

Team: 102-4

Group: Wolf-Gang Amadeus

Members: Tyler Fansler, Dazong Chen, Titus Bard, Chinwei Yang, Ryan Swanson

Project Name: Come To CO!

Check	Criteria	Details
<input type="checkbox"/>	Revised List of Features	<ul style="list-style-type: none"><li>• This is an updated list of your FEATURES inventory (from Milestone 2).</li><li>• It is normal for feature lists to change during the course of a project.</li><li>• Some features may have been dropped. Some features may have been added.</li><li>• This revised features list should reflect these changes.</li><li>• This revised features list should identify the PRIORITY order of how the features will be developed.</li></ul>
<input type="checkbox"/>	Architecture Diagram	<ul style="list-style-type: none"><li>• This deliverable is a picture or diagram that shows each architectural component of your application.</li><li>• The diagram should identify how your application's front-end, integration layer, and backend processes will be hosted.</li><li>• This diagram should identify the flow of data from one layer to another.</li><li>• This diagram should identify the protocols being used to/from each component layer.</li></ul>
<input type="checkbox"/>	Front End design	<ul style="list-style-type: none"><li>• This deliverable is a series of diagrams that show the basic design of your application's front end.</li><li>• Typically, this design is most easily presented in terms of a WIREFRAME.</li><li>• This may be hand drawn, or created using a web page wireframe drawing tool.</li><li>• The front end design should identify each major feature of the application's front end</li></ul>
<input type="checkbox"/>	Web Service Design	<ul style="list-style-type: none"><li>• If your application is using Web Services via APIs, this deliverable should list the Web Service being used along with a description of the Web Service's API including the data being passed to and received from the API.</li></ul>
<input type="checkbox"/>	Database design	<ul style="list-style-type: none"><li>• This deliverable provides a summary design of your application's database.</li><li>• The design document should identify each type of data being stored in your database.</li><li>• This may be documented in terms of a schema definition, showing data entities ("files") and attributes ("fields")</li><li>• This may be documented via an Entity Relationship Diagram showing database tables and columns.</li><li>• The document should identify the specific DBMS technology being used to store your application data (PostgreSQL, MySQL, Firebase, etc.)</li></ul>

## Project Milestone 4

### Project Features List

- Presentable Homepage
  - A beautiful homepage will greet users with a getting started button and links to our contact information
- Preference and Interest Selection
  - The customer will select their favorite activities and types of food from a list and we will tailor suggestions to the user's preferences
- Log-In Database
  - A user will be required to log in before accessing the website. Information about a customer will be stored in a database. This way a person's recommendations will be saved when they return to the website.

- Website Hosting on Github
  - The website will be accessible to anyone on the internet and will be hosted through GitHub
- Website Recommendation
  - Page that takes information received via our other features and will suggest tourist attractions. Suggestions will be calculated via Javascript
- Background Slideshow
  - While a user is on our website we will present a slideshow of the fun activities that Boulder has to offer. Bootstrap has a carousel attribute that can display this.

