Senior Design Design Document

Panther Shuttle App

Team Members:

Joseph Hilte (jhilte2022@my.fit.edu)

Tony Arrington (tarrington2022@my.fit.edu)

Jonathan Suo (jsuo2022@my.fit.edu)

Chase Monigle (cmonigle2022@my.fit.edu)

Faculty Advisor:

Professor Khaled Ali Slhoub (kslhoub@fit.edu)

Client:

Vincent Borrelli (vborrelli2022@my.fit.edu)

Florida Institute of Technology

September 4, 2025

Table of Contents

1. Introduction

1.1 Purpose

This document presents the system design for the Shuttle App, ensuring it satisfies the requirements defined in the Requirements Document. It outlines the system's architecture, modules, data flows, GUI layout, and algorithms necessary to provide real-time shuttle tracking and schedule access for students.

1.2 Scope

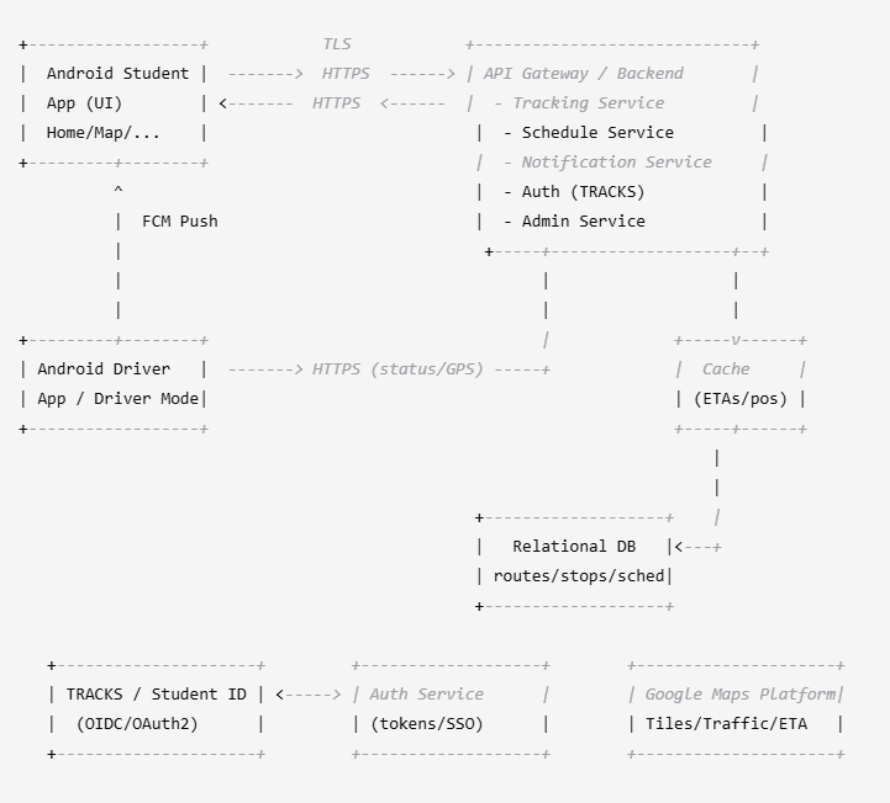
The Shuttle App is an Android-based mobile application with backend cloud services. It provides:
- Real-time shuttle tracking on a campus map.
- Schedule and route details.
- Push notifications for delays and status updates.
- Administrative tools for managers and drivers.

1.3 References

- IEEE Standard for Information Technology – Software Design Descriptions.

2. System Architecture

2.1 High-Level Architecture Diagram



Quick mock-up of what it could look like

2.2 Component Overview

- Mobile App (Android): Provides student-facing UI, receives real-time data, shows schedules, sends/receives notifications.
- Driver Interface: Allows shuttle drivers to update shuttle status and view assigned routes.
- Manager/Admin Portal: Manages schedules, routes, and reporting.
- Backend Services: API gateway, shuttle tracking, notification manager, authentication, and reporting modules.
- Database: Stores shuttle schedules, routes, logs, and user preferences.
- External APIs: Google Maps API for routing, university TRACKS authentication.

