Santa Clara University

Computer Engineering Department

Software Engineering

Alumni Engagement Recording System

By

Ronak Gajrawala

Yutong Li

Mickey Hoang

Santa Clara, California
October 8th, 2018

Table of Contents

Abstract

As of now Santa Clara University does not have a system in which alumni engagement and attendance is recorded for both official and unofficial alumni events. To better understand what aspects of an event affect alumni engagement Santa Clara University needs an online system in which alumni can find events and submit whether or not they will attend as well as any feedback about the event. This report describes a project to create a web interface for alumni events and engagement recording. The basic system is a website where SCU can post official events and approve alumni-submitted events so that alumni can see and interact with these events online. This new system will add a new tool for the SCU Alumni office as well as replace the current event calendar system.

1. Introduction

This report details the various design aspects of the Alumni Engagement Recording System for Santa Clara University. These aspects include the several use-cases of the system, conceptual models, technologies chosen, and architectural design.

This design has three main objectives: create an event calendar with weighted event display, have interactive events for engagement data, and provide a system for SCU administrators to approve and edit events and to view collected data from the interactive events.

The first objective is to create an event calendar for alumni events that displays events with a weighted system in which more important events are displayed first. We plan to achieve this by automatically assigning weights to events based on several factors such as whether the event is official, if the event is recurring, and the type of event. SCU administrators will also be able to modify this weighting system as they seem fit.

The next objective is to have the events on the calendar be interactive to record engagement data. This includes attendance records and satisfaction scores after the event is over.

Lastly, SCU alumni will be provided with a system in which they can approve, edit, or remove events on the alumni calendar. This system will also provide access to the data collected from the interactive events.

This report details all the design choices we have made regarding these objectives.

2. Requirements

There are a multitude of requirements that our system must meet to be considered fully-featured and functional. The two sections of requirements are user-interface requirements and administrative control requirements.

The system must provide several functionalities to users. Users must be able to view upcoming alumni events chronologically with significant events shown first. Users must also be able to engage and interact with the events by indicating plans to attend, giving satisfaction scores after events, and requesting email reminders for events. Lastly, this must be achieved without the need for user logins. Users of this system should be able to use all the functionalities by only providing their email address.

The system must also provide a framework for SCU administrators to modify and approve events that show up on the alumni event calendar. Administrators need to be able to change event descriptions and information and approve and post unofficial events that are submitted by SCU alumni. This administrative tool will also need to be able to give all of the data recorded by the interactive events so that administrators can analyze and learn about user engagement.

3. Use-Cases

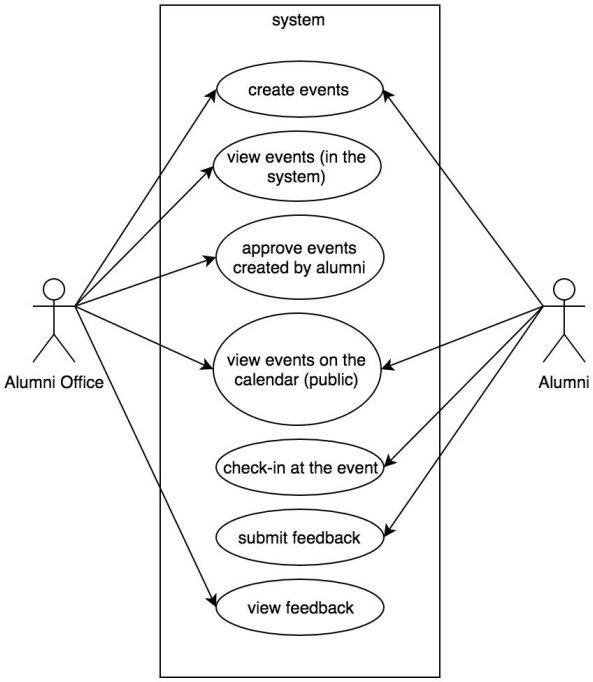

Figure 1: Case diagram of the system

With the system, the Alumni Office is able to create events and approve events created by alumni, while the alumni can use the system to create their own hosted events.

For the events that is created by the admin, it will be on the calendar automatically after creation.

For the events created by the alumni, they will be on the calendar if they are approved by the admin.

For all events, the alumni can check-in using the system and submit feedback afterwards.

The admin can then use the feedback collected to decide what kind of events are more popular and therefore hold more of this kind of events.

