TA Emergency Coverage System

Final Report

COEN 174L Th 2:15 Team 1

by

Isabela Figueira
Sonali Chaudhry
Kai Schmidt

Santa Clara University

Santa Clara, California
December 1, 2017

TA Emergency Coverage System

Final Report

COEN 174L Th 2:15 Team 1

Isabela Figueira
Sonali Chaudhry
Kai Schmidt

Santa Clara University
December 1, 2017

ABSTRACT

When Teaching Assistants get sick or become otherwise unavailable, they must contact their supervisor and ask them to find another TA to fill the spot. This process is cumbersome since a qualified, available substitute TA can be difficult to find. Our solution provides a way for TAs to easily see who is available to cover their shift. The TA is then able to contact the potential replacements without using their supervising professor as a middleman.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Currently, when Teaching Assistants are sick or otherwise unavailable, they must contact their supervisor and ask them to find another TA to fill in. The supervisor then has to contact the rest of the qualified TAs to figure out who is available. Having to use an intermediary to fill labs makes the entire process more burdensome for all involved.

However, there is a better way. We propose a solution that improves the current method of finding replacement TAs to teach a class or lab. When someone needs a class filled, our system will tell them who is available to fill in so that they can directly search for a replacement. This system will significantly improve the process of finding replacements to fill in classes by making it more efficient for TAs and professors. Before, a TA did not have a way to contact other TAs since they had to contact the professor in charge of the TAs. Utilizing our solution, a TA will have access to information regarding other TAs' availabilities and contact information, given that they use the system.

We will build a webpage with which TAs can find replacement TA for the lab section they will have to miss. An administrator will be able to keep track of authorized professors. Professors will be able to keep track of their current TAs.

Chapter 2

Requirements

2.1 Functional Requirements

- Critical
  - Allow administrators to invite and remove professors, administrators, and TAs to the system
  - Display all professors and TAs on the admin's home page
  - Allow professors to invite and remove TAs from the system
  - Display all TAs on the professor's home page
  - Display the TA's schedule on the TA's home page
  - Allow TAs to add and remove events from their schedules
  - Allow TAs to view the contact information of all qualified TAs who are available to substitute a selected class
  - Allow all types of users to sign up and log in/out
- Recommended
  - Allow professors to modify TAs' qualifications
  - Allow each user to modify their profile data, such as contact information
- Suggested
  - Allow TAs to click a button that automatically contacts one or all available substitute TAs, or their professor
  - Allow professors to view TAs' schedules

2.2 Non-Functional Requirements

- Critical
  - Usable and easy to understand
- Recommended
  - Efficient: will not perform unnecessary queries or tasks
  - Responsive: the system will not be sluggish while using it
- Suggested
  - Easy to maintain
  - Pretty: Attractive to users

2.3 Design Constraints

- Web-based
- Hosted on Linux DC computers
- Must be able to run on both Firefox and Chrome

Chapter 3

Use Cases

The system has three different actors: the administrator, the TA, and the professor. To simplify the figure and increase readability, we split the use case diagram into two separate diagrams: specific user actions (Figure 3.1) and actions all users can do (Figure 3.2). Only main functions are listed to keep things simple. More detailed information about specific use cases can be found in Table 3.1.