3. Detailed Design

3.1 Mobile Application Modules

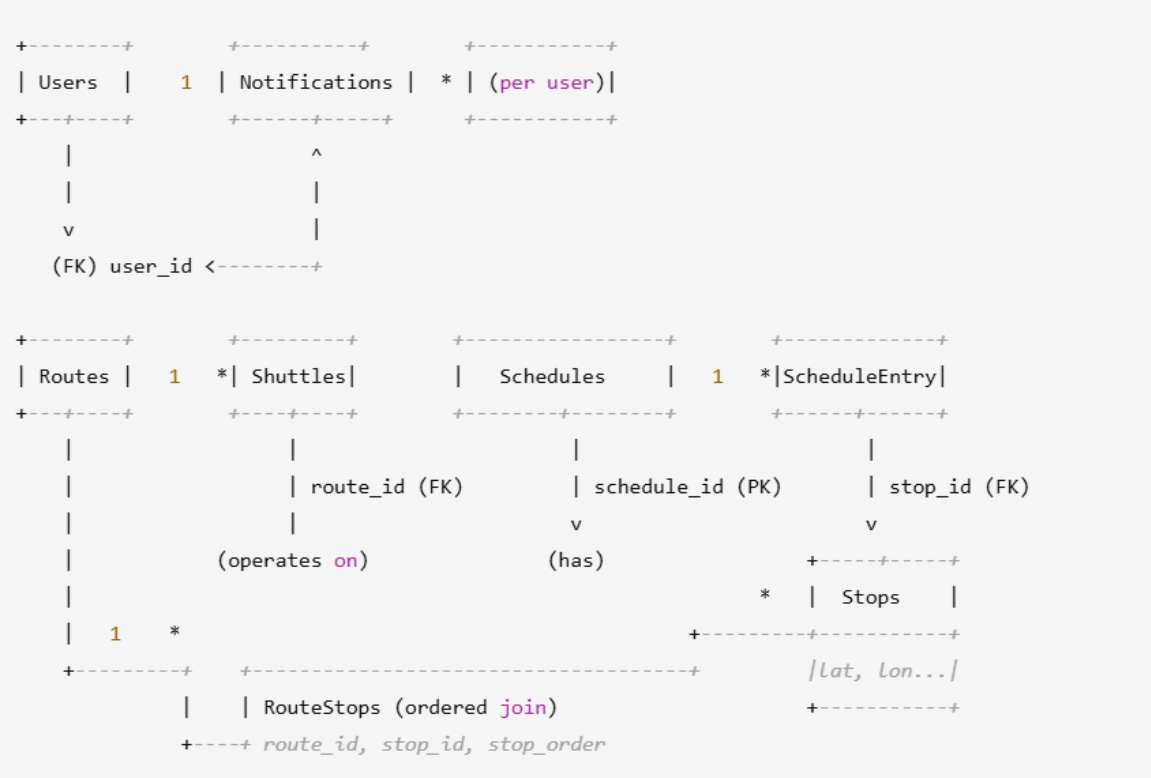
- Live Map Module – Interfaces with GPS feed; shows shuttles and ETAs.
- Schedule Module – Displays static schedules (cached for offline use).
- Notification Module – Subscribes to alerts, delay messages.
- Settings Module – Allows customization (favorite stops, units, dark mode).

3.2 Backend Services

- Tracking Service – Polls shuttle GPS, updates location database.
- Notification Service – Pushes updates to students and drivers.
- Authentication Service – University login (TRACKS/Student ID).
- Admin Service – CRUD for routes, schedules, and reports.

