CPEN 321 Software Engineering

M3: Requirements

UBC Explore

Group members:

Jane Shi	37998283
Xiaoqin Mei	71315980
Dylan Pither	64661267
Akshat Hari	28768299

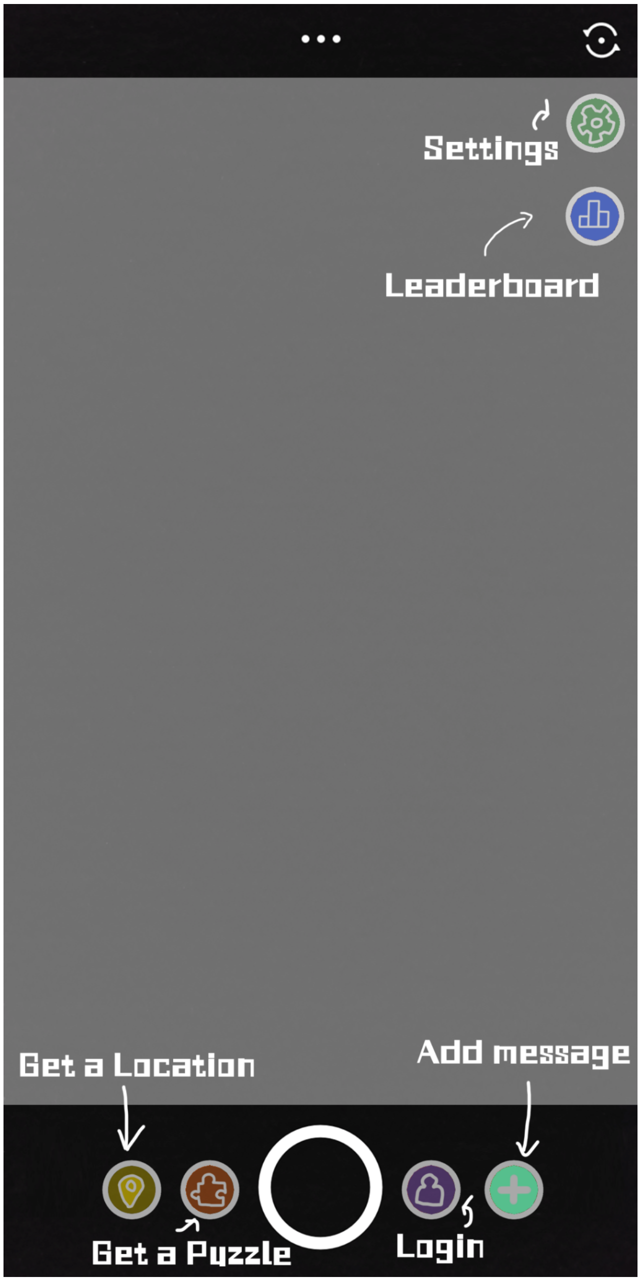
Description:

The UBC campus is large and it takes effort to explore and understand every part of it. Our app proposes to decrease the difficulty through showing the history and fun facts of UBC by the tip of your fingers. If a student or a tourist wants to know more about a part of the campus they are in, they can direct their phone camera to the location and our app would give them a brief summary about the location and direct them to related resources. To encourage people to explore, we will hide away AR caches that can be found around the campus for leaderboards and prizes.

