

Slopes Design Document

Annalies Turley, Sean MacDonald, Jan Evangelista, David Ma, Shavon Zhang, Ben Miller

Introduction

Slopes is intended to be a an easy to use, streamlined way for skiers and snowboarders to check mountain conditions for their favourite ski hills on iOS and Android. The app will provide the user with the most up to date snow base information, current weather, and snowfall forecasts for their desired mountains.

This document will detail how we will implement Slopes.

Design

Software Architecture and Rationale

The Slopes app will use the Ionic Framework in order to produce an app for both iOS and Android mobile systems. The Ionic Framework will work in conjunction with Cordova, with Ionic providing the front end user interface of the application and Cordova acting as a native wrapper to provide access to the mobile operating systems' individual native SDK's.

The Ionic Framework will reduce the development time needed to produce a fully functional app for both mobile platforms. Using Ionic and Cordova as the underlying architecture will allow the Slopes app to use only one codebase for both platforms, allowing the focus of the development process to shift towards creating a more refined and functional application. The unification of the codebase will also allow for a similar look and equal functionality for the application across both mobile platforms.

The Ionic Framework will use the following technologies in order to produce a fully functional Slopes app:

- HTML5 and CSS for the user interface of the application
- Javascript for UI scripting
- AngularJS for application logic and access to Cordova API's and native SDK's
- MySQL database to store the database of mountains
- Apache Cordova as the native wrapper for each mobile OS

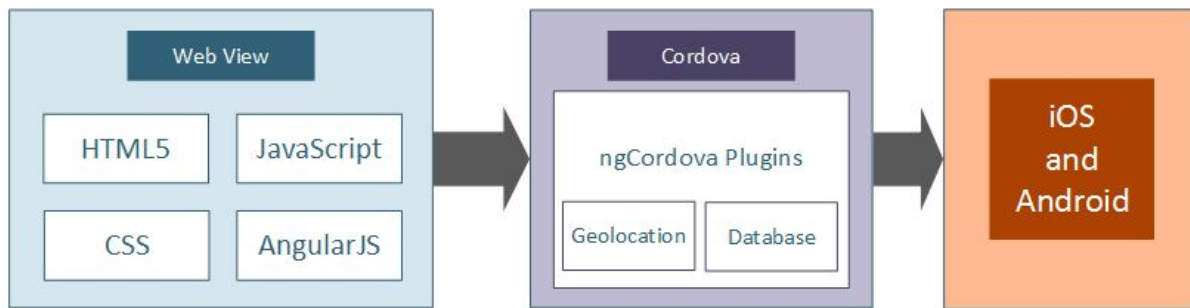


Figure 1: Technologies used by the Slopes App

HTML5, CSS and Javascript

Slope's user interface will be written in HTML5, CSS, and Javascript. Rather than developing locally, our group has decided to deploy our app to iOS and Android using Ionic. Ionic uses web languages instead of native languages to allow for deployment to multiple platforms to create apps with identical interfaces regardless of operating system on the user's device. HTML is used to design the basic layouts, CSS is used to assign styles to the HTML, and JavaScript is used for interactive, dynamic, and more complex design elements.

AngularJS

AngularJS is an open-source front-end web application framework that is based on Javascript. It is a single page application framework which is considered the best practice for Apache Cordova applications as it provides great performance. It will be used for navigating to the different pages within our Slopes application as well as the view for each of these pages.