3.3 Database Design

Tables:
- Users(user_id, role, auth_id, preferences)
- Shuttles(shuttle_id, status, capacity, route_id)
- Routes(route_id, name, stops[])
- Schedules(schedule_id, route_id, times[])
- Notifications(notification_id, type, timestamp, user_id)



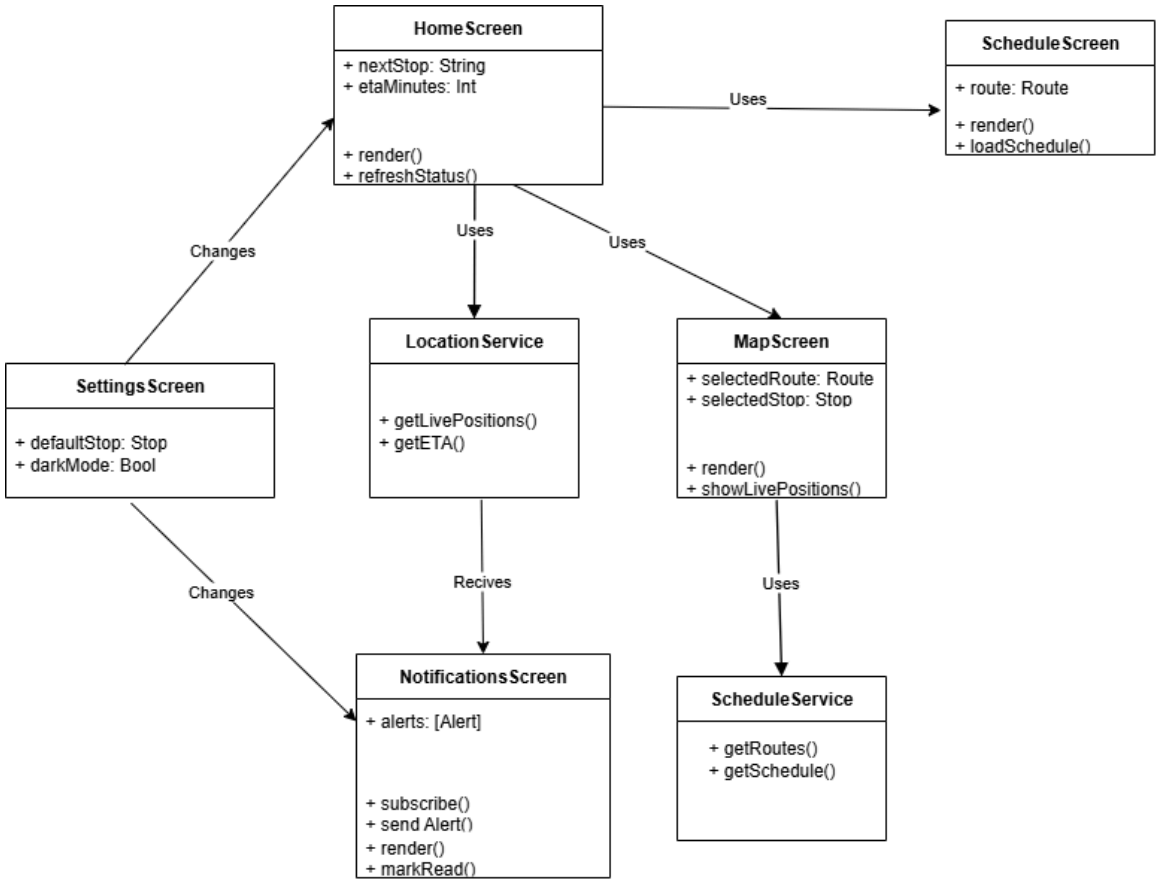
Quick mock-up of what it could look like

3.4 Communication & Security

- All traffic secured with TLS.
- Credentials hashed and salted.
- Mobile app communicates with backend REST APIs.

