

Wade Johnson (wmj2108)
Charlie Summers (cgs2161)
Nick Clark (ntc2120)
Wenhe Henry Qin (whq2000)
Magikarp
COMS W4156
Team Assignment 3

Part 1: GitHub Repository Link

<https://github.com/hwqin16/Magikarp>

Git Commit SHA: 99dcad2ed6f79bfa21e7c2f3d9b7ec992c734d0a

Part 2: User Stories and Acceptance Testing Plan

User Stories:

1. As a traveler, I want to discover crowdsourced travel tips and advice directly on my phone so that I can be best informed during my vacations.

My conditions of satisfaction are:

- An application on a mobile device that I can carry with me while travelling.
- Ability to read posts created by other users.
- If the application cannot communicate with the backend or fails, provide the ability to view the most recently accessed data offline or inform the user if no such data is available.

2. As a traveler, I want to be able to find the best travel locations near me so that I know where to go in order to have the best experience.

My conditions of satisfaction are:

- An interactive map of posts that shows my location relative to various crowdsourced destinations.
- Ability to read the posts that I select from the map.
- If the post cannot be selected (for example, due to network connectivity issues) inform the user that the post cannot be viewed right now.

3. As a content creator, I want to share geotagged text descriptions and pictures of areas of interest so that I can help other travelers.

My conditions of satisfaction are:

- Create a post and add a text description and picture of an area of interest.
- Geotag a post using the current location on the device.
- If a post cannot be created, ask the user to try again.
- If the current location cannot be accessed via GPS, inform the user and ask them to try again.

4. As a new content creator, I want to be able to log in using Google OAuth so I can create posts.

My conditions of satisfaction are:

- Can create an account by linking their Google account and then create posts.
- If registration fails (for example, because the user already has an account linked) then inform the user.
- If the user attempts to create a post without a linked Google account, respond with an error message saying they cannot do that.

5. As a content creator, I want to auto-save partially complete post drafts so that I can finish them later without losing my work.

My conditions of satisfaction are:

- Auto-save drafts of posts while the user is creating them so they can be finished and published later.
- If the app closes and then is re-opened, the user should return to the post they were writing.
- If the auto-save fails for any reason (for example, because the user has not granted access to the required space) then inform the user so they know their current input cannot be saved.

Acceptance Testing Plan :

User Story 1 and 2:

1. Valid Input
 - a. Open the app
 - b. Navigate to my current location
 - c. Be able to select on interesting posts in my area
 - d. Turn off wifi, and still be able to see values
2. Invalid Input (No GPS or Network to fetch posts and/or location)
 - a. Disable network and GPS
 - b. Open the app
 - c. Verify the map is blank with no pins.

User Story 3 and 4:

1. Valid Input
 - a. Open the app
 - b. Login to my google account on the app
 - c. Verify that the views shown are that of an authenticated user
 - d. Navigate to my current location
 - e. Create a post and publish
 - f. Verify the post was posted on the App
2. Invalid Input
 - a. Disable network and GPS
 - b. Open the app
 - c. Login to my account
 - d. Fails due to lack of network access
 - e. Notifies user that that there is no network connection
 - f. Verify that the views shown are still that of a non-authenticated user

User Story 5:

