Software Implementation and Testing Document

For

Group 10

Version 1.0

Authors:
Kirra Orndorff
Alexander Jubran
Lauren Chang
Riley Galpin
Michael Greenberg

1. Programming Languages (5 points)

C#: This is our primary language for application logic. We chose C# because it is the native language for the .NET MAUI framework, allowing us to share a single codebase across multiple platforms while maintaining high performance.

XAML: Used in conjunction with C# within the .NET MAUI framework to define the user interface. It allows for a clean separation between the UI design and the underlying business logic (MVVM pattern).

2. Platforms, APIs, Databases, and other technologies used (5 points)

Platform: .NET MAUI: We are using this framework to build a cross-platform application that can run as a web application while retaining the ability to deploy to mobile devices in the future. We will use .NET SDK version 8.0.

Weather API (National Weather Service/OpenWeatherMap): This API is used to retrieve real-time environmental data including temperature, wind speed, and humidity to power our clothing recommendation engine.

Google Maps API: Integrated to allow users to view, create, and save custom running routes with specific terrain data.

OpenRouteService: for routing API

OpenStreetMapAPI: contributors for map data

MapLibreAPI: for the mapping library

Turf.jsAPI: for geospatial calculations

Google CalendarAPI: for weekly planner and past workout log.

Local Storage (.NET MAUI Built-in): We are utilizing the framework's internal storage capabilities to ensure user privacy and provide offline access to saved routes and shoe mileage.

UraniumUI.Material & Community Toolkit.Mvvm: These libraries are used to enhance the UI experience and implement the Model-View-ViewModel architecture for better code maintainability.

3. Execution-based Functional Testing (10 points)

*Describe how/if you performed functional testing for your project (i.e., tested for the **functional requirements** listed in your RD).*

4. Execution-based Non-Functional Testing (10 points)

*Describe how/if you performed non-functional testing for your project (i.e., tested for the **non-functional requirements** listed in your RD).*

5. Non-Execution-based Testing (10 points)

Describe how/if you performed non-execution-based testing (such as code reviews/inspections/walkthroughs).