COEN 174 Design Document

by

Ruiwen Li
Zheqing Li
Stephen Chuang

Submitted in partial fulfillment of the requirements
for the degree of
Bachelor of Computer Science and Engineering
School of Engineering
Santa Clara University

Santa Clara, California
October 11, 2017

COEN 174 Design Document

Ruiwen Li
Zheqing Li
Stephen Chuang

Department of Computer Engineering
Santa Clara University
October 11, 2017

ABSTRACT

Teaching assistants of lab sessions may sometimes become ill or encounter emergencies in conflict with the lab session, but there isn't an efficient way for professors and TAs to communicate for finding a substitute. We designed a web-based application to provide user a list of available TAs to facilitate the process of finding a suitable substitute to cover the lab session. The TAs will login to the system and provide availability schedules. Both TAs and professors will provide a list of lab sessions they covered this quarter and their contact information. All information is stored in a database. Our system has high reliability, security, and performance, supporting both Chrome and Firefox on Linux operating system. The application effectively saves time in communication.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Teaching assistants of lab sessions may sometimes become ill or encounter emergencies that conflict with the lab session they are in charge of, therefore becoming unable to cover their assigned labs. When these situations arise, Santa Clara University lacks a way for TAs and professors to communicate with each other efficiently in case coverage is needed for a lab session. The current solution is merely for the particular case at hand and without consideration of wider applications. Whoever is looking for a substitute does not know the schedule availability of others beforehand, and thus would have to contact each TA one by one to check if others are available. This method of communication is very time consuming, which would affect the progress of the lab session and the students lab attendance if a substitute has not been found.

1.2 Solution

We present a web-based application that solves this problem by providing a list of available TAs for the TA who needs a substitute or the professor in charge, so that they do not have to waste time looking for substitutes themselves. They can then contact each TA according to the list provided, and ask them if they want to cover the lab session. The potential users of our application are teaching assistants and professors in charge of lab sessions.

Chapter 2

Requirements

2.1 Functional Requirements

- Critical:
    - A sign-up and login system for teaching assistants and professors
    - Collect schedules of all teaching assistants and store in text format
    - Collect contact information of all users
    - Store all information in a database for future retrieval
    - Users can query the system to get a list of available TA's names during the time period of a specific lab
- Recommended:
    - When there is no available TA during the requested time, the system will display a warning message to the user and send an email to the professor in charge
    - The system indicates which professor is in charge of each TA's lab session
- Suggested:
    - Provide available TA's contact information, such as email addressed and cellphone numbers to facilitate users' efforts to reach out to them
    - Notify the user if the message has been sent or not

2.2 Non-functional Requirements

- Critical:
    - Security: the login system requires SCU school email, so the TAs' and professors' information will be protected
    - Reliability: support multiple queries in case more than one TA needs substitutes
- Recommended:
    - Usability: when finding the replacement, other TAs who teach the same course might be listed at the beginning, then branching out from there
    - Performance: the software should provide quick responses
- Suggested:
    - Visualization: The theme color of the user interface will be consistent with the main color of Santa Clara University, which is dark red.

2.3 Design Constraints

- Web-based software
- Support Linux operating system
- Work on both Chrome and Firefox
- Finish by the end of 2017 Fall Quarter of Santa Clara University

Chapter 3

Use Cases

The use case represents the list of actions and event steps which define the interactions among TAs, professors, and the system.

