Highlands Ranch Backcountry Membership

Architecture Notebook

There is guidance within this template that appears in a style named InfoBlue. This style has a hidden font attribute that allows you to toggle whether it is visible or hidden in this template. Use the Microsoft® Word® menu **Tools > Options > View > Hidden Text** check box to toggle this setting. There is also an option for printing: **Tools > Options > Print**.

# Purpose

The intended outcome of this document is to provide a detailed outline of the architecture involved in our solution for the client (HRCA). The proposed solution ultimately remedies the long-term concerns surrounding the current Excel spreadsheet system. This new solution will include a locally hosted website that is only accessible to the workers so they can keep better track of Membership levels, dates of expiration, and relevant information. As a stretch goal, a log in system will be implemented so employees can log onto the website in line with security guidelines.

# Architectural goals and philosophy

This solution to meet HRCA – Backcountry’s must be scalable for in the future. This means whatever system is implemented must have room for more members to be added. With this in mind, there must to be a way to ensure the system will stay robust enough to need minimal IT intervention.

The main goals for this system will include:

* Import a CSV record into the database.
* Compare the newly imported information in with the older information on the database and update the records.
* Add in a member ID as a primary identifier that will stay in the backend and not be displayed to the client.
* Make a local webserver so the database can be accessed and communicated with through a local web page.
* Form reports and export those records to the user directly, as well as allow for InSite editing of the information.
* Make a user database to allow an “Admin” on the site to add and remove users as well as grant privileges to other users of the database.
* Allow for a quick/easy integration of the system onto the client’s systems so production work will not be majorly affected.

# Assumptions and dependencies

## Assumptions

* A Local server that is running Windows Server 2012 – 2022 or Windows 10 Pro/Enterprise
* Less than 15 total users in the environment
* All working out of one building or having a stable VPN connection

## Dependencies

* Administrator Access to the network and server
* 2022’s data in order to make the databases
* Staff IT in order to make a seamless integration

# Architecturally significant requirements

The main significance change is as follows:

* Any meaningful improvement to their current workflow situation

The reasoning behind this outcome is because HRCA – Backcountry is a small sub section of the greater HRCA. The Backcountry area is growing at a significant rate, and because they get their lists from the master section of HRCA they will only receive data in a .CSV format. This is not something Meebers can change or influence. Consequently, Meebers will implement a database and Web UI solution so HRCA have some redundancy for their stored information, as well as insuring HRCA will be able to make their own lists and be more productive in the future.

# Decisions, constraints, and justifications

* Not changing the delivery of information from the master system
  + This is out of our scope as this comes from the greater HRCA group. .CSV/s will be used and can be adjusted as needed to fit into the database system.
* Using a database and a Web UI
  + People that are working with Backcountry want to be able to do their work quickly and effectively. Currently, the membership projects are tedious ones to get done. With a Web UI, multiple people can work on these lists at once to accomplish multiple tasks simultaneously.
* Meebers will not have admin access to the local server.
  + According to the client, Meebers will not have admin access to the server. HRCA has an IT department to help with implementation.

# Architectural Mechanisms

## Architectural Mechanism 1

HTML – This will be the backbone of the website and what will display the results.

## Architectural Mechanism 2

CSS – This is what will make the website look and feel like an actual application and not just some text thrown at the screen.

## Architectural Mechanism 3

PHP – Natively, HTML and MySQL can’t talk to each other, so a server-side language which can act as a catalyst to transport over information will be needed. This is what will the website to be dynamic instead of static.

## Architectural Mechanism 4

MySQL – MySQL will store the membership information.

## Architectural Mechanism 5

Python – Python will interpret master .CSV files and import it them into the database automatically.

# Key abstractions

* New User
  + New Users can be added for if/when their department grows.
* Reset Password
  + People forget their passwords. In case of this event, the proposed solution includes a password reset functionality.
* Login
  + Users must be properly authenticated.
* Add Member
  + A script will exist to add a new member in case the web UI is nonfunctional.
* Edit Member
  + Member information can be updated to reflect any real life changes.
* View Member
  + Members must be able to check on their current membership status.
* View Lists
  + Database can view and sort lists as needed.
* Export Lists
  + Lists can be exported as a CSV file for ease of access.

# Layers or architectural framework

The solution will consist of a Node.js and a React framework. The database will contain tables as needed that are linked via Primary and foreign keys.

Note: Test data has not been provided, therefore the required tables in the databasehave have not been listed in this architecture document.

# Architectural views

