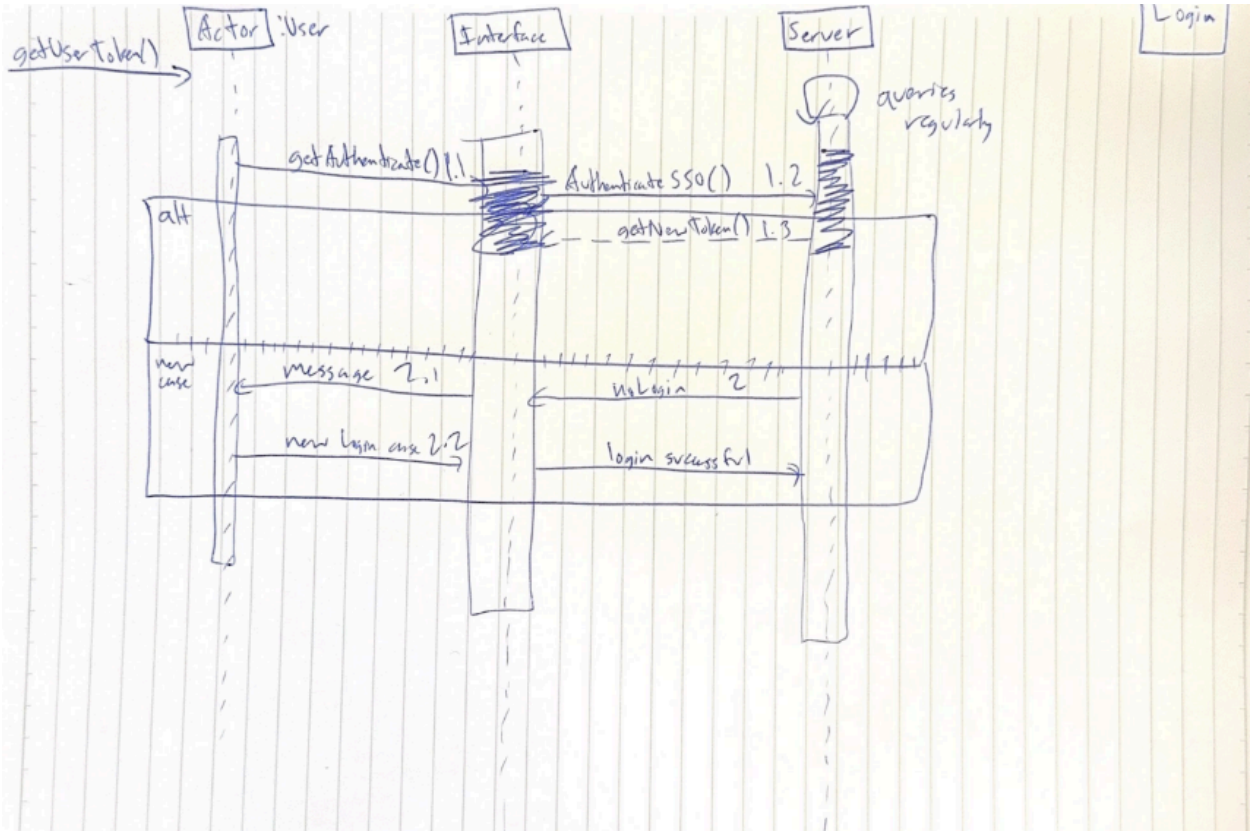
SDD and Contract D3

Kevin Brigham and Aidan McKittrick

SDD #1



Contracts:

App Connection and getNewToken()[1.3] (related to SDD #2):

Preconditions:

- The app is connected to the backend via the internet.
- The app has access to the student's GPS location (with user consent).

Inputs:

- Request for user login verification.

