN_DISTANCE_METERS, and number of uncollected trophies is more than MAX_TROPHIES] “UserTooMany”, 123, 123	Return: the combination of UserMany’s collected and uncollected trophies Nothing is actually updated	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/e2d6cf472ad8af67ace43884b7efe8de090e51bf/backend/src/services/trophies/__tests__/trophydetails.test.js#L499
The userId has no uncollected trophies, and there is no famous locations around the user’s location	[assume: “5213.1” and “5213.2” represents the location that no famous location around, and “UserNoUncollect” has no uncollected trophies] “UserNoUncollect”, “5213.1”,“5213.2”	Return: []	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/e2d6cf472ad8af67ace43884b7efe8de090e51bf/backend/src/services/trophies/__tests__/trophydetails.test.js#L533
Invalid lat or/and lon	“User”, null, null	Throws Input Error	https://github.com/CPEN-321-World-Exploration-Action/CPEN-3

			21-World_Exploration_Action/blob/e2d6cf472ad8af67ace43884b7efe8de090e51bf/backend/src/services/trophies/__tests__/trophydetails.test.js#L540
Invalide userId	null, 123, 123	Throws Input Error	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/e2d6cf472ad8af67ace43884b7efe8de090e51bf/backend/src/services/trophies/__tests__/trophydetails.test.js#L549
userId is not found in TrophyUser database	[assume: “UserNotInDB” cannot be found in TrophyUserDB] “UserNotInDB”, 49.264320, -123.251574	Throws Not In DB Error	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/e2d6cf472ad8af67ace43884b7efe8de090e51bf/backend/src/services/trophies/__tests__/trophydetails.test.js#L556
userId has no uncollected trophy	[assume: “UserGood” has no uncollected trophy] “UserGood”, 49.264320, -123.251574		https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/e2d6cf472ad8af67ace43884b7efe8de090e51bf/backend/src/services/trophies/__tests__/trophydetails.test.js#L563

- TrophyTrophy getTrophyDetails(String TrophyID)

Scenario	Input	Expected Result	Location in Git
Successfully retrieve the TrophyTrophy associates to the TrophyID	“Trophy”	Return: the TrophyTrophy Associates to the “Trophy”	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/e2d6cf472ad8af67ace43884b

			7efe8de090e51bf/backend/src/services/trophies/__tests__/trophydetails.test.js#L585
Successfully retrieve the TrophyTrophy associates to the TrophyID (when there are multiple IDs)	“Trophy”, “Troph2”	Return: the TrophyTrophy Associated	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/e2d6cf472ad8af67ace43884b7efe8de090e51bf/backend/src/services/trophies/__tests__/trophydetails.test.js#L610
Invalid TrophyID	[assume: no entry indexed by null exists in TrophyTrophy] null	Throws Input Error Nothing happens to TrophyTrophyDB	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/e2d6cf472ad8af67ace43884b7efe8de090e51bf/backend/src/services/trophies/__tests__/trophydetails.test.js#L651
TrophyID is not found in TrophyTrophyDB	[assume: “TrophyNotInDB” cannot be found in TrophyTrophyDB] “TrophyNotInDB”	Return: Nothing Nothing happens to TrophyTrophyDB	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/e2d6cf472ad8af67ace43884b7efe8de090e51bf/backend/src/services/trophies/__tests__/trophydetails.test.js#L657

1.15.2. Interface Tests Design and Locations

- Login Use Case:

Scenario	Action	Expected Behavior	Location in GitHub
	Login for a non-existing User with a valid tokenID	Return newly created user's userID Status Code: 201	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Ex

Login			ploration_Action/blob/8727aaa2987530f71d282f47245156acaeffffd4/backend/test/interfaces/login.test.js#L48
	Login for an existing User with a valid tokenID	Return existing user's userID Status Code: 201	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/8727aaa2987530f71d282f47245156acaeffffd4/backend/test/interfaces/login.test.js#L57
	Login for User (existing or non-existing) without a valid tokenID	Invalid tokenID Error Status Code: 400	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/8727aaa2987530f71d282f47245156acaeffffd4/backend/test/interfaces/login.test.js#L68
	Login for User without a tokenID	Bad Request Error Status Code: 400	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/10b7eb56580bed47f369534619b4df6385e7e0ef/backend/test/interfaces/login.test.js#L76
Upload FCM Token	Upload the FCM token of the user’s device	The token is successfully uploaded and stored in the database Status Code: 200	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/10b7eb56580bed47f369534619b4df6385e7e0ef/backend/test/interfaces/login.test.js#L93
	The FCM token is not provided	Not Found Error (the token is provided as a path parameter) Status Code: 404	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/10b7eb56580bed47f369534619b4df6385e7e0ef/backend/test/interfaces/login.test.js#L101

	Unauthenticated user uploads a token	Authorization Error Status Code: 401	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/10b7eb56580bed47f369534619b4df6385e7e0ef/backend/test/interfaces/login.test.js#L108

- Browse Trophies on Map Use Case:

Scenario	**Action**	**Expected Behavior**	**Location in GitHub**
Get User Trophies	Retrieve List of Trophies for authenticated User	Return list of collected and uncollected trophies Status Code: 200	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/8727aaa2987530f71d282f47245156acaeffffd4/backend/test/interfaces/browse-trophies.test.js#L35
	Retrieve List of Trophies for authenticated User, who moved far away from the last location	Return list of collected and uncollected trophies Status Code: 200	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/1bb3b015aec0d89008473a4a6fcc4be8e4d907f3/backend/test/interfaces/browse-trophies.test.js#L43
	Retrieve List of Trophies for authenticated User without any trophies	Return list of newly created uncollected trophies Status Code: 200	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/8727aaa2987530f71d282f47245156acaeffffd4/backend/test/interfaces/browse-trophies.test.js#L46
	Retrieve List of Trophies for non-authenticated User	Authentication Error Status Code: 401	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/8727aaa2987530f71d282f47245156acaeffffd4/backend/test/interfaces/browse-trophies.test.js#L63

	Retrieve List of Trophies for User in location without any nearby points of interest	Return an list with original uncollected trophies if there are uncollected trophies are within MIN_DISTANCE_METERS Status Code: 200	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/8727aaa2987530f71d282f47245156acaeffffd4/backend/test/interfaces/browse-trophies.test.js#L69
	Retrieve List of Trophies for User without specifying user location	User Location Error Status Code: 400	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/8727aaa2987530f71d282f47245156acaeffffd4/backend/test/interfaces/browse-trophies.test.js#L85
	Retrieve List of Trophies for User with incorrect location format	User Location Error Status Code: 400	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/8727aaa2987530f71d282f47245156acaeffffd4/backend/test/interfaces/browse-trophies.test.js#L90

- View Profile Use Case:

Scenario	**Action**	**Expected Behavior**	**Location in GitHub**
Get User Profile	Retrieve Profile for existing authenticated user	Return User object from Database. Contains name, email, profile picture URI, list of friends, score, and location. Status Code: 200	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/1f09ae40c8241f231035754a7340301e71278677/backend/test/interfaces/view-profile.test.js#L34
	Retrieve Profile for existing but unauthenticated user	Authentication Error Status Code: 401	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/1f09ae40c8241f231035754a7340301e71278677/backend/test/interfaces/view-profile.test.js#L50
	Retrieve Profile for non-existing user	User not found Error Status Code: 404	https://github.com/CPEN-321-World-Exploration-Action/

			CPEN-321-World_Exploration_Action/blob/1f09ae40c8241f231035754a7340301e71278677/backend/test/interfaces/view-profile.test.js#L57
Get User Photos	Retrieve Photos for Existing and Authenticated User who has taken photos	Return list of photo IDs Status Code: 200	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/1f09ae40c8241f231035754a7340301e71278677/backend/test/interfaces/view-profile.test.js#L64
	Retrieve Photos for Non-existing User	User not Found Error Status Code: 404	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/1f09ae40c8241f231035754a7340301e71278677/backend/test/interfaces/view-profile.test.js#L78
	Retrieve Photos for User who has never taken photos	Return empty list Status Code: 200	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/1f09ae40c8241f231035754a7340301e71278677/backend/test/interfaces/view-profile.test.js#L92
Logout	Logout Valid User with authenticated session	Delete User Session in the backend. Status Code: 201	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/1f09ae40c8241f231035754a7340301e71278677/backend/test/interfaces/view-profile.test.js#L101
	Logout Valid User with expired Sessions	Do Nothing Status Code: 401	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/1f09ae40c8241f231035754a7340301e71278677/backend/test/interfaces/view-profile.test.js#L111

- Watch Leaderboard Use Case:

Scenario	Action	Expected Behavior	Location in GitHub
See Global Leaderboard	Valid, Authenticated User requests global leaderboard.	Return list of User Objects ordered by Trophy Score Status Code: 200	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/1f09ae40c8241f231035754a7340301e71278677/backend/test/interfaces/watch-leaderboard.test.js#L139
	Valid, non-authenticated User requests Global Leaderboard	Authentication Error Status Code: 401	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/1f09ae40c8241f231035754a7340301e71278677/backend/test/interfaces/watch-leaderboard.test.js#L156
	Valid, authenticated User requests empty global leaderboard	Return empty list Status Code: 200	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/1f09ae40c8241f231035754a7340301e71278677/backend/test/interfaces/watch-leaderboard.test.js#L161
See Friends Leaderboard	Valid, Authenticated User requests friends leaderboard.	Return list of User Objects corresponding to Friends ordered by Trophy Score Status Code: 200	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/1f09ae40c8241f231035754a7340301e71278677/backend/test/interfaces/watch-leaderboard.test.js#L170
	Valid, non-authenticated User requests Friends Leaderboard	Authentication Error Status Code: 401	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/1f09ae40c8241f231035754a7340301e71278677/backend/test/interfaces/watch-leaderboard.test.js#L181
	Valid, authenticated User with no Friends requests Friends leaderboard	Return a list only contain himself Status Code: 200	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/1f09ae40c8241f231035754a7340301e71278677/backend/test/interfaces/watch-leaderboard.test.js#L186

Subscribe to Leaderboard	Valid, Authenticated User with valid FCM token subscribes to leaderboard updates	Add User to list of subscribers and return the expiry time Status Code: 200	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/1f09ae40c8241f231035754a7340301e71278677/backend/test/interfaces/watch-leaderboard.test.js#L206
	Valid but unauthenticated User subscribes to leaderboard updates	Authentication Error Status Code: 401	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/1f09ae40c8241f231035754a7340301e71278677/backend/test/interfaces/watch-leaderboard.test.js#L213
	Valid, Authenticated User subscribes to leaderboard updates without providing a FCM token	Missing FCM Token Status Code: 400	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/1f09ae40c8241f231035754a7340301e71278677/backend/test/interfaces/watch-leaderboard.test.js#L220

- Review Trophy Details and Photos Use Case:

Scenario	**Action**	**Expected Behavior**	**Location in GitHub**
View Trophy Details	Get Details of an existing Trophy with a TrophyID	Returns details of Trophy such as name, location, associated tags, quality, list of photoIds associated. Status Code: 200	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/8727aaa2987530f71d282f47245156acaeffffd4/backend/test/interfaces/review-trophy-details.test.js#L280
	Get Details of a non-existing Trophy	Trophy Not Found Error Status Code: 404	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/8727aaa2987530f71d282f47245156acaeffffd4/backend/test/interfaces/review-trophy-details.test.js#L287
	Get Photos Associated with a valid Trophy	Return list of photo Ids for the given Trophy sorted randomly Status Code: 200	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/8727aaa2987530f71d282f47245156acaeffffd4/backend/test/interfaces/review-trophy-details.test.js#L295
	Get Photos	Error Trophy not	https://github.com/CPEN-321-World-Expl

