

CMPT 470 - Project Report

SFU Sports Clubs Website

INTRODUCTION

Most universities have recreational clubs, aimed at promoting extracurricular activities for students, with some of the clubs focusing on sports branches such as Football or Soccer. Sports clubs often have a recreational member base as well as a competitive team which consists of more advanced players. The ideal dynamics within sports clubs is to support and improve recreational players with potential, so that they can become an asset for the team in the future.

However, these clubs are run by students and managing both the team and the club is often difficult. Moreover, the clubs don't have a platform that facilitates management and maintenance. Most clubs use different platforms for different tasks, which is not only inefficient but also hard to keep track of. For example, some clubs keep their documents in Google Drive and update them manually. Documents eventually pile up and become difficult to manage. For communication with members, social media is the primary choice, but not everyone has social media accounts or check their account frequently enough.

Considering the problems described and many more similar problems, a platform which facilitate managing and maintaining these clubs can be an asset for executive members. Moreover, this platform would provide a robust source of information for members/players. We intend to implement such platform for the group project of this course.

FEATURES

Shortly described, the platform allows students to login using their universities' login system. After logging in, students can either search for a specific club or see a list of registered clubs. Students can learn more about the clubs by visiting club profiles, register/subscribe to clubs and choose to receive various notifications about the clubs. Students can also update their profiles to provide information about themselves.

So far, the features listed are nothing special. The actual problem lies in creating, managing and maintaining the clubs. To facilitate these processes, the platform follows a permission-based membership system. In the platform, there are pre-defined actions for managing the clubs, such as posting an announcement, recording an expense or viewing certain documents and members need permission to perform each of these actions.

Club admins can create new roles or use role presets to encapsulate a set of permissions and assign roles to members, determining what actions each member is allowed to take. These roles follow a hierarchical structure, building on top of each other.

The principals in the membership system is extended to all other services for managing clubs. For example, types of expenses are created in a similar manner to roles. To create a type of expense, members need to specify “input fields” as well as “label fields”. In the future, any expense of the same kind can be recorded by filling in the appropriate information in the input fields. For example, a tennis club might have court booking expenses. To record these expenses, the club creates an expense called “Court Booking”. The expense can have input fields for, hourly fee, provider facility and the total booking time. By doing so, any court booking expense in the future can be recorded by simply providing a provider facility, hourly fee and the total booking time.

To wrap up, the focus of this platform is abstraction and automation. The platform aims to be abstract enough to fit the needs of every club; be highly customizable and provide enough tools to facilitate the setup/modification processes for management and maintenance. Ideally, once a club is set up, managing it (e.g. posting announcements, events, registration, assigning roles to members, tracking finances, etc.) is very easy and highly autonomous. Therefore significantly reducing the work over long periods with a slightly longer set up time.

IMPLEMENTATION

Summary

Application implements a LAMP stack, using GCP Compute Engine VM Instances for the web-host and database servers. Once enough features are implemented, a clean production branch is prepared and pulled from the web-host server. Composer is used to manage PHP dependencies and NPM/Yarn is used to manage front-end packages. The source code on web-host server is linked to web-host’s document root via a symbolic link.

The web application is implemented using PHP using Laravel MVC framework. Laravel’s “Auth” is modified redirect users to cas.sfu.ca (SFU’s Central Authentication Service). Upon successful authentication, SFU redirects back to the application specified service link with a ticket indicating the user is a valid SFU user. Application then validates the ticket and registers the user to its database. Restricting page access to only valid users is done through a middleware.

The core features of the application are; creating a club, recording expenses for clubs, posting events and sharing posts on the club dashboard. In order to use these features, the user must be logged in. If the user is not logged in, the application will redirect to SFU’s login page.

Creating a club effectively registers that club to the application. Users can then interact with said club. In current implementation, only club founders can use the core features for a specific club. However, users can see a list of all registered clubs and search for a specific club.

In order to use the features, the user needs to navigate to ‘Manage’ tab. From the dropdown menu, the user can choose which feature to experiment with. Expense to manage expenses, events to manage events and dashboard to browse or add posts.