

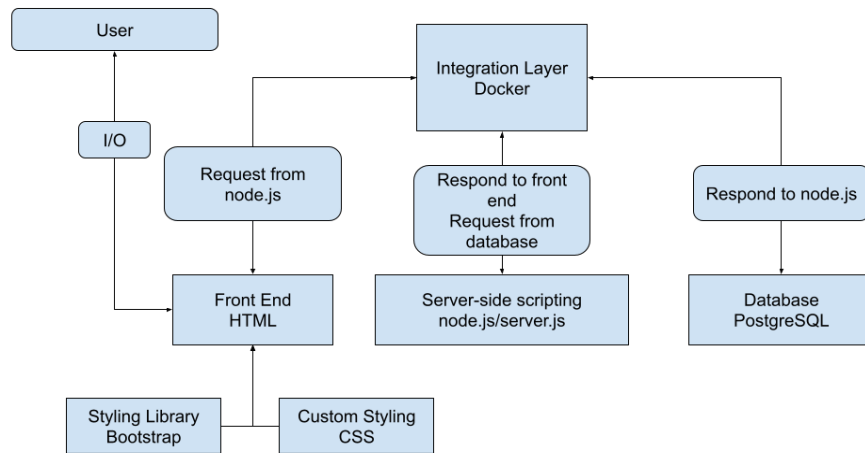
Project Milestone 4

Revised List of Features:

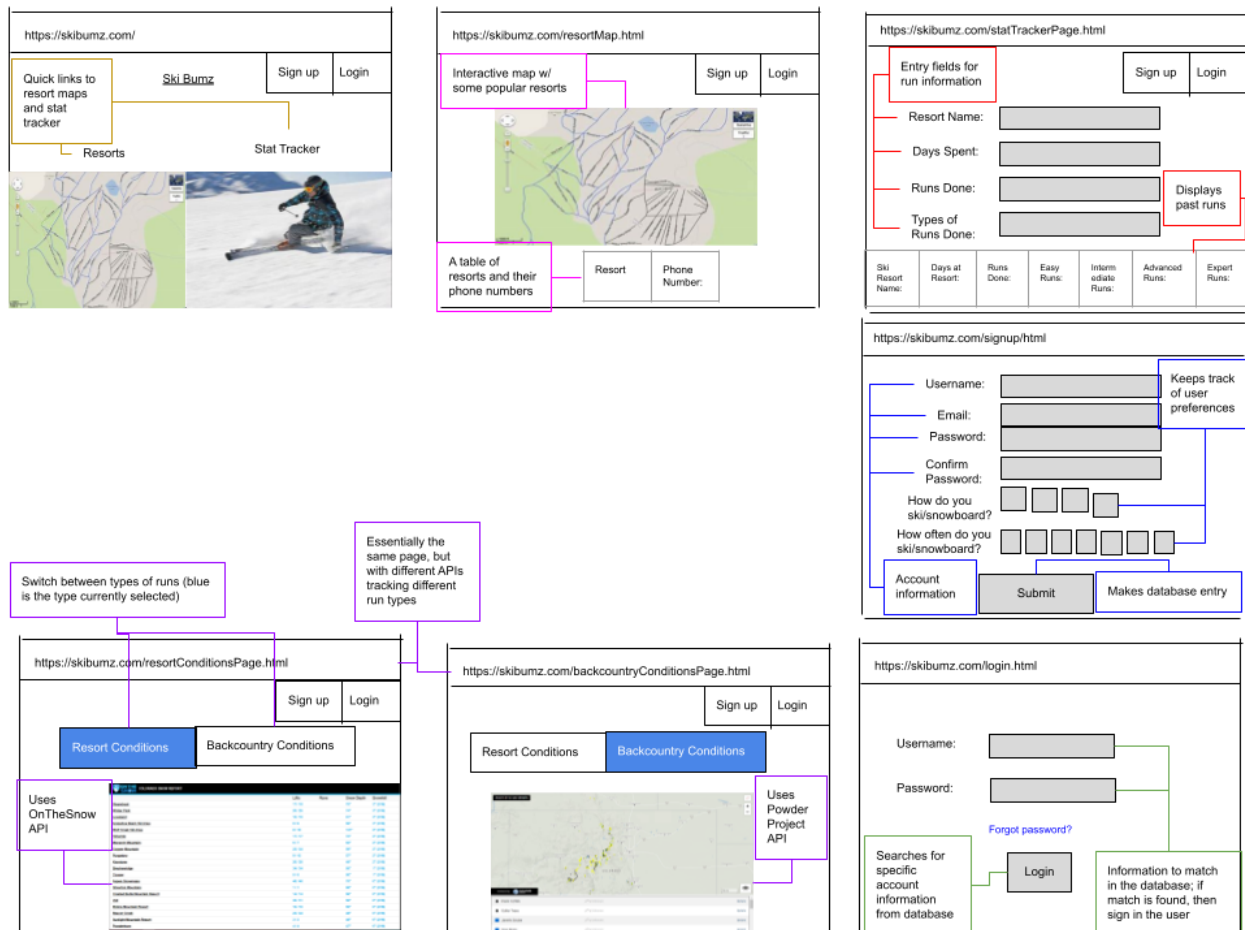
Priority (1)-(5) with 1 being the highest priority

