We need an Android app which will be used to collect ground truth data for indoor positioning system ML model.

Ground truth application will have two tabs at bottom of screen.

1. Home page: On this page, user will simply see below button

Entered Restaurant Building

Entered Elevator

Climbing Stairs 3 floors

Going up 8 floors in Lift

Reached Restaurant Corridor

Reached Restaurant

Left Restaurant

Coming Down 3 floors

Left Restaurant Building

Entered Delivery Building

Reached Delivery Corridor  
Reached Doorstep

Left Doorstep

Going down 8 floors in Lift

Left Delivery Building

Whenever user tap Entered Restaurant Building button, app should call backend API to start a session, and Session should be closed when they tap Left Delivery Building.

**Buttons rule**

1. If Last pressed button is **Entered Restaurant Building,** then only show **Entered Elevator, Climbing Stairs 3 floors**
2. If Last pressed button is **Entered Elevator**, then show **Going up 8 floors in Lift**
3. If Last pressed button is **Climbing Stairs 3 floors** and **Entered Delivery Building** was not pressed for this session, then show **Reached Restaurant Corridor** otherwise show **Reached Delivery Corridor**
4. If Last pressed button is **Going up 8 floors in Lift** and **Entered Delivery Building** was not pressed for this session, then show **Reached Restaurant Corridor** otherwise show **Reached Delivery Corridor**
5. If Last pressed button is **Reached Restaurant Corridor**, then only show **Reached Restaurant**
6. If Last pressed button is **Reached Restaurant**, then only show **Left Restaurant**
7. If Last pressed button is **Left Restaurant**, then only show **Going down 8 floors in Lift**, **Coming Down 3 floors by stairs**
8. If Last pressed button is **Coming Down 3 floors** and **Entered Delivery Building** was not pressed for this session, then show **Left Restaurant Building** otherwise show **Left Delivery Building**
9. If Last pressed button is **Going down 8 floors in Lift** and **Entered Delivery Building** was not pressed for this session, then show **Left Restaurant Building** otherwise show **Left Delivery Building**
10. If Last pressed button is **Left Restaurant Building**, then only show **Entered Delivery Building**
11. If Last pressed button is **Entered Delivery Building**, then only show **Entered Elevator**, **Climbing Stairs 3 floors**
12. if Last pressed button is **Reached Delivery Corridor**, then only show **Reached Doorstep**
13. If Last pressed button is **Reached Doorstep**, then only show **Left Doorstep**
14. If Last pressed button is **Left Doorstep**, then show **Left Delivery Building**
15. If Last pressed button is **Left Delivery Building**, then show **Entered Restaurant Building**

This screen will also allow user to see their Open Session and buttons they pressed along with timestamp. Make it interactive so that it looks like they are completing a task based on the buttons they pressed.

1. Payment Status

This screen will list all the sessions of user with basic details + status (Approved/rejected) + Payment Status (Paid or Unpaid) + Remarks: so, they know why it was Rejected.

1. Bonus Screen: This screen will show the bonus received from each day. Basically, they will get bonus if they submit all the data correct.

API Flow

1. - When user will click on any button then it will check if Session is already created or not, if not it will create a session Id (unique) locally and start session and store session Id, button clicked and timestamp in local storage. Session will start when user click Entered Restaurant Building and it will be closed when user clicks Left Delivery Building. If session is already opened, then it will simply store values in local storage of device.
2. There will be foreground service (always running even when phone is locked), it will check the queue of pending requests in local storage and keep sending to backend one by one. This foreground service will also read IMU data batches stored in local storage and will push those to a separate backend API.

IMU Data:   
 1) A foreground service will be running which will keep capturing GPS + IMU data as per maximum frequency. It will create 5 seconds batch and push that to local storage.

To Save Battery:

- We want to start recording GPS only when user speed is slowed down. We will store capturing GPS and other IMUs if user speed goes beyond the threshold. Which mean we need to continuously pool accelerometer readings.