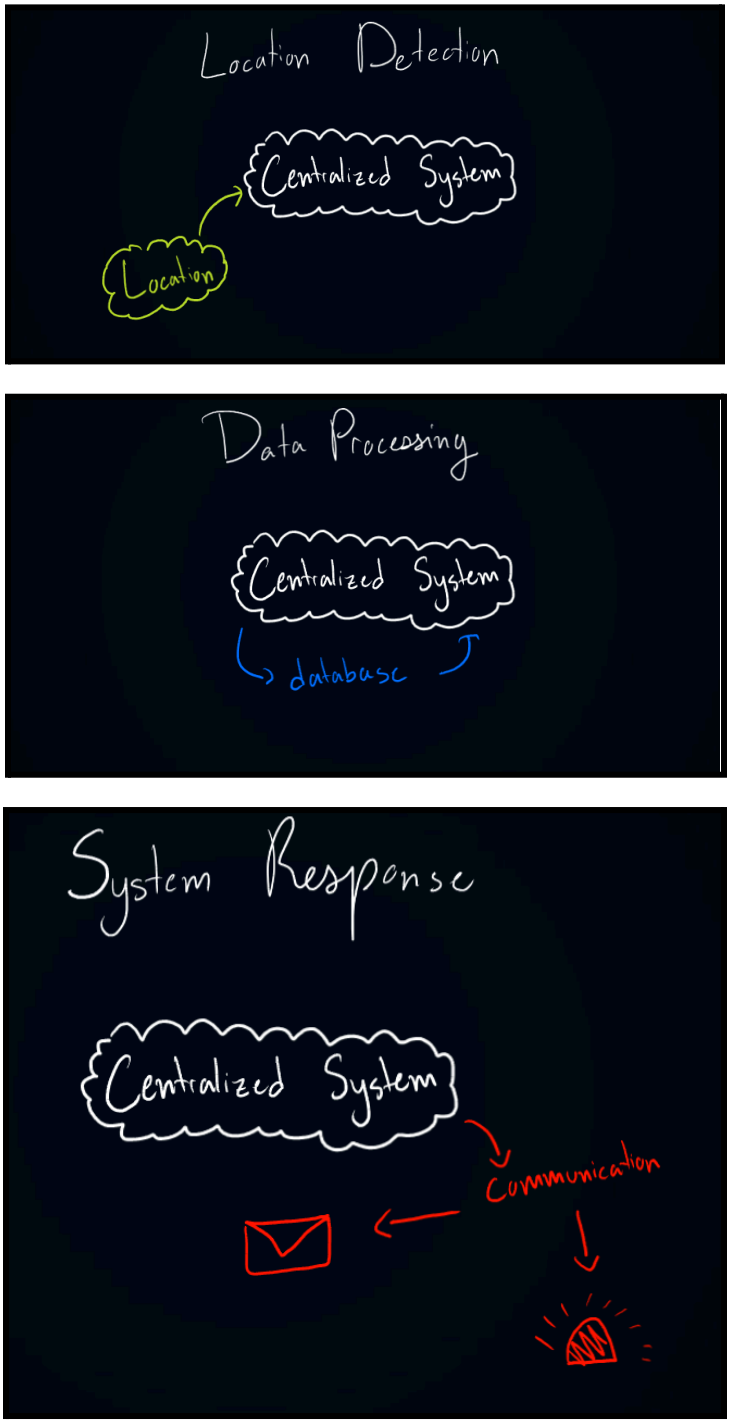
Project Deliverable E: Project Schedule and Cost

User Equipment Tracking System - Project Group 10

Objective:

Develop a system to monitor UE devices on a Wi-Fi network, restricting or reducing their network access if they leave designated zones, using Shabodi's API for network management.

1. **Design Drawing**



Components:

- *Shabodi API:* Manages network restrictions and controls access.

- *Python Backend:* Handles network monitoring, zone definition, and restriction application.

- *Database*: Stores device MAC addresses, allowed zones, and restriction rules.

- *Wi-Fi Network Data Collection:* Uses `scapy` for device detection and zone assignment based on network presence.

- *User Interface:* Optional mobile or web app for managing devices and monitoring status.