3.5 UML Class Diagram (Mockup)

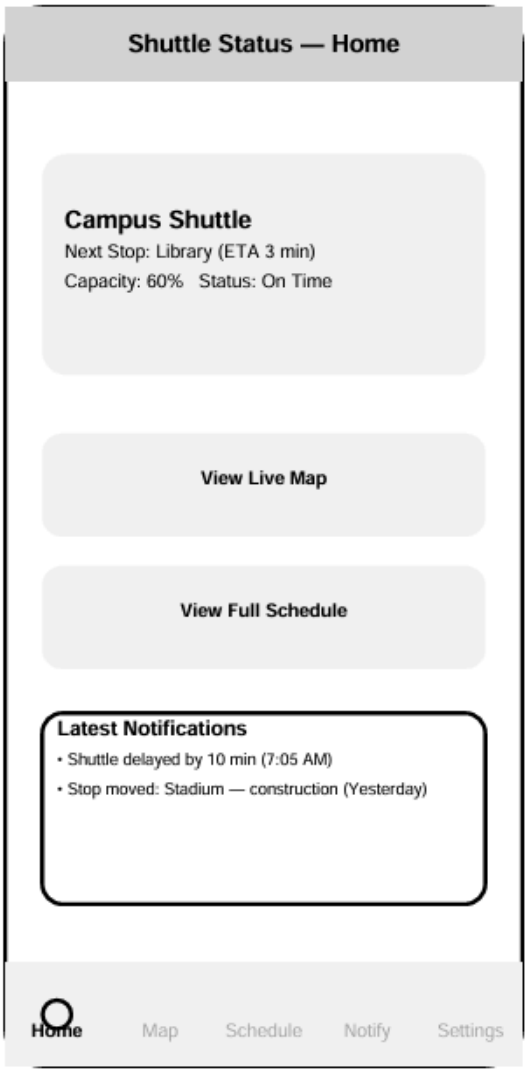
A simplified UML-style class diagram derived from the provided GUI wireframes, https://www.geeksforgeeks.org/system-design/unified-modeling-language-uml-introduction/showing key screens, services, and models and their relationships.