1. Valid Input
 - a. Post with image, no text
 - i. Open the app
 - ii. Login to my google account on the app
 - iii. Navigate to my current location
 - iv. Create post with image, but no text
 - v. Send application to background
 - vi. Reopen application
 - vii. Navigate back to the create post view
 - viii. Verify the image is still in the editor
 - b. Post with text, no image
 - i. Open the app
 - ii. Login to my google account on the app
 - iii. Navigate to my current location
 - iv. Create post with text, but no image
 - v. Send application to background
 - vi. Reopen application
 - vii. Navigate back to the create post view
 - viii. Verify the text is still in the editor
 - c. Post with text and image
 - i. Open the app
 - ii. Login to my google account on the app
 - iii. Navigate to my current location
 - iv. Create post with text and image
 - v. Send application to background
 - vi. Reopen application
 - vii. Navigate back to the create post view
 - viii. Verify the text and image is still in the editor
2. Invalid Input (Explicit kill of application)
 - a. Post with image, no text
 - i. Open the app
 - ii. Login to my google account on the app
 - iii. Navigate to my current location
 - iv. Create post with image, but no text
 - v. Kill application
 - vi. Reopen application
 - vii. Navigate back to the create post view
 - viii. Verify the image is not in the editor
 - b. Post with text, no image
 - i. Open the app
 - ii. Login to my google account on the app
 - iii. Navigate to my current location
 - iv. Create post with text, but no image
 - v. Kill application
 - vi. Reopen application
 - vii. Navigate back to the create post view
 - viii. Verify the text is not still in the editor
 - c. Post with text and image
 - i. Open the app
 - ii. Login to my google account on the app

- iii. Navigate to my current location
- iv. Create post with text and image
- v. Kill application
- vi. Reopen application
- vii. Navigate back to the create post view
- viii. Verify the text and image is not still in the editor

Discussion of Bugs:

- User Stories 1 and 2:
 - When working with the maps, we found an issue with the way that the google maps API key was configured, which would not allow some instances of the application to view the map. After reconfiguring the key and tweaking the settings on the emulators, we were able to get the application to draw the map for all users.
 - With regard to map ease of use, we found that many posts in a single place on a map are difficult to distinguish and access. We plan to implement a clustering algorithm that will show the number of posts in a particular area if they are close together.
- User Story 3:
 - When performing testing for posting stories, we found that the application crashed on a post if the GPS location could not be resolved due to an unchecked null pointer exception. In order to fix this, we added a null check on the GPS location before continuing.
 - When selecting an image, we found a bug where some images were too large for the Parcelable data type in a saved argument bundle if the user left the app in the middle of creating a post. This led to application crashing. To fix this, we only store a URL of the image in the bundle and reload the image from storage when the UI is restored.
- User Story 4:
 - There was initially a learning curve with the Google Sign-In functionality regarding SHA-1 fingerprints which delayed sign in functionality testing. Once we determined that each developer needed to upload their own SHA-1 fingerprint to build the application on their local device we were able to have the entire team create successful builds. In reality, users would not be building their own app, and APK files would have been distributed by a developer with a valid fingerprint.
 - The UI is designed to prevent users from being able to create a post if they are not signed in, so implementing an error is not necessary. We were not able to fully implement everything we wanted to by the deadline so it is still possible through a roundabout process, however we plan to have this completed in the next iteration.
- User Story 5:
 - Initially, we were not successfully retrieving the last known image and text from the saved arguments. This was because the arguments were not being saved correctly in the event the user left the app while creating a post. Once that was resolved, we successfully were able to retrieve values from the saved state.
 - One outstanding known bug is in regard to editing existing posts. If the user does not select a new picture then the post cannot be uploaded. Replacing the picture with a picture on the device will allow correct functionality. We will fix this in the next iteration.

Part 3: Automated Build and Unit Testing

Automated Build File:

<https://github.com/hwqin16/Magikarp/blob/master/.github/workflows/magikarp.yml>

Unit Test Files:

<https://github.com/hwqin16/Magikarp/tree/master/server/src/test/java> (All sub directories)

<https://github.com/hwqin16/Magikarp/tree/master/app/src/test/java> (All sub directories)

<https://github.com/hwqin16/Magikarp/tree/master/app/src/androidTest/java/com/magikarp/android> (All sub directories)

Part 4: Checkstyle and Spot Bugs

<https://github.com/hwqin16/Magikarp/tree/master/app/build/reports/checkstyle>

<https://github.com/hwqin16/Magikarp/tree/master/app/build/reports/spotbugs>

<https://github.com/hwqin16/Magikarp/tree/master/server/build/reports/checkstyle>

<https://github.com/hwqin16/Magikarp/tree/master/server/build/reports/spotbugs>

Wade Johnson (wmj2108)
Charlie Summers (cgs2161)
Nick Clark (ntc2120)
Wenhe Henry Qin (whq2000)
Magikarp
COMS W4156
Magikarp API

POST /messages

Get a collection of posts based on a geographic area.