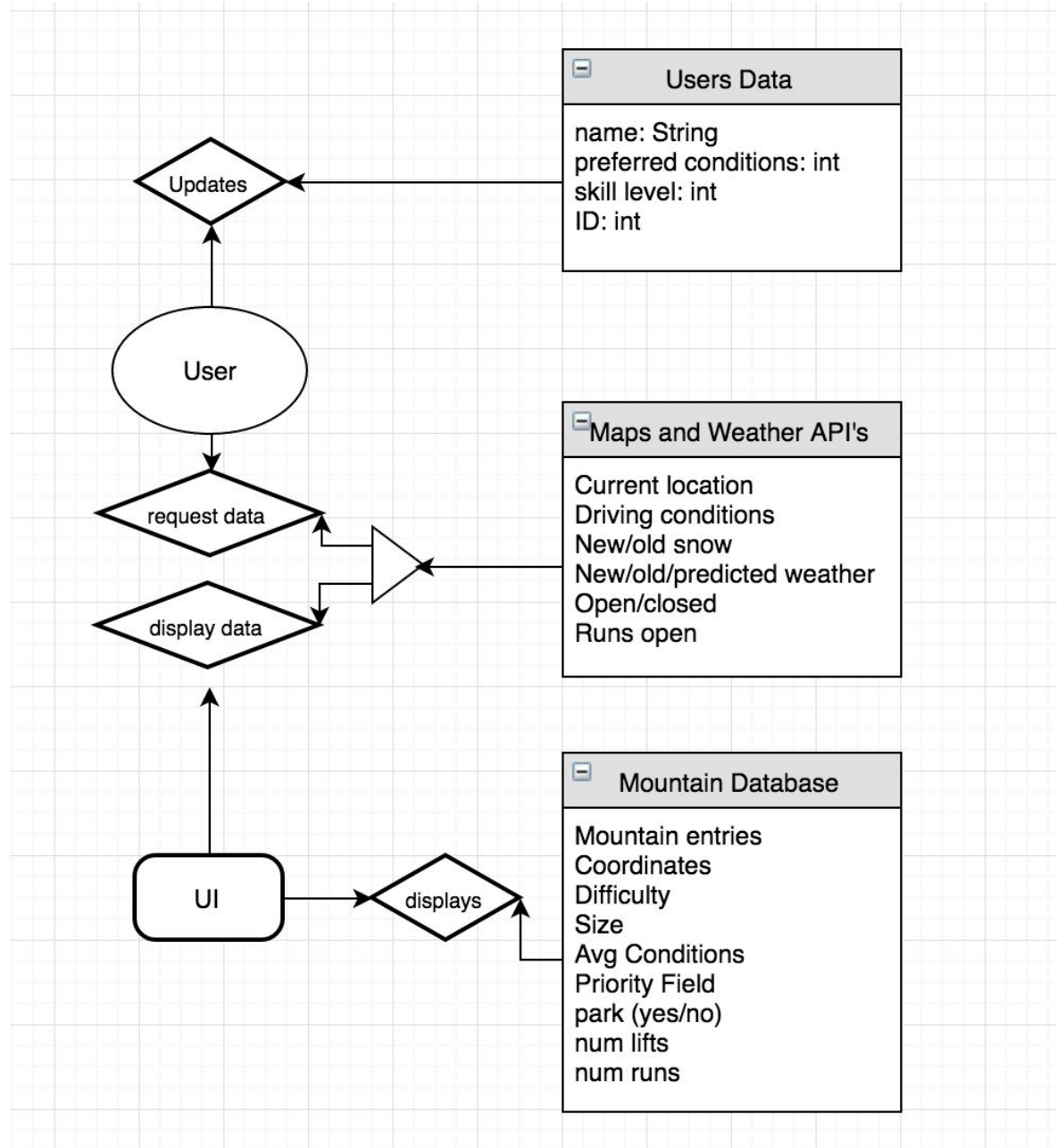
MySQL Database

The Ionic Framework supports, and has excellent documentation for MySQL databases. We will use PHP to connect with the database. MySQL is a database technology that stores information passed through PHP script to a table style database. It gives the user the power to store information in a customizable table with columns of their choice.

Apache Cordova

Apache Cordova is a mobile application development framework for hybrid applications (meaning that they are neither truly native nor Web-based applications). It allows one to access native APIs through Javascript since it is not purely Web-based. Many application platforms can be targeted with one code base in Javascript, HTML and CSS. It is very appealing to web developers as it enables access to many useful libraries such as JQuery. However, Cordova does not supply one with what a lot of native SDKs do. For instance, UI components are not provided by Cordova. This is where Ionic fits in for developing applications for the different platforms. Cordova runs in a WebView. The WebView is slightly different for IOS and Android, therefore Ionic is used to mitigate the differences.