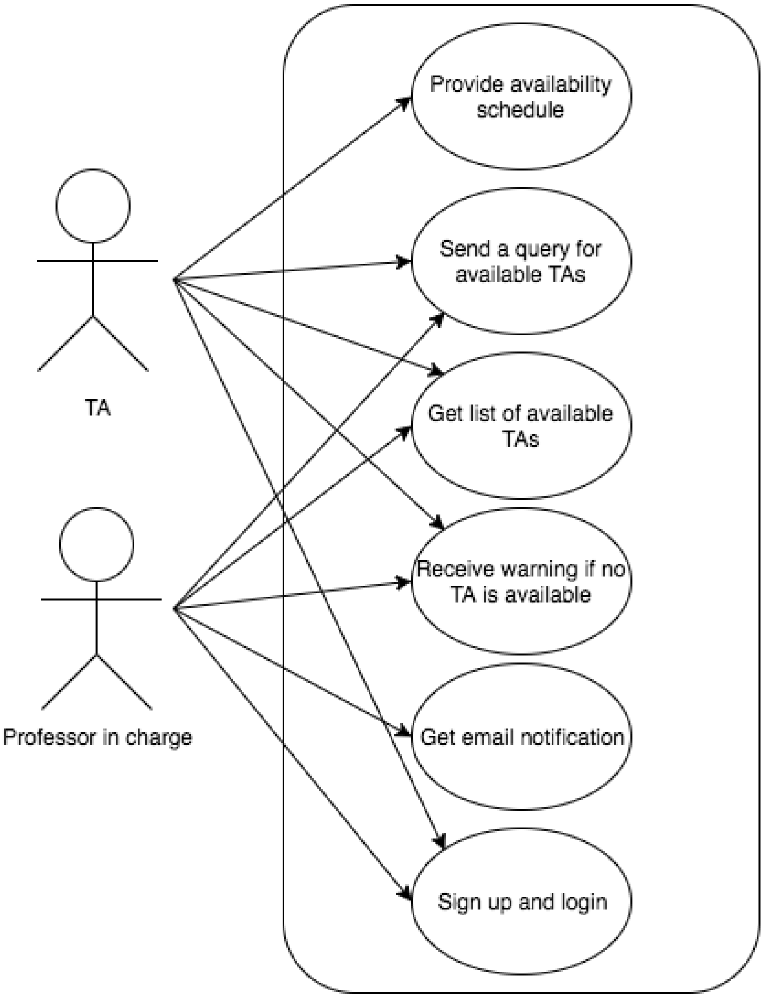

Provide availability schedule
Send a query for available TAs
Get list of available TAs
Receive warning if no TA is available
Get email notification
Sign up and login
TA
Professor in charge

Figure 3.1: Use case diagram of the system

Table 3.1: User behavior description

	Sign up and login	**Provide availability schedule**	**Send a query for available TAs**	**Get list of available TAs**	**Receive warning**	**Get email notification**
Goal	Provide identification of each user	Collect the availability information of all TAs	Tell the system to retrieve data	Know which TA is available during each lab session	Notify the user if no TA is available	Notify the professor in charge if no TA is available
Actor	All Users	Teaching assistants	TA who needs a substitute and the professor in charge			The professor in charge
Preconditions	User must open the web page	TAs are logged in	User has logged into the system	The user has sent the query to the system		The TA queries the system and there is no available TA
Steps	User creates a username and password	TAs fill in the their available times	User chooses the lab session that need to be covered and click the button	User reads through the list provided on the screen	User reads the warning message	A warning email is sent to the professor in charge
Postconditions	Users are logged in	TAs availability schedule is stored in the database	The query is sent to the system	The user knows which TA to contact	User potentially finds a different solution	The professor in charge receives the email notification and finds another solution
Exceptions	N/A	If a TA is not available at all times, leave the table blank	The user needs to resend the query if the query failed	No TA is available	N/A	N/A