2. **Prototyping and Testing Plan**

a. **Task List and Schedule**

Task Number	Task Description	Assigned To	Start Date	End Date	Status
1	Design system architecture and create diagrams	TBD	2024-10-27	2024-10-29	Pending
2	Integrate Shabodi API with backend for device restriction	TBD	2024-10-29	2024-10-31	Pending
3	Set up database to store device address, zones, and restrictions	TBD	2024-10-31	2024-11-02	Pending
4	Implement device detection (using scapy)	TBD	2024-11-04	2024-11-06	Pending
5	Prototype testing for zone	TBD	2024-11-06	2024-11-08	Pending

	accuracy and restriction response times				

b. **Risk Management Plan**

- *Shabodi API latency or restrictions:* Test response time early and adjust system polling frequency.
- *Network congestion affects device detection:* Use priority filtering for critical devices to ensure real-time tracking.
- *Unexpected disconnections from Shabodi API:* Implement automated reconnection attempts and fail-safes.
- *Inaccurate zone definitions:* Perform testing with different signal ranges to calibrate zones accurately.

c. **Budget Estimate (Bill of Materials - BOM)**

Item	Description	Quantity	Unit Cost ($)	Total Cost ($)
Shabodi's APIs	Access to Shabodi's APIs for location, and other network capabilities	1	0	0
Python Development Environment	Open-source software for backend development and API integration	1	0	0
Database Software	Open-source database software (e.g., MySQL, Postgr, eSQL)	1	0	0

Wi-Fi/Bluetooth Testing Tools	Open-source tools (ex. scapy) for device detection and network control scripting (ex. psutil)	1	0	0
Server Hosting	Free tier cloud server	1	0	0
Development Tools	Open-source tools for developing an optional user interface (e.g. React, Flutter, Figma)	1	0	0
Miscellaneous	Utilizing existing resources and tools for testing and prototyping	1	0	0

d. Software and Tools

- *Python:* For backend monitoring and API integration.
- *Shabodi API:* For network management and restriction capabilities.
- *Database:* Stores device details, zone configurations, and network policies.
- *Wi-Fi/Bluetooth Testing Tools:* scapy for device detection, and `psutil` for network control scripting.