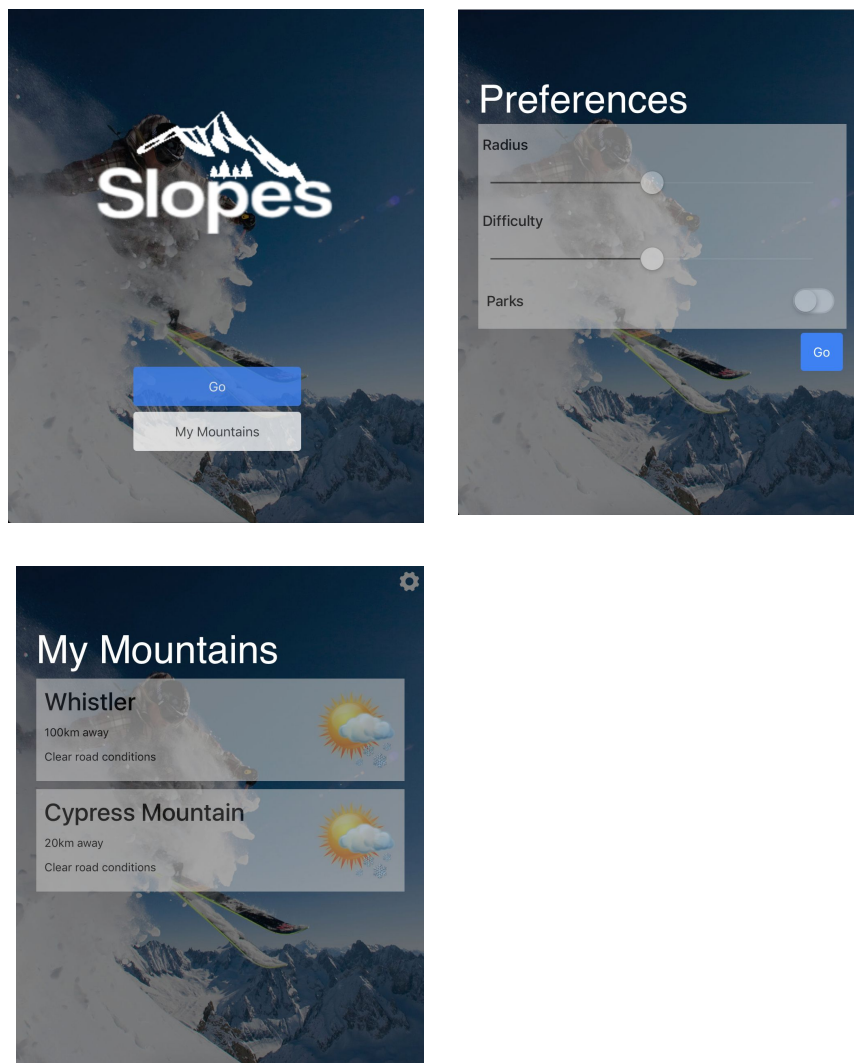
Data



GUI

Our GUI will also be developed through Ionic Framework. This allows us to design the front-end portion of our app through the use of HTML and CSS. We decided that this would be a better choice than making two separate teams for each native platform (Android and iOS) because this limits the differences that will be apparent between the two apps.

Furthermore, Ionic has an extensive CSS library already implemented, allowing use to use their buttons, and icons in our app.



Our users will prefer our app over checking the websites of each individual ski mountain, because of our app's quick display of information as well it's visual appeal. Therefore we aim to maximize the ease of navigation through our app through creating a simplistic user interface.

To make for easier navigation, we have limited our app to four different pages. When the app is launched, the user will be greeted by our home page. There will be a page

allowing the user to input their preference as to difficulty of slopes, or driving distance. There will also be a results pages, listing nearby mountains. From this page, users can click on individual mountains leading them to a page containing more details on that particular mountain.

Validation

In order to maintain a valid design that fully meets our users needs, we will incorporate our potential user-base in our UI validation. Getting input on what users prefer to see throughout the design process will allow us to make our app easy to use, and maximize usability. Therefore, we will validate our initial UI design with potential users, as well as ensuring that all changes made in the development process are also run by users.

Additionally, we have ensured that our preliminary design choices have been made to best fulfill our requirements. Since our main requirement is to produce a product that is simple and convenient for mountain-goers to use, we have opted to develop the app using the Ionic unified framework. Doing this will allow us to develop for both platforms simultaneously, which will help meet our requirement of having an app available to all mountain-goers, regardless of their phone platform. Additionally, developing using the unified framework will help keep both platforms consistent in meeting our requirements, and make our development process more efficient, so that we have more time for testing. This will enable us to produce a consistent, robust app that will be simple, accurate, and accessible.

Changes to Design Specification

Currently we are using myWeather2 to access the weather conditions of our mountains. However, myWeather2 only provides information about the general conditions and does not contain the actual amount of snowfall. Accordingly, we have decided to scrape the amount of snowfall from Weather Underground. In order to do this we plan on using ThingSpeak.