Chapter 4

Activity Diagrams

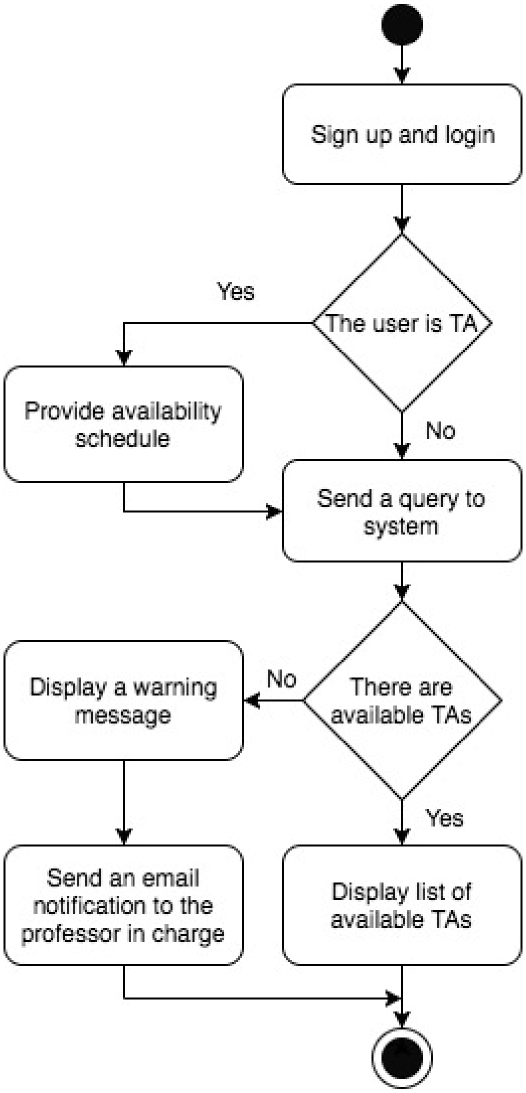

Sign up and login
Yes
The user is TA
Provide availability schedule
No
Send a query to system
Display a warning message
No
There are available TAs
Yes
Send an email notification to the professor in charge
Display list of available TAs

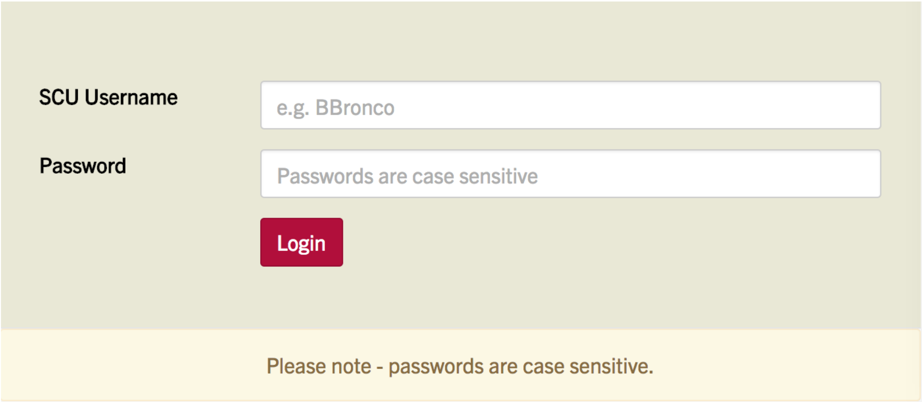
Figure 4.1: High level view of user activities

Chapter 5

Conceptual Model

5.1 Login page

The login page (See figure 5.1) is very similar to the SCU login page.



Please note - passwords are case sensitive.

Figure 5.1: Signin page of the user

5.2 Information page

After the TA logged in, he or she will fill out the information page (See figure 5.2) with basic contact information, lab session, and availability schedule.

5.3 User page

5.3.1 TA

The user interface of the TA (See figure 5.3) consists of three parts: personal profile, lab session, and available TAs. The personal profile allows the TA to view and edit his or her schedule, and change the lab session if necessary. The lab session part displays all the lab sessions that the TA covers this quarter, and the request for substitution button is bellow every lab session. The most right part is used to display the list of available TAs and contact information when a request button is clicked.