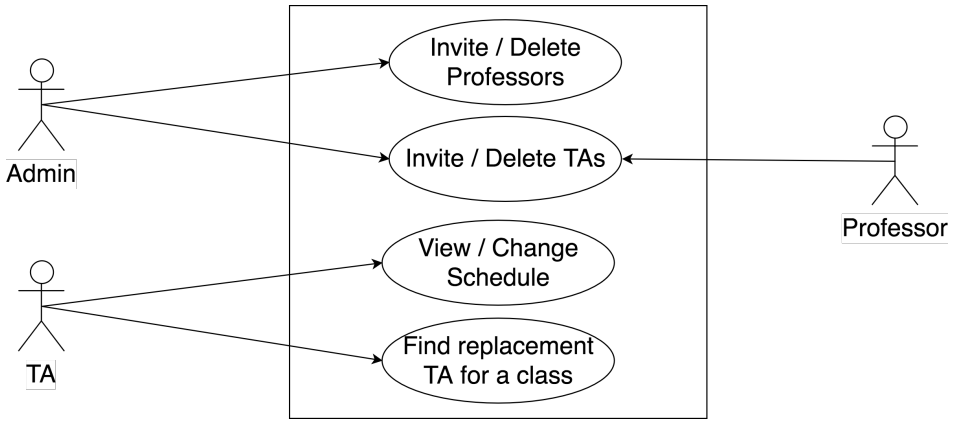

Figure 3.1: Specific Use Cases

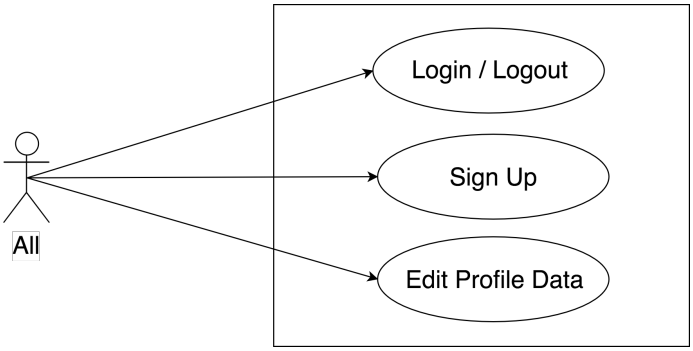

Figure 3.2: All Use Cases

Table 3.1: Description of Use Cases

Name	Goal	Actors	Pre-Conditions	Post-Conditions	Exceptions
Login	Sign in to account.	Admins, Professors, and TAs	User has created an account.	User is either logged in or given an error message.	Account does not exist. Invalid or forgotten credentials.
Logout	Sign out of account.	Admins, Professors, and TAs	User is logged into account.	User is not logged in.	Not signed in.
Sign Up	Create an account.	Admins, Professors, and TAs	User had been invited (i.e. given permission code).	Account is created and user is redirected to the login page, or error message is displayed.	Invalid or missing permission code. Email field does not match email on invitation. Required fields not filled out properly.
Edit Profile	Update personal data, such as contact info.	Admins, Professors, and TAs	User is logged in.	User is taken back to his or her home page.	User cancels edit profile.
Invite or Delete Professor	Generate permission code to allow the specified user to create a professor account, or remove a professor from the system.	Admins	Admin is logged in and has invitee's email address, or there is a professor in the system	User returns to his or her home page.	User cancels invite or delete.
Invite or Delete TA	Generate permission code to allow the specified user to create a TA account or remove a TA from the system.	Admins and Professors	User is logged in and has invitee's email address, or there is a TA in the system.	User returns to his or her home page.	User cancels invite or delete.
View Schedule	See the TA's schedule.	TAs	TA is logged into his or her account.	TA sees calendar and can navigate to different weeks.	TA has not added any events to his / her schedule.
Add Event	Add an event to a TA's schedule.	TAs	TA is logged into his or her account.	TA returns to home page.	TA cancels add event.
Remove Event	Remove an event from a TA's schedule.	TAs	TA is logged into the system. TA has input events in his or her schedule.	Event is removed from TA's schedule.	There are no events in the TA's schedule.
Find Replacement	Display all qualified TAs who are available to substitute the selected class.	TAs	TA is logged into the system. TA has a TA event in his or her schedule.	Popup displays possible substitutes or tells the TA to contact his or her supervisor or professor.	There are no TAs in the system that are both qualified and available to substitute the selected class.

Chapter 4

Activity Diagram

Figures 4.1 through 4.3 show the activity flow of our system for each user type specifically. All users will start at the black filled dot and progress to the half-filled dot. Arrows indicate that the system's execution is able to progress in that direction. Even though double ended arrows are not standard for activity diagrams, we use them to make the figures look cleaner.

Figure 4.1 depicts the administrator's activity flow. After an admin logs in, they see a list of all professors and TAs in the system. From here, the admin can edit their own profile and invite or delete TAs and professors.

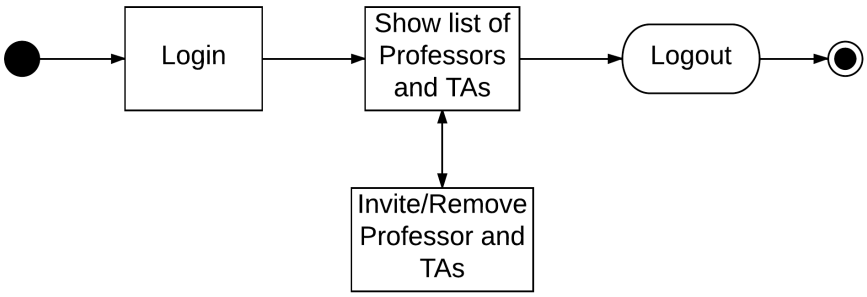

Figure 4.1: Administrator Activity Diagram

Figure 4.2 shows a Professor's activity flow. After a professor logs in, they see a list of all TAs in the system. From this page, they can edit their own profile and invite or remove TAs.

Figure 4.3 represents a TA's activity flow. After a TA logs in, they first see their weekly schedule. From this page, the TA can edit their profile, add events, find substitutes for TA events, and delete events.

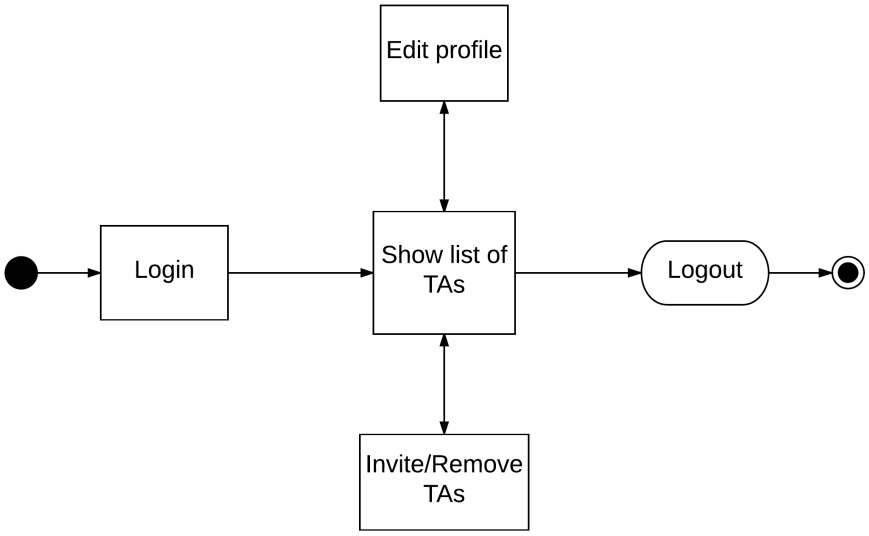

Figure 4.2: Professor Activity Diagram

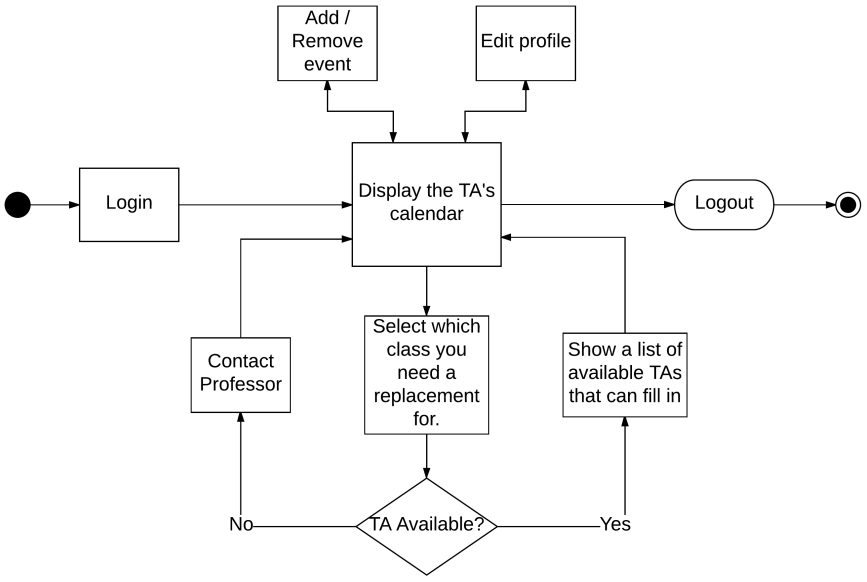

Figure 4.3: Teaching Assistant Activity Diagram

Chapter 5

Architectural Diagram

As shown in Figure 5.1, the system uses a data-centric architecture and manages user accounts with a DBMS. The TA's schedules were originally going to be stored in the the database as well, but to improve efficiency, we decided to store them directly on the server. This architecture allows the potentially large amount of data required by the site to be stored both efficiently and persistently.