4. Activity Diagrams

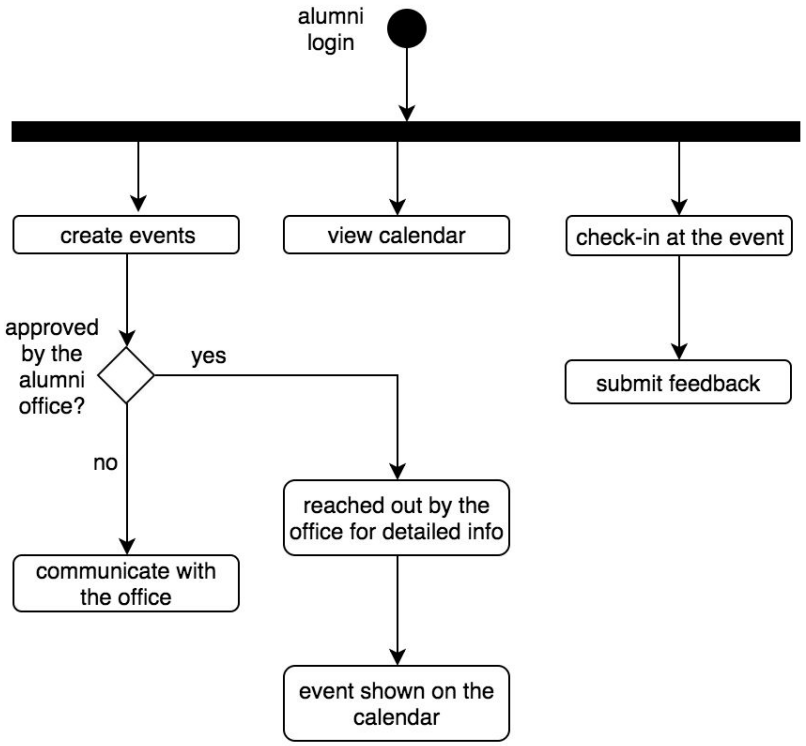

Figure 2: Activity diagram for alumni use

This activity diagram is what the alumni can do with the system.

Through the system, the alumni can create their own events as well as view the event calendar. When the events that the alumni create is approved by the admin, the admin will reach out to the alumni for more detail about the events. After that, the events will be posted on the calendar. If the events are rejected, the alumni can choose to communicate with the office privately to discuss.

When the alumni attend any event, they can check-in at the event using the system, and submit feedback afterwards.

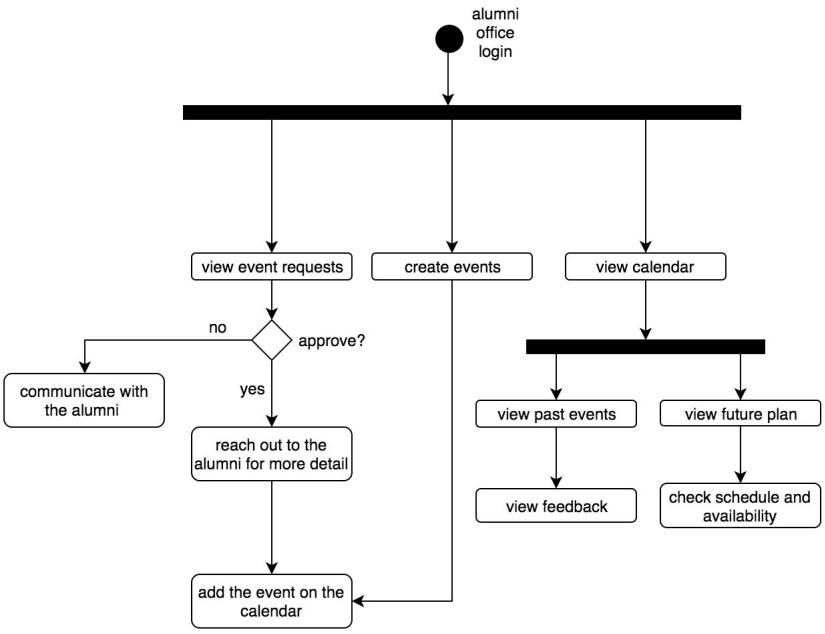

Figure 3: Activity diagram for alumni office use

This activity diagram is what the alumni office can do with the system.

Through the system, the admin can view event requests by the alumni, view the event calendar, and create official events.

When the admin decides to approve certain event, the admin will reach out to the alumni who created it for more information about the event. When both sides agree on all details about the event, the admin can then add the event on the calendar. When the admin decides to reject an event, they can communicate with the alumni privately.

By viewing feedback of the past events on the calendar, the admin can learn what kind of events is more popular or immersive and then hold more of these events in the future. By view the calendar for future plan, the admin can check availability and therefore avoid conflict while adding events to the system.

