LifeAtHere – The Backend Experience

Allan La, Lukas Munoz, Lucas Nicolois and Anushikha Sharma

CSCI 476: Senior Design

Computer Science, Bucknell University, Lewisburg, Pa.

Abstract

Bucknell is always buzzing with a wide range of activities that cater to students, faculty, and staff. LifeAtHere aims to enhance the experience of viewing and managing campus events. Our team has focused on building a server that can search events, a database that stores events, and a web application that allows anyone with a Bucknell email to view, create, save, and edit events on the application. The server and database also provide backend support to the mobile application built by the LifeAtHere Mobile Team.

Background

With many events happening and no cohesive platforms to effectively summarize and communicate the information, Bucknellians have a difficult time keeping track of when, where, and what events there are.

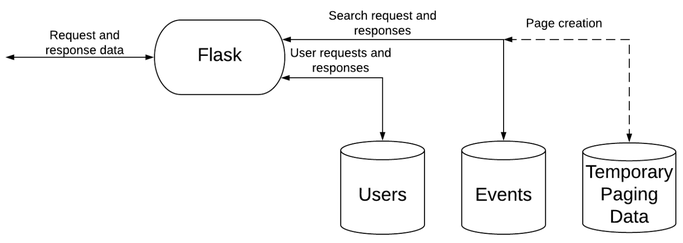
Our client, Leo Botinelly from the Library and IT Department, has this information and wanted to consolidate it into a usable format. Thus, we focused on building a server-side solution to the problem.

Flask Server

The backend server runs on Python Flask. The main features that run on the server include-

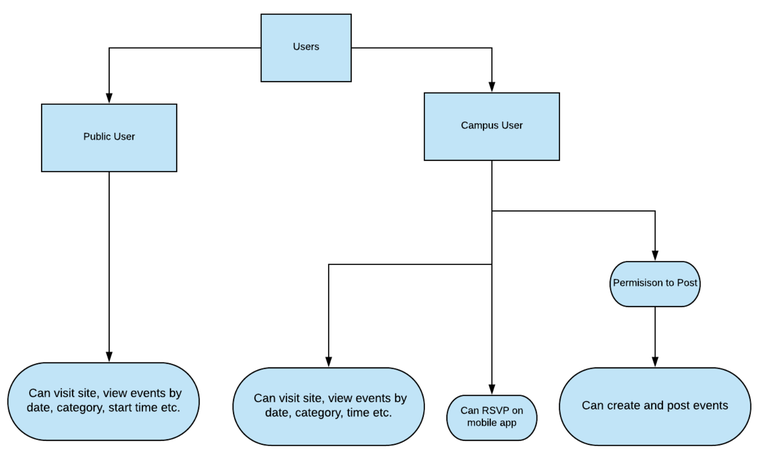
- Search
- Pagination
- Add events
- Create users
- Edit user settings
- RSVP to events

The server runs using Amazon Web Services. It runs on a t2 micro instance to make use of the free tier of Amazon's



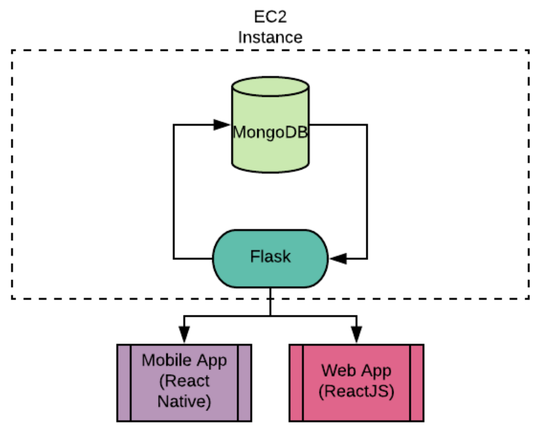
The Flask server interacts with three collections in the database: Users, Events, and Temporary Paging Data.

Web App (ReactJS)



There are two types of users in our system: campus users and public users. Public users have limited access to our system's functionality.

Design



The mobile and web applications interact with our database ad server by connecting to the EC2 instance they run on.