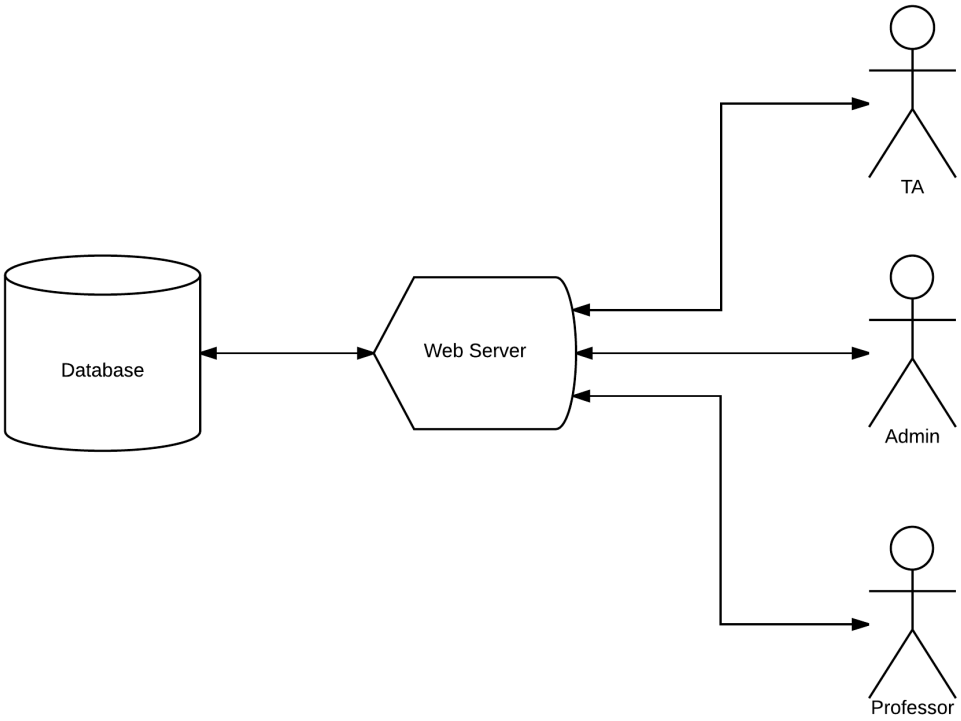

Figure 5.1: Architectural Diagram

Chapter 6

Technologies Used and Design Rationale

For version control in our project, we used GitHub. We chose this tool over other version control technologies because it works well for projects with multiple developers and one of our teammates was already familiar with Git.

We used an Oracle database to store users and their data. The users' data were easily organized into tables, so it made sense to use a relational database. We chose the Oracle database because relational SQL databases are not significantly different from one another in terms of functionality, it was integrated into the ECC system, and we were familiar with it.

We used PHP to connect to the database, as this is the most common way to link databases to websites and to display information stored in a database on a website.

We used HTML5 and CSS to build the backbone of the site and make the pages presentable. These are the standard markup and style sheet tools for creating web pages and applications.

We utilized JavaScript—along with JS libraries such as JQuery—for event handling. This made the site more usable and interactive. Like HTML and CSS, JavaScript is considered a core technology for web development and is supported by modern web browsers without needing additional plug-ins.

Chapter 7

Description of System Implemented

We were able to implement all of the critical functional requirements as well as some recommended functional requirements such as Edit Profile.

Administrators and professors can invite new users (Figure 7.1). New users use the provided permission code to create an account (Figure 7.2).

Users can login via the login page (Figure 7.3). If the user is an administrator, they will be taken to the administrator home page (Figure 7.4). Professors will be taken to the professor home page (Figure 7.5), and TAs will be taken to the TA home page (Figure 7.6).

All users are able to edit their profiles by clicking the "Edit Profile" button in the top right corner, which will bring them to the Edit Profile page (Figure 7.7).

TAs can add new events to their calendars by clicking the "Add Event" button on their home page. A pop-up appears after clicking the "Add Event" button, and here the TA can input details of the new event to be added (Figure 7.8). On the calendar, TAs can find substitutes by clicking on a TA event block and then clicking "Find Replacement" on the pop-up menu (Figures 7.6 and 7.9). TAs can navigate through weeks by clicking "Next Week", "Previous Week", or "Current Week" (Figure 7.6).

Using our system, TAs can more easily find replacements for classes they teach that they suddenly cannot attend.