5. Conceptual Model

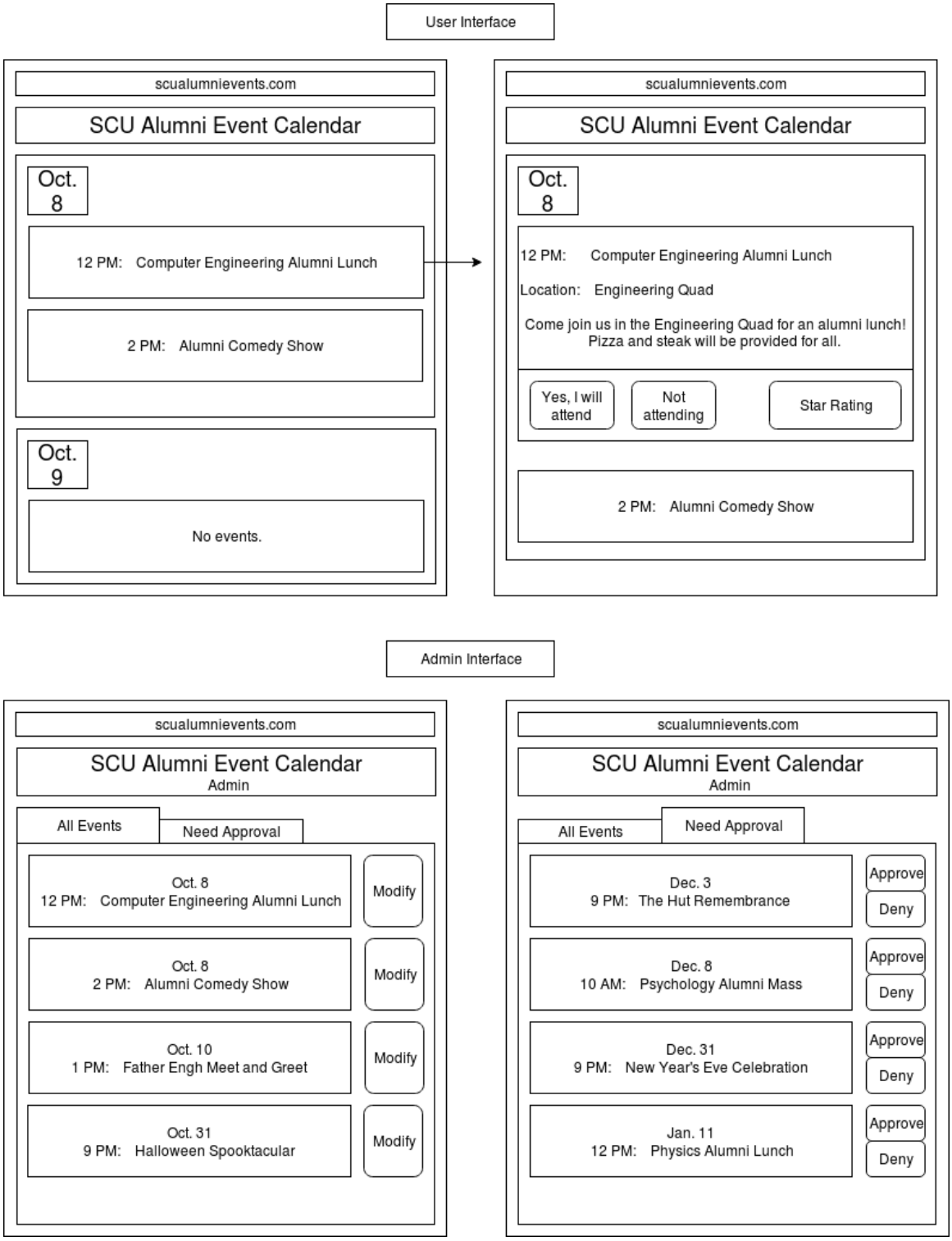

Figure 4: Conceptual Models of Website

In Fig. 4 there are four models of what the interface will look and behave like. The first interface is the user interface in which the alumni can view events in the calendar and interact with these events. The first page shows what the calendar should look like when displaying events in chronological order and then by importance within the same dates. When clicking on an event the event will expand giving more detail to the user and showing buttons for the user to show attendance plans and to rate the event.

On the administrative side of the interface the events will show up in chronological order. Next to the events there is an option to modify the information shown on the event. Clicking these events will also give any recorded data from attendance plans or satisfaction surveys. Public events will show up to the administrators to view or edit on the ‘All Events’ tab. The ‘Needs Approval’ tab shows administrators all events that have been submitted and are waiting to be approved and shown on calendar. Here there will be buttons to approve or deny these events.

