Wade Johnson (wmj2108)
Charlie Summers (cgs2161)
Nick Clark (ntc2120)
Wenhe Henry Qin (whq2000)
Magikarp
COMS W4156
Team Assignment 2

Part 1 Overview

In this project, our goal is to create a crowdsourced travel assistance app where users can search and view travel tips and advice based on geotagged posts created by other users. Posts would be short first-person experiences on a specific geographic area of interest (perhaps an out of the way coffee shop or the trailhead to a hidden hiking path) that provides unique information that is not usually found in a commercial travel guide or website.

Content creators would create posts by writing short descriptions and adding a picture and geotag of their location. Content consumers would use a map interface to search and filter on geographic areas of interest to find posts. All crowdsourced content would be available to the public without needing to authenticate with credentials, however to develop and post content a user must create and log into an account. We plan to create a mobile Android app as a front end with account credentials securely stored on the device, so a login timeout is not required. An explicit logout will be available using the built-in account management features of the operating system.

We plan to demo the app using an Android emulator on a computer, which can be done with screen sharing on any video conferencing application that supports it. Since the emulator can perform most of the major functions of a real device, including simulating location services, we should be able to demo all of the primary application front end components.

Content will be stored on a cloud server accessible from anywhere on the internet. Temporary working data will be stored on client devices themselves, for example if a creator wants to stop working on a post and come back to it later. Data stored on the server and the device will consist primarily of text, as well as some metadata such as geotagging information.

We plan to use the Google Maps SDK external API in order to handle retrieving maps and to query for layers of the map on the App. We will interact with Google OAuth in order to register users. *For the MVP we will only allow users to login with a Google Account.*

Part 2 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 current GPS location is not available, inform the user that they cannot connect to the GPS and show a standard map that the user will have to scroll to their own location manually.
   - 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.

Part 3 Acceptance Testing

1. Internal acceptance testing for User Story #1 includes planning and designing test cases to cover all the functional methods within the API dealing with accessing and viewing posts. External acceptance testing would involve interviewing a series of users to walk through the process of discovering and reading posts associated with geographic areas of user interest on the front end Android app. Acceptance testing would include:
    - Confirmation that the application runs on multiple Android devices of varying screen sizes and operating system versions
    - Confirmation that based on a user's choice of geographic area of interest, the app loads a desirable amount of information, including a screen that is not overloaded with text or pictures, and data that is relevant to the area of interest.
    - Confirmation that if the app loses connection it will gracefully handle the lack of connection and display last-available information or a message that no information is available.

2. Internal acceptance testing includes planning and designing test cases to cover all the functional methods within the API dealing with interacting with a map UI and accessing location services. External acceptance testing would involve users querying the frontend Android app for a series of locations to see what kind of locations of interest are nearby. Acceptance testing would include:
    - The user is able to search for a location on the map and see a list of crowdsourced posts from that location
    - Ability to read the posts from the search area and view images from that location if available
    - If the current location does not have any posts, inform the user that no posts were found and allow them to refine the search
    - If the post cannot be selected (for example, due to network connectivity issues) inform the user that the search cannot be done at the time due to network errors.

3. Conducting acceptance testing on user story #3 involves internal acceptance testing including planning and designing test cases to cover all the functional methods within the API dealing with creating new posts. External acceptance testing would involve users querying the frontend Android app for a series of locations to see what kind of locations of interest are nearby. Acceptance testing would include:
    - The user is able to create a text post and have it geotagged to the location when the content creation began.
    - If the user created content fails to completely upload, the user is notified and the incomplete post is not shown on the content feed.
    - If the user does not have internet access, the user is notified that there is no connection and to try again later. When the user attempts to try again, the previously written content is reloaded to attempt to publish again.
    - If the user does not have access to the location service, the user is notified that the user's location could not be determined and to try to create the content again later.

4. Conducting acceptance testing on user story #4 involves internal acceptance testing including planning and designing test cases to cover all the functional methods within the API. External acceptance testing would involve attempting to create a creator login and then track my previous post. Acceptance testing would include:
    - Allow the user to create an account and log in

- Show error if incorrect username or password is provided
- Revert the user to register again if the registration fails and display an error message
- Allow the user to view there old posts
- Do not allow creators to post without logging in

5. Conducting acceptance testing on user story #5 involves internal acceptance testing including planning and designing test cases to cover all the functional methods within the API. External acceptance testing would involve attempting to create content under a variety of conditions to ensure that the behavior of the auto-save feature works as expected. Acceptance testing would include:
    - If the user starts writing a post, then closes the app, when they next open the app they'll see that their post still exists in a draft state.
    - If the user starts writing a post, then their phone powers off, when they next open the app they'll see their post was saved and can resume writing it.
    - If the user writes a post, then navigates to other apps on their phone and doesn't return for a long while, the post should still be present in a draft state.
    - If the app detects that it cannot auto-save the post, it should alert the user so that they know that their post will not be saved should the application state refresh.

Part 4 Technology Chosen

Front End
Platform: Android
Language: Java
IDE: Android Studio (IntelliJ)
Build Tool: Gradle
Code Standardization: CheckStyle
Testing: Junit
Test Coverage: Emma
Bug Analysis: Spotbugs
Storage: SQLite

Back End
Platform: Linux VM on Google Cloud Platform
Language: Java
IDE: IntelliJ
Build Tool: Gradle
Code Standardization: CheckStyle
Testing: Junit
Test Coverage: Emma
Bug Analysis: Spotbugs
User Data Storage: Cloud Firestore
Image Storage: Google Cloud Storage
High-Level Firestore Schema of Post Table:

- Post ID and User ID
- Google Cloud Storage link for image
- Geotag
- Description
- Timestamp