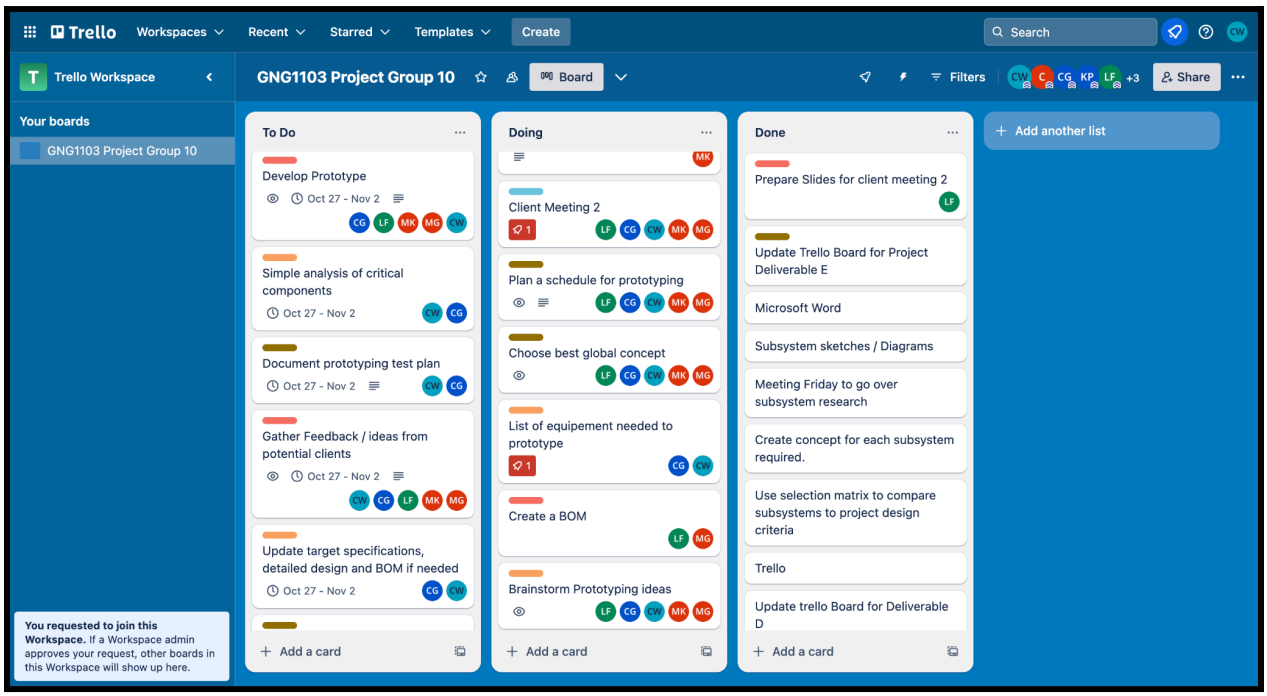
3. Prototyping Test Plan

Design Concept	Reason for Prototype	Describe the Concept to be Prototyped and Tested	Level of Prototype	Kind of Prototype	Metrics	Test Description	Analysis Method	Results	Interpretation	Notes

Prototype 1 - Proof of Concept	Device Detection and Logging	Demonstrate basic device detection and ability to log devices within specified zones using addresses	HiFi Comprehensive	Analytical	Device detection accuracy: 80% or higher over a 30-minute interval	Scan for devices within a specified Wi-Fi range	Monitor device presence within the zone and log data in the database; verify accuracy by comparing manuel logs with system logs	At least 80% detection accuracy in device logging	Pass if device detection is accurate 80% of the time or higher over a 30-minute testing interval	Testing conducted using controlled Wi-Fi network setup to ensure signal consistency
Prototype 2 - Critical Subsystem Testing	Access Restriction and Bandwidth Control	Verify access restriction and bandwidth control capabilities using Shabodi's API	HiFi Comprehensive	Analytical	Response time for access restriction: 2 seconds or less	Set up restricted zones and apply bandwidth limits when a device moves out of bounds	Measure response time by timing delay between device detection out of bounds and restriction activation	Restriction applied within 2 seconds of device detection out of zone	Pass if the system limits access within 2 seconds and maintains restriction consistently	Testing conducted in different zones to verify response stability across varied network conditions
Prototype 3 - Database and Zone Management	Data Management for Zones and Devices	Test storage and retrieval of device data in a database, as well as setting up, updating, and removing zones.	HiFi Comprehensive	Analytical	Database retrieval time <1 second; accurate zone creation and updates	Test storing device data (MAC Address zones) and updating zone settings in database, ensuring data is quickly accessible and up-to-date	Query the database and measure retrieval times; validate zone accuracy by checking stored vs. displayed data	Database retrieval in <1 second and correct zone updates	Pass if retrieval time is under 1 second, and zone management functions (add, update, delete) work as expected	Testing includes adding multiple devices and zones to stimulate actual usage
Prototype 4 - Full System Integration	User Interaction and System Functionality	Test full system integration, including backend, API, database, and optional user interface functiona	HiFi Comprehensive	Physical and analytical	System operates without crashes; UI Displays data within 2 seconds	Test complete system with all integrated components, allowing device management via optional	Interact with the system through the UI (if implemented) and validate end-to-end functionality for different	System stable and UI updates within 2 seconds	Pass if system operates as intended with no crashes, and all components work in sync	Testing include various device entries/exits and user actions via the UI functionality and stability

		lity for monitoring				UI and ensuring real-time updates	user scenarios			

These four prototypes give a full view of the UE tracking system's testing, from initial detection to complete system functionality.

4. **Task Plan Update**

1. Trello Board Update: Added specific subtasks for integrating Shabodi's API and optimizing device detection accuracy.
2. Assigned Responsibilities: Each developer has been assigned distinct parts of the backend and testing phases to streamline the development process.
3. Detailed Subtasks:
    - Backend integration with Shabodi for real-time control.
    - Separate subtask for setting up and testing database interactions.
4. Project Timeline: Adjusted to account for testing and refining detection accuracy across varied network conditions.