Get Trophy Photos	Associated with an invalid Trophy	found Status Code: 404	oration-Action/CPEN-321-World_Exploration_Action/blob/8727aaa2987530f71d282f47245156acaeffffd4/backend/test/interfaces/review-trophy-details.test.js#L305
	Get Photos for a valid Trophy without any photos	Return empty list Status Code: 200	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/8727aaa2987530f71d282f47245156acaeffffd4/backend/test/interfaces/review-trophy-details.test.js#L313
	Get Photos for a valid Trophy ordered by time	Return list of photo Ids for the given Trophy sorted in descending order by time they were taken. Status Code: 200	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/8727aaa2987530f71d282f47245156acaeffffd4/backend/test/interfaces/review-trophy-details.test.js#L321
	Get Photos for valid Trophy ordered by likes	Return list of photo Ids for the given Trophy sorted by the number of likes they have. Status Code: 200	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/8727aaa2987530f71d282f47245156acaeffffd4/backend/test/interfaces/review-trophy-details.test.js#L332
	Get Photos for valid Trophy, explicitly ordered randomly	Return list of photo Ids for the given Trophy ordered randomly. Status Code: 200	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/8727aaa2987530f71d282f47245156acaeffffd4/backend/test/interfaces/review-trophy-details.test.js#L343
	Get Photos for valid Trophy with an invalid ordering (not time, likes, or random)	Invalid Order Error Status Code: 400	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/8727aaa2987530f71d282f47245156acaeffffd4/backend/test/interfaces/review-trophy-details.test.js#L353

- Evaluate Photos Use Case:

Scenario	**Action**	**Expected Behavior**	**Location in GitHub**
Open Photo	Download existing image (valid PhotoID)	Return Image Status Code: 200	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/8727aaa2987530f71d282f4724515

			6acaeffffd4/backend/test/interfaces/evaluate-photos.test.js#L36
	Download non-existing image (invalid PhotoID)	Photo not Found Status Code: 404	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/8727aaa2987530f71d282f47245156acaeffffd4/backend/test/interfaces/evaluate-photos.test.js#L41
Press Like Photo Button	Valid, authenticated User likes a valid photo that hasn't been liked by the user.	Add User ID to Photo's list of likedUsers Status Code: 200	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/8727aaa2987530f71d282f47245156acaeffffd4/backend/test/interfaces/evaluate-photos.test.js#L48
	Valid, authenticated User likes a valid photo that has already been liked by the user.	Remove User ID from Photo's List of likedUsers Status Code: 200	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/8727aaa2987530f71d282f47245156acaeffffd4/backend/test/interfaces/evaluate-photos.test.js#L55
	Non-authenticated User likes a valid photo	Authentication Error Status Code: 401	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/8727aaa2987530f71d282f47245156acaeffffd4/backend/test/interfaces/evaluate-photos.test.js#L63
	Valid, authenticated User likes a non-existing photo	Photo Not found Error Status Code: 404	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/8727aaa2987530f71d282f47245156acaeffffd4/backend/test/interfaces/evaluate-photos.test.js#L68
	No picID is provided	Bad Request Error	https://github.com/CPEN-3

		Status Code: 400	21-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/8727aaa2987530f71d282f47245156acaeffffd4/backend/test/interfaces/evaluate-photos.test.js#L73

- Collect Trophy Use Case:

Scenario	**Action**	**Expected Behavior**	**Location in GitHub**
Collect Trophy	Collect existing Trophy for valid userID	User Collects Trophy - User List of collected and uncollected trophies are updated along with the user score. Status Code: 201	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/8727aaa2987530f71d282f47245156acaeffffd4/backend/test/interfaces/collect-trophy.test.js#L47
	Collect non-existing Trophy for valid userID	Trophy not found Error Status Code: 404	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/8727aaa2987530f71d282f47245156acaeffffd4/backend/test/interfaces/collect-trophy.test.js#L61
	Collect existing Trophy for invalid userID	User not found Error Status Code: 404	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/8727aaa2987530f71d282f47245156acaeffffd4/backend/test/interfaces/collect-trophy.test.js#L75
	Collect non-existing Trophy for invalid userID	User not found Error Status Code: 404	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/8727aaa2987530f71d282f47245156acaeffffd4/backend/test/interfaces/collect-trophy.test.js#L80
	Collect existing Trophy for Non-Authenticated User	Authentication Error Status Code: 401	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/8727aaa298

			7530f71d282f47245156acaeffffd4/backend/test/interfaces/collect-trophy.test.js#L85
Upload Photo to Trophy	Existing User uploads photo to existing Trophy	New entry in Photo database created linked to user and relevant Trophy Status Code: 201	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/8727aaa2987530f71d282f47245156acaeffffd4/backend/test/interfaces/collect-trophy.test.js#L92
	Non-Authenticated User Uploads photo	Authentication Error Status Code: 401	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/8727aaa2987530f71d282f47245156acaeffffd4/backend/test/interfaces/collect-trophy.test.js#L101
	Existing User Uploads empty file to existing Trophy	No Photo Error Status Code: 400	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/8727aaa2987530f71d282f47245156acaeffffd4/backend/test/interfaces/collect-trophy.test.js#L107
	Existing User Uploads txt file to existing Trophy	Bad Photo Status Code: 400	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/8727aaa2987530f71d282f47245156acaeffffd4/backend/test/interfaces/collect-trophy.test.js#L112

- Manage Friends Use Case:

Note: Friends' profiles (name, email & picture URL) are also returned by Get User's Friends so viewing a friends's profile does not have a separate test case.

Scenario	**Action**	**Expected Behavior**	**Location in GitHub**
	Retrieve list of Friends for Valid	Return List of Friends' UserIds	https://github.com/CPEN-321-World-Exploration-Action/CPEN

Get User's Friends	User with Friends	Status Code: 200	-321-World_Exploration_Action/blob/8727aaa2987530f71d282f47245156acaeffffd4/backend/test/interfaces/manage-friends.test.js#L30
	Retrieve list of Friends for Valid User with no Friends	Return empty list Status Code: 200	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/8727aaa2987530f71d282f47245156acaeffffd4/backend/test/interfaces/manage-friends.test.js#L37
	Retrieve list of Friends for a user who did not logged in	Unauthorized Error Status Code: 401	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/8727aaa2987530f71d282f47245156acaeffffd4/backend/test/interfaces/manage-friends.test.js#L47
Search for Friends	Retrieve list of Users that match (partial or completely) the search query	Return list of UserIds for Users matching the query Status Code: 200	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/8727aaa2987530f71d282f47245156acaeffffd4/backend/test/interfaces/manage-friends.test.js#L55
	Retrieve list of Users where the query doesn't match any User	Return empty list Status Code: 200	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/8727aaa2987530f71d282f47245156acaeffffd4/backend/test/interfaces/manage-friends.test.js#L62
	Retrieve list of Users without providing a search query	Missing Query Error Status Code: 400	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/8727aaa2987530f71d282f47245156acaeffffd4/backend/test/interfaces/manage-friends.test.js#L69
	Retrieve non-empty list of Users who made requests to Valid, Authenticated	Return list of User Objects corresponding to Users that sent	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/8727aaa2987530f71d282f

See Friend Requests	User	requests Status Code: 200	47245156acaeffffd4/backend/test/interfaces/manage-friends.test.js#L77
	Retrieve Friend Requests when no Users made requests	Return empty list Status Code: 200	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/8727aaa2987530f71d282f47245156acaeffffd4/backend/test/interfaces/manage-friends.test.js#L84
	Retrieve Friend Requests for Non-authenticated User	Authentication Error Status Code: 401	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/8727aaa2987530f71d282f47245156acaeffffd4/backend/test/interfaces/manage-friends.test.js#L93
Send Friend Request	Valid Authenticated User sends Friend Request to existing Non-Friended User	The friend request is successfully sent and stored in the friend's module Status Code: 201	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/8727aaa2987530f71d282f47245156acaeffffd4/backend/test/interfaces/manage-friends.test.js#L101
	Unauthenticated, User sends Friend Request	Authentication Error Status Code: 401	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/8727aaa2987530f71d282f47245156acaeffffd4/backend/test/interfaces/manage-friends.test.js#L109
	Valid and Authenticated User sends Friend Request to non-existing User	Not Found Error The request is not stored Status Code: 404	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/8727aaa2987530f71d282f47245156acaeffffd4/backend/test/interfaces/manage-friends.test.js#L115
	Valid and Authenticated User sends Friend Request to Friend	Bad Request Error The request is not stored Status Code: 400	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/8727aaa2987530f71d282f47245156acaeffffd4/backend/test/interfaces/manage-friends.te

			st.js#L123
	The user sends a request to themselves	Bad Request Error The request is not stored Status Code: 400	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/8727aaa2987530f71d282f47245156acaeffffd4/backend/test/interfaces/manage-friends.test.js#L131
	Valid and Authenticated User sends Friend Request to Non-Friended User after already sending a request	Request already Sent Error Status Code: 400	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/8727aaa2987530f71d282f47245156acaeffffd4/backend/test/interfaces/manage-friends.test.js#L140
Accept Friend Request	Valid, Authenticated User accepts request of from valid Sender	Remove Request, and set User and Sender as friends, mutually. Status Code: 201	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/8727aaa2987530f71d282f47245156acaeffffd4/backend/test/interfaces/manage-friends.test.js#L157
	Non-authenticated User accepts request from any sender	Authentication Error Status Code: 401	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/8727aaa2987530f71d282f47245156acaeffffd4/backend/test/interfaces/manage-friends.test.js#L166
	Valid and authenticated user accepts request, but Sender ID was not provided	Bad Request Error Status Code: 400	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/8727aaa2987530f71d282f47245156acaeffffd4/backend/test/interfaces/manage-friends.test.js#L172
	Valid and authenticated User accepts a request that does not exists	Bad Request Error Status Code: 400	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/8727aaa2987530f71d282f47245156acaeffffd4/backend/test/interfaces/manage-friends.test.js#L179
	Valid, Authenticated	Remove Request,	https://github.com/CPEN-321-

Deny Friend Request	User declines request of from valid Sender	and notify Sender of Decline. Status Code: 201	World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/8727aaa2987530f71d282f47245156acaeffffd4/backend/test/interfaces/manage-friends.test.js#L188
	Non-authenticated User declines request from any sender	Authentication Error Status Code: 401	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/8727aaa2987530f71d282f47245156acaeffffd4/backend/test/interfaces/manage-friends.test.js#L197
	Valid and authenticated user declines request, but Sender ID was not provided	Missing Sender Error Status Code: 400	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/8727aaa2987530f71d282f47245156acaeffffd4/backend/test/interfaces/manage-friends.test.js#L203
	Valid and authenticated User declinesa request that does not exists	Not Found Error Status Code: 400	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/8727aaa2987530f71d282f47245156acaeffffd4/backend/test/interfaces/manage-friends.test.js#L210
Delete Friend	Non-authenticated user attemps to delete a friend	Authentication Error Status Code: 401	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/8727aaa2987530f71d282f47245156acaeffffd4/backend/test/interfaces/manage-friends.test.js#L219
	The user does not provide the friend's user ID	Bad Request Error Status Code: 400	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/8727aaa2987530f71d282f47245156acaeffffd4/backend/test/interfaces/manage-friends.test.js#L225
	The provided user ID does not match any user in the	Not Found Error Status Code: 400	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/

	friend list		blob/8727aaa2987530f71d282f47245156acaeffffd4/backend/test/interfaces/manage-friends.test.js#L232
	The provided friend ID is valid and matches on of the user's friend	The friend is successfully deleted Set User and Sender as non-friends, mutually Status Code: 200	https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/8727aaa2987530f71d282f47245156acaeffffd4/backend/test/interfaces/manage-friends.test.js#L239

1.15.3. Front-end Tests Design and Locations

- Review Trophy Details and Photos Use case:

Action	**Expected Behavior**	**Location in Git**	**Success/Failure criteria**
1. User opens "Review Trophy details" screen	The screen shows the trophy's name, 9 or less pictures added by other users for this trophy if any, a sorting button, a map button, a "collect trophy" button, and a text of "Number of users who collected this trophy:" followed by the number	144 - 154 in TrophyDetailsAndPhotosTests.java	We encoded the success/failure criteria by checking if the trophy's name, the images grid, the textView of "Number of users who collected this trophy:", the number of collectors textView, and the "collect trophy" button are displayed on the screen using their id. The test is succeed only if all the checks return true and failed otherwise.
2. User opens "Review Trophy details" screen of a far trophy	The "collect trophy" button is inactive (grey)	239- 245 in TrophyDetailsAndPhotosTests.java	We encoded the success/failure criteria by adding a new ToastMatcher and checking if the toast message is displayed on the screen after performing a click on

			the inactive “collect trophy” button. The test is succeeded only if the check of the toast message returns true.
3. User tries to press the “collect trophy button” for a far trophy	A toast message of “You are too far away” is displayed and the trophy isn’t collected		
4. User opens “Review Trophy details” screen of a trophy that no other users added pictures and user not at the location of the trophy	A text of “Seems nobody has added a photo here yet” is displayed instead of the pictures	181- 185 in TrophyDetailsAndPhotosTests.java	We encoded the success/failure criteria by checking if the expected text is displayed. The check is done using the string of the TextView. The test passed if and only this check returns true.
5. User opens “Review Trophy details” screen of a trophy that is close	The “collect trophy” button is active (purble)	188- 205 in TrophyDetailsAndPhotosTests.java	We encoded the success/failure criteria by adding a new ToastMatcher and checking if the toast message is displayed on the screen. We also check if the text of the “collect trophy” button is changed to “Take a photo” using the id of the button. Additionally, we check if the images grid is displayed on the screen using its id. The test passed only if all the checks return true.
6. User presses “collect trophy button” for a	A toast message of “You have collected this trophy” is		

trophy that has pictures	displayed, the trophy is successfully collected and the number of users collected the trophy is incremented, the “collect trophy” button is replaced with “Take a photo” button, and the pictures uploaded by users to this trophy are displayed under “Pictures by others” textView		
7. User presses “collect trophy button” for a trophy that doesn’t have pictures	A toast message of “You have collected this trophy” is displayed, the trophy is successfully collected and the number of users collected the trophy is incremented, the “collect trophy” button is replaced with “Take a photo” button, and a text of “Take a picture now to be the first one” is shown under “Pictures by others” textView	163- 179 in TrophyDetailsAndPhotosTests.java	We encoded the success/failure criteria by adding a new ToastMatcher and checking if the toast message is displayed on the screen. We also check if the text of the “collect trophy” button is changed to “Take a photo” using the id of the button. Additionally, we check if the textView of “Take a picture now to be the first one” is displayed using the string value. The test passed only if all the checks return true.
8. User presses “Take a photo” button again	There is a dialog “You have already taken a photo for this trophy. Do you want to replace the old one?” with 2 buttons: Cancel - Replace	208- 237 in TrophyDetailsAndPhotosTests.java	We encoded the success/failure criteria for this test by comparing the the id of the image before and after performing the replacing and using Assert.assertNotEquals
9. User presses	The camera is		

“Replace” button in the alert dialog	opened and the taken photo replaces the old photo taken by this user		
10. User presses “Cancel” button in the alert dialog	The camera is not opened and the alert dialog is disappeared	256- 282in TrophyDetailsAndPhotosTests.java	We encoded the success/failure criteria for this test by not adding stubImageCapture() and so confirming that the camera intent is not opened. Also, we check that the dialog is disappeared using its id and doesNotExist(). The test is passed only if doesNotExist() returns true.
11. User presses the map button	The trophy’s location is displayed on a Google map window	352- 376 in TrophyDetailsAndPhotosTests.java	We encoded the success/failure criteria by making a stub for google maps intent and returns true if it successfully opens to use Assert.assertTrue.
12. User presses the “Sort By” button	There is a dialog “Sort By Time Like Number”	284- 294 in TrophyDetailsAndPhotosTests.java	We encoded the success/failure criteria by checking if the dialog is displayed using text inside the dialog (Time, Like number). The test passes only if the check of displaying the dialog returns true and fails otherwise.
13. User presses on “Time” option on the dialog	The photos are sorted by the time in which the photo is posted in a descending order	296- 319in TrophyDetailsAndPhotosTests.java	We encoded the success/failure criteria by checking if the two images were not switched (as they are already in the correct order) their id

			and Assert.assertEquals.
14. User presses on “Like Number” option on the dialog	The photos are sorted by the number of like in a descending order	321- 350 in TrophyDetailsAndPhotosTests.java	We encoded the success/failure criteria by checking if the two images were switched using their id and Assert.assertEquals.

- Evaluate Photos Usecase:

Action	**Expected Behavior**	**Location in Git**	**Success/Failure criteria**
1. User opens “Evaluate Photos” screen	The screen shows the trophy’s name, the photo that the user clicked on of the “Review trophy details and photos” screen, a like button, and a TextView displays the number of likes of this photo	104-121 in EvaluatePhotosTests.java	We encoded the success/failure criteria by checking if the trophy’s name, the photo, the number of likes textView, and the like button are displayed on the screen using their id. The test is successful if all the checks return true and failed otherwise.
2. The user presses the like button	The number of likes is incremented	123-152 in EvaluatePhotosTests.java	This test passes only if the number of likes of textView is changed from “0” to “1” when pressing the like button and from “1” to “0” when pressing it again (unlike). The test failed with any other behavior not like that.
3. The user presses the like button again	The number of likes is decremented		

- Manage Friends Usecase:

Action	Expected Behavior	Location in Git	Success/Failure criteria
1. User opens “Manage Friends” screen	The screen shows search bar, and a text of “Seems you have no friends”	119-129 in ManageFriendsTests.java	We encoded the success/failure criteria by checking if editText and the search bar are displayed on the screen using their ids. The test succeeds only if the checks return true and fail otherwise.
2. User receives a friend request from Test User 4	The friend request is appeared on the screen with “Review Request” button	160-187 in ManageFriendsTests.java	We encoded the success/failure criteria by adding a new ToastMatcher and checking if the toast message of accepting a friend is displayed on the screen and if the friend is displayed on the friends list.
3. User presses “Review Request” button	There’s a dialog “Accept Test User 4’s friend request?” with 2 buttons” ACCEPT - DECLINE		
4. User presses “ACCEPT” button	The added friend’s name, image, and score shown on the screen. A toast message “Succesfully accepted Test User 4’s friend request” is shown		
5. User presses “DECLINE” button	The friend request is disappeared, the person isn’t added to the screen, and a toast message “Succesfully declined Test User 4’s friend	189-208 in ManageFriendsTests.java	We encoded the success/failure criteria by adding a new ToastMatcher and checking if the toast message of declining a friend

	request” is shown		request is displayed on the screen.
6. User presses on the friend’s icon (John)	There is a dialog with the name of the added person and 2 choices of: View Profile, Delete	210- 234 in ManageFriendsTests.java	We encoded the success/failure criteria by checking if the id of the profile layout matches the id of the displayed layout. The test passes if and only if the ids are similar.
7. User presses on “View Profile”	The app navigates to the profile screen of John		
8. User presses on “Delete” for	The friend is deleted and disappeared from “Manage Friends” screen, and a toast message “succesfully deleted friend John” is shown	237- 261 in ManageFriendsTests.java	We encoded the success/failure criteria by adding a new ToastMatcher and checking if the toast message of deleting a friend is displayed on the screen.
9. User types J/Jo/Joh/John in the search bar and no user with this name in the app	A text of “No user found” is shown in the screen	302- 309 in ManageFriendsTests.java	We encoded the success/failure criteria by checking if the id of the expected text matches the id of what is displayed on the screen. The test is successful only if the ids are the same.
10. User searches for a user already in the friends list	A text of “No user found” is shown in the screen	311- 337 in ManageFriendsTests.java	We encoded the success/failure criteria by checking if the id of the expected text matches the id of what is displayed on the screen. The test is successful only if the ids are the same.
11. User types Test User 3 and there is at	The users with these names, which are not in the user’s friends	339- 345 in ManageFriendsTests.java	We encoded the success/failure criteria by checking if

least a user with this name	list, are shown in the screen with “Send Request” button		there is a user row, which includes the “Send Request” button, displayed on the screen using the id after performing the search. The test is succeed only if there’s an element on the screen with this id.
12. User presses on “Send Request” button	There is a dialog “Send a friend request to John?” with 2 buttons: NO - YES	131- 157 in ManageFriendsTests. java	We encoded the success/failure criteria by adding a new ToastMatcher and checking if the toast message of sending a friend request is displayed on the screen.
13. User presses “YES” button	The friend request is send and there is a toast message “Succesfully sent a friend request to John”		
14. User presses “NO” button	The friend request isn’t sent and the dialog is disappeared		We encoded the success/failure criteria by checking if the dialog disappeared usingdoesNotExist() and the text inside the dialog instead of the id. The test fails if the dialog doesn’t disappear.

1.15.4. Non-functional Requirements Verification Plan

- Responsiveness

  We plan to verify this non-functional requirement by adding UI tests that measure the delay between each Espresso operation and compare the delay to a predefined threshold. Since Espresso automatically waits for resources to be idle, this will correctly measure the app’s response time.
- Usability

- We plan to verify this non-functional requirement by writing a UI test for each action. In each of these UI tests, we use assertions to ensure the number of clicks after logging in is less than 5.

1.16. Test Results

1.16.1. Backend Tests

1.16.1.1. Location and Description of Back-end Test Suites

- Path
  - Module Tests
    - Trophies
      - backend\src\services\trophies__tests__\trophycollection.test.js
      - backend\src\services\trophies__tests__\trophydetails.test.js
      - backend\src\services\trophies__tests__\trophyfilter.test.js (deleted, due to the missing complexity idea)
    - Users
      - backend\src\services\users__tests__\friends.test.js
      - backend\src\services\users__tests__\leaderboard.test.js
      - backend\src\services\users__tests__\useraccounts.test.js
    - Location of individual tests:
      - 1.15.1. Backend Module Tests Design and Locations
      - 1.15.2. Interface Tests Design and Locations
  - Per-Use Case Interface Tests
    - backend/test/interfaces/
- How to Run
  - All backend tests require a running MongoDB on port 27017
  - First “cd backend”, then do “npm install”
- Module Tests
  - Trophies
    - npm test src\services\trophies__tests__\trophycollection.test.js
    - npm test src\services\trophies__tests__\trophydetails.test.js
    - npm test src\services\trophies__tests__\trophyfilter.test.js (deleted, due to the missing complexity idea)
  - Users
    - npm test src\services\users__tests__\friends.test.js
    - npm test src\services\users__tests__\leaderboard.test.js
    - npm test src\services\users__tests__\useraccounts.test.js
- Per-Use Case Interface Tests
  - Put these two files in the backend folder https://drive.google.com/drive/folders/146s-5rEzo96geB-ks9RDlyl1y0A3DxrP?usp=sharing

- All interface tests
  - npm test .\test\interfaces*
- Login
  - npm test .\test\interfaces\login.test.js
- Browse Trophies on Map
  - npm test .\test\interfaces\browse-trophies.test.js
- View Profile
  - npm test .\test\interfaces\view-profile.test.js
- Watch Leaderboard
  - npm test .\test\interfaces\watch-leaderboard.test.js
- Review Trophy Details and Photos
  - npm test .\test\interfaces\review-trophy-details.test.js
- Evaluate Photos
  - npm test .\test\interfaces\evaluate-photos.test.js
- Collect Trophy
  - npm test .\test\interfaces\collect-trophy.test.js
- Manage Friends
  - npm test .\test\interfaces\manage-friends.test.js

1.16.1.2. Location of Github Action File

- .github/workflows/run-backend-tests.yml
- Note: our tests require a running MongoDB on port 27017

1.16.1.3. Jest Coverage Reports For Module Tests

- Note: The coverage is 0 for files not shown in the following coverage reports. They are hidden by Jest automatically.
- Note: Because we do not have many dependencies between main modules, we instead run tests for each sub-module and mock everything else.
- Trophies
  - Backend\test\services\trophies\trophycollection.test.js

```
----------------------|---------|----------|---------|---------|-------------------
File                  | % Stmts | % Branch | % Funcs | % Lines | Uncovered Line #s
----------------------|---------|----------|---------|---------|-------------------
All files             |   71.18 |    68.42 |   61.11 |   71.18 |
 data/db              |   61.11 |       60 |      50 |   61.11 |
  trophy.db.js        |   61.11 |       60 |      50 |   61.11 | 41,43,65-70,91-103
 services/trophies    |     100 |      100 |     100 |     100 |
  trophycollection.js |     100 |      100 |     100 |     100 |
 utils                |      70 |    33.33 |    62.5 |      70 |
  database.js         |     100 |      100 |     100 |     100 |
  errors.js           |   57.14 |    33.33 |      50 |   57.14 | 16-28
----------------------|---------|----------|---------|---------|-------------------
Test Suites: 1 passed, 1 total
Tests:       5 passed, 5 total
Snapshots:   0 total
Time:        0.849 s, estimated 1 s
```