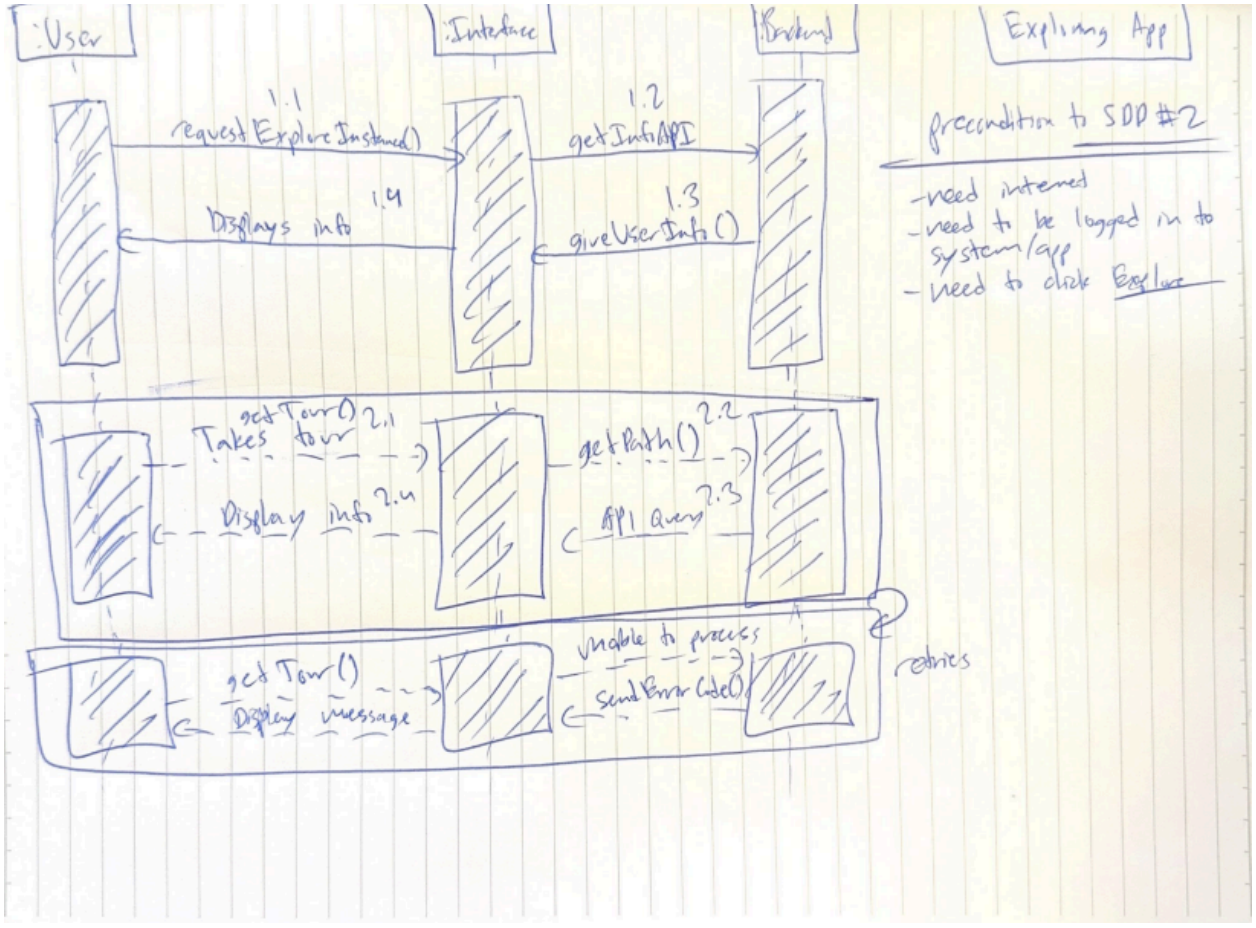
Outputs:

- **API/App Response**: Provides details of user info and a successful login summary.

  Error Handling:

- If the backend fails to respond (due to network issues), the app retries the request and informs the user with an appropriate message.

SDD #2: Exploring the App:



Target Goal:

To allow students to explore the app and request a tour, or take a tour.

Preconditions:

1. The user has installed the AR app on their mobile device.
2. The user has enabled location services (GPS) and camera access.
3. The app is connected to the internet (Wi-Fi or mobile data).
4. The backend system and APIs are available and responsive.
5. User has authenticated (if required by the app) and is within the target location (College of Charleston campus or the city).