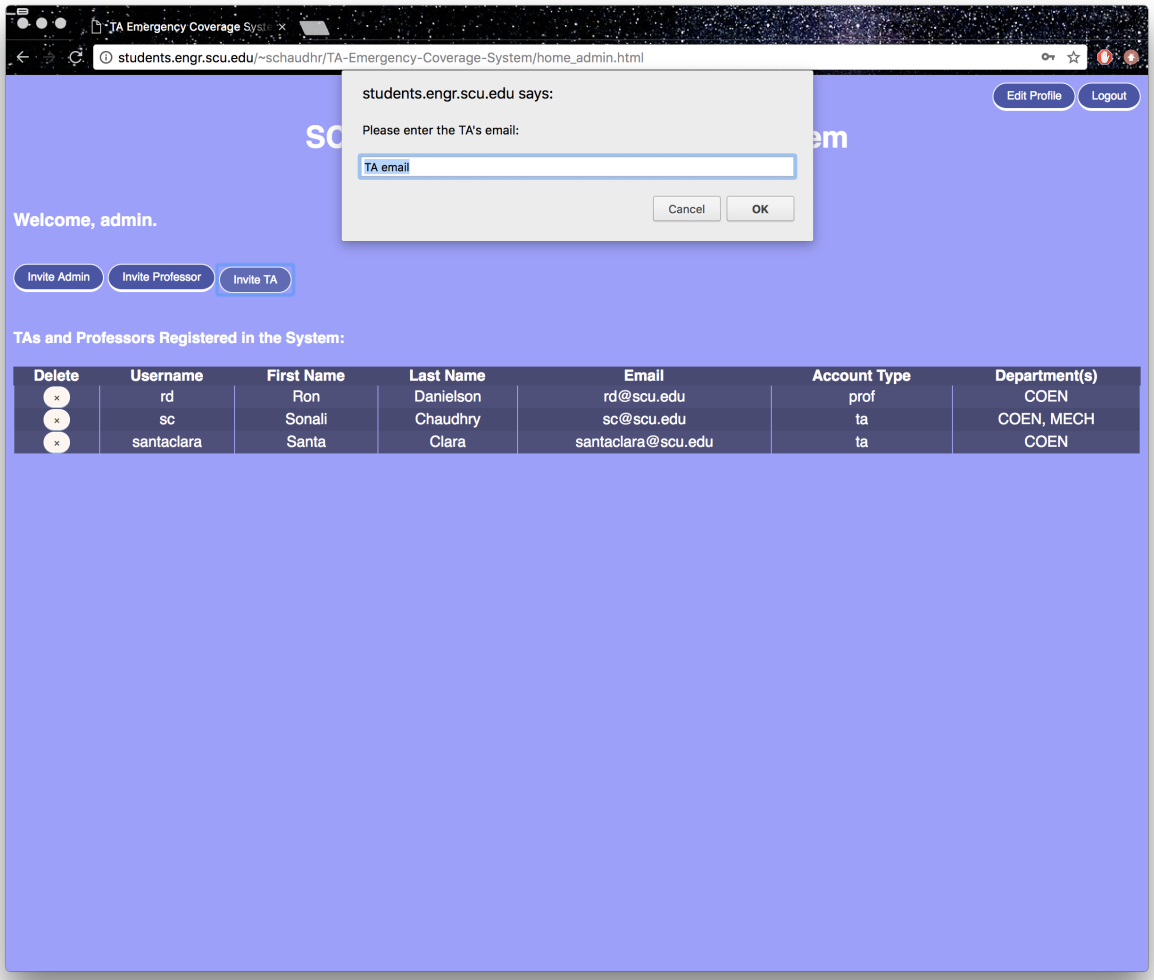

Figure 7.1: Inviting a New User

Figure 7.2: The Account Creation Page

Figure 7.3: The Login Page

Figure 7.4: The Administrator Home Page

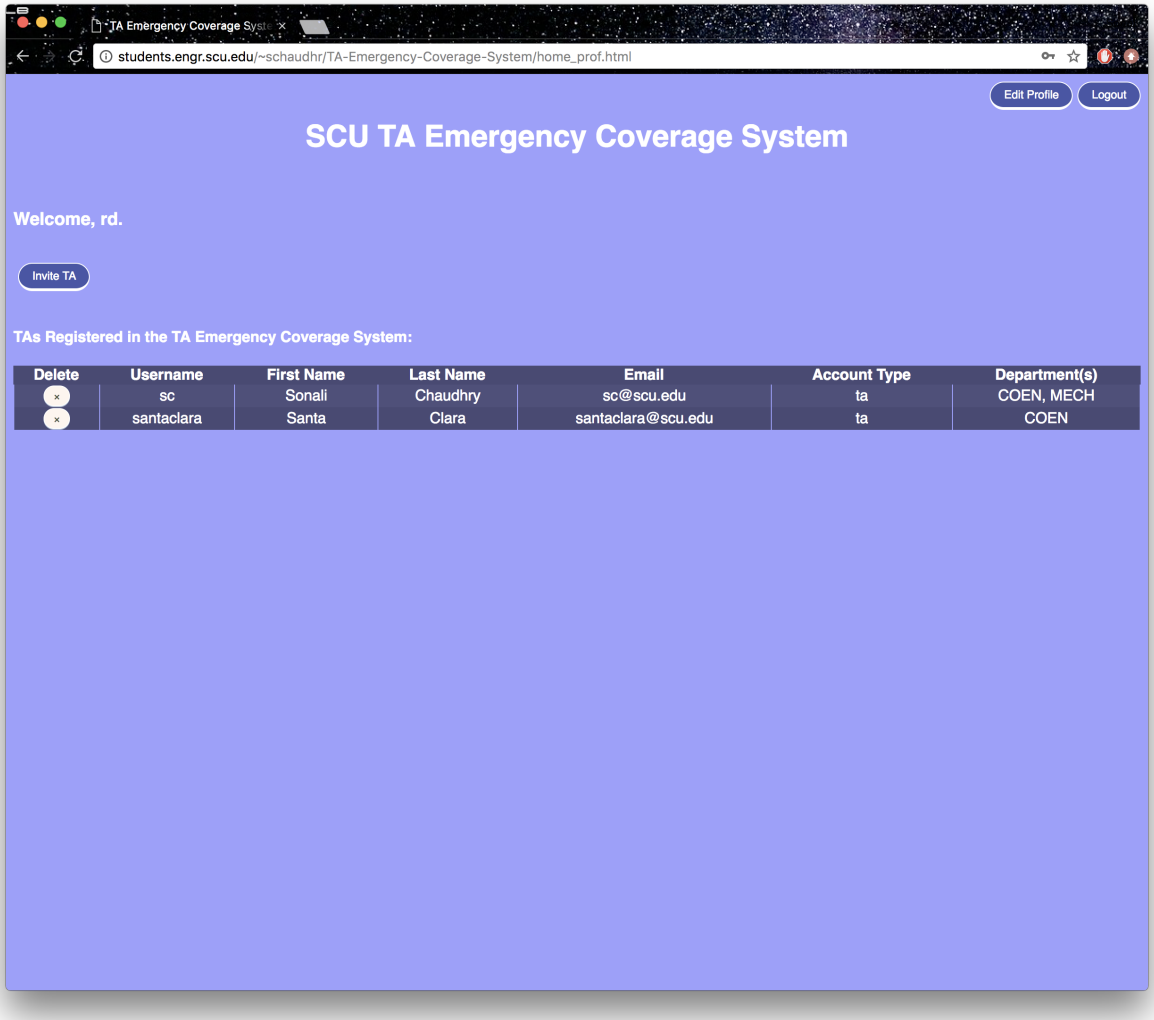

Figure 7.5: The Professor Home Page

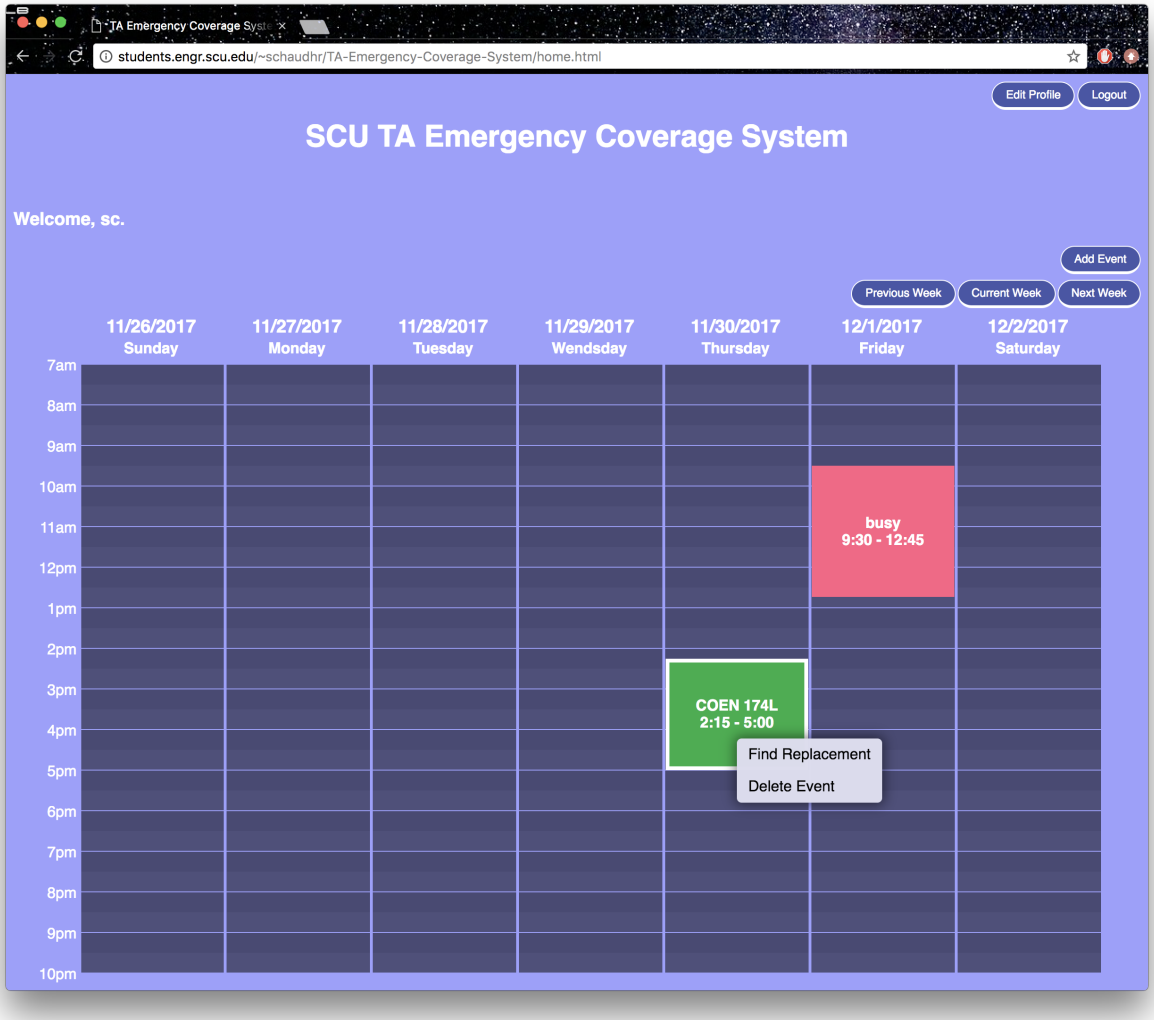

Figure 7.6: The TA Home Page

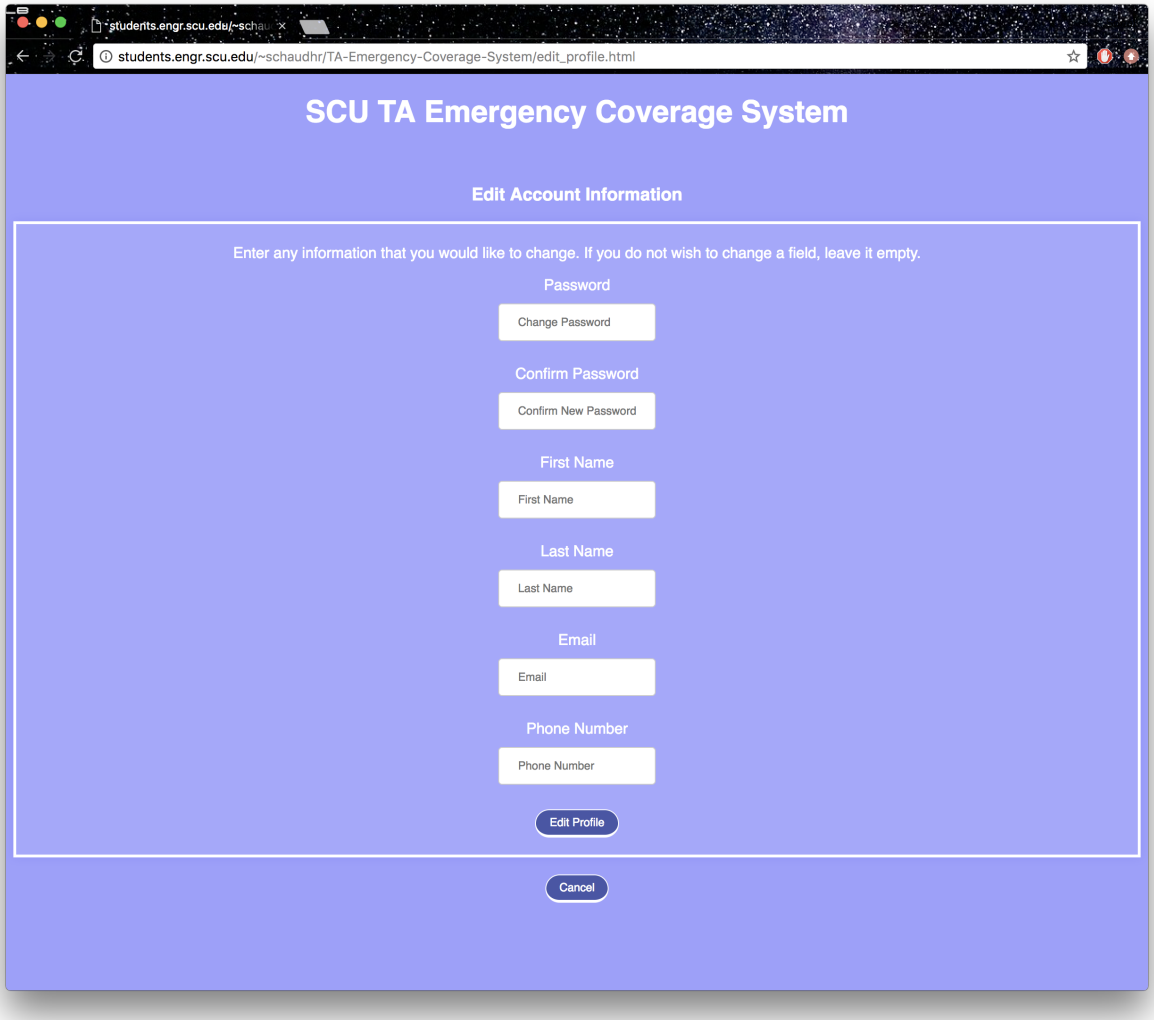

Figure 7.7: The Edit Profile Page

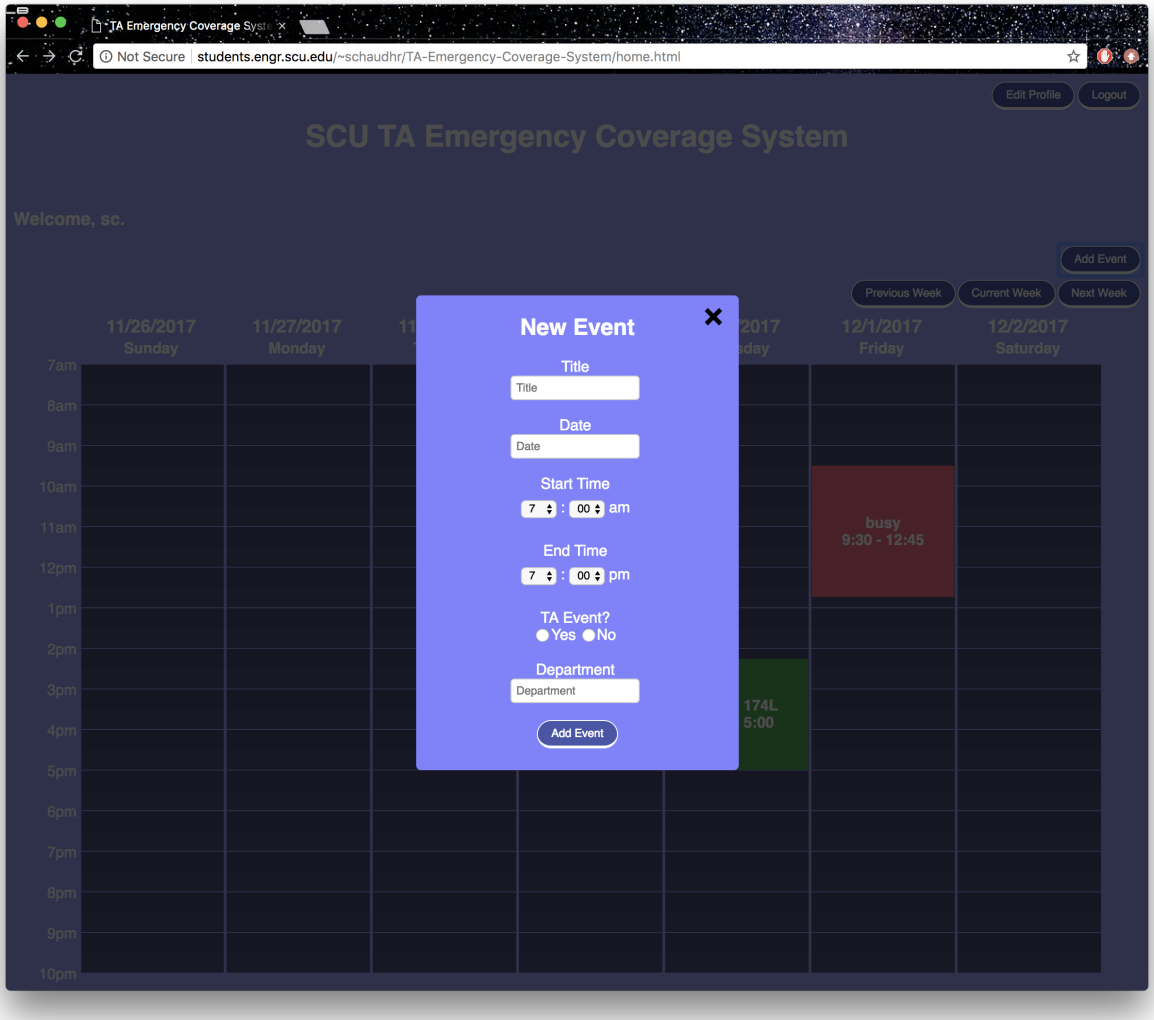

Figure 7.8: The Add Event Pop-Up

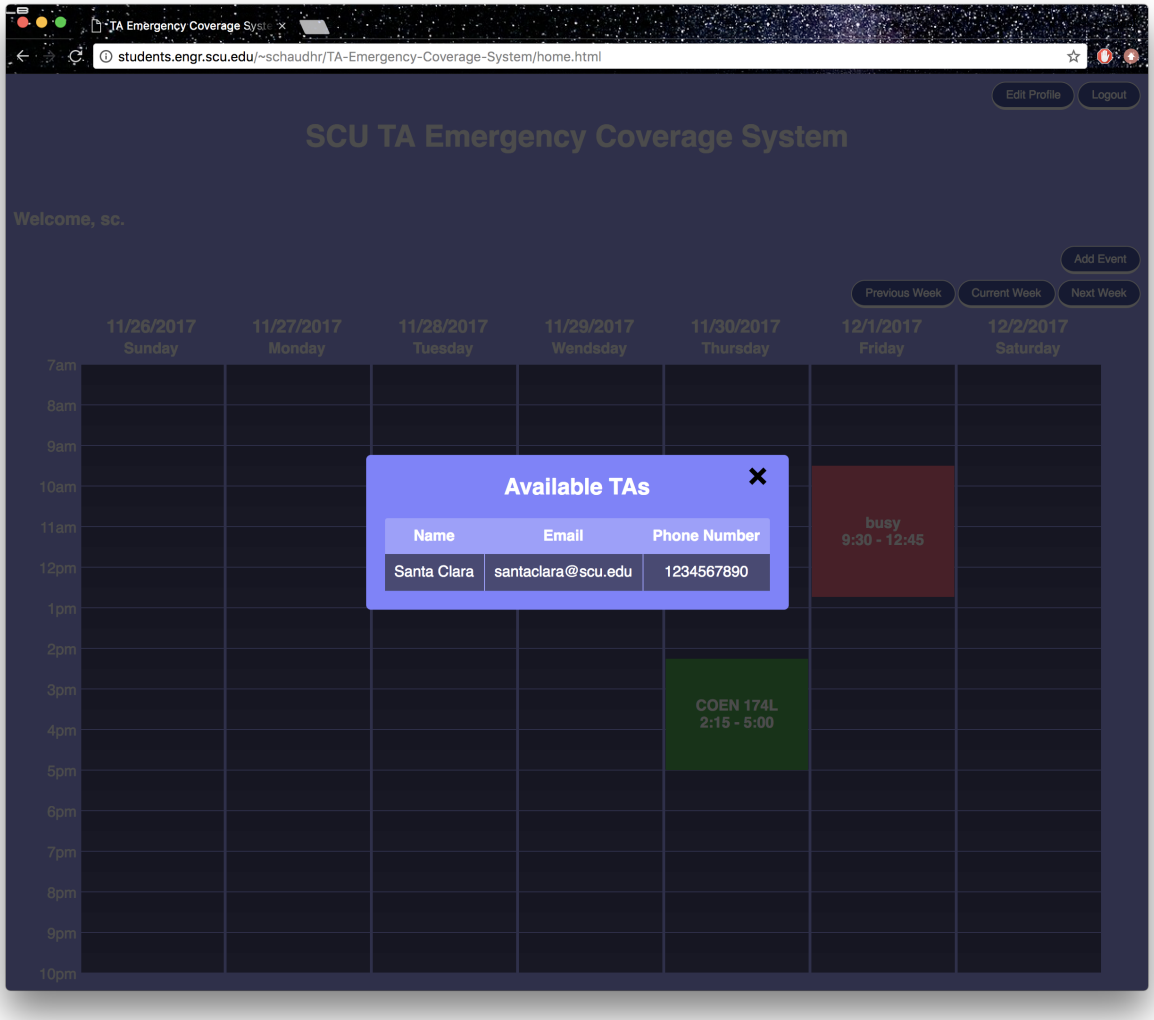

Figure 7.9: The Find Replacement Pop-Up

Chapter 8

Suggested Changes to Implemented System

If given more time on this project we would like to add the following functionality:

- Professors are able to edit TAs' qualifications (i.e. Departments)
- Improved site security—especially for login and data storage.
- In "Add Event," make it possible to repeat events weekly or on multiple days.
- If an event has repeats, have the option to delete subsequent iterations of this event or only delete the clicked event.
- After a permission code for account creation has been used to create an account, we would like to have it expire.