## Architecture Diagram

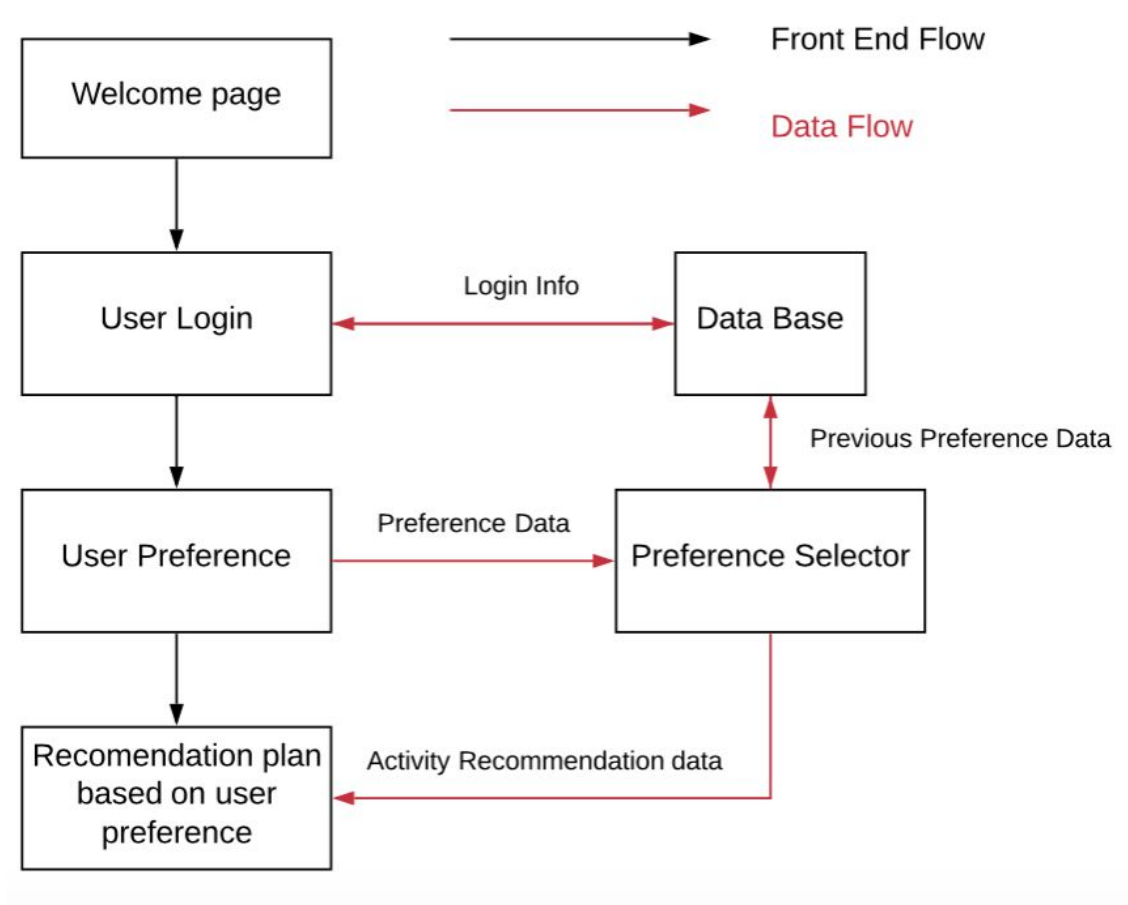
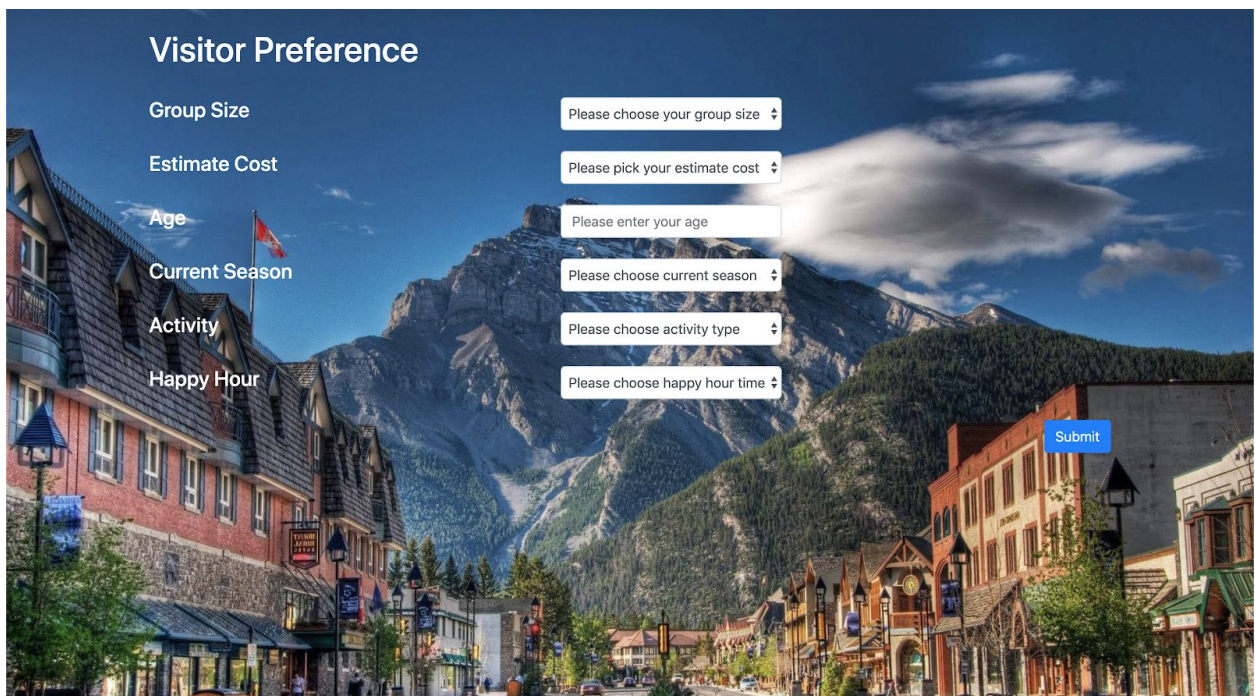
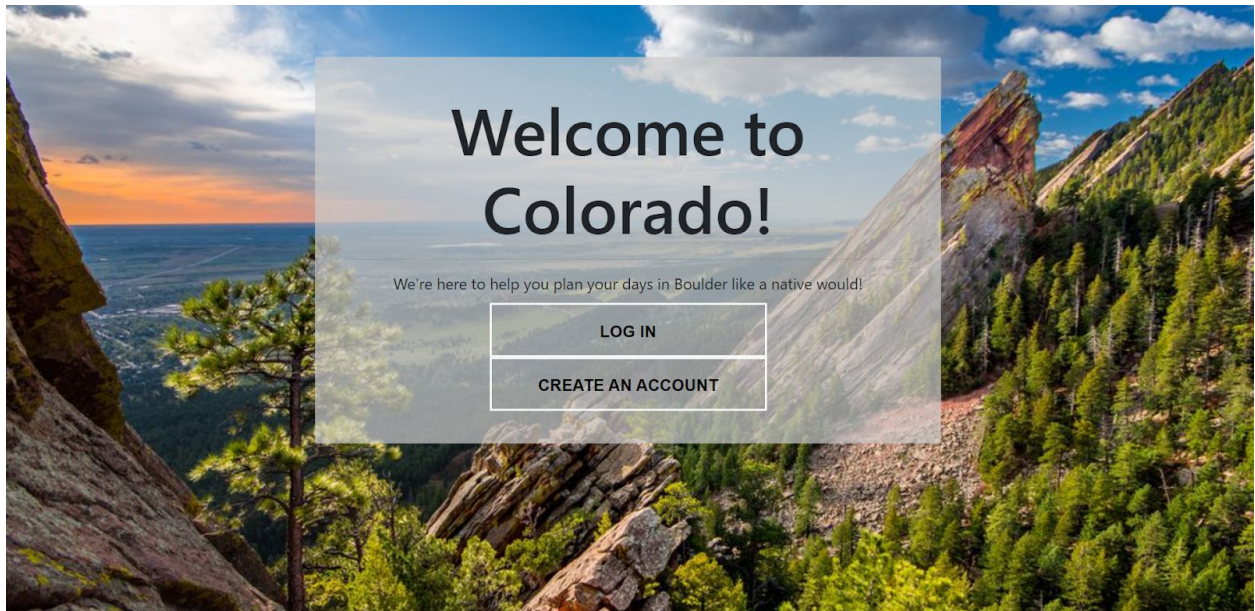


