Senior Design Test Plan

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 Test Plan describes the strategy and test cases for verifying the Shuttle App, ensuring it satisfies all requirements defined in the Requirements Document. The goal is to validate that the system reliably enables students to track campus shuttles in real time, view schedules, receive notifications, and access related features.

1.2 Scope

Testing will cover functional, performance, and non-functional requirements. The focus is on verifying real-time map updates, notifications, route/schedule displays, and overall usability.

1.3 References

- IEEE Std 829-1998, Software Test Documentation.
- Shuttle App Requirements Document (v1.0).

2. Test Items

- Shuttle App mobile application (Android).
- Backend services (shuttle tracking, authentication, notifications).
- Database of shuttle schedules and routes.

3. Features to be Tested

- Real-time shuttle tracking and refresh rate ([FR-1], [FR-2]).
- Schedule and route display ([FR-3]).
- Estimated arrival times ([FR-4]).
- Push notifications ([FR-5], [FR-6]).
- Driver status updates ([FR-7], [FR-8]).
- Manager tools for schedules and map ([FR-9], [FR-10]).
- Reports generation ([FR-11]).
- Performance under concurrent load ([PR-1]–[PR-4]).
- Security, reliability, and usability ([NFR-1]–[NFR-12]).

4. Features Not to be Tested

- Future ride-sharing feature (not in current release).
- Cross-platform support for iOS (not in current scope).

5. Test Approach

A mix of black-box testing, stress testing, and usability testing will be used. Test data will include both normal input (valid schedules, routes, login credentials) and unusual input (invalid credentials, corrupted GPS signals, network failures) to uncover edge cases.

6. Test Cases

6.1 Functional Requirements

- - [FR-1] Test Case

  * Input: Open live map with GPS feed active

  * Method: Black-box testing using real GPS data and simulated feeds

  * Usual Output: Shuttle icons appear at correct locations in real time

  * Unusual Output: If GPS signal is lost, system displays error or fallback message

- - [FR-2] Test Case

  * Input: Track a moving shuttle across campus

  * Method: Time-based observation with logging

  * Usual Output: Shuttle marker updates every ≤10 seconds

  * Unusual Output: If network delay occurs, system retries gracefully and warns user

- - [FR-3] Test Case

  * Input: Access 'Schedules' tab with valid route data

  * Method: Black-box functional test

  * Usual Output: Correct times and routes are displayed

  * Unusual Output: If no internet, cached schedules are shown with warning

- - [FR-4] Test Case

  * Input: Select a stop on the map

  * Method: Compare app ETA against simulated travel time

  * Usual Output: Arrival times within ±1 minute accuracy

  * Unusual Output: If GPS erratic, system shows 'ETA unavailable'

- - [FR-5] Test Case
  * Input: Driver marks shuttle as 'delayed'
  * Method: Push notification testing with Firebase
  * Usual Output: Students receive notification within 3 seconds
  * Unusual Output: If phone muted, banner still appears; if offline, queued until online

- - [FR-6] Test Case
  * Input: Manager modifies schedule
  * Method: Integration test backend to device
  * Usual Output: Student receives notification about schedule change
  * Unusual Output: If update fails, system retries and logs error

- - [FR-7] Test Case
  * Input: Driver marks shuttle 'Out of Service'
  * Method: Manual input from driver device
  * Usual Output: App reflects change within 10 seconds
  * Unusual Output: If offline, syncs update once online

- - [FR-8] Test Case
  * Input: Assign route in backend to driver account
  * Method: Backend-to-device push test
  * Usual Output: Driver sees assigned route on phone
  * Unusual Output: If push fails, app retries and shows 'awaiting assignment'

- - [FR-9] Test Case
  * Input: Manager edits route in admin interface
  * Method: Database update verification
  * Usual Output: Changes propagate to student app within 1 minute
  * Unusual Output: Invalid entries rejected with error logs

- - [FR-10] Test Case
  * Input: Manager sets shuttle 'unavailable'
  * Method: End-to-end functional test
  * Usual Output: Shuttle icon disappears from live map
  * Unusual Output: If sync fails, app shows last known status with warning

6.2 Performance Requirements

- - [PR-1] Test Case

  * Input: Simulate 1,000 concurrent users

  * Method: Load testing with JMeter

  * Usual Output: System handles 1,000 users smoothly

  * Unusual Output: Spike to 1,500 causes graceful degradation

- - [PR-2] Test Case

  * Input: Request shuttle map or schedules

  * Method: Black-box timing tests

  * Usual Output: Loads in ≤2 seconds

  * Unusual Output: If server overloaded, warning displayed

- - [PR-3] Test Case

  * Input: Feed GPS simulator every 5 seconds

  * Method: Integration timing test

  * Usual Output: Updates visible within 10 seconds

  * Unusual Output: If delayed, user notified of stale data

- - [PR-4] Test Case

  * Input: Run uptime monitor for 30 days

  * Method: Automated monitoring tools

  * Usual Output: ≥99.5% uptime achieved

  * Unusual Output: If downtime exceeds threshold, alerts triggered

6.3 Non-Functional Requirements

- - [NFR-7] Test Case

  * Input: Intercept traffic between app and server

  * Method: Penetration testing and packet sniffing

  * Usual Output: All traffic encrypted via TLS

  * Unusual Output: If misconfigured, traffic visible → fail

- - [NFR-10] Test Case

  * Input: Install app on new device

  * Method: Usability testing with first-time users

  * Usual Output: Live map accessible in ≤2 steps

  * Unusual Output: If >2 steps, redesign required

- - [NFR-12] Test Case

  * Input: Enable dark mode and large text

  * Method: Accessibility testing on Android

  * Usual Output: UI scales correctly and remains usable

  * Unusual Output: If misaligned, report defect

7. Pass/Fail Criteria

- Pass if the test case meets requirement thresholds (time, correctness, or usability).
- Fail if the system does not meet requirements, crashes, or misbehaves with invalid input.

8. Test Deliverables

- Test Plan
- Test Cases Document
- Test Execution Report
- Bug Reports
- Final Test Summary

9. Testing Schedule

- Once a feature is implemented it will go through a testing process as follows:
- Initial Test → Fix bugs or errors → repeat → test with client → add additional changes.

10. Risks and Contingencies

- Risk: Server overload under peak usage.
  * Mitigation: Load balancing, monitoring.

- Risk: GPS inaccuracies.
  * Mitigation: Use average position smoothing.