- Allow TAs to edit exiting events.
- Allow TAs to click a button that automatically contacts available TAs.
- When inviting a new user, automatically email the invitee the permission code.

Chapter 9

Testing

9.1 Testing Methods

- White Box Testing
  - Unit Testing: Throughout development, we tested each JavaScript and PHP function individually to ensure valid unit functionality.
- Black Box Testing
  - System Testing: At the end of development, we tested each piece of the system's functionality, as detailed in Table 9.1.

9.2 Test Cases

At the end of development, we tested every piece of functionality, as shown in Table 9.1.

Table 9.1: All Test Cases

Feature	Test Procedure	Expected Outcome	Result
First admin creation	Sign up for a new account with the database empty	An admin account is created	Success
New user invite	Use the "Invite x" pages	Generate a permission code associated with an email. Check in the database.	Success
New user sign up	Sign up as a new user using a previously generated permission code.	Account is created and user is redirected to the login page	Success
Add a new event	Create multiple events at different times, with different repeat statuses some regular events and some TA events	All events are created successfully and behave appropriately	Success
Delete an event	Click and event an select "delete"	The event is deleted.	Success
Find replacement TAs.	Click a TA event and select "Find Replacement".	A list of available TAs and their contact information is displayed.	Success
Logout	Click the logout button	The user is return to the login page and cannot access their homepage until they log in again.	Success
Edit profile information	Click the Edit Profile button in the top right corner of the home page. Change all information, click "Edit Profile".	The user's information is changed.	Success
Delete a user as an admin or a professor.	Click the "x" next to a user's name on the home page and verify the deletion.	The user is removed from the system	Success

Chapter 10

Difficulties and Lessons Learned

10.1 Difficulties Encountered

We encountered the following difficulties:

- Delegating tasks during development was confusing because we did not assign tasks specifically enough while planning.
- Due to lack of communication with the client, we were unclear about how the client wanted specific features implemented. As a result we spent too much time on certain parts of the project and left little time for other important parts.

10.2 Experiences and Lessons Learned

- Need to assign actual tasks off the get go - delegating tasks was hard at times
- Web development is difficult
- Software engineering is a lot more than just coding

Appendix A

Installation Guide

A.1 Pre-Installation Requirements

Before installing, you will need:

- A web server with PHP CGI enabled
  - If you do not know how to set up PHP CGI, contact your school's technology help group. For Santa Clara University, this info can be found here:

    http://wiki.helpme.engr.scu.edu/index.php/Webpage#PHP-CGI.
- An Oracle SQL database server account with no tables named "Account" or "Permission"
  - If you do not have an Oracle database account, contact your school's technology help group or set one up yourself. You can find out how to set up an account here:

    http://wiki.helpme.engr.scu.edu/index.php/Oracle

A.2 Installation Process

Follow these steps to install the TA Emergency Coverage System.