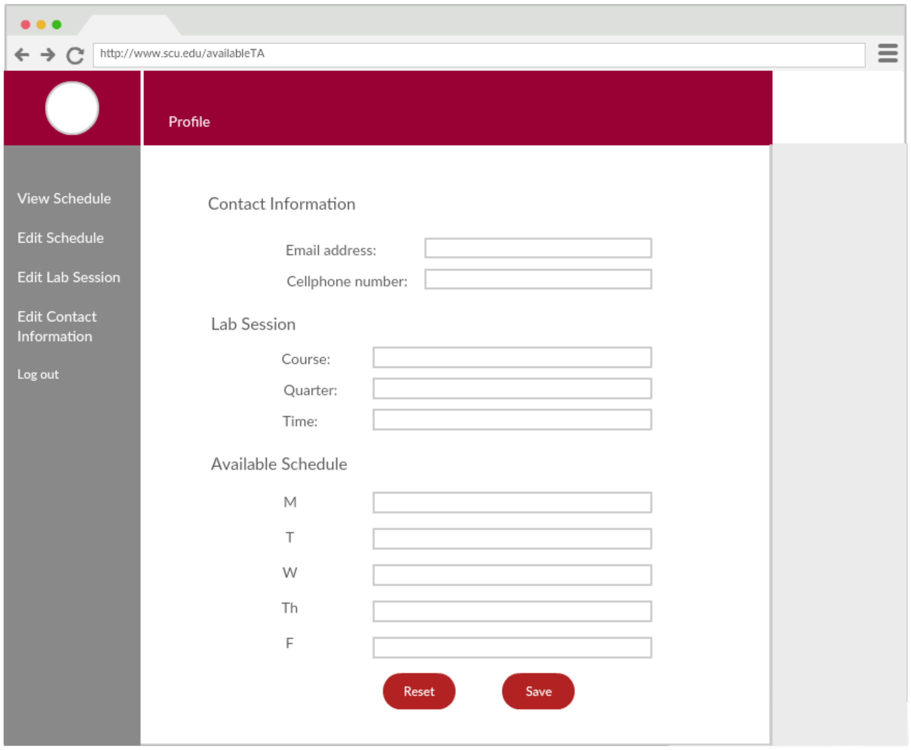


Figure 5.2: Information page of the TA

5.3.2 Professor in charge

The user interface of the professor (See figure 5.4) in charge is really similar to the TAs', except that the professor does not have availability schedule and cannot edit the schedule.

