Project Checkpoint 01

**Team name:** Reimei

|  |  |  |
| --- | --- | --- |
| **Team members** | | |
|  | Student Number | Email |
| Shawn Nabizada | 2333349 | 2333349@champlaincollege.qc.ca |
| Nathan Roos | 2332907 | 2332907@champlaincollege.qc.ca |

**Project Description: "Reimei Compass"**

**App Concept:**

Our project, *Reimei Compass*, is an Android application that leverages geofencing and mapping technologies to deliver smart, location-based reminders and actions. The app is designed to help users remember tasks or important events when they arrive at or leave designated locations. By combining multiple alarms with customizable actions and location-based to-do lists, the app caters to busy professionals, students, and anyone who wants to integrate their reminders with their everyday locations.

**Key Features:**

1. **Multiple Alarms**  
   Users can create and manage several location-based alarms simultaneously. This allows them to set distinct reminders for different locations (e.g., “Pick up dry cleaning at the laundromat” and “Grocery shopping at the supermarket”).
2. **Customizable Alarm Actions**  
   Each alarm can have tailored actions such as:
   * **Notification Types:** Options for sound, vibration, or a combination.
   * **Custom Sounds:** Users can choose their own alarm tones.
   * **Custom Messages/Notes:** Users can attach specific messages or task details that display when the alarm is triggered.
3. **Entry and Exit Triggers**  
   Alarms can be set to trigger when a user enters or exits a defined geofence, enabling reminders for both arrival (e.g., “Time to check in with your colleague”) and departure (e.g., “Remember to lock the door when leaving”).
4. **Actionable Notifications**  
   When an alarm is triggered, the notification includes actionable buttons (e.g., “Mark as Done,” “Snooze,” “View Task”) that allow users to interact directly with the reminder without needing to open the app.
5. **Location-Based To-Do Lists**  
   Users can attach a to-do list to a specific location. For instance, arriving at your school could automatically show you a list of school related tasks.
6. **Custom Geofence Shapes**  
   Beyond simple circular geofences, users will be able to draw custom polygonal shapes on the map to define more complex areas (e.g., campus boundaries, park areas), providing greater flexibility and precision for location triggers.
7. **Visited Location History**  
   The app maintains a log of triggered alarms and visited locations, which can be used for personal tracking, reviewing past activities, or refining future reminders.

**Potential Users and Vendors**

* **Potential Users:**
  + **Busy Professionals & Students:** Need reminders for tasks associated with work or study locations.
  + **Individuals with Memory Challenges:** Benefit from contextual reminders linked to familiar locations.
  + **Everyday Consumers:** Anyone interested in integrating location-based reminders with their day-to-day activities.
* **Potential Vendors/Integration Partners:**
  + **Mapping and Geofencing Services:** Google Maps and Google Play Services for location data and geofencing.
  + **Notification Services:** Android’s native notification framework.
  + **Third-Party API Providers:** Optional integration with productivity or task management APIs if we choose to expand functionality later.

**Development Plan**

**Event 1: Project Setup and Initial Integration**

* **Project Environment:**
  + Set up the Android Studio project with the required dependencies (Google Maps, Play Services, etc.).
  + Configure necessary permissions and integrate the Google Maps API.
* **Basic UI:**
  + Implement a simple map-based interface where users can see their current location.

**Event 2: Core Geofencing and Single Alarm Functionality**

* **Geofence Creation:**
  + Allow users to tap the map to create a basic circular geofence.
* **Alarm Triggering:**
  + Implement the mechanism to trigger an alarm (notification) when the user enters the geofenced area.
* **Visited Locations Log:**
  + Record triggered alarms with basic information (location and timestamp) in a simple history log.

**Event 3: Multiple Alarms and Management Interface**

* **Alarm Management:**
  + Expand the app to allow users to create, edit, and delete multiple location-based alarms.
* **Management Screen:**
  + Build a user-friendly list or dashboard view that displays all active alarms.

**Event 4: Customizable Alarm Actions and Dual Triggers**

* **Custom Alarm Settings:**
  + Add functionality for users to customize alarm actions, including:
    - Choosing different notification types (sound, vibration, custom tone).
    - Attaching custom messages or notes to each alarm.
* **Entry and Exit Triggers:**
  + Extend the geofence logic to allow alarms to trigger both when entering and when exiting a geofenced area.

**Event 5: Actionable Notifications**

* **Interactive Alerts:**
  + Enhance notifications by incorporating actionable buttons (e.g., “Snooze”, “Mark as Done”).
  + Ensure that these actions update the app state appropriately without requiring the user to open the app.

**Event 6: Integration of Location-Based To-Do Lists**

* **Task Management Integration:**
  + Enable users to attach to-do lists or task checklists to specific location alarms.
  + Display the associated tasks when the alarm is triggered and allow users to mark tasks as complete.

**Event 7: Custom Geofence Shapes**

* **Advanced Geofencing:**
  + Implement functionality for users to draw custom polygonal geofences on the map.
  + Adjust the geofencing logic to handle non-circular areas, providing more flexibility in defining trigger zones.

**Event 8: Enhanced Visited Location History**

* **Detailed Log:**
  + Expand the visited location history to include detailed data such as custom alarm messages, whether the alarm was triggered on entry or exit, and any user interactions.
* **Filtering and Sorting:**
  + Provide options for users to filter or sort the history by date, location, or alarm type.

**Event 9: UI/UX Refinements and Performance Optimization**

* **Interface Polishing:**
  + Refine UI elements and transitions for a smoother user experience.
* **Optimization:**
  + Optimize background location tracking to minimize battery usage.
  + Perform usability testing and iterate on feedback to ensure the app is intuitive and responsive.

**Event 10: Integration Testing, Documentation, and Final Presentation**

* **Comprehensive Testing:**
  + Integrate all features into a cohesive app and perform thorough testing across different scenarios and devices.
* **Bug Fixes and Enhancements:**
  + Address any issues identified during testing and fine-tune the app’s performance.
* **Documentation:**
  + Prepare both technical documentation (code comments, architecture overview) and user documentation (how-to guides, feature explanations).
* **Final Presentation:**
  + Assemble a final demo that clearly showcases all functionalities, including multiple alarms, customizable actions, entry/exit triggers, actionable notifications, location-based to-do lists, custom geofences, and visited location history.