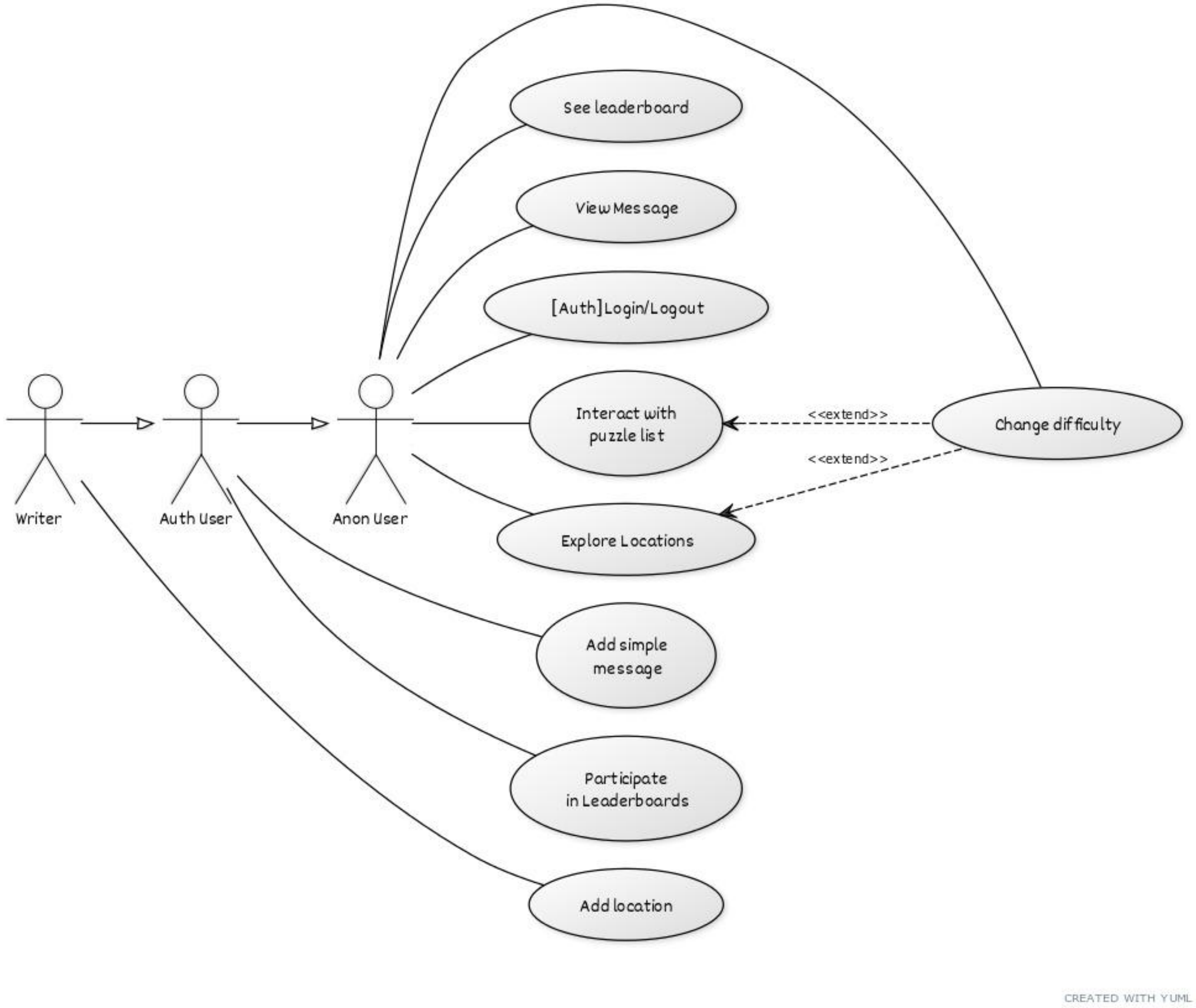
Interesting Features:

- Physical Live2D stickers around campus that people can find that will interact with the app
- Our app will have Artificial Reality/Live 2d creatures/items for the user to collect when they get to the location.

Main Screen Sketch:



Use Case Diagram:



Formal Use Case Specifications:

Title: Explore locations

Description: The user browses explorable locations and selects one they wish to explore. After following the provided directions to the location, information about the location is displayed.