- backend\test\services\trophies\trophydetails.test.js

File	% Stmts	% Branch	% Funcs	% Lines	Uncovered Line #s
All files	78.44	66.66	72	77.47	
data/db	44.44	10	50	44.44	
trophy.db.js	44.44	10	50	44.44	23-45,73-88,93
services/trophies	97.14	87.5	100	96.92	
trophydetails.js	97.14	87.5	100	96.92	96,166
utils	70	33.33	62.5	70	
database.js	100	100	100	100	
errors.js	57.14	33.33	50	57.14	16-28

```
Test Suites: 1 passed, 1 total
Tests:       12 passed, 12 total
Snapshots:   0 total
Time:        1.331 s, estimated 2 s
```

  - Line 96: this is the default break statement, which is not reachable if the program is running correctly
  - Line 166: this statement is for handling upsert errors; they are not reachable because of not mocking the databse
- backend\test\services\trophies\trophyfilter.test.js

The trophyfilter is considered to be a complexity idea, which is missing

- Users
  - backend\src\services\users__tests__\friends.test.js

File	% Stmts	% Branch	% Funcs	% Lines	Uncovered Line #s
All files	88.23	96.36	78.12	88.23	
data/db	64.28	83.33	54.54	64.28	
user.db.js	64.28	83.33	54.54	64.28	26,35-48,92
services/users	100	100	100	100	
friends.js	100	100	100	100	
utils	83.33	66.66	80	83.33	
database.js	100	100	100	100	
errors.js	71.42	66.66	66.66	71.42	22-28
utils.js	100	100	100	100	

```
Test Suites: 1 passed, 1 total
Tests:       29 passed, 29 total
Snapshots:   0 total
Time:        0.904 s, estimated 1 s
```

  - backend\src\services\users__tests__\leaderboard.test.js

File	% Stmts	% Branch	% Funcs	% Lines	Uncovered Line #s
All files	81.35	84.78	76.92	80.18	
data/db	25	0	27.27	25	
user.db.js	25	0	27.27	25	26,32,38-48,61-92
services/users	100	100	100	100	
leaderboard.js	100	100	100	100	
utils	90	66.66	87.5	90	
database.js	100	100	100	100	
errors.js	85.71	66.66	83.33	85.71	22

```
Test Suites: 1 passed, 1 total
Tests:       16 passed, 16 total
Snapshots:   0 total
Time:        2.88 s, estimated 3 s
```

- backend\src\services\users__tests__\useraccounts.test.js

File	% Stmts	% Branch	% Funcs	% Lines	Uncovered Line #s
All files	81.7	87.87	78.57	81.7	
data/db	50	50	54.54	50	
user.db.js	50	50	54.54	50	32-35,57-83,92
services/users	100	100	100	100	
useraccounts.js	100	100	100	100	
utils	90	66.66	87.5	90	
database.js	100	100	100	100	
errors.js	85.71	66.66	83.33	85.71	22

```
Test Suites: 1 passed, 1 total
Tests:       24 passed, 24 total
Snapshots:   0 total
Time:        0.864 s, estimated 1 s
```

1.16.1.4. Jest Coverage Reports For Use-case Tests

- Note: The coverage is 0 for files not shown in the following coverage reports. They are hidden by Jest automatically.

- Login

```
----------------------------|---------|----------|---------|---------|--------------------
All files                   |   26.02 |    11.81 |    20.3 |   26.73 |
 src                        |     100 |       50 |     100 |     100 |
  app.js                    |     100 |       50 |     100 |     100 | 40
 src/controllers            |   14.28 |     8.33 |    12.5 |   14.28 |
  photos.controllers.js     |       0 |        0 |       0 |       0 | 8-69
  trophies.controllers.js   |       0 |        0 |       0 |       0 | 6-46
  users.controllers.js      |   27.08 |       50 |      20 |   27.08 | 19-22,32-111
 src/data/db                |      24 |    18.75 |   14.81 |      24 |
  photo.db.js               |   27.27 |      100 |       0 |   27.27 | 17-64
  trophy.db.js              |   13.88 |        0 |       0 |   13.88 | 23-45,65-103
  user.db.js                |   35.71 |       50 |   36.36 |   35.71 | 32-35,43-83,92
 src/data/external          |   33.33 |       10 |   14.28 |   34.48 |
  fcm.external.js           |   16.66 |        0 |       0 |   18.18 | 11-55
  googleplaces.external.js  |   16.66 |     12.5 |       0 |   16.66 | 5-28
  googlesignin.external.js  |     100 |      100 |     100 |     100 |
 src/middleware             |   33.33 |       50 |   33.33 |   35.29 |
  auth.middleware.js        |     100 |      100 |     100 |     100 |
  upload.middleware.js      |    7.69 |        0 |       0 |    8.33 | 8-39
 src/routes                 |     100 |      100 |     100 |     100 |
  photos.routes.js          |     100 |      100 |     100 |     100 |
  trophies.routes.js        |     100 |      100 |     100 |     100 |
  users.routes.js           |     100 |      100 |     100 |     100 |
 src/services/photos        |       0 |        0 |       0 |       0 |
  photomanaging.js          |       0 |        0 |       0 |       0 | 7-47
 src/services/trophies      |    2.38 |        0 |       0 |    2.53 |
  trophycollection.js       |       0 |        0 |       0 |       0 | 9-40
  trophydetails.js          |    2.81 |        0 |       0 |    3.03 | 12-168
 src/services/users         |   26.34 |    13.08 |      25 |   27.37 |
  friends.js                |   20.96 |    13.04 |   27.27 |   20.96 | 9-94,116-130
  leaderboard.js            |    8.75 |      2.7 |       5 |    9.58 | 16,19-22,31-162
  useraccounts.js           |    65.9 |    29.16 |   66.66 |    65.9 | 7-16,24,38,42,69-81
 src/utils                  |   44.44 |    42.85 |   46.15 |   44.44 |
  database.js               |     100 |      100 |     100 |     100 |
  errors.js                 |   71.42 |      100 |   66.66 |   71.42 | 28-34
  message-manager.js        |       0 |        0 |       0 |       0 | 5-15
  utils.js                  |       0 |      100 |       0 |       0 | 2-3
----------------------------|---------|----------|---------|---------|--------------------
Test Suites: 1 passed, 1 total
Tests:       7 passed, 7 total
Snapshots:   0 total
Time:        1.382 s
```

- Browse Trophies on Map

```
----------------------------|---------|----------|---------|---------|--------------------------
All files                   |   35.29 |    24.09 |   26.31 |   35.34 |
 src                        |    92.3 |       50 |      50 |    92.3 |
  app.js                    |    92.3 |       50 |      50 |    92.3 | 32
 src/controllers            |   14.28 |       25 |    12.5 |   14.28 |
  photos.controllers.js     |       0 |        0 |       0 |       0 | 8-69
  trophies.controllers.js   |      50 |       60 |      50 |      50 | 6-9,33-40
  users.controllers.js      |    8.33 |        0 |    6.66 |    8.33 | 7-111
 src/data/db                |   33.33 |       25 |   22.22 |   33.33 |
  photo.db.js               |   27.27 |      100 |       0 |   27.27 | 17-64
  trophy.db.js              |   44.44 |       10 |      50 |   44.44 | 23-45,73-88,93
  user.db.js                |   21.42 |       50 |   18.18 |   21.42 | 26,32-83,92
 src/data/external          |   46.66 |       60 |   28.57 |   48.27 |
  fcm.external.js           |   16.66 |        0 |       0 |   18.18 | 11-55
  googleplaces.external.js  |   83.33 |       75 |     100 |   83.33 | 5,23
  googlesignin.external.js  |   33.33 |      100 |       0 |   33.33 | 7-13
 src/middleware             |   33.33 |       50 |   33.33 |   35.29 |
  auth.middleware.js        |     100 |      100 |     100 |     100 |
  upload.middleware.js      |    7.69 |        0 |       0 |    8.33 | 8-39
 src/routes                 |     100 |      100 |     100 |     100 |
  photos.routes.js          |     100 |      100 |     100 |     100 |
  trophies.routes.js        |     100 |      100 |     100 |     100 |
  users.routes.js           |     100 |      100 |     100 |     100 |
 src/services/photos        |       0 |        0 |       0 |       0 |
  photomanaging.js          |       0 |        0 |       0 |       0 | 7-47
 src/services/trophies      |    73.8 |       60 |   81.81 |   72.15 |
  trophycollection.js       |       0 |        0 |       0 |       0 | 9-40
  trophydetails.js          |   87.32 |    70.58 |     100 |   86.36 | 18,37,66,92-93,98-101,168
 src/services/users         |   16.12 |     6.54 |      15 |   16.75 |
  friends.js                |   20.96 |    13.04 |   27.27 |   20.96 | 9-94,116-130
  leaderboard.js            |    8.75 |      2.7 |       5 |    9.58 | 16,19-22,31-162
  useraccounts.js           |   22.72 |        0 |   22.22 |   22.72 | 7-85
 src/utils                  |      50 |    42.85 |   53.84 |      50 |
  database.js               |     100 |      100 |     100 |     100 |
  errors.js                 |   85.71 |      100 |   83.33 |   85.71 | 28
  message-manager.js        |       0 |        0 |       0 |       0 | 5-15
  utils.js                  |       0 |      100 |       0 |       0 | 2-3
----------------------------|---------|----------|---------|---------|--------------------------
Test Suites: 1 passed, 1 total
Tests:       6 passed, 6 total
Snapshots:   0 total
Time:        1.942 s, estimated 2 s
```

- View Profile

```
----------------------------|---------|----------|---------|---------|--------------------
All files                   |   23.88 |       10 |   18.79 |   24.54 |
 src                        |    92.3 |       50 |      50 |    92.3 |
  app.js                    |    92.3 |       50 |      50 |    92.3 | 32
 src/controllers            |   16.48 |        0 |   16.66 |   16.48 |
  photos.controllers.js     |      16 |        0 |      20 |      16 | 8-38,52-69
  trophies.controllers.js   |       0 |        0 |       0 |       0 | 6-46
  users.controllers.js      |   22.91 |        0 |      20 |   22.91 | 7-15,26-28,53-111
 src/data/db                |      24 |    18.75 |   14.81 |      24 |
  photo.db.js               |   36.36 |      100 |    12.5 |   36.36 | 20-64
  trophy.db.js              |   13.88 |        0 |       0 |   13.88 | 23-45,65-103
  user.db.js                |   32.14 |       50 |   27.27 |   32.14 | 26,32-35,43-83,92
 src/data/external          |      20 |       10 |       0 |   20.68 |
  fcm.external.js           |   16.66 |        0 |       0 |   18.18 | 11-55
  googleplaces.external.js  |   16.66 |     12.5 |       0 |   16.66 | 5-28
  googlesignin.external.js  |   33.33 |      100 |       0 |   33.33 | 7-13
 src/middleware             |   33.33 |       50 |   33.33 |   35.29 |
  auth.middleware.js        |     100 |      100 |     100 |     100 |
  upload.middleware.js      |    7.69 |        0 |       0 |    8.33 | 8-39
 src/routes                 |     100 |      100 |     100 |     100 |
  photos.routes.js          |     100 |      100 |     100 |     100 |
  trophies.routes.js        |     100 |      100 |     100 |     100 |
  users.routes.js           |     100 |      100 |     100 |     100 |
 src/services/photos        |    5.26 |        0 |   16.66 |    5.55 |
  photomanaging.js          |    5.26 |        0 |   16.66 |    5.55 | 7-20,31-47
 src/services/trophies      |    2.38 |        0 |       0 |    2.53 |
  trophycollection.js       |       0 |        0 |       0 |       0 | 9-40
  trophydetails.js          |    2.81 |        0 |       0 |    3.03 | 12-168
 src/services/users         |   20.96 |    12.14 |      20 |   21.78 |
  friends.js                |   20.96 |    13.04 |   27.27 |   20.96 | 9-94,116-130
  leaderboard.js            |    8.75 |      2.7 |       5 |    9.58 | 16,19-22,31-162
  useraccounts.js           |   43.18 |       25 |   44.44 |   43.18 | 7-16,24,37-73,78,85
 src/utils                  |   44.44 |    28.57 |   46.15 |   44.44 |
  database.js               |     100 |      100 |     100 |     100 |
  errors.js                 |   71.42 |    66.66 |   66.66 |   71.42 | 28-34
  message-manager.js        |       0 |        0 |       0 |       0 | 5-15
  utils.js                  |       0 |      100 |       0 |       0 | 2-3
----------------------------|---------|----------|---------|---------|--------------------
Test Suites: 1 passed, 1 total
Tests:       8 passed, 8 total
Snapshots:   0 total
Time:        1.415 s
```

