

Software Test Plan (STP)

1. Introduction -

This Software Test Plan defines the testing strategy for the TourTrack application — a tourism and social networking app developed in collaboration with the Lower Galilee Regional Council.

The app provides real time information, trail conditions, weather updates, and traveler interaction tools to enhance the hiking experience in Israel.

2. Test Items-

- User Authentication Module
- Registration & Profile Preferences
- Trail Recommendation Engine
- Map Integration – your real time location and routes
- Nearby Users Detection Module
- Real Time Weather Interface
- Environmental Hazard Reporting
- Interface for Messaging and Communication Module

3. Features to be Tested-

- User registration via Google, email, or phone
- Personalized trail recommendation based on preferences and weather
- Display of trail maps
- Live updates about weather
- Real-time connection between users (using WhatsApp and phone call)
- Environmental hazard reporting
- Display of relevant local advertisements along the route

4. Features Not to be Tested-

- Integration with third-party weather APIs
- Integration with third-party maps APIs
- External authentication systems (Google login) – we assume the authentication provider handles validation securely.
- Route and location data correctness provided by local authorities – data integrity is assumed and will not be validated by the team.

5. Testing Strategy

- Unit Testing: Performed on individual backend components (e.g., API endpoints, recommendation logic)
- Integration Testing: Testing between frontend, backend, and MongoDB
- System Testing: End-to-end testing of user scenarios, including registration, route selection, and alerts
- Acceptance Testing: Final stage testing with real users and council representatives.

6. Test Environment

- Frontend: Android application (Java/XML)
- Backend: Node.js (Express) server on localhost
- Database: MongoDB Atlas
- APIs: OpenWeatherMap API, Google Maps API, WhatsApp API
- Devices: Android smartphones with GPS and active internet connection
- Tools: Android Studio, MongoDB, VSCode

7. Responsibilities-

Frontend: We created the mockup together, and then divided the coding work between us – each of us was responsible for several screens in the app.

Backend: We divided the development into tasks, and each of us was responsible for different areas.

For example, Ori was responsible for the map – including real-time location, detecting nearby users, and saving the data to the database.

Shani was responsible for the authentication system and the development of the communication feature between users.

We shared the rest of the development equally so that each of us could learn and gain experience in all parts of the project.

8. Schedule

Unit Testing: Weeks 8–9

Integration Testing: Week 10

System Testing: Weeks 11–12

Acceptance Testing: Week 13

9. Risks and Contingencies

Risk: Delays in map data from local authorities

Mitigation: Use sample static data for development and testing

Risk: Performance bottlenecks during high usage

Mitigation: Simulate high user load and optimize queries and data access