SP11 – Happy Trails App

SOFTWARE REQUIREMENTS SPECIFICATION (SRS)

CS 4850 - Section 01 - Spring 2024

2/13/2024 - Sharon Perry

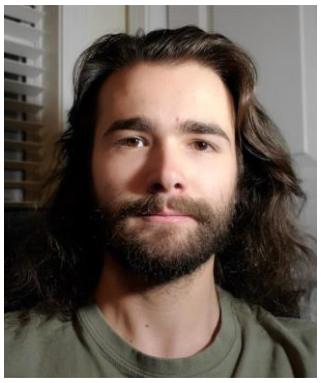		
Jalen	**Justin**	**Ian**

Table of Contents

1.0 Introduction

The System Requirements Document establishes the functional and non-functional requirements for this project. This project is a mobile app designed to enhance the hiking experience for both hiking enthusiasts and newer hikers. This document outlines the requirements and specifications for the app, detailing the features, functionalities, and constraints.

1.1 Overview

Many people face challenges in discovering engaging hiking trails in their area. Despite abundant local trails, most hikers tend to frequent a limited selection due to the lack of easily accessible information about alternative trails. This limitation diminishes the potential for exploration and hinders the overall hiking experience for experienced and new hikers.

To address this issue, our team will develop a user-friendly mobile application with comprehensive information on local hiking trails. This application aims to inform users of the local hiking trails to enhance their outdoor experience.

1.2 Project Goals

1. Provide a centralized platform for accessing detailed information about hiking trails including trail length, difficulty, proximity, and accessibility.
2. Prioritize user experience by implementing intuitive interface design and functionality that facilitates easy navigation and efficient trail selection.
3. Integrate national and state-level databases to ensure the accuracy and reliability of trail information presented in the app.
4. Implement a review system that allows users to share their trail experiences and engage with fellow hiking enthusiasts.

1.3 Definitions and Acronyms

UI	User Interface	An app's overall design to be intuitive and visually appealing
UX	User Experience	The overall experience of a user when interacting with a product, including ease of use, efficiency and satisfaction
API	Application Programming Interface	A set of rules and protocols that allows different software applications to communicate with each other

API Key	Application Programming Interface Key	An alphanumeric code issued by an API provider that allows access to their services and data
NPS	National Park Service	An agency of the U.S. federal government responsible for managing national parks and national monuments.
SDK	Software Development Kit	A set of tools, libraries and documentation that developers use to create software applications
OSM	Open Street Map	A free-to-use map that allows users to retrieve geospatial data.
OAuth2	Open Authorization	An open-standard authorization framework that allows users to grant third-party applications limited access to a software program
OIDC	OpenID Connect	Authentication protocol built on the OAuth2 framework
	Flutter	An open-source UI SDK created by Google.

1.4 Assumptions

Throughout the requirements, assumptions were made on user preferences, technical capabilities, and resource availability. This may impact the implementation of certain features.

2.0 Design Constraints

The design of the mobile app is influenced by several constraints that must be considered early in the development process. This section outlines these constraints to better guide the development of the application.

2.1 Environment

The app will be developed using the Flutter SDK; this means that whatever sort of restrictions Flutter has regarding supported programming languages will be applicable to the development of the app. Flutter allows developers to launch apps for both Android and iOS so the app will support both OS platforms.

The website associated with the app will not have many constraints and will be accessible on most modern browsers; Google Chrome, Mozilla Firefox, Microsoft Edge, etc. will be supported.

Performance regarding the app will not be a concern; CPU and memory requirements will be relatively low and most modern devices will easily be able to browse the database.

Regardless of the platform – desktop or mobile – the website will provide an optimized user experience.

The app will need network capabilities to function; if any device using the application doesn't have access to the Internet, the utility of the app will be greatly reduced. The app will require network communication with a database to properly populate the trail list.

The app's database provides the location of the trails; the user will have the ability to use Google Maps to navigate to the trail. This contributes to a clear constraint; the app must have the ability to communicate with the Google Maps app.