Postconditions:

1. The user can see nearby buildings and structures overlaid through the AR interface.
2. The app provides real-time information on buildings and tours, retrieved via API.
3. The user can select and begin a tour, with guided navigation to each stop.
4. All interactions with the system are logged (optional) for tracking user progress or analytics.

Class Names and Methods:

1. `requestExploreInstance()`

- **Description**: This method initializes the AR exploration mode, where the app requests and displays AR overlays for nearby buildings and structures.
- **Inputs**: None (triggered when the user selects the "Explore" feature).
- **Preconditions**:
    - The user has opened the app and selected "Explore."
    - GPS location and camera access are available.
- **Postconditions**:
    - The system fetches a list of nearby buildings based on GPS coordinates.
    - AR content (overlays) is displayed for the nearby buildings on the user's screen.
- **System Actions**:
    - Query the backend API to get a list of nearby buildings.
    - Fetch AR content for each building and overlay it on the camera feed.

---

2. `getTour()`

- **Description**: This method retrieves a list of available tours based on the user's location or preference.
- **Inputs**:
  - `location` (GPS coordinates) or user-selected tour category.
- **Preconditions**:
  - User has opened the app and selected the "Take Tour" option.
  - The app is connected to the backend API.
- **Postconditions**:
  - The app presents the user with a list of available tours.
  - The user can choose a tour to begin navigation.
- **System Actions**:
  - Query the API for available tours using the `location` parameter or user preferences.
  - Display the available tours with brief descriptions, estimated time, and points of interest.

---

3. `getAPIInfo()`

- **Description**: This method interacts with the backend API to fetch building details, tour data, and additional information about points of interest.
- **Inputs**:
  - `buildingID` or `tourID` (depending on the request).
- **Preconditions**:
  - User selects a building or starts a tour.
  - The app is connected to the internet and has access to the backend API.
- **Postconditions**:
  - Detailed information (such as historical facts, department info, or navigation instructions) is retrieved and displayed to the user.
- **System Actions**:
  - Send a query to the backend API with the selected `buildingID` or `tourID`.
  - Display the retrieved information (e.g., building history, floor plans, or tour details).

---

4. `giveUserInfo()`

- **Description**: This method provides feedback to the user, either as AR overlays, text, or audio descriptions, based on their interaction with buildings or points of interest in the app.
- **Inputs**:
    - User actions (e.g., tapping on AR markers or selecting a building/tour).
- **Preconditions**:
    - The user has selected a specific building, AR marker, or tour stop.
- **Postconditions**:
    - The system presents relevant information in AR format (text overlays, images, or audio) to the user.
- **System Actions**:
    - Retrieve relevant information (via `getAPIInfo()`).
    - Display the information in the AR environment as overlays or in another form.

---

5. `getPath()`

- **Description**: This method provides AR navigation to guide the user to the next point of interest (building or tour stop).
- **Inputs**:
    - `startLocation` (current GPS coordinates).
    - `endLocation` (next point of interest's coordinates).
- **Preconditions**:
    - The user has selected a tour or building to navigate to.
    - The app has access to the user's location and the point of interest's coordinates.
- **Postconditions**:
    - The app generates and displays a path from the current location to the target point of interest using AR.
    - The user is guided step-by-step via the AR interface until they reach the target location.
- **System Actions**:
    - Calculate the path between the `startLocation` and `endLocation` using the maps service or backend navigation API.
    - Overlay directional arrows or markers in the AR environment, guiding the user to the destination.

Contract Summary:

- **Target Goal**: To allow students to explore core features of the app.
- **Preconditions**: The app is installed, permissions are granted, and backend APIs are responsive.
- **Postconditions**:
    - The user receives AR overlays for nearby buildings.
    - They can explore building details and take guided tours with AR-based navigation.
    - Information is fetched from the backend and displayed in real time.
- **Key Methods**:
    - `requestExploreInstance()`: Initializes the AR exploration mode.
    - `getTour()`: Retrieves available tours based on location or preference.
    - `getAPIInfo()`: Fetches detailed information from the backend API.
    - `giveUserInfo()`: Presents relevant information to the user in AR format.
    - `getPath()`: Guides the user with AR navigation to points of interest.