1. Copy the entire contents of the installation flash drive onto your web server. (i.e. Move the folder "TA-Emergency-Coverage-System" into your root web server directory) If you need help setting up your SCU web directory click here: http://wiki.helpme.engr.scu.edu/index.php/Webpage

2. In the terminal, navigate into the directory into which you copied the files. Then navigate into the TA-Emergency-Coverage-System folder.

3. Change the permissions of the file set_perms.sh by typing "chmod 700 set_perms.sh" into Terminal and pressing Enter. This script is used to set the proper permissions for the rest of the files.

4. Run the shell script *set_ perms.sh* by typing "./set_perms.sh" into Terminal and pressing Enter.

5. Open *db_ config.php* in a text editor. Replace the example database address and credentials with those of your own Oracle SQL server. Refer to the Pre-Installation Requirements for help with setting up your Oracle database if you have not done so already.

6. You should now be ready to initialize the system. Your site will be accessible at:

   <YOUR_SERVER_ADDRESS>/TA-Emergency-Coverage-System/login.html

   For example, at SCU your site URL would look like: http://students.engr.scu.edu/~<USERNAME>/TA-Emergency-Coverage-System/login.html. Refer to Section B.2.1 of the User Manual in Appendix B for instructions on how to create the first user who will automatically be registered as a system administrator.

Appendix B

User Manual

B.1 Introduction

This document details the functionality and usage of the TA Emergency Coverage System.

B.2 Administrator Functionality

B.2.1 Creating the First Administrator

To create the first administrator for the system, simply create an account. The first account created in the system automatically becomes an administrator. In the "Permission Code" field, simply enter any text. After creating the administrator account, log in to access the administrator home page.

B.2.2 Signing up as an Administrator

To sign up as an administrator, click "sign up" from the login page. Enter your information, and in the "Permission Code" field, enter the code given to you by another system administrator.

B.2.3 The Administrator Home Page

After logging in to an administrator account, you will be taken to the administrator home page. From here, you can see a list of all professors and TAs in the system. You can also invite new TAs, professors, and administrators to the system.

B.2.4 Inviting New Users

To invite a new administrator, professor, or TA, simply click the corresponding "Invite" button in the top left corner of the administrator home page. A pop-up will appear, and in the prompt box you will enter the email address of the person you would like to invite and click "submit". A message will then appear with a permission code. Give this permission code to the person you are inviting, and when they sign up using the email you provided, they will use that permission code to create a valid account.

Note: If you re-invite a user using the same email, you will need to provide the invitee with the new permission code, as the old one will be invalid.

B.2.5 Deleting Users

To delete a user, click the "x" to the left of their name on the administrator home page. Then confirm that you are sure you would like to delete this suer, and click "Okay" on the box that confirms the user has been deleted.

B.2.6 Editing Your Profile

To edit your account information, click the "Edit Profile" button in the top right corner of your home page. Change the information you would like to change and click "Edit Profile". Leave fields blank if you do not wish to change the corresponding information. Click "cancel" to return to the previous page.

B.3 Professor Functionality

B.3.1 Signing Up as a Professor

To sign up as a professor, click "sign up" from the login page. Enter your information, and in the "Permission Code" field, enter the code given to you by your system administrator.

Note: You must sign up with the same email address with which you were invited.

B.3.2 The Professor Home Page

After logging into a professor account, you will be taken to the professor home page. Here, you can see a list of all TAs in the system. You can also invite new TAs to the system.

B.3.3 Inviting New TAs

To invite a new TA, click the "Invite TA" button in the top left corner of the professor home page. A pop-up will appear prompting you to enter the email address of the TA you would like to invite. Click "Submit" once you've entered this email. A message will appear with a permission code. Give this permission code to

the TA you are inviting. When they sign up using the corresponding email, they may use that permission code to create a valid account.

Note: If you invite a TA using an email that has previously been invited to the system, you will need to provide the invitee with the new permission code, as the old one will be invalid.

B.3.4 Deleting TAs

To delete a TA, click the "x" to the left of their name on the professor home page. Then confirm that you are sure you would like to delete this suer, and click "Okay" on the box that confirms the user has been deleted.

B.3.5 Editing Your Profile

To edit your account information, click the "Edit Profile" button in the top right corner of your home page. Change the information you would like to change and click "Edit Profile". Leave fields blank if you do not wish to change the corresponding information. Click "cancel" to return to the previous page.

B.4 TA Functionality

B.4.1 Signing Up as a TA

To sign up as a TA, click "Sign Up" on the login page. Enter your information, and in the "Permission Code" field, enter the code given to you by your system administrator or professor.

Note: You must sign up with the same email address with which you were invited.

B.4.2 The TA Home Page

After logging in to a TA account, you will be taken to your TA homepage. This page allows you to view and edit your weekly calendar. Here, you can find replacement TAs to fill in for labs you cannot be present for.

B.4.3 Creating a New Event

Events in your calendar determine when you are and are not available. Looking for replacements to fill in for a lab section you TA for will only list TAs who are available at that time.

To add a new event to your schedule, click the "Add Event" button on the right above the weekly schedule on the TA home page. This will open the new event interface. Enter the name, date, start, and end times of the event. Finally, decide if this is a TA event or a regular event. A regular event is one that you are not a TA for, and this will appear as a pink block on your calendar. A TA event is one that you are a TA for a lab, and this event will appear as green on your calendar. TA events can be selected to find replacement TAs, while regular events cannot.

Note: TA events for which you would like to be able to find a replacement should be named starting with the lab's department code, i.e. COEN 174L

B.4.4 Deleting an Event

To delete an event from your calendar, click on the specific event block you would like to delete and select "Delete" from the pop-up menu. This will remove the one event you selected.

B.4.5 Finding a Replacement TA

To find a replacement TA for your lab, click the TA event for which you need the replacement. Click "Find Replacement" on the pop-up menu. This will display a list of TAs in the system who are both qualified to teach that lab and are available at that time. You can use the listed TAs' contact information to contact them directly and determine if they would be willing to fill in for you. If no TAs are listed as available at this time, contact your professor.

B.4.6 Editing Your Profile

To edit your account information, click the "Edit Profile" button in the top right corner of your home page. Change the information you would like to change and click "Edit Profile". Leave fields blank if you do not wish to change the corresponding information. Click "cancel" to return to the previous page.