2.2 User Characteristics

The mobile app will meet the needs of many different types of users. The design will be intuitive for those familiar will mobile apps and for those who may struggle to navigate to make the experience as seamless and easy as possible.

2.3 System

The app may not run at fully optimized speeds if the user's software is outdated. Most devices allow users to access content with an older OS that is still being supported, however the older OS may not come with all the bug fixes that have been added in the more recent updates. Without these updates, the device itself potentially could run into problems whilst using the mobile app. Some features that have been added in more recent updates may not be available to those using older software.

3.0 Functional Requirements

R.1.0 Launch Page

R 1.1 Login button redirects to login page.

R 1.2 Register button redirect to create account page.

R 1.3 Browse button that directs to the website's trail database; this allows a user to browse trails without having an account.

R 2.0 Account Login Page

R 2.1 Field for username (email associated with the account).

R 2.2 Field for password.

R 2.3 "Forgot Password" button will allow user to reset password.

R 2.4 "Forgot Username" button will allow user to retrieve a forgotten username.

R 3.0 Account Creation Page

R 3.1 Field to enter email; this will be the username for each account.

R 3.2 Field to enter password. Another field will be used to confirm the password.

R 3.3 Field to enter first name.

R 3.4 Field to enter last name.

R 3.5 Field to enter phone number.

R 3.6 Button that confirms account creation; all fields must be filled to submit.

R 3.7 Button that redirects back to the login page.

R 4.0 Account Settings Page

R 4.1 Field and button to change the password of the account.

R 4.2 Field and button to change the phone number of the account.

R 4.3 Button that redirects to the website's trail database.

R 5.0 Trail List Page

R 5.1 Area to display the list of trails; each trail will display the name and an image associated with it.

R 5.2 Ability to select a trail and visit a page detailing information about the trail. The image and name associated with each trail will allow the user to do so.

R 5.3 Drop down menu that allows the user to filter the search.

R 5.4 Search button that applies the selected filters.

R 5.5 Field that allows the user to search for specific trails by name.

R 6.0 Trail Information Page

R 6.1 Area to display an image associated with the selected trail.

R 6.2 Area to display the trail name.

R 6.3 Button that redirects back to the list of trails.

R 6.4 Area that displays the user score of the trail; this will be based on its reviews.

R 6.5 Area to display user reviews.

R 6.6 Field to leave a review.

R 6.7 Button that submits the review; can only be submitted if the field is entered and a rating for the trail is selected.

R 6.8 Field that displays the location of the trail.

4.0 Non-Functional Requirements

The Happy Trails mobile app must meet a set of non-functional requirements to ensure the overall quality, performance, and UX of the system. This section outlines the guide for the design and development for the non-functional requirements of the app.

4.1 Security

The mobile app must implement OIDC for secure user authentication supported by the SDK, Flutter. All communications between the app and external services must be encrypted using industry standard protocols such as HTTPS.

4.2 Capacity

The app must be designed to handle many users, making low response times essential for app functions. Amazon's DynamoDB will be used to store information about accounts and reviews.

4.3 Usability

The UI must be intuitive and UX should be positive. UI must be easily accessible to use for those who struggle with understanding general app structure.

4.4 Other

Performance of the application shall be timely with minimal latency to user interactions. Time taken to load and display trail information must be optimized to minimize wait times and provide a positive UX.

5.0 External Interface Requirements

5.1 User Interface Requirements

Compatible w/ Microsoft Edge, Google Chrome, OperaGX. Should be easy to use; a UI that fights back is a worthless UI.

5.2 Hardware Interface Requirements

iPhone 6 or higher | Any device supported by Android OS 10 or higher (Google pixel, Samsung Galaxy series, LG, Motorola, etc.) The website will be accessible by any desktop using a modern browser; most desktop OSes will be supported: Windows, MacOS, Linux, etc.

5.3 Software Interface Requirements

IOS 9 or higher | Android OS 10 or higher. The app will use the NPS Data API to retrieve data from the NPS Database and the Overpass API to retrieve data from the OSM Derivative database; this data will be used to present trail information to the user. Amazon DynamoDB will be used to store accounts and review information for the app.