- Watch Leaderboard

```
---------------------------|---------|----------|---------|---------|---------------------------------
All files                  |   24.59 |    12.27 |    20.3 |   25.09 |
 src                       |    92.3 |       50 |      50 |    92.3 |
  app.js                   |    92.3 |       50 |      50 |    92.3 | 32
 src/controllers           |   14.28 |     8.33 |   16.66 |   14.28 |
  photos.controllers.js    |       0 |        0 |       0 |       0 | 8-69
  trophies.controllers.js  |       0 |        0 |       0 |       0 | 6-46
  users.controllers.js     |   27.08 |       50 |   26.66 |   27.08 | 7-90
 src/data/db               |   22.66 |    18.75 |   14.81 |   22.66 |
  photo.db.js              |   27.27 |      100 |       0 |   27.27 | 17-64
  trophy.db.js             |   13.88 |        0 |       0 |   13.88 | 23-45,65-103
  user.db.js               |   32.14 |       50 |   36.36 |   32.14 | 26,32,38-48,61-83,92
 src/data/external         |      20 |       10 |       0 |   20.68 |
  fcm.external.js          |   16.66 |        0 |       0 |   18.18 | 11-55
  googleplaces.external.js |   16.66 |     12.5 |       0 |   16.66 | 5-28
  googlesignin.external.js |   33.33 |      100 |       0 |   33.33 | 7-13
 src/middleware            |   33.33 |       50 |   33.33 |   35.29 |
  auth.middleware.js       |     100 |      100 |     100 |     100 |
  upload.middleware.js     |    7.69 |        0 |       0 |    8.33 | 8-39
 src/routes                |     100 |      100 |     100 |     100 |
  photos.routes.js         |     100 |      100 |     100 |     100 |
  trophies.routes.js       |     100 |      100 |     100 |     100 |
  users.routes.js          |     100 |      100 |     100 |     100 |
 src/services/photos       |       0 |        0 |       0 |       0 |
  photomanaging.js         |       0 |        0 |       0 |       0 | 7-47
 src/services/trophies     |    2.38 |        0 |       0 |    2.53 |
  trophycollection.js      |       0 |        0 |       0 |       0 | 9-40
  trophydetails.js         |    2.81 |        0 |       0 |    3.03 | 12-168
 src/services/users        |   25.26 |    14.01 |    27.5 |   25.69 |
  friends.js               |   20.96 |    13.04 |   27.27 |   20.96 | 9-94,116-130
  leaderboard.js           |      30 |    24.32 |      30 |    31.5 | 16,19-22,31-50,62,67,77,81,96-162
  useraccounts.js          |   22.72 |        0 |   22.22 |   22.72 | 7-85
 src/utils                 |   44.44 |    42.85 |   46.15 |   44.44 |
  database.js              |     100 |      100 |     100 |     100 |
  errors.js                |   71.42 |      100 |   66.66 |   71.42 | 28-34
  message-manager.js       |       0 |        0 |       0 |       0 | 5-15
  utils.js                 |       0 |      100 |       0 |       0 | 2-3
---------------------------|---------|----------|---------|---------|---------------------------------
Test Suites: 1 passed, 1 total
Tests:       9 passed, 9 total
Snapshots:   0 total
Time:        1.421 s
```

- Review Trophy Details and Photos

File	% Stmts	% Branch	% Funcs	% Lines	Uncovered Line #s
All files	35.47	24.54	26.31	35.53	
src	92.3	50	50	92.3	
app.js	92.3	50	50	92.3	32
src/controllers	14.28	25	12.5	14.28	
photos.controllers.js	0	0	0	0	8-69
trophies.controllers.js	50	60	50	50	6-9,33-40
users.controllers.js	8.33	0	6.66	8.33	7-111
src/data/db	33.33	25	22.22	33.33	
photo.db.js	27.27	100	0	27.27	17-64
trophy.db.js	44.44	10	50	44.44	23-45,73-88,93
user.db.js	21.42	50	18.18	21.42	26,32-83,92
src/data/external	46.66	60	28.57	48.27	
fcm.external.js	16.66	0	0	18.18	11-55
googleplaces.external.js	83.33	75	100	83.33	5,23
googlesignin.external.js	33.33	100	0	33.33	7-13
src/middleware	33.33	50	33.33	35.29	
auth.middleware.js	100	100	100	100	
upload.middleware.js	7.69	0	0	8.33	8-39
src/routes	100	100	100	100	
photos.routes.js	100	100	100	100	
trophies.routes.js	100	100	100	100	
users.routes.js	100	100	100	100	
src/services/photos	0	0	0	0	
photomanaging.js	0	0	0	0	7-47
src/services/trophies	75	62.5	81.81	73.41	
trophycollection.js	0	0	0	0	9-40
trophydetails.js	88.73	73.52	100	87.87	18,66,92-93,98-101,168
src/services/users	16.12	6.54	15	16.75	
friends.js	20.96	13.04	27.27	20.96	9-94,116-130
leaderboard.js	8.75	2.7	5	9.58	16,19-22,31-162
useraccounts.js	22.72	0	22.22	22.72	7-85
src/utils	50	42.85	53.84	50	
database.js	100	100	100	100	
errors.js	85.71	100	83.33	85.71	28
message-manager.js	0	0	0	0	5-15
utils.js	0	100	0	0	2-3

```
Test Suites: 1 passed, 1 total
Tests:       7 passed, 7 total
Snapshots:   0 total
Time:        1.97 s, estimated 2 s
```

- Evaluate Photos

```
----------------------------|---------|----------|---------|---------|-------------------
All files                   |   25.49 |     10.9 |    21.8 |      26 |
 src                        |    92.3 |       50 |      50 |    92.3 |
  app.js                    |    92.3 |       50 |      50 |    92.3 | 32
 src/controllers            |   16.48 |     12.5 |   16.66 |   16.48 |
  photos.controllers.js     |      44 |       30 |      60 |      44 | 23-46,53
  trophies.controllers.js   |       0 |        0 |       0 |       0 | 6-46
  users.controllers.js      |    8.33 |        0 |    6.66 |    8.33 | 7-111
 src/data/db                |      24 |    18.75 |   22.22 |      24 |
  photo.db.js               |   63.63 |      100 |      50 |   63.63 | 17,38-51
  trophy.db.js              |   13.88 |        0 |       0 |   13.88 | 23-45,65-103
  user.db.js                |   21.42 |       50 |   18.18 |   21.42 | 26,32-83,92
 src/data/external          |      20 |       10 |       0 |   20.68 |
  fcm.external.js           |   16.66 |        0 |       0 |   18.18 | 11-55
  googleplaces.external.js  |   16.66 |     12.5 |       0 |   16.66 | 5-28
  googlesignin.external.js  |   33.33 |      100 |       0 |   33.33 | 7-13
 src/middleware             |   94.44 |       75 |     100 |   94.11 |
  auth.middleware.js        |     100 |      100 |     100 |     100 |
  upload.middleware.js      |    92.3 |       50 |     100 |   91.66 | 13
 src/routes                 |     100 |      100 |     100 |     100 |
  photos.routes.js          |     100 |      100 |     100 |     100 |
  trophies.routes.js        |     100 |      100 |     100 |     100 |
  users.routes.js           |     100 |      100 |     100 |     100 |
 src/services/photos        |   36.84 |       40 |   33.33 |   38.88 |
  photomanaging.js          |   36.84 |       40 |   33.33 |   38.88 | 25-47
 src/services/trophies      |    2.38 |        0 |       0 |    2.53 |
  trophycollection.js       |       0 |        0 |       0 |       0 | 9-40
  trophydetails.js          |    2.81 |        0 |       0 |    3.03 | 12-168
 src/services/users         |   16.12 |     6.54 |      15 |   16.75 |
  friends.js                |   20.96 |    13.04 |   27.27 |   20.96 | 9-94,116-130
  leaderboard.js            |    8.75 |      2.7 |       5 |    9.58 | 16,19-22,31-162
  useraccounts.js           |   22.72 |        0 |   22.22 |   22.72 | 7-85
 src/utils                  |      50 |    28.57 |   53.84 |      50 |
  database.js               |     100 |      100 |     100 |     100 |
  errors.js                 |   85.71 |    66.66 |   83.33 |   85.71 | 34
  message-manager.js        |       0 |        0 |       0 |       0 | 5-15
  utils.js                  |       0 |      100 |       0 |       0 | 2-3
----------------------------|---------|----------|---------|---------|-------------------
Test Suites: 1 passed, 1 total
Tests:       7 passed, 7 total
Snapshots:   0 total
Time:        1.426 s
```