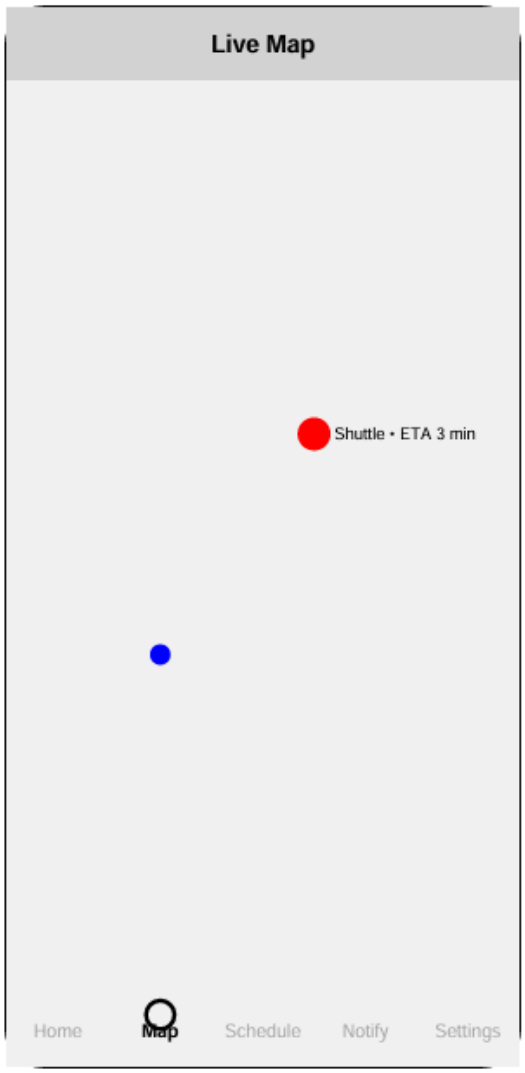


4. Graphical User Interface (GUI)

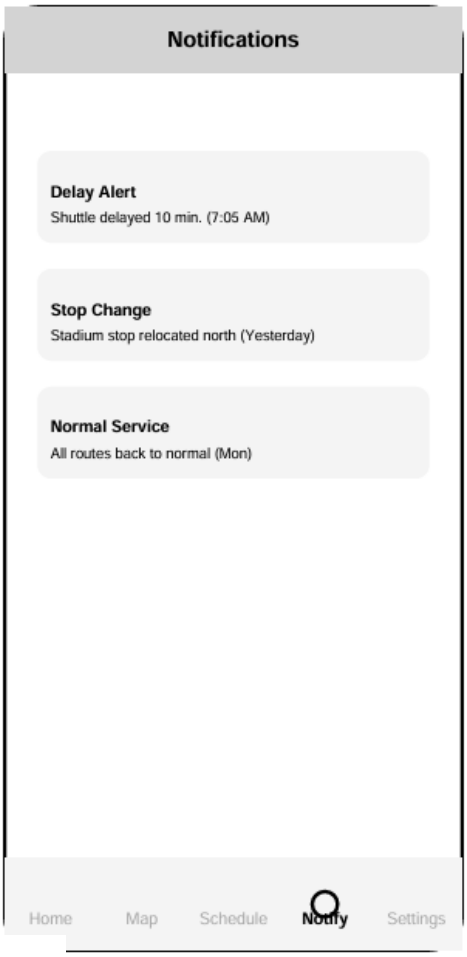
4.1 Mock-ups of Key Screens

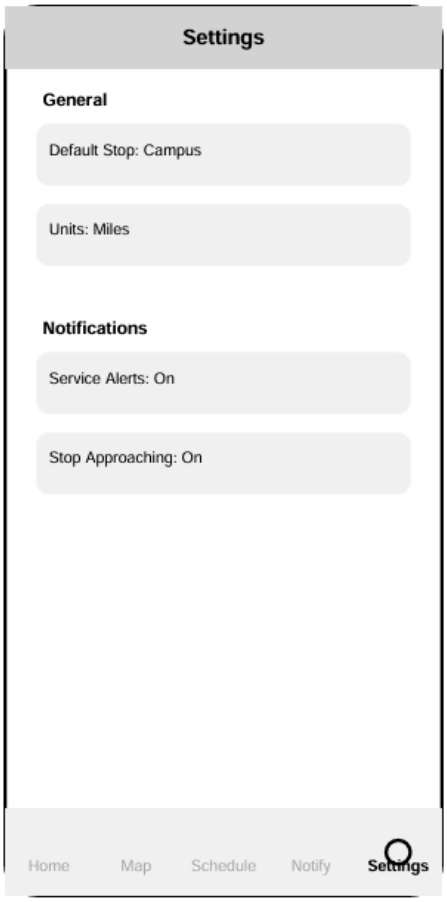
Based on provided wireframes:
- Home/Status Screen: Next shuttle ETA, shuttle capacity, “View Live Map,” “View Full Schedule,” latest notifications.
- Live Map Screen: Campus map with shuttles and stops.
- Schedule Screen: Timetable of shuttles per route.
- Notifications Screen: Alerts for delays, stop changes, cancellations.
- Settings Screen: Default stop, notification toggles, units, dark mode.






Schedule
7:00am Campus
7:15am Panther Bay
7:30am Mary Star
7:45am Campus
8:00am Campus
8:15am Panther Bay
8:30am Mary Star
8:45am Campus
9:00am Campus


Notifications
Delay Alert
Shuttle delayed 10 min. (7:05 AM)
Stop Change
Stadium stop relocated north (Yesterday)
Normal Service
All routes back to normal (Mon)
Home
Map
Schedule
Notify
Settings


Settings
General
Default Stop: Campus
Units: Miles
Notifications
Service Alerts: On
Stop Approaching: On
Home
Map
Schedule
Notify
Settings

4.2 Navigation Flow

Bottom tab navigation: Home → Map → Schedule → Notify → Settings.

5. Algorithms & Pseudocode

ETA Calculation (simplified pseudocode):

```
function calculateETA(currentLocation, stopLocation, avgSpeed):
    distance = getDistance(currentLocation, stopLocation)
    time = distance / avgSpeed
    return round(time, minutes)
```

- Uses GPS + Google Maps traffic API.
- Provides fallback if GPS unavailable (last known speed).

6. Conclusion

The design integrates modular components to meet functional, performance, and usability requirements. The GUI follows student-friendly navigation with clear maps, schedules, and alerts. By adhering to IEEE standards, this design ensures reliability, scalability, and maintainability of the Shuttle App.