Primary Actor: Anon User

Preconditions: None

Postconditions: The user is given information about their selected location

Main Success Scenario:

1. The user selects the location list button on the main page.
2. The app displays a list of explorable locations.
3. The user selects a location.
4. The app asks the user if they would like to explore the selected location.
5. The user clicks “Yes”.
6. The app checks for user difficulty.
7. Easy:
   - 7.1. The app displays directions to the selected location.
   - 7.2. The user follows the directions to the location.
8. Medium:
   - 8.1. The app displays pictures to hint at the location.
   - 8.2. The app shows the general location.
   - 8.3. The user parses clues and arrives at the location.
9. When the user arrives at the location the app displays information about the location.

Extensions:

5a. User clicks “No”.

- 5a1. The app returns to the list of explorable locations.

7a. User does not follow the directions or cannot find the location.

- 7a1. The user clicks cancel on the directions and returns to the main screen.

Title: Interact with puzzle list

Description: The user chooses a puzzle and solves it to get achievements and unlock secret locations.

Primary Actor: Anon User

Preconditions: The user is at the difficulty level: hard.

Postconditions: The user solves the puzzle and gets achievements, a score and unlocks a secret location.

Main Success Scenario:

1. The user selects the puzzle list on the main page.
2. The app displays the puzzle list.
3. The user selects a puzzle on the list.
4. The user solves the selected puzzle.
5. The user gets achievements, a score and unlocks a secret location.

Extensions:

1a. The user is not on the right difficulty.

- 1a1. The app displays a message “please change your difficulty level to hard.”
- 1a2. The user goes to settings and changes the difficulty.

4a. The user fails to solve the puzzle.

- 4a1. The app displays a failure message.
- 4a2. The user goes back to the main page.

Title: Login/logout

Description: The user uses Google authentication to log into or out of the app.

Primary Actor: Anon user

Preconditions: To login, the user must be logged out of the app; to logout, the user must be logged into the app.

Postconditions: The user is logged into/out of the app.

Main Success Scenario:

1. The user selects the login button from the main page.
2. The app displays a dialog showing Google authentication.
3. The user can add their Google account to login.
4. The app shows the message “logged in successfully” and returns to the main page.

Extensions:

3a. Wrong credentials are entered.

- 3a1. The app displays an error message.

Title: Change difficulty

Description: The user can change the app’s difficulty from easy, medium, and hard.

Primary Actor: Anon user

Preconditions: None.

Postconditions: The user has their difficulty set to the desired level.

Main Success Scenario:

1. The user selects the settings button on the main page.
2. The app displays the settings page.
3. The user selects the “change difficulty” button.
4. The user picks between one of the three difficulty levels.
5. The user clicks “Ok”.
6. The app alerts the user of the new difficulty level that has been selected.

Extensions:

5a. The user clicks submit without clicking one of the three buttons.

- 5a1. The app requests the user to pick a difficulty level.

Title: See leaderboard

Description: The user views a leaderboard that displays the number of achievements authenticated users have collected.

Primary Actor: Anon User

Preconditions: None

Postconditions: The leaderboard is presented.

Main Success Scenario:

1. The user selects the leaderboards button on the main page.
2. The app displays the leaderboard to the user.

Extensions: None

Title: View message

Description: The user can read the message left by another user at a certain location.

Primary Actor: Anon User

Preconditions: The user must be at a location where a user has left a message.

Postconditions: The message at the marked location is displayed to the user.

Main Success Scenario:

1. While walking around, the app will notify the user if any messages are nearby.
2. The user clicks on the notification.
3. The app displays the message to the user.

Extensions:

2a. The user does not click anything.

- 2a1. The app does nothing, the notification will disappear after 5 seconds.

Title: Participate in leaderboards

Description: Authenticated users can share their achievements to compete with other users.

Primary Actor: Auth user

Preconditions: None.

Postconditions: The user’s number of achievements is collected and ranked with other users.