Figure 1: Proposed Data Flow and User Front End Flow

## Frontend Design



- For front end design we decided to go with a minimalistic, transparent jumbotron so that we could make the design focus on pictures of boulder. This was to show that the focus of the website is to connect our users to Boulder, and be able to experience everything that it has to offer
- The buttons and dropdowns that we use, have the same background as to not draw too much attention away from the pictures, while lighting up when you hover over them to create a feel of interactivity and ease of use.



## Web Service Design

Our website will be utilizing the Google Places API to make our webpage more dynamic. When we recommend a restaurant to a user, we will perform an HTTP request to get restaurant hours of operation, location information, and photos of the destination. This data will be displayed on the recommendation page and will keep our website up to date for years to come.

## Database Design

The website will store log-in data in our website's database. The database will hold information for each user. Their login, password, and preferences all need to be stored here. The database isn't going to be very large. It is used for storing user account info and preferences specific to that user. It will not contain information relating to the recommended locations. That is handled in the Google API. This will make sure that the information is always as updated as possible and will be done automatically. Storing the business information in our database would be inefficient because it could easily change. Preferences include normal price range, age, activity type, etc. We will be using PostgreSQL because that is what is most familiar to our group.

### Basic ER Diagram Example

Tyler Fansler | November 8, 2019

