SRS

SRS Revision History
2/17 - Document Created

Concept of Operations / ConOps
Software Description
TrailGrade is a Python-based system that analyzes hiking trail data to generate multidimensional difficulty ratings using various components like cardio intensity, trail terrain, etc. Users can be provided recommendations based on previously completed trails.

Current System / Situation
There are currently a few versions of our system that are available to users. However, many users have expressed complaints over current systems with current information being unhelpful or unavailable. The goal of our system is to create a better system for users that provides them with accurate and helpful data.

Justification for a new System
Other versions of trail difficulty and information apps have a lack of depth and poor difficulty ratings that fail to accurately depict trail conditions for users, which has led to poor reviews. Our version will attempt to rectify current issues by implementing a better difficulty system and providing better information about each trail.

Operational Features of the Proposed System
A UI for users to filter and view trail difficulty, elevation change, distance, terrain, and location for trails nearby.

User Classes + Modes of Operation
Hikers/Anyone: The system is designed for anyone interested in hiking/going on a hike that wants to view nearby options for difficulty and information. There would be little variation regardless of who the user is.

Use Cases

1. A user wants to start hiking to get in shape. They know what trailhead they want to start at, but they need to make sure they pick a trail that won't be too tiring for them. They choose a trailhead / start location from the system and are presented with a list of options of trails sorted by difficulty, with easiest first. If they have other specifications as well, they can define them using a filter system.
2. An avid hiker is on vacation and is looking for a difficult trail near them. They give the program a radius (5 km, 10 miles, etc.) and general difficulty preference, and are given a list of nearby trails, their locations, and their difficulty.

Requirements

SDS

SDS Revision History
2/17 - Document Created

System Overview
The system ingests GPS trail data, processes it through various analytical modules, stores the results in a structured database, and presents the information through interactive visualizations. The system is designed to be modular, for incremental development purposes.

Software Architecture
Components
1. Trail Data Ingestion Engine
    - Handles GPS data input
    - Validates and standardizes incoming formats
    - Performs initial data cleaning
    - Converts to internal data structure
    - Provides clean data to other components
2. Trail Database Manager
    - Manages database operations
    - Stores trail data and analysis results
    - Handles data persistence and retrieval
    - Provides data access interface
3. Geographic Analysis Core
    - Processes trail segments
    - Calculates elevation changes and metrics
    - Analyzes terrain characteristics
    - Generates difficulty ratings
4. API Service
    - Provides endpoints for data access
    - Handles request validation
    - Manages API response formatting
    - Coordinates between components
5. Visualization Generator
    - Creates interactive trail maps using Matplotlib
    - Generates elevation profiles and difficulty overlays
    - Produces visual analysis reports

Component Interactions
- The visualization generator interacts with other components through the API service

- The API service coordinates all data flow between components
- The geographic analysis core processes data from the ingestion engine and stores results through the database manager

Key Flows:
1. The ingestion engine processes raw gps data and passes it to the database
2. The analysis core pulls trail data for processing and returns difficulty ratings
3. The database manager maintains all persistent data and provides it to other components
4. The API service coordinates requests between components
5. The visualization generator creates visual representations based on analysis results

Design Rationale

Software Modules
Trail Data Ingestion Engine
Serves as the system's data intake processor, accepts various formats of GPS trail data and converts them into a standardized internal format
Geographic Analysis Core
Processes trail segments and generates difficulty ratings based on factors including elevation change, terrain, exposure, etc.
Trail Database Manager
Handles all storage operations, maintains trail data, analysis results, and user interaction history in a structured database
API Service
Provides an interface for external applications to access trail data, analyses, and recommendations
Visualization Generator
Creates interactive visualizations of trail data, difficulty analyses, and recommendations using Matplotlib

Project Timeline
Milestones (February 19 - March 12):
1. Research Data
2. Database Setup
3. Frontend/UI Design
4. API
5. Algorithm
6. Data Analysis/Geographical system

Resources
https://www.usgs.gov/national-digital-trails/seven-ways-access-or-view-usgs-trails-dataset