DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CSE 489: Mobile Application Development
Lab Exam

Objective:

Develop an Android mobile application that interacts with a REST API to manage and display geographic entities. The app should allow users to create, edit, and view entities on a map centered on Bangladesh. Each entity will have an ID, title, latitude, longitude, and image.

REST API Details:

The provided REST API enables you to interact with the SQLite database and perform CRUD (Create, Read, Update, Delete) operations on entities.

Base URL:

https://labs.anontech.info/cse489/t3/api.php

Endpoints:

1. **Create Entity**:
   - **URL**: `POST /api.php`
   - **Description**: Create a new entity.
   - **Parameters**:
     - `title` (String): Title of the entity.
     - `lat` (Double): Latitude of the entity.
     - `lon` (Double): Longitude of the entity.
     - `image` (File): Image file for the entity.
   - **Response**: JSON object containing the ID of the created entity.
2. **Retrieve Entities**:
   - **URL**: `GET /api.php`
   - **Description**: Retrieve all entities.
   - **Response**: JSON array of entities, where each entity has `id`, `title`, `lat`, `lon`, and `image` fields.

3. **Update Entity**:
   - **URL**: `PUT /api.php`
   - **Description**: Update an existing entity. Use the **x-www-form-encoded** format to submit a PUT request to update the entry.
   - **Parameters**:
     - `id` (Integer): ID of the entity to update.
     - `title` (String): Updated title of the entity.
     - `lat` (Double): Updated latitude of the entity.
     - `lon` (Double): Updated longitude of the entity.
     - `image` (File, optional): Updated image file for the entity. You can skip it, or you need images to use: enctype="**multipart/form-data**"
   - **Response**: JSON object indicating success.

Requirements:

1. **User Interface (Drawer menu with options)**:
   - **Main Screen**: Display a map centered on Bangladesh with markers for each entity.
   - **Entity Form**: Provide a form to create or edit an entity, including fields for the title, latitude, longitude, and image upload.
   - **Entity List**: Display a list of all entities with options to edit or delete each entity.
2. **Functionality**:
   - **Fetch and Display Entities**: Fetch entities from the API and display them as markers on the map. Clicking a marker should display the entity's title and image. Clicking the image will show an enlarged version of the image with the title. (image path should have base url as: https://labs.anontech.info/cse489/t3/ So having image path from api as "image": "images/image2.jpg" should produce a full image path as: " https://labs.anontech.info/cse489/t3/images/image2.jpg"
   - **Create Entity**: Allow users to create a new entity by filling out the form with current GPS information (latitude, longitude) and uploading an image (resize image to 800x600 before submission). Send a POST request to the API.
   - **Edit Entity**: Allow users to edit an existing entity. Send a PUT request to the API. (Respect others' entries, and you should manipulate your own entry only.)
3. **Map Integration**:
   - Use Google Maps or OpenStreetMap for the map view.
   - Ensure the map is centered on Bangladesh (latitude: 23.6850, longitude: 90.3563) and zoomed appropriately.

4. **Error Handling**:
   – Handle API errors gracefully, showing appropriate messages to the user. 5. **Testing**:
   – Test the app thoroughly to ensure all CRUD operations work as expected and the map displays correctly.

Submission:

- **Code Repository**: Submit the complete Android project without the build folder in a form. Include a README file with instructions on how to build and run the app.
- **Documentation**: Provide documentation detailing how the app interacts with the API, challenges you have faced, and how you have overcome them. Put screenshots of your app. (not more than 2-3 pages.)
- **YouTube URL of project:** You need to upload a video demonstrating your app on YouTube and give the URL. You may upload as unlisted or public (according to your preference). Video checklist:
  1) App demo.
  2) Which file is related to which screen?
  Video duration: Max 7 minutes.
- **Submission Form**: https://forms.gle/ogaqgETqhKHXPJfV7
- **Submission Deadline**: **August 13, 2025 (Wednesday) 10.00 PM.** Submission after the deadline will not be considered (Even for a minute). You can submit the form only once, so be careful.

Bonus (Optional):

- Implement offline caching to allow viewing of previously fetched entities when there is no network connection. (Use Room DB or local SQLite)
- Add user authentication to secure the API endpoints and restrict CRUD operations to authenticated users only. (Ask Adnan sir for the server code in the language you prefer. Currently, it is in PHP.

If you are stuck, ask for code Snippets in the group, no problem with it. Others can help with. But do not share the full project. It will not only destroy your own integrity, but also destroy your friend's future. Share wisely.

Additional Resources:

1. Google Maps Android API
2. OpenStreetMap Integration in Android
3. Retrofit, A type-safe HTTP client for Android
4. Save data in a local database using Room
5. CameraX

6. Photo picker
7. Get the last known location

Good luck with your exam!

Screenshots for Sample Application