Main Success Scenario:

1. The user presses the leaderboard button on the main screen.
2. The app displays the leaderboards.
3. The user presses the “Participate in Leaderboards” button to join the ranks.
4. The app collects the user’s achievement information and updates the leaderboard.

Extensions: None

Title: Add a simple message

Description: The user is able to choose and add a simple 240-character message that can only be viewed by going to the same location.

Primary Actor: Authenticated User

Preconditions: None

Postconditions: A message is added to the location successfully.

Main Success Scenario:

1. The user clicks the “add message” button.
2. The app shows a form to add a message.
3. The user adds the message.
4. The user presses submit.
5. The app shows a message saying that the submission is successful.

Extensions:

1a. The user is an anonymous user.

- 1a1. The user gets an error message saying that they have to login to use this feature.
- 1a2. The user goes back to the main page.

4a. The user submits an empty message.

- 4a1. The app shows a warning message: “please add something before submitting.”
- 4a2. The user rectifies and submits.

5b. The app fails to submit.

- 5b1. Asks the user to wait a while and retry later.
- 5b2. The user waits and retries the submission.

Title: Add a location

Description: The user adds a location to the list of explorable locations.

Primary Actor: Writer

Preconditions: None

Postconditions: The provided location is added to the location list.

Main Success Scenario:

1. The user selects the add location list button on the location list.
2. The app displays a form to add a location.
3. The user enters a location name, coordinates, a short description and provides an image.
4. The user clicks “Add location”.
5. The app alerts the user that the location has been successfully added.

Extensions:

4a. The user leaves any field empty

- 4a1. The app alerts that all fields must be filled out, displaying which fields are empty.

4b. The user provides invalid coordinates.

- 4b1. The system alerts the user that the provided coordinates are invalid and a range that they should be in.

4c. The user enters illegal characters in the location name or description.

- 4c1. The system alerts the user of the illegal characters that the fields contain.

4d. The user provides an image that is too large.

- 4d1. The system alerts the user that the provided image is too large and how large the image is allowed to be.

4e. The user provides a location that already exists.

- 4e1. The app alerts the user that the location already exists.

5a. The app fails to add the location

- 5a1. The app tells the user to wait to try again.

Non-Functional Requirements:

[Usability] The user should not need more than 5 clicks to perform any action.

- This requirement is relevant because it ensures that the user has fast access to all of the functionality of the app which is important for the user experience.
- We plan to test this requirement by going through each of the use cases and make sure they all need no more than 5 clicks to perform.

[Performance] AR/relevant information should show up within 1 second after the user points their phone in a certain direction at a certain location.

- This requirement is relevant because showing AR/relevant information about the location is a major functionality of our app and we don’t want there to be a significant delay when showing this information.
- We plan to test this requirement by going to each of the locations and measuring the time it takes for the AR/information to show up.

[Energy efficiency] The battery life should not drop by more than 1% after continuous usage of the app for 5 minutes.

- This requirement is relevant since our app is an outdoor-based app requiring our app to be constantly active, while requiring location and camera services at the same time, which both have high energy consumption.
- We plan to test the drainage using Battery Historian for virtual deployment and battery manager apps like AccuBattery for live deployment. We will test the app under heavy use (Constantly display AR image).

[User-friendly Interface] The UI should be simple and easy to use. The icons of the buttons should be able to suggest what they are used for.

- This requirement is relevant since our app’s target users are students and tourists on UBC campus. The app’s interface should be easy for them to understand.
- We plan to find several students who have never used the app before and introduce the general functionality of the app. After that, we will ask them to match the usages to each button.

Member Contributions:

- Akshat: Wrote description, and use case diagram and some parts of formal use case.
- Jane: Edited description, wrote part of formal use case specifications and non-functional requirements.
- Dylan: Wrote and edited some formal use cases.
- Mei: Wrote part of the use cases and non-functional requirements. Made the main screen sketch.

Android Device:

Our group has three Android devices running Android versions 9,10 and 12 which can run the front-end app.