- Resort Conditions
 - The resort conditions page should show a short description of the weather and the current temperature, temperature it feels like, and wind speed. **(1)**
 - This page should also display the high and low temperatures for the next 7 days. **(2)**
 - Ideally the page will show the 24, 48, and 72 hour snowfall but, we are having a bit of trouble finding a free API for this. **(4)**
- Backcountry Conditions
 - The backcountry page should show information on some local backcountry lines with a map. It will also have a map with the current avalanche risk levels in colorado. **(2)**
 - This is not as high of a priority but ideally the user should be able to enter a longitude and latitude and the page will display the current weather conditions at that location. **(3)**
- Map of resorts
 - The information page should display a map of colorado resorts and allow the user to check how far away different resorts are from them. **(2)**
- Information of the resorts
 - This page should have information on the different Colorado ski resorts. The page should allow the user to easily find the resort's website and phone number. **(2)**
 - The page should show more specific information about the resorts such as the prices of lift tickets, reviews, and/or number of runs. **(4)**
- Stat Tracker
 - This feature will allow the user to track how many days they have skied, what resorts they skied at, and the difficulty of the runs they did. **(1)**
- User Management
 - The website should be able to keep track of the people who have registered to the site, as well as when they're logged in and what they have access to. **(1)**
 - If a user is logged in they should be able to see their saved stats. **(2)**
 - If user should not be able to submit stats unless they are logged in. **(3)**
- Homepage
 - The homepage should clearly allow the user to see some basic site information, log in, and create an account. **(2)**

Architecture Diagram:



Front End Design:



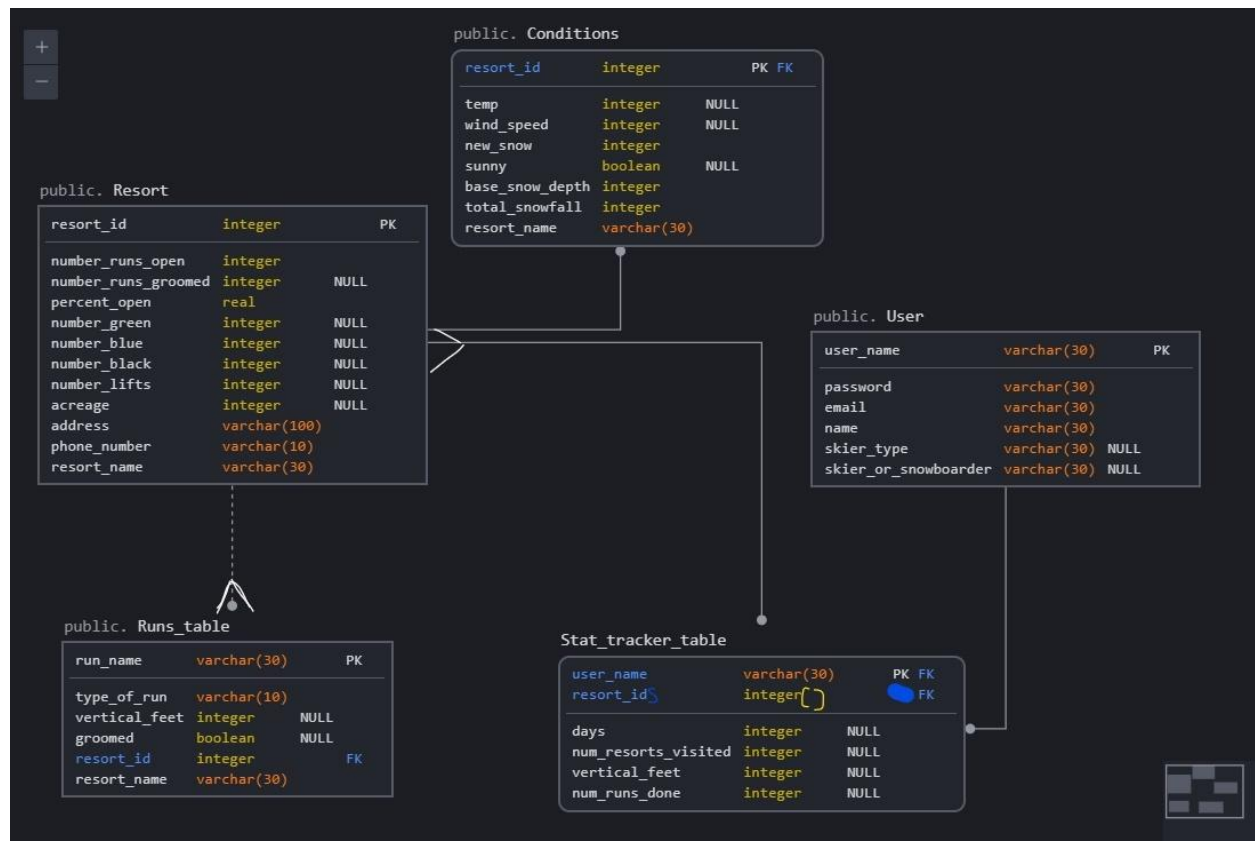
Web Service Design:

APIs Used:

- OpenWeather
 - OpenWeather allows us to access weather data for latitudes and longitudes. This is nice because it allows us to pinpoint specific resorts. OpenWeather takes in the key, longitude, latitude, and the units used. We are using it to return a weather description, temperatures, wind speed, and the weekly forecast. We aren't currently using these calls but you can also access things like humidity, sunrise time, and visibility from this api.
- Google Maps
 - Google maps API allows us to embed maps into our website. The API takes the key, longitude, and latitude as input and returns the satellite map for that location.

Database Design:

Database is made using Postgres



Individual Contributions

- Caleb Caulk
 - Challenges

- Database Design
 - Setting up the database, creating tables and inserting values
- Drew Hockstein
 - Made conditions pages
 - Did the revised features list
 - Did the web service design
- Bill Black
 - Created architecture diagram
 - Combining login with registration
- Neo Rieck
 - Created stat tracker page
 - Created the wireframe

Challenges

1. Setting up the database
 - a. Backup plan is to talk to a TA or professor about how to do it since it is necessary
2. Figuring out Node js
 - a. Backup plan is to use a different language such as python or c++ for the application server
3. Figuring out how to set up an application server
 - a. Backup plan is to have the front end connect directly to the database