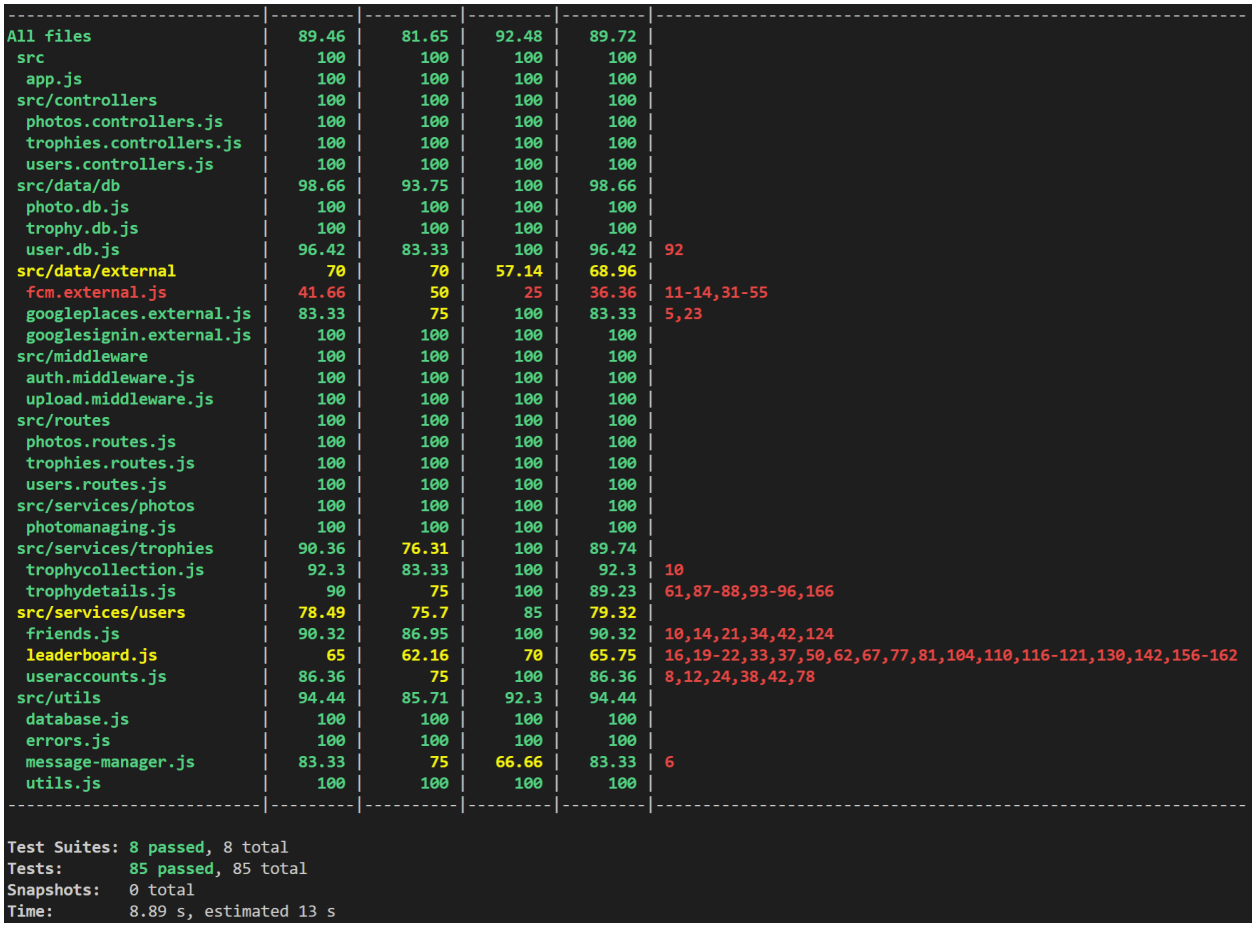
- Collect Trophy

```
---------------------------|---------|----------|---------|---------|-------------------------------------------------------------------
File                       | % Stmts | % Branch | % Funcs | % Lines | Uncovered Line #s
---------------------------|---------|----------|---------|---------|-------------------------------------------------------------------
All files                  |   41.35 |    29.09 |   39.84 |   41.39 |
 src                       |    92.3 |      100 |      50 |    92.3 |
  app.js                   |    92.3 |      100 |      50 |    92.3 | 32
 src/controllers           |   16.48 |     8.33 |   16.66 |   16.48 |
  photos.controllers.js    |      16 |       20 |      20 |      16 | 8-46,68-69
  trophies.controllers.js  |   22.22 |        0 |      25 |   22.22 | 13-46
  users.controllers.js     |   14.58 |        0 |   13.33 |   14.58 | 7-28,53-111
 src/data/db               |   66.66 |    81.25 |   51.85 |   66.66 |
  photo.db.js              |   45.45 |      100 |      25 |   45.45 | 20,38-64
  trophy.db.js             |   88.88 |      100 |    87.5 |   88.88 | 65-70
  user.db.js               |   46.42 |       50 |   45.45 |   46.42 | 26,32,48-83,92
 src/data/external         |      30 |       20 |   14.28 |   27.58 |
  fcm.external.js          |   41.66 |       50 |      25 |   36.36 | 11-14,31-55
  googleplaces.external.js |   16.66 |     12.5 |       0 |   16.66 | 5-28
  googlesignin.external.js |   33.33 |      100 |       0 |   33.33 | 7-13
 src/middleware            |     100 |      100 |     100 |     100 |
  auth.middleware.js       |     100 |      100 |     100 |     100 |
  upload.middleware.js     |     100 |      100 |     100 |     100 |
 src/routes                |     100 |      100 |     100 |     100 |
  photos.routes.js         |     100 |      100 |     100 |     100 |
  trophies.routes.js       |     100 |      100 |     100 |     100 |
  users.routes.js          |     100 |      100 |     100 |     100 |
 src/services/photos       |    5.26 |        0 |   16.66 |    5.55 |
  photomanaging.js         |    5.26 |        0 |   16.66 |    5.55 | 7-14,25-47
 src/services/trophies     |   16.66 |     12.5 |   18.18 |   17.72 |
  trophycollection.js      |    92.3 |    83.33 |     100 |    92.3 | 10
  trophydetails.js         |    2.81 |        0 |       0 |    3.03 | 12-168
 src/services/users        |   39.24 |    28.03 |    47.5 |   38.54 |
  friends.js               |   20.96 |    13.04 |   27.27 |   20.96 | 9-94,116-130
  leaderboard.js           |      50 |    43.24 |      60 |   49.31 | 16,19-22,33,37,50,61-86,104,110,116-121,130,142,156-162
  useraccounts.js          |   45.45 |    33.33 |   44.44 |   45.45 | 8,12,24,29,37-85
 src/utils                 |   72.22 |    85.71 |   61.53 |   72.22 |
  database.js              |     100 |      100 |     100 |     100 |
  errors.js                |   71.42 |      100 |   66.66 |   71.42 | 28-34
  message-manager.js       |   83.33 |       75 |   66.66 |   83.33 | 6
  utils.js                 |       0 |      100 |       0 |       0 | 2-3
---------------------------|---------|----------|---------|---------|-------------------------------------------------------------------
Test Suites: 1 passed, 1 total
Tests:       10 passed, 10 total
Snapshots:   0 total
Time:        3.06 s
```

- Manage Friends

```
----------------------------|---------|----------|---------|---------|-------------------
All files                   |   34.04 |       25 |   31.57 |   34.98 |
 src                        |     100 |       50 |     100 |     100 |
  app.js                    |     100 |       50 |     100 |     100 | 40
 src/controllers            |   25.27 |        0 |   33.33 |   25.27 |
  photos.controllers.js     |       0 |        0 |       0 |       0 | 8-69
  trophies.controllers.js   |       0 |        0 |       0 |       0 | 6-46
  users.controllers.js      |   47.91 |        0 |   53.33 |   47.91 | 7-34,96-111
 src/data/db                |      36 |    31.25 |   25.92 |      36 |
  photo.db.js               |   27.27 |      100 |       0 |   27.27 | 17-64
  trophy.db.js              |   13.88 |        0 |       0 |   13.88 | 23-45,65-103
  user.db.js                |   67.85 |    83.33 |   63.63 |   67.85 | 26,35-45,92
 src/data/external          |      20 |       10 |       0 |   20.68 |
  fcm.external.js           |   16.66 |        0 |       0 |   18.18 | 11-55
  googleplaces.external.js  |   16.66 |     12.5 |       0 |   16.66 | 5-28
  googlesignin.external.js  |   33.33 |      100 |       0 |   33.33 | 7-13
 src/middleware             |   33.33 |       50 |   33.33 |   35.29 |
  auth.middleware.js        |     100 |      100 |     100 |     100 |
  upload.middleware.js      |    7.69 |        0 |       0 |    8.33 | 8-39
 src/routes                 |     100 |      100 |     100 |     100 |
  photos.routes.js          |     100 |      100 |     100 |     100 |
  trophies.routes.js        |     100 |      100 |     100 |     100 |
  users.routes.js           |     100 |      100 |     100 |     100 |
 src/services/photos        |       0 |        0 |       0 |       0 |
  photomanaging.js          |       0 |        0 |       0 |       0 | 7-47
 src/services/trophies      |    2.38 |        0 |       0 |    2.53 |
  trophycollection.js       |       0 |        0 |       0 |       0 | 9-40
  trophydetails.js          |    2.81 |        0 |       0 |    3.03 | 12-168
 src/services/users         |   40.86 |    40.18 |    37.5 |   42.45 |
  friends.js                |   90.32 |    86.95 |     100 |   90.32 | 10,14,21,34,42,124
  leaderboard.js            |    8.75 |      2.7 |       5 |    9.58 | 16,19-22,31-162
  useraccounts.js           |   29.54 |     8.33 |   33.33 |   29.54 | 7-64,77-85
 src/utils                  |   61.11 |    42.85 |   69.23 |   61.11 |
  database.js               |     100 |      100 |     100 |     100 |
  errors.js                 |   85.71 |      100 |   83.33 |   85.71 | 28
  message-manager.js        |       0 |        0 |       0 |       0 | 5-15
  utils.js                  |     100 |      100 |     100 |     100 |
----------------------------|---------|----------|---------|---------|-------------------
Test Suites: 1 passed, 1 total
Tests:       28 passed, 28 total
Snapshots:   0 total
Time:        1.64 s
```

- All all use cases tests together



```
---------------------------|---------|----------|---------|---------|-------------------------------------------------------------------
All files                  |   89.46 |    81.65 |   92.48 |   89.72 |
 src                       |     100 |      100 |     100 |     100 |
  app.js                   |     100 |      100 |     100 |     100 |
 src/controllers           |     100 |      100 |     100 |     100 |
  photos.controllers.js    |     100 |      100 |     100 |     100 |
  trophies.controllers.js  |     100 |      100 |     100 |     100 |
  users.controllers.js     |     100 |      100 |     100 |     100 |
 src/data/db               |   98.66 |    93.75 |     100 |   98.66 |
  photo.db.js              |     100 |      100 |     100 |     100 |
  trophy.db.js             |     100 |      100 |     100 |     100 |
  user.db.js               |   96.42 |    83.33 |     100 |   96.42 | 92
 src/data/external         |      70 |       70 |   57.14 |   68.96 |
  fcm.external.js          |   41.66 |       50 |      25 |   36.36 | 11-14,31-55
  googleplaces.external.js |   83.33 |       75 |     100 |   83.33 | 5,23
  googlesignin.external.js |     100 |      100 |     100 |     100 |
 src/middleware            |     100 |      100 |     100 |     100 |
  auth.middleware.js       |     100 |      100 |     100 |     100 |
  upload.middleware.js     |     100 |      100 |     100 |     100 |
 src/routes                |     100 |      100 |     100 |     100 |
  photos.routes.js         |     100 |      100 |     100 |     100 |
  trophies.routes.js       |     100 |      100 |     100 |     100 |
  users.routes.js          |     100 |      100 |     100 |     100 |
 src/services/photos       |     100 |      100 |     100 |     100 |
  photomanaging.js         |     100 |      100 |     100 |     100 |
 src/services/trophies     |   90.36 |    76.31 |     100 |   89.74 |
  trophycollection.js      |    92.3 |    83.33 |     100 |    92.3 | 10
  trophydetails.js         |      90 |       75 |     100 |   89.23 | 61,87-88,93-96,166
 src/services/users        |   78.49 |     75.7 |      85 |   79.32 |
  friends.js               |   90.32 |    86.95 |     100 |   90.32 | 10,14,21,34,42,124
  leaderboard.js           |      65 |    62.16 |      70 |   65.75 | 16,19-22,33,37,50,62,67,77,81,104,110,116-121,130,142,156-162
  useraccounts.js          |   86.36 |       75 |     100 |   86.36 | 8,12,24,38,42,78
 src/utils                 |   94.44 |    85.71 |    92.3 |   94.44 |
  database.js              |     100 |      100 |     100 |     100 |
  errors.js                |     100 |      100 |     100 |     100 |
  message-manager.js       |   83.33 |       75 |   66.66 |   83.33 | 6
  utils.js                 |     100 |      100 |     100 |     100 |
---------------------------|---------|----------|---------|---------|-------------------------------------------------------------------

Test Suites: 8 passed, 8 total
Tests:       85 passed, 85 total
Snapshots:   0 total
Time:        8.89 s, estimated 13 s
```

- Justifications for the lower coverage
  - src/data/db
    - user.db.js
      - 92: this statement is for handling MongoDB upsert failures
  - src/data/external
    - fcm.external.js:
      - 11-14,31-55: our per-use case tests do not test sending notifications to the frontend, so these statements are not reached
    - googleplaces.external.js
      - 5: this checks if the GOOGLE_MAPS_API_KEY environment variable exists, which is always true if the server is configured correctly
      - 23: this statement is for handling unknown errors from an external service
  - src/services/trophies
    - trophycollection.js
      - 10: this statement is for handling missing inputs; they are not reachable because the inputs come from URL path parameters, which must exist for a valid route

        - trophydetails.js
            - 61: this statement is for handling an missing input to the function, which will not occur if the program is correct
            - 87-88,93-96: these are for handling different qualities of trophies; we do not have control over which qualities we receive as they are from an external service
            - 166: this statement is for handling MongoDB upsert failures
    - src/services/users
        - friends.js
            - 10,14,21,34,42: these statements are for handling invalid user id or non-existing user; they are not reachable in interface tests because the user id comes from the session, and the user must log in before using the interfaces
            - 124: our per-use case tests do not test sending notifications to the frontend, so these statements are not reached
        - leaderboard.js
            - 16,19-22: these statements are for cleaning up stale subscriptions; we did not include them in our tests because the first clean-up would not take place until 15 minutes after the server is started
            - 33,37,62,67,77,81: these statements are for handling invalid user id or non-existing user; they are not reachable in interface tests because the user id comes from the session, and the user must log in before using the interfaces
            - 50,104,110,116-121,130,142: our per-use case tests do not test sending notifications to the frontend, so these statements are not reached
            - 156,162: these are helper functions for module tests
        - useraccounts.js
            - 8,12,24,38,42,78: these statements are for handling invalid user id or non-existing user; they are not reachable in interface tests because the user id comes from the session, and the user must log in before using the interfaces
    - src/utils
        - message-manager.js
            - 6: this statement is for logging all unknown exceptions from the subscriber callbacks; it is not reached if the server is running correctly