MongoDB Database

We chose to use Mongodb for our database system for a number of factors:

- JSON data format matched the data given to us from our client
- Pymongo, a python library for Mongodb scripting, interfaces well with Flask
- Fast Response times
- Self enforced data integrity gives us more flexibility

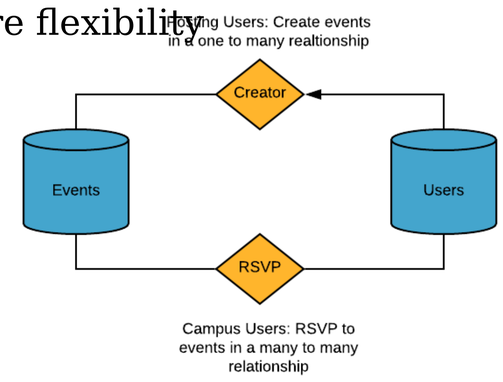
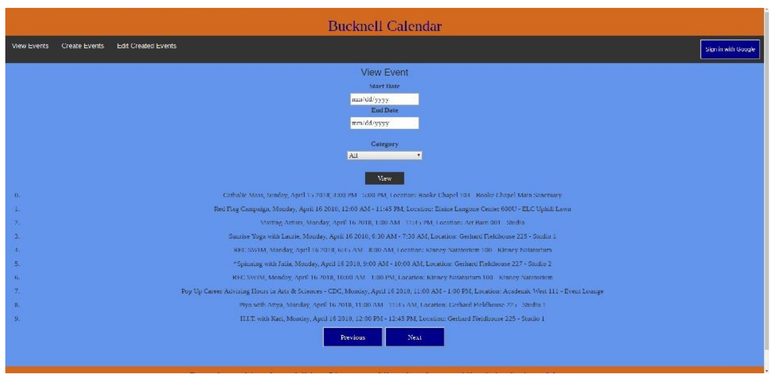


Illustration of the creator relationship between our databases

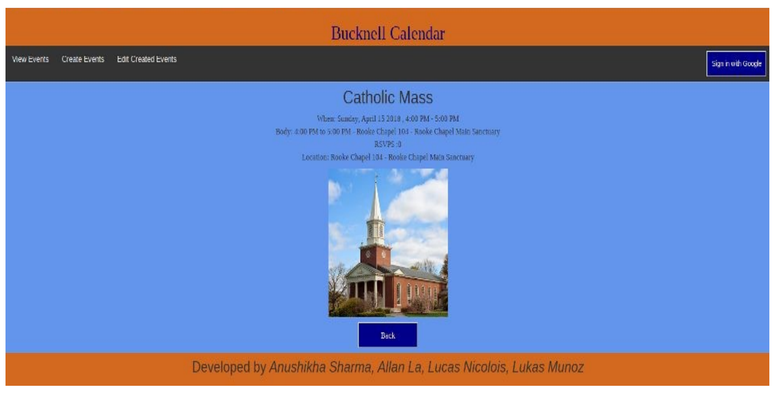
Our database is formatted around two key collections:

- **Events**: This database stores data for displaying and managing events. This includes information such as total RSVPs and event descriptions.
- **Users**: This database stores data for managing user profiles and activity. This includes data such as mobile app settings and a user's RSVPs.

Results

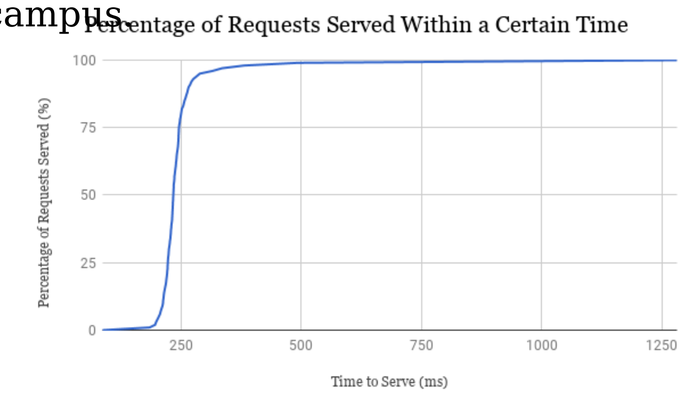


The frontend layout of the view events page



The frontend layout of an event page

We use ApacheBench to load test our server. The test consisted of 20 concurrent requests performing 500 total requests. This yields 80 requests per second, which is enough to serve the campus.



ApacheBench test results

Challenges

- Deploying AWS instances and handling AWS security
- Learning ReactJS and implementing a web application
- Refactoring our codebase
- Coordinating with the mobile team

Acknowledgements

We would like to thank our client Leo Botinelly, from L&IT, Prof. Alan Marchiori and the LifeAtHere Mobile Team