**Colorado Mesa University Outdoor Program Reservation Software  
Technical Documentation**

Owner: Ryan Dutch – Colorado Mesa University Outdoor Program  
Developers: Michael Sweeney, Drew Collins, Luke Schaefer

**Development Login**  
User: dev  
Password: w'}'KALp<9)f

**Project Motivation:**   
Providing a means for employees at the Outdoor Program to quickly and efficiently manage gear reservations and rentals.

The purpose of this document is to provide any future developers of this project with a starting point for understanding the code base.

**Links:**  
This is a private repository, so you will need to contact [michael.sweeney5@gmail.com](mailto:michael.sweeney5@gmail.com) to clone the repo or transfer ownership of the repository  
git clone url is: [git@bitbucket.org](mailto:git@bitbucket.org):cmuop/cmuop.git

Github for the database class we used for back end to minimize sql vulnerabilities and increase readability, security and writability of code.  
<https://github.com/joshcam/PHP-MySQLi-Database-Class>

Mustache is a logic less templating engine used to dynamically generate web page content from data generated in php scripts. The basic idea is simple: query the database with php. Stuff the results into the $data array (i.e. $data[‘Customers’] = $dbresults). Then in the mustache templates, use a mustache hash loop to display all the customers contained in the array $data[‘Customers’]. View the php and mustache files in the repo to learn more. Mustache information here:  
<https://mustache.github.io/>

Material Design icons:   
<http://zavoloklom.github.io/material-design-iconic-font/icons.html>

Other technologies used:  
Javascript, AJAX, Jquery, Bootstrap, Material Design - Use google-fu to find documentation on these.

Directory structure of the application:

Root web folder (subject to change)  
- dev/  
 - (contains base php scripts for each page)  
 - css/ (self explanatory)  
 - js/  
 - (contains javascripts we wrote)  
 - vendor (contains plugin javascripts such as jquery and bootstrap)  
 - fonts/ (self explanatory)  
 - include/ (contains php code used in all scripts, and contains php files handling AJAX requests)  
 - views/ (contains mustache templates used in rendering dynamic content from php scripts)

1. **Design decisions**
   1. The developers made a few design decisions regarding certain aspects of the project that might need clarification. The purpose of this section is to shed light on how to use the product from a general standpoint.
      1. Carts
         1. Non-finalized reservations. In the cart stage, items are not officially reserved, nor are they excluded from other carts based on date conflicts. Carts (think Amazon) are a place to build a reservation. You can add and delete inventory items to and from a cart. When you submit a cart (think checking out with Amazon), the cart is finalized and a reservation is created.
      2. Reservations
         1. Finalized reservations. You cannot add or delete items from a reservation.
         2. You can edit dates should the time frame of a reservation change, although date conflicts with inventory items in other reservations may prevent it.
      3. Inventory
         1. Directory structure much like the old file system.
         2. Categories may have subcategories, or children, that contain either inventory items or other children.
         3. Only Dutch can alter the structure of inventory.