1.16.1.5. A table listing fully qualified names of all files for each of the backend modules

Module	Name	Use	Path
Photos	photomanaging	Sub-module	backend/src/services/

			photos/photomanaging.js
Photos	photomanaging_mock	mock	backend/src/services/photos/__mocks__/photomanaging.js
Trophies	trophcollection	Sub-module	backend/src/services/trophies/trophycollection.js
Trophies	trophydetails	Sub-module	backend/src/services/trophies/trophydetails.js
Trophies	trophyfilter	Sub-module	backend/src/services/trophies/trophyfilter.js
Trophies	trophcollection_test	Test	backend/src/services/trophies/__tests__/trophycollection.test.js
Trophies	trophydetails_test	Test	backend/src/services/trophies/__tests__/trophydetails.test.js
Trophies	trophcollection_mock	Mock	backend/src/services/trophies/__mocks__/trophycollection.js
Trophies	trophydetails_mock	Mock	backend/src/services/trophies/__mocks__/trophydetails.js
Trophies	trophyfilter_mock	Mock	backend/src/services/trophies/__mocks__/trophyfilter.js
User	friends	Sub-module	backend/src/services/users/friends.js
User	leaderboard	Sub-module	backend/src/services/users/leaderboard.js
User	useraccounts	Sub-module	backend/src/services/users/useraccounts.js
User	friends_mock	Mock	backend/src/services/users/__mocks__/friends.js

User	leaderboard_mock	Mock	backend/src/services/users/__mocks__/leaderboard.js
User	useraccounts_mock	Mock	backend/src/services/users/__mocks__/useraccounts.js
User	friends_mock	Test	backend/src/services/users/__test__/friends.js
User	leaderboard_mock	Test	backend/src/services/users/__test__/leaderboard.js
User	useraccounts_mock	Test	backend/src/services/users/__test__/useraccounts.js
Message-Manager	message-manager	Module	backend/src/utils/message-manager.js
Message-Manager	message-manager_mock	Mock	backend/src/utils/__mocks__/message-manager.js
Photo	PhotoDB	Database	backend/src/data/db/photo.db.js
Trophy	TrophyDB	Database	backend/src/data/db/trophy.db.js
User	UserDB	Database	backend/src/data/db/user.db.js

1.16.2. Frontend Tests Results

1.16.2.1. Location and Description of Front-end Test Suites

- Location:
  The front-end tests directory contains all the front-end test suites. Each use case has one java class file for all its tests as followed:
  - Review Trophy Details and Photos Usecase: https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/devel/frontend/app/src/androidTest/java/com/worldexplorationaction/android/frontend/TrophyDetailsAndPhotosTests.java

- Evaluate Photos Usecase: https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/devel/frontend/app/src/androidTest/java/com/worldexplorationaction/android/frontend/EvaluatePhotosTests.java

- Manage Friends Usecase: https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/devel/frontend/app/src/androidTest/java/com/worldexplorationaction/android/frontend/ManageFriendsTests.java

Location of individual tests: 1.15.3. Front-end Tests Design and Locations

- Running tests:
  1. Make sure the secrets are entered:
     a. Download frontend/app/google-services.json: https://drive.google.com/file/d/1hcmKg505pXg1P-lm_PSCoCV4ghFIB20k/view?usp=sharing
     b. Enter in frontend/local.properties:
     ```
     MAPS_API_KEY=AIzaSyAtLaDaObbjpXy1-MWc0YMMZQERBrjSjJM
     ```
     c. Add this line in frontend/app/src/main/res/values/strings.xml:
     ```
     <string name="google_client_id">507381724545-lpf2ba1sve9lm4ihkisq087m7ivepsg3.apps.googleusercontent.com</string>
     ```
  2. You can run all the tcv ests in each file by pressing right click on the file from the project tab and choosing “run” or pressing Ctrl+Shift+F10. You can also run each test in the file individually by clicking on “run test” besides the test name.

1.16.2.3. Espresso Execution Log

- Review Trophy Details and Photos Usecase

- Evaluate Photos Usecase:

- Manage Friends Usecase:

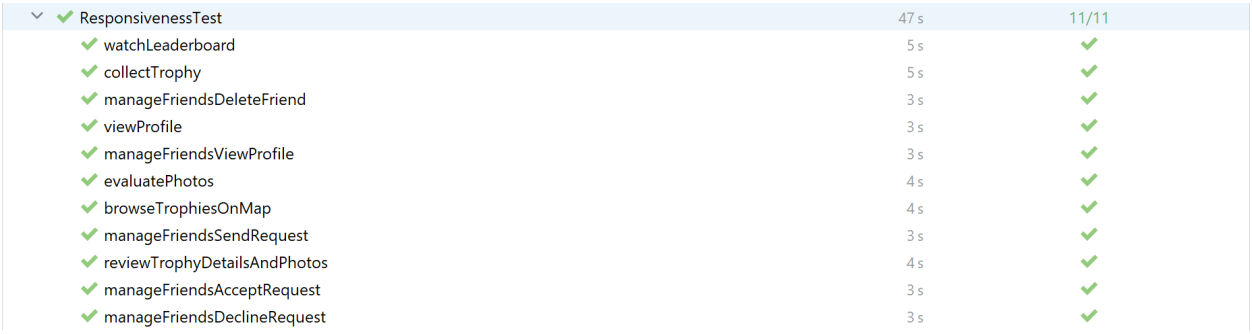
1.16.3. Non-Functional Requirement Verification

1.16.3.1 Responsiveness

- Location:
  - https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/e2d6cf472ad8af67ace43884b7efe8de090e51bf/frontend/app/src/androidTest/java/com/worldexplorationaction/android/nonfunctional/ResponsivenessTest.java
- Explanation:
  - This non-functional requirement is verified by UI tests that perform all actions, measure the runtime of each user operation, and compare the runtime to a predefined threshold.

```
104     private static void runWithRuntimeCheck(Runnable task) {
105         long startTime = System.nanoTime();
106         task.run();
107         long endTime = System.nanoTime();
108         long duration = (endTime - startTime);
109         System.out.println("runtime: " + duration);
110         Assert.assertTrue("Not Sufficiently responsive, delay=" + duration, duration < MAX_DELAY);
111     }
```

  - Since Espresso automatically waits for resources to be idle, this will correctly measure the app's response time. (To ensure Espresso also waits for network requests, the HTTP client is also registered as an idling resource.)



1.16.3.2 Usability

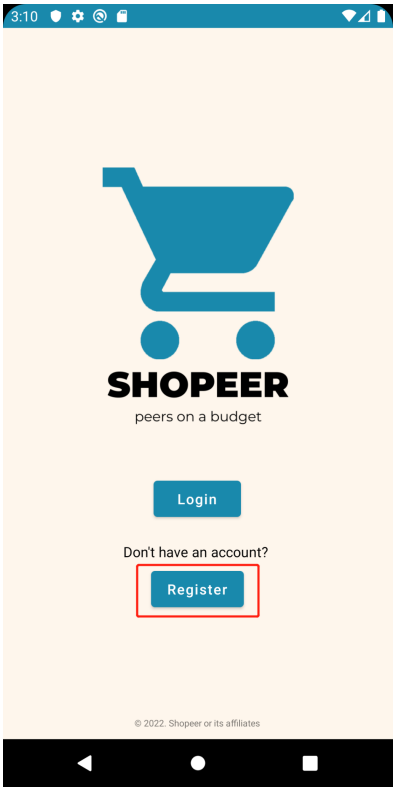
- Location:
  - https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action/blob/e2d6cf472ad8af67ace43884b7efe8de090e51bf/frontend/app/src/androidTest/java/com/worldexplorationaction/android/nonfunctional/UsabilityTest.java
- Explanation:
  - We verify this non-functional requirement by writing a UI test for each action. For each action, we use assertions to ensure the number of operations is not greater than 5.

```
110     private void increaseOperationCount(int value) {
111         int newCount = operationCount.addAndGet(value);
112         Assert.assertTrue("Too many operations", newCount <= OPERATION_COUNT_LIMIT);
113     }
114
115     private void runOneOperation(Runnable task) {
116         InstrumentationRegistry.getInstrumentation().waitForIdleSync();
117         increaseOperationCount(1);
118         task.run();
119     }
```

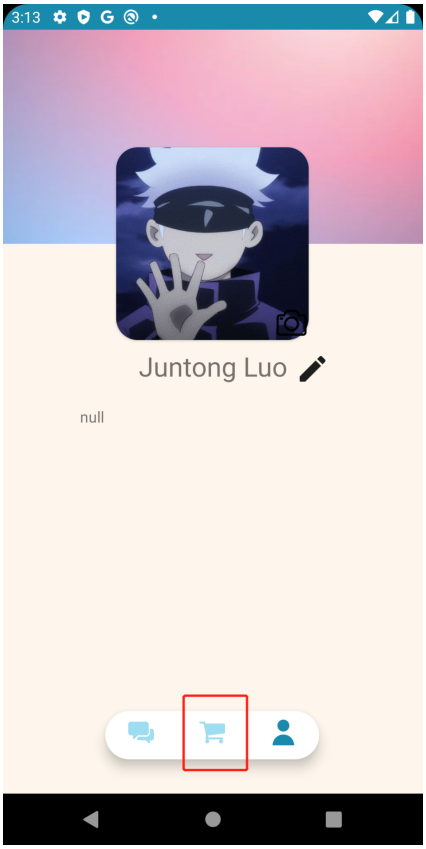
1.16.4. Customer Acceptance Testing

1.16.4.1. One major fault in the peer-group's app

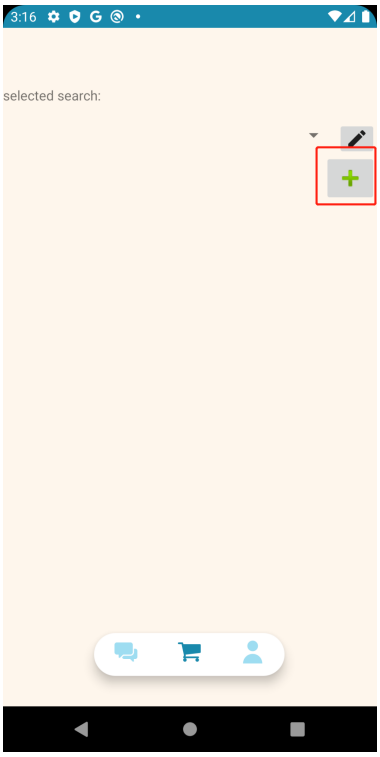
- Major Fault:
  When the user enters any character in the search view, the search view is immediately closed and no result is presented.
- Sequence of events:
  1. Click Register and login to the google account



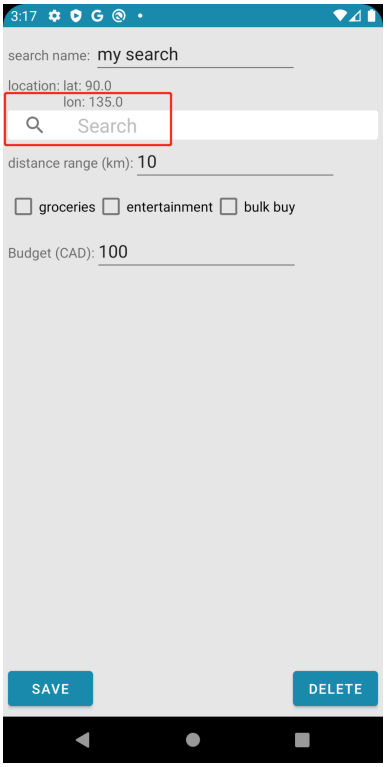
2. Click the shopping cart icon at the bottom



