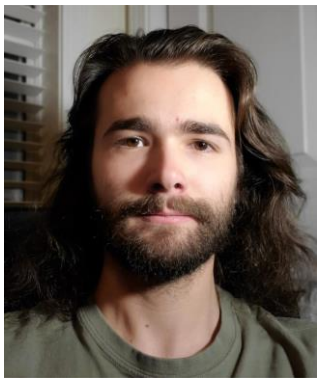
# SP11 – Happy Trails App

## SOFTWARE DESIGN DOCUMENT (SDD)

## CS 4850 - Section 01 - Spring 2024

## 2/13/2024 - Sharon Perry

| | | |
|---|---|---|
|  |  |  |
| **Jalen** | **Justin** | **Ian** |

# Table of Contents

# 1.0 Introduction

The Happy Trails mobile app is a comprehensive solution designed to enhance the outdoor experience for hikers of all experience ranges. This app aims to provide a user-friendly platform that facilitates trail discovery, navigation, and community engagement. This SDD serves as the blueprint for the development, outlining the architectural design, system components, and technical specifications necessary.

## 1.1 Document Outline

1. Introduction: Provides an overview of the Happy Trails application, its purpose, scope, audience, and organization.

2. Design Considerations: Discusses the key design considerations, such as user experience, scalability, and security.

3. Architectural Strategies: Outlines the overarching architectural strategies and principles guiding the design and implementation of Happy Trails.

4. System Architecture: Describes the high-level system architecture, including main components, their interactions, and overall system flow.

5. Policies and Practices: Details the policies, practices, and methodologies used in the development and deployment of Happy Trails.

6. Detailed System Design: Provides a comprehensive breakdown of the system design, including component-level design, data flow diagrams, and interfaces specifications.

## 1.2 Document Description

This SDD for the Happy Trails application serves as a comprehensive guide for developing the application. This document provides a clear and structured overview to ensure that the project's scope, objectives, and technical requirements are met by the development team.

# 2.0 Design Considerations

Design considerations play a critical role in deciding the architecture and functionality of the Happy Trails app. This section outlines a range of factors that influence design decisions.

## 2.1 Assumptions and Dependencies

We are leveraging the cross-platform development framework, Flutter, to ensure the application works on both iOS and Android. Dependencies on external services like the National Park Service's Trail API and OpenID, are assumed for the functionality of the application.

### 2.2 General Constraints

The Happy Trails application must deliver responsive and seamless user experiences even when under challenging network conditions. We will also adhere to industry standards for data encryption and authentication.

### 2.3 Goals and Guidelines

Our primary goal is to create an intuitive and enjoyable user experience that enhances the outdoor experience for hikers of all kinds. The app will also feature community feedback, providing the ability to rate trails and share

### 2.4 Development Methods

For this project we are using an Agile style of development by following a Gantt chart where we plan, develop, test, and deploy versions of the application. We will constantly test as we develop the app, ensuring a continuous integration of versions as we finalize the development.