6. Technologies Used

Since we will be utilizing the Design Center to host our website, we do not have many technology choices to choose from. We will be using a SQL database and MySQL for database management since these are the only database technologies that the Design Center utilizes. For the backend of the server, we are forced to use the Apache server and PHP backend language that are already configured to work for the Design Center. We will incorporate a PHP framework, i.e. Laravel, to make backend development more structured and easier to implement. The backend will server as an internal API for the frontend to query. The frontend of the website will be a single-page application implemented using a JavaScript framework, namely React. In addition, we will be using the Bootstrap framework to help style our website. Finally, we will use Webpack as our frontend asset manager to simplify the frontend development process.

7. Architectural Diagram

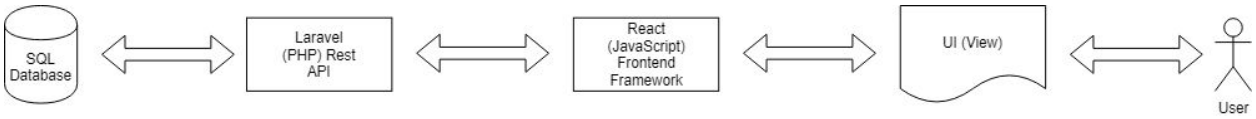

Figure 5: Architectural Diagram

8. Design Rationale

We are forced to use SQL, MySQL, Apache, and PHP to implement the backend because other technologies are not supported by the Design Center. We are using the Laravel PHP framework to significantly simplify and bolster backend development, keep our backend code clean and easy to understand, and ensure that our internal API is REST-compliant. We are implementing an internal REST API and SPA frontend so that the initial load time for our website is fast, as many users will probably load the website over a mobile connection weekly or biweekly. We chose to use the Bootstrap frontend framework to make our site's UI mobile-friendly without having to waste time writing our own mobile-friendly code and styles. Finally, we decided to use Webpack as our asset pipeline manager so we can write modular and loosely-coupled code without having to worry about writing HTML code to reference all of the source files we create.

9. Test Plan

Functional Test Plan: When we are testing, we need to make sure that the users are able to create events and view the calendar. Additionally, for the faculties in the alumni office, they should be able to approve or reject events created by the alumni.

Security Test Plan: Since the system is only for the Alumni Office and alumni, users that aren't in either category should not have access to the website. If the name, year of graduation that the user inputs cannot be found in the system, s/he can't access the website. Only faculties in the Alumni Office should be authorized to approve events. The email addresses of these users will be authorized will full access to the system.

10. Risk Analysis

Table 1: Risk Analysis

Risk	Consequences	Probability	Severity (0-10)	Impact (Probability * Severity)	Mitigation Strategies
Time	System not fully completed on time	5%	7	0.35	Set deadlines Prioritize core features
Bugs	Users are unable to use system properly Can extend production time	99%	3	2.97	Write clean, readable code and have peers review code Implement functional tests to make sure all aspects of the site work
Group member missing	Loss of productivity, development may take longer	12%	4	0.48	Make sure group members know what all other group members are doing Check-in (commit) code as often as possible
Design Center downtime	Loss of productivity. development may take longer Complete site failure	0.01%	10	0.001	Have a mirror site on another server Use personal computer to develop website in the meantime
Too many users in DC	Site slows down	0.1%	9	0.009	Have a mirror site on another server

(resource starvation)					
Data breach	User data can be used to send targeted spam/fraud/phishing emails Fake events can be created; alumni could end up in a dangerous situation	0.001%	10	0.0001	Make sure passwords are secured and cannot be read by other DC users. Install database and runtime security updates.

11. Development Timeline

	Task Name	week1 (Sep. 17-21)							week2 (Sep. 24-28)							week3 (Oct. 1-5)							week4 (Oct. 8-12)						
		M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S
1	Select the project and work on problem statement	Group																											
2	Design report															Group													
3	Design review presentation (slides)																						Group						
4	Front-End																												
5	- Calendar																												
6	- Interactive Events																												
7	- Weight based event showing																												
8	Admin Side																												
9	- Modify/Remove/Add Events to Calendar																												
10	- Approval of Events																												
11	- View collected data																												
12	Testing																												
13	Final Revisions																												

	Task Name	week5 (Oct. 15-19)							week6 (Oct. 22-26)							week7 (Oct. 29-Nov. 2)							week8 (Nov. 5-9)						
		M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S
1	Select the project and work on problem statement																												
2	Design report																												
3	Design review presentation (slides)																												
4	Front-End	Group																											
5	- Calendar	Mickey																											
6	- Interactive Events	Ronak																											
7	- Weight based event showing	Yutong																											
8	Admin Side								Group																				
9	- Modify/Remove/Add Events to Calendar								Ronak																				
10	- Approval of Events								Yutong																				
11	- View collected data								Mickey																				
12	Testing															Group													
13	Final Revisions																						Group						

Figure 6: Development Timeline

12. Table of Figures