Authorization

No authentication is required to get messages.

Request Parameters

Query Parameters

- latitude_top (double)
 - The northernmost latitude of the requested geographic boundary.
 - Format: **(TBD)** DDMM.SSS
- longitude_left (double)
 - The leftmost longitude of the requested geographic boundary. Left does not necessarily imply the western-most point (ex. crossing International Date Line).
 - Format: **(TBD)** DDDMM.SSS
- latitude_bottom (double)
 - The southernmost latitude of the requested geographic boundary.
 - Format: **(TBD)** DDMM.SSS
- longitude_right (double)
 - The rightmost longitude of the requested geographic boundary. Right does not necessarily imply the eastern-most point (ex. crossing International Date Line).
 - Format: **(TBD)** DDDMM.SSS
- max_records (integer)
 - The maximum number of messages located in the geographic boundary to return

Responses

200 application/json

```
{
  "record_count" : "integer",
  "records" : [
    {
      "id" : "integer",
      "user_id" : "integer",
      "image_url" : "string",
      "text" : "string",
      "latitude" : "double",
      "longitude" : "double",
      "timestamp" : "string [date-time]"
    }
  ]
}
```

204 application/json

```
{
}
```

POST /messages/{user_id}

Get a collection of posts created by a specific user.

Authorization

No authentication is required to get messages.

Request Parameters

Path Parameters

- user_id (integer)
 - The user ID.

Responses

200 application/json

```
{
  "record_count" : "integer",
  "records" : [
    {
      "id" : "integer",
      "user_id" : "integer",
      "image_url" : "string",
      "text" : "string",
      "latitude" : "double",
      "longitude" : "double",
      "timestamp" : "string [date-time]"
    }
  ]
}
```

204 application/json

```
{
}
```


POST /messages/{user_id}/new

Create a new message.

Request Parameters

Path Parameters

- user_id (integer)
 - The user ID.

Request Body

- image_url (string)
 - The url of the image
- text (string)
 - The message text.
- latitude (double)
 - The message geotag latitude.
 - Format: **(TBD)** DDMM.SSS
- longitude (double)
 - The message geotag longitude.
 - Format: **(TBD)** DDDMM.SSS

Responses

201 application/json

```
{  
  "record_id" : "string"  
}
```

401 application/json

```
{  
  "responseCode": "integer"  
  "error" : "string"  
}
```

POST /messages/{user_id}/update/{record_id}

Update an existing message.

Request Parameters

Path Parameters

- user_id (integer)
 - The user ID.
- record_id (integer)
 - The ID of the record to update.

Request Body

- image (binary)
 - The message image.
- text (string)
 - The message text.
- latitude (double)
 - The message geotag latitude.
 - Format: **(TBD)** DDMM.SSS
- longitude (double)
 - The message geotag longitude.
 - Format: **(TBD)** DDDMM.SSS

Responses

200 application/json

```
{  
  "responseCode": "integer"  
}
```

401 application/json

```
{  
  "responseCode": "integer"  
  "error" : "string"  
}
```

POST /messages/{user_id}/delete/{record_id}

Delete an existing message.

Request Parameters

Path Parameters

- user_id (integer)
 - The user ID.
- record_id (integer)
 - The ID of the record to delete.

Responses

200 application/json

```
{  
  "responseCode": "integer"  
}
```

401 application/json

```
{  
  "responseCode": "integer"  
  "error" : "string"  
}
```