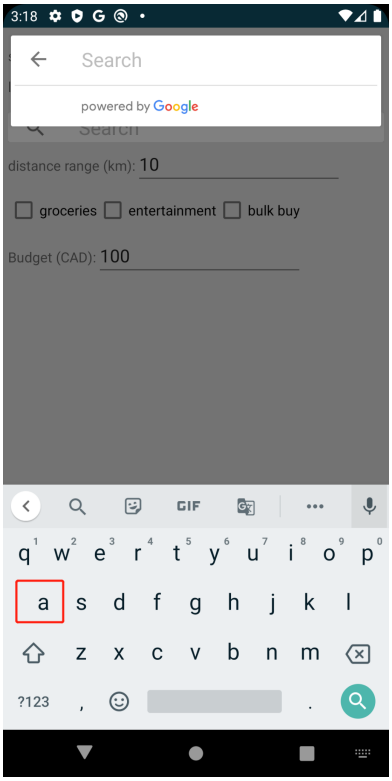
3. Click the green add button



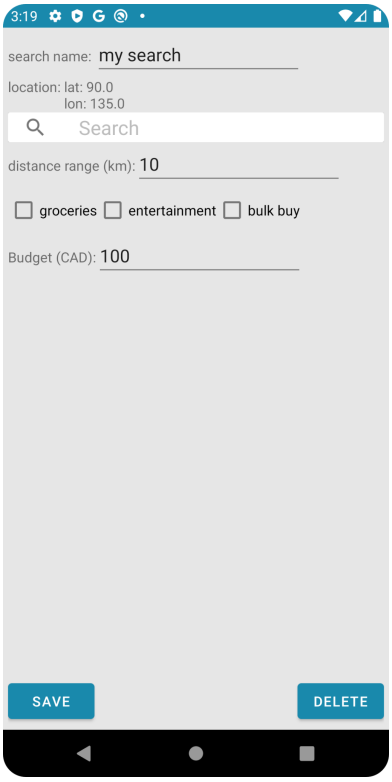
4. Click the search bar



5. Enter “a”

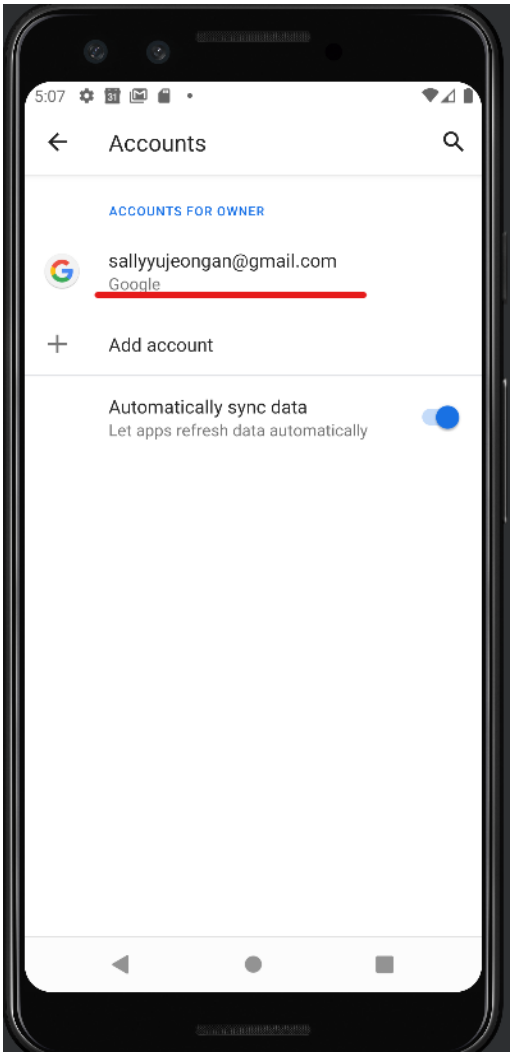


6. Fault: The search view is immediately closed

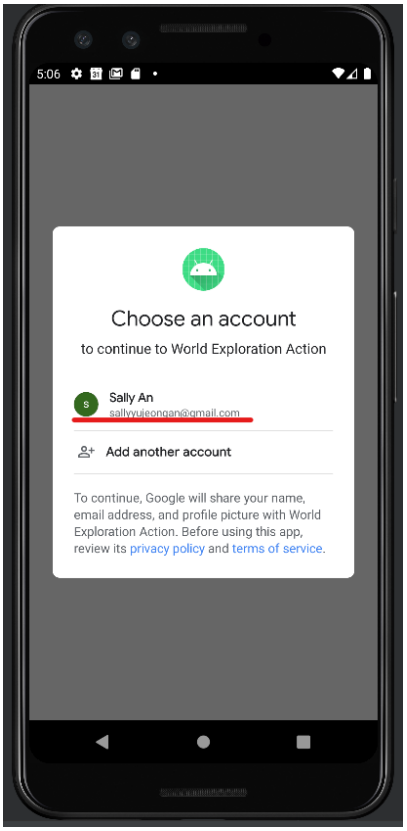


1.16.4.2. One major fault that the peer-group found in our app

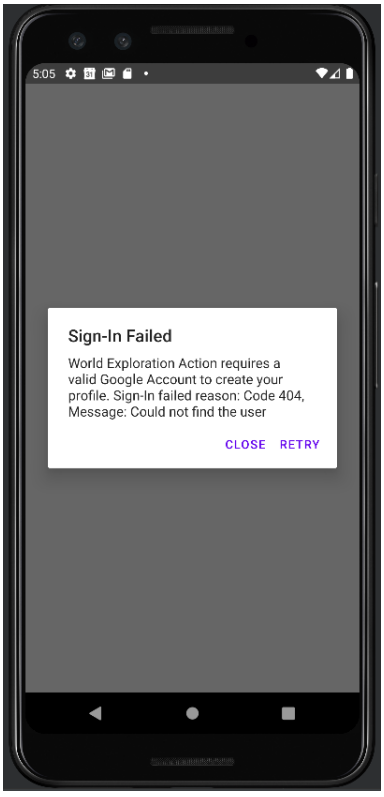
- Major Fault:
  When user first enters the app, they are unable to login using an existing Google account linked to the device.
- Sequence of events:
    1. Before entering the app, make sure a Google account is already added to the device


5:07
Accounts
ACCOUNTS FOR OWNER
sallyyujeongan@gmail.com
Google
Add account
Automatically sync data
Let apps refresh data automatically

2. Open the app and select the Google account



3. Fault: sign-in failed



2. Change in Project

2.1 Scope

None

2.2 Requirements

None

2.3 Design

- Updated sequence diagrams to match the current design

2.4 Test Plan

- Fixed errors in the Manage Friends Interface tests
- Added Upload FCM Token tests
- Test cases about session with non-existing user's ID are removed, as a logged in user must exist

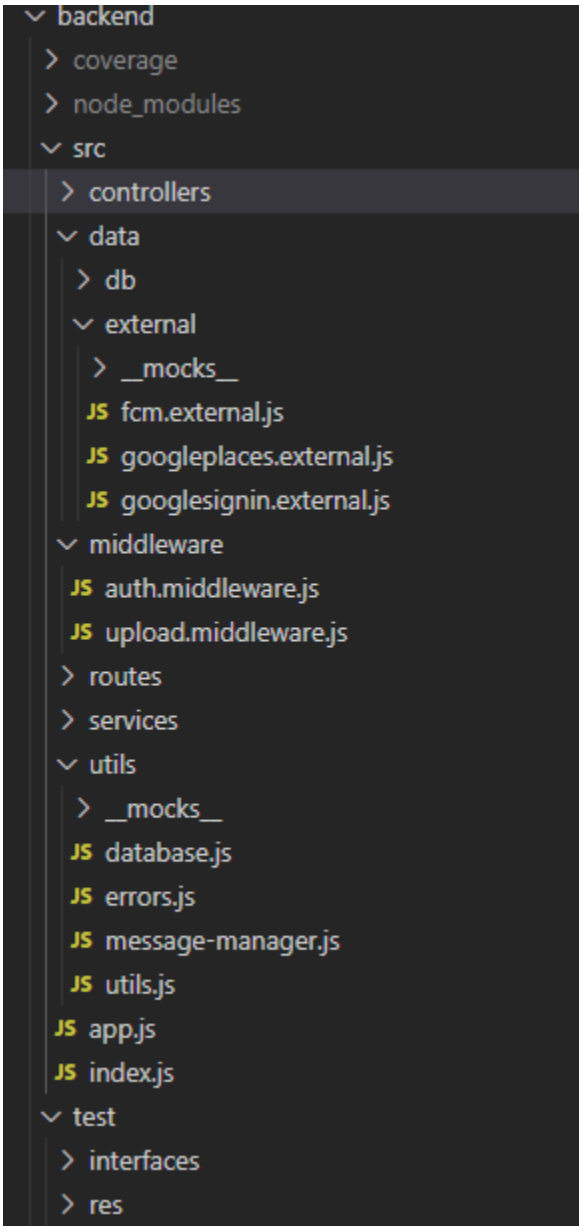
2.4 Test Results

- Updated test results

3. "Humblebrag"

- We insist on producing code of high quality.
- Backend Code Structure and Quality:

- All the codes are well formatted. Each file only contained the corresponding codes. The coding style is clean and efficient.



- Almost all lines of codes are used

All files	97.5	95.41	96.99	97.43	
src	100	100	100	100	
app.js	100	100	100	100	
src/controllers	100	100	100	100	
photos.controllers.js	100	100	100	100	
trophies.controllers.js	100	100	100	100	
users.controllers.js	100	100	100	100	
src/data/db	98.66	93.75	100	98.66	
photo.db.js	100	100	100	100	
trophy.db.js	100	100	100	100	
user.db.js	96.42	83.33	100	96.42	92
trophydetails.js	97.14	87.5	100	96.92	96,166
src/services/users	99.46	99.06	100	99.44	
friends.js	100	100	100	100	
leaderboard.js	98.75	97.29	100	98.63	22
useraccounts.js	100	100	100	100	
src/utils	94.44	85.71	92.3	94.44	
database.js	100	100	100	100	
errors.js	100	100	100	100	
message-manager.js	83.33	75	66.66	83.33	6
utils.js	100	100	100	100	

Logout

- External Services Usage
    - The auth middleware is deliberately used to keep the security of the project
    - Only necessary parts of external APIs are extracted and used

4. The contributions of each team member to the work done on the project (2-3 sentences for each team member).

- Houlin (Mike) - Use-Case and Sequence Diagram designs. Backend modules and databases implementation. Mock the backend modules, and external services. Module test designs, and module test and use-case test implementation.

- Sara - Designed the User Interface and Prototype of the app, implemented the frontend of Trophy Details, evaluate photos, and profile page, and developed all the frontend tests for the central use cases.

- Juntong - Setup and created skeletons of the frontend and the backend, implemented the UI for the map, leaderboard, and friends page, implemented the backend Users module, wrote all NFRs tests and some backend tests, setup GitHub Action and Codacy, and helped other teammates

- Francisco (Copied from M5-M7)
    - Backend Trophy routes, Backend token authentication and sessions (login), Backend Trophy functions for both users and trophies (view map, collect trophy, view trophy details), backend places api setup (get user trophies)
    - Design Tests for the exposed interfaces between backend and frontend
    - Fixed some problems reported by Codacy. Found security issues in peer group backend. Fixed lazy class code smell reported by peer group.

5. GitHub repository and Location of Tests

- Link to the GitHub repository:
    - https://github.com/CPEN-321-World-Exploration-Action/CPEN-321-World_Exploration_Action
- Location of backend tests:
    - 1.16.1.1. Location and Description of Back-end Test Suites
- Location of frontend tests:
    - Use Case tests
        - 1.16.2.1. Location and Description of Front-end Test Suites
    - Non-Functional Requirements Tests
        - 1.16.3. Non-Functional Requirement Verification

6. Physical device

- Manufacturer: Xiaomi
- Model: M2011K2C

7. Details needed to test the app

- Please refer to 1.16.2.1. Location and Description of Front-end Test Suites for how to run front-end tests and non-functional requirements tests, and 1.16.1.1. Location and Description of Back-end Test Suites for how to run backend tests.

PART II. Project Presentation

PART III. Project Videos

PART IV. Mobile Application