## 3.0 Architectural Structure

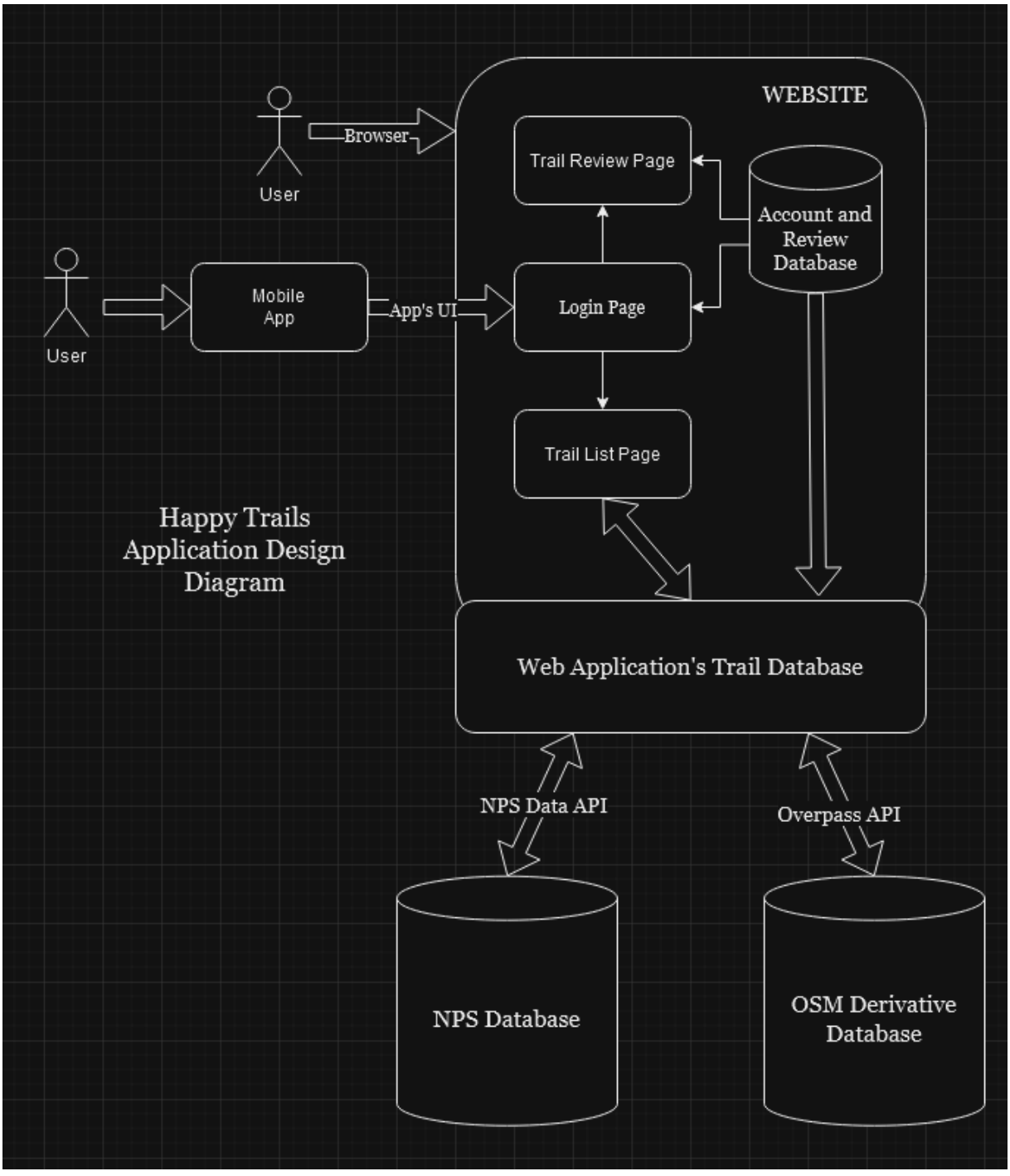
The Happy Trails mobile application will utilize service-oriented architecture that will easily sort newly created accounts. Service-oriented architecture creates a user layer that allows them to interact with their accounts (create new account or change password).

A website will be used in tandem with a database to store user account information. Trail information will be provided by external databases communicated with via APIs.

## 4.0 System Architecture

The system architecture of the Happy Trails app follows a client-server model, with the client-side application running on the user's devices. The client-side application is responsible for presenting the user interface, handling user interactions, and communicating with the server-side components such as the NPS Data API and the Overpass API. The server-side application is responsible for managing the core logic, data storage, and service integration.

The following is a design diagram that outlines the general structure of the application.




Users will be able to access the website via the mobile app or any browser. Through the app's UI, the user will be able to access the login page of the website. This login page will then retrieve data from the website's account database to verify the user's login information. From here, the user may navigate to the trail list page or the trail review page.

The trail review page will retrieve data from the website's database to display the user reviews. The website's database will be created via Amazon's DynamoDB.

The trail list page will retrieve data from a collection of data from multiple outside databases. Both outside databases used – the NPS Database and the OSM Derivative Database – provide much more data than will be required by the app and website. Therefore, all desired data will be filtered through the website; only desired data will be accessed by the trail list page. The NPS Data API will be used to communicate with the NPS Database. The Overpass API will be used to communicate with the OSM Derivative Database.

### 4.1 Subsystem Architecture

The Happy Trails app will have a database for storing user account information allowing for easy access on runtime of the application.

## 5.0 Policies and Tactics

Our app will impose a censor setting to prevent users or bots from posting obscene or inappropriate content in the reviews section of the application. The app must remain user-friendly to all ages to keep user interactions safe and respectful.

## 6.0 Detailed System Design

This section outlines the critical aspects of each design aspect to facilitate seamless development and integration of the Happy Trails application.

### 6.1. Classification

Due to the nature of our project being a mobile and web application, we rely on two main APIs and an external database management tool for the Happy Trails app to function.

### 6.2. Definition

The Happy Trails app requires an outlining of the features and functionalities it provides for users as well as its interactions within subsystems and external services

### 6.3. Responsibilities

The Happy Trails mobile app provides the user with a database capable of acquiring information on the user-selected hiking trail. The interactions between user reviews on the app will be kept family friendly to keep the environment tame and respectful.

Databases will be secured from outside sources. Account information will only be accessible to developers and those who manage the application.

## 6.4. Composition

The app will be composed via the Flutter SDK. The source code for the app will be stored online via GitHub; this will also be the version control method.

## 6.5. Uses/Interactions

Happy Trails will be used to acquire information on a specific hiking trail in the state of Georgia selected by the user. The user can create an account to continuously log in to access the database from the website. Interactions between the databases and the website will occur frequently and will be mediated by various APIs.

## 6.6. Resources

The Flutter SDK will be used as a resource to develop the app itself. The app will rely on two external resources for its trail data: the NPS Database and the OSM Derivative Database.

Amazon DynamoDB will be used to store accounts and review information for the app. This database will not interact with the external ones and will function independently of them.