Lab Assignment #4 – Developing Native Android Apps using Jetpack Compose with Location, Mapping

Due Date: Mid-night (11.59 pm) **20th Nov 2025** **Marks/Weightage: 30/15%**

End Date: Mid-night (11.59 pm) **23d Nov *with 20% late penalty. No Exceptions***

Note: You are required to demonstrate the assignment as per scheduled lab session as announced by your teacher. 25% penalty for not demonstrating the assignment

IDE: Android Studio – Meerkat Feature Drop Version and Kotlin Jetpack Compose

Purpose: The purpose of this lab assignment is to:

- Integrate Google Maps SDK to display interactive maps in their applications.
- Retrieve and manage user location data while handling necessary permissions.
- Update and track real-time location changes
- Customize map appearance and manage user interactions with map events.
- Manage the lifecycle of map components for a seamless user experience.
- Implement advanced location-based features for enhanced app functionality.

.

References: Textbook, ppt slides, class examples, and Android tutorials (**https://developer.android.com/develop/ui/views/theming/look-and-feel**). This material provides the necessary information that you need to complete the exercises.

Be sure to read the following general instructions carefully:
- This assignment must be completed in pairs by all the students. This is based on pair programming.
- You will have to **demonstrate your solution in a scheduled lab session** and upload the solution on Luminate through the **assignment link under Assessments**.

Android Project Naming Rules:

Step 01: You must name your Android Studio **project** according to the following rule:
yourfullname_COMP304*SectionNumber*_Labnumber
For Example: **johnsmith_COMP304Sec001_Lab04. Save location drive name can be C:\COMP304\Assignments or D:\COMP304\Assignments etc.**
If you have more than one exercise in the assignment, then you need to create separate project for each exercise.

Step 02: Submission rules

Once you complete, run and test projects for all the exercises, then submit your projects as one **zip file and it should be** named according to the following rule:
yourfullname_COMP304SectionNumber_Labnumber.zip.
Example: **johnsmith_COMP304Sec001_Lab04.zip (if your section is 001)**

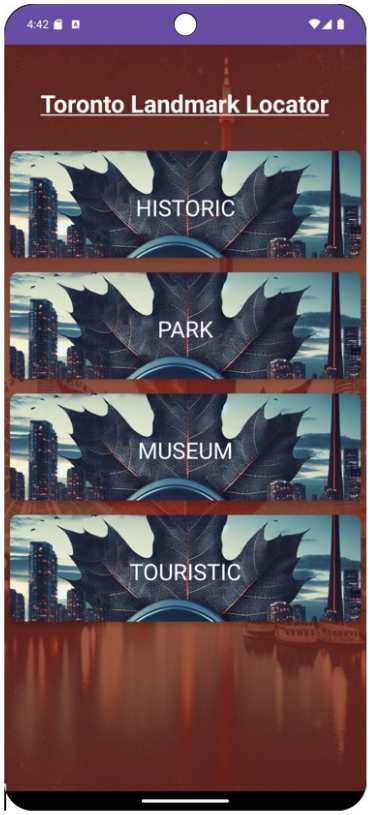
Exercise 01:

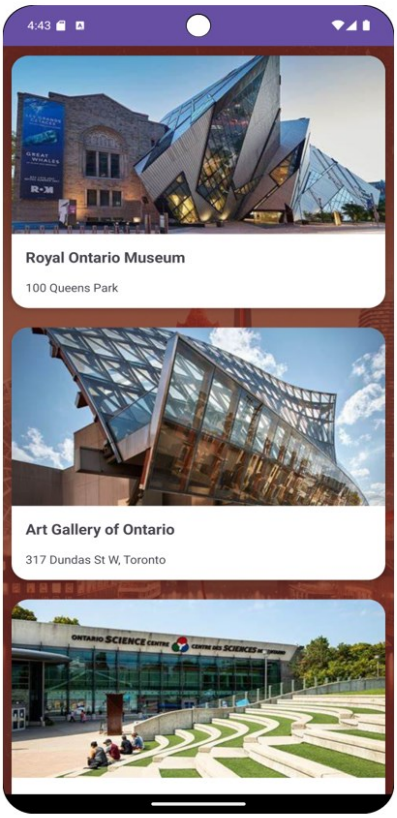
You will develop a new app to help users plan and manage their travel routes with interactive maps and real-time location tracking. The app should include the following functionalities:

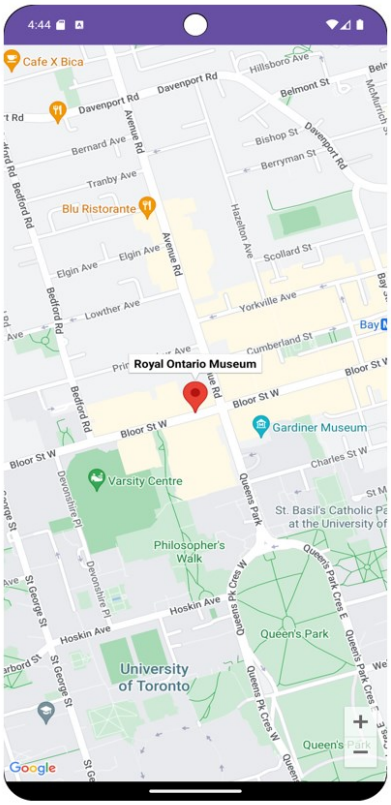
- Displaying interactive maps with the Google Maps SDK.
- Retrieving and managing user location data.
- Tracking real-time location changes and implementing geofencing.
- Customizing map appearance and handling user interactions.
- Implementing advanced location-based features such as route planning and location-based notifications (optional).

Features and Implementations ***(Refer the screen shots below, you can have your own layout, design and info.)***

NOTE: You can choose landmarks of city of choice, or you can choose restaurants of your favorites cuisines such as Italian, Indian, Mexican etc.







In the first activity (Main.kt), it should show list of Categories of Attractions, or Categories of Cuisine, NBA Division, Teams etc. as per your choice. And when you click on any one category type, it should navigate to second activity, in the second screen/activity (it should be named after the first name of your first team member), it should show at least three attractions/items under that category (as shown above), and finally when you click any one of the item/attraction in the second screen/activity, then it should display that in the map. Name 3rd activity after the first name of second team member.

1. **Maps and Location APIs**:
   - Integrate Google Maps SDK to display interactive maps in the application.
   - Retrieve and manage user location data while handling necessary permissions.
   - Update and track real-time location changes, including implementing geofencing.
2. **Customizing Map Appearance and Interactions**:
   - Customize the appearance of maps (e.g., markers, polylines).
   - Manage user interactions with map events (e.g., taps, long presses).
   - Manage the lifecycle of map components for a seamless user experience.
3. **Advanced Location-based Features (optional)**:
   - Implement advanced location-based features such as route planning.
   - Use geofencing to trigger notifications based on user location.
   - Provide location-based recommendations or alerts to enhance app functionality.

Evaluation Table:

Item	Percentage of Total Mark	Details
Functionality:	**80%**	
Correct implementation of Maps and Location APIs:		
Integration of Google Maps SDK and location management	60%	Display interactive maps and manage user location data, handling necessary permissions.
Customization and management of map components:		
Customizing map appearance and handling user interactions	10%	Customize map appearance and manage user interactions with map events.
Lifecycle management of map components	10%	Ensure seamless user experience by managing the lifecycle of map components.
Friendliness:	**15%**	
Alignments of UI controls	10%	UI controls should be properly aligned and organized, providing a visually appealing layout.
Friendly I/O	5%	The app should provide a user-friendly interface with intuitive input/output operations.
Comments, Correct Naming of Variables, Methods, Classes, etc.	**5%**	Code should be well-documented with appropriate comments. Variables, methods, and classes should follow proper naming conventions.
Total	**100%**	