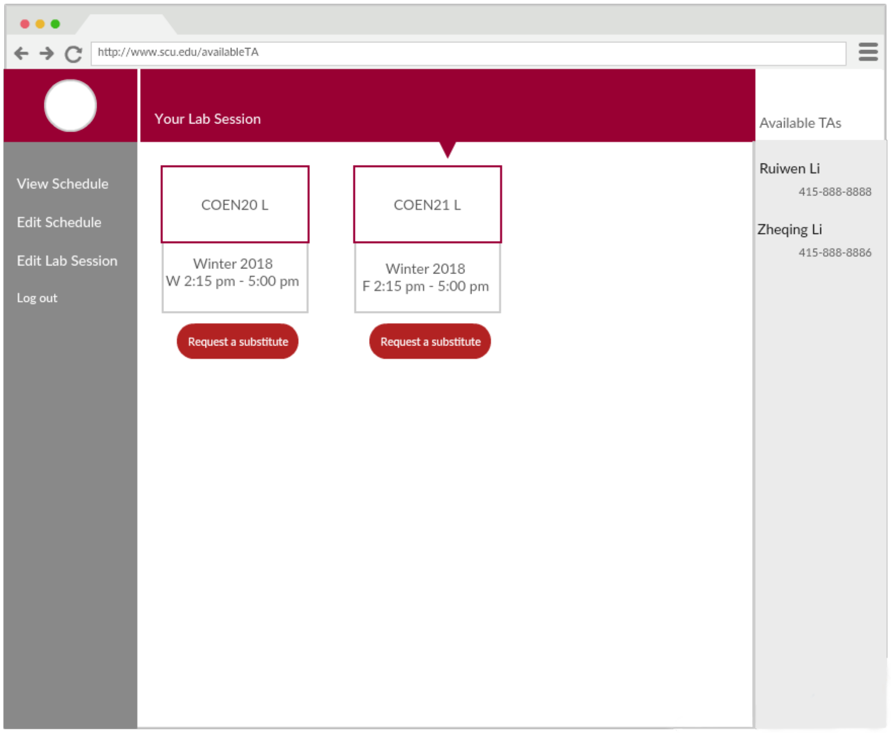


Figure 5.3: User page of the TA

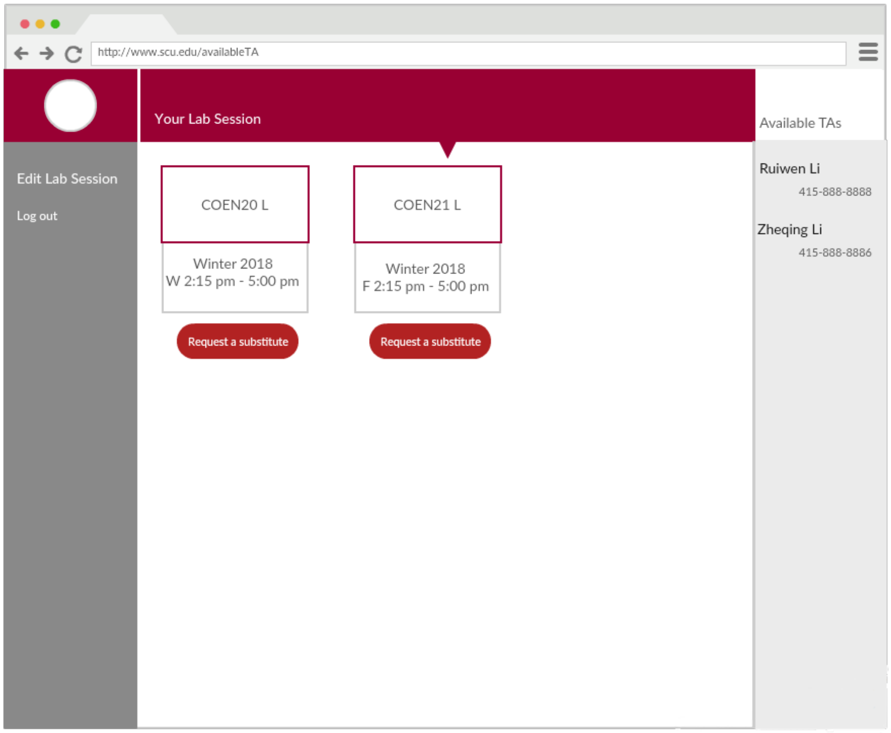


Figure 5.4: User page of the professor in charge

Chapter 6

Architecture

6.1 Design rationale

We decided to use a data-centric model to construct our software because nearly every action throughout the process needs the data retrieved from the database. For example, the usernames and passwords are stored in the database, along with TAs' availability schedules and their contact information. When the user sends a query to the system, the system searches through the database to find available TAs and returns their names and contact information.

6.2 Architectural diagram

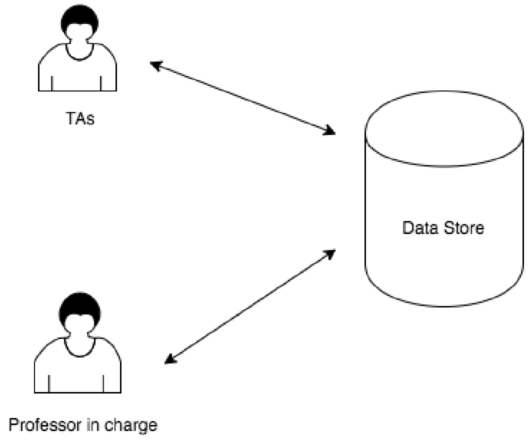


Figure 6.1: Architecture diagram of the system

Chapter 7

Technology Used

7.1 Programming Languages

- HTML5

HTML5 is used to create documents on the web page. It defines the structure and layout of a Web document by using a variety of tags and attributes.

- CSS3

CSS3 is used to describe the presentation of Web pages. It also makes the web page responsive to different devices.

- PHP

PHP is a general-purpose scripting language for server-side web development. It generally runs on a web server to create dynamic web page content or dynamic images.

- JavaScript

JavaScript is used as a client side scripting language. Its code is written into an HTML page. When a user requests an HTML page, the script is sent to the browser.

- SQL

SQL is used to communicate with a database. As the standard language for relational database management systems, SQL statements are used to perform tasks such as update data on a database, or retrieve data from a database.

7.2 Applications

- Github

Github is used for version control of the project. It allows programmers to update, revise, or restore a history version of the project. It also facilitates team collaboration on a project.

- MySQL

MySQL is an open source Relational Database Management System. It is used to store all the information of the users.

Chapter 8

Test Plan

8.1 Alpha

8.1.1 White Box

For the white box testing, we will test each portion of the code to ensure that they function as intended. We will design a series of varied test cases to use on the system to verify that all of the constraints are met in any situation. The test cases will assert the system's ability to maintain its functions in standard use situations, as well as potential edge cases.

8.1.2 Black Box

Black box testing will enlist the help of individuals not involved with the development of our application, which will likely include classmates and friends. They will understand what the system is designed to do, but not how it is done. The testers will then use the system as if they were a professor and his/her lab TAs, attempting to cover all possible scenarios.

8.2 Beta

Finally, our beta test will enlist the help of a Santa Clara University engineering professor and his or her TAs to test the system and confirm that it works in a real life scenario. While the program should be mostly tested and confirmed to work as intended by this point, we will continue to monitor its performance as there is always the possibility of new issues appearing during live use over an extended period of time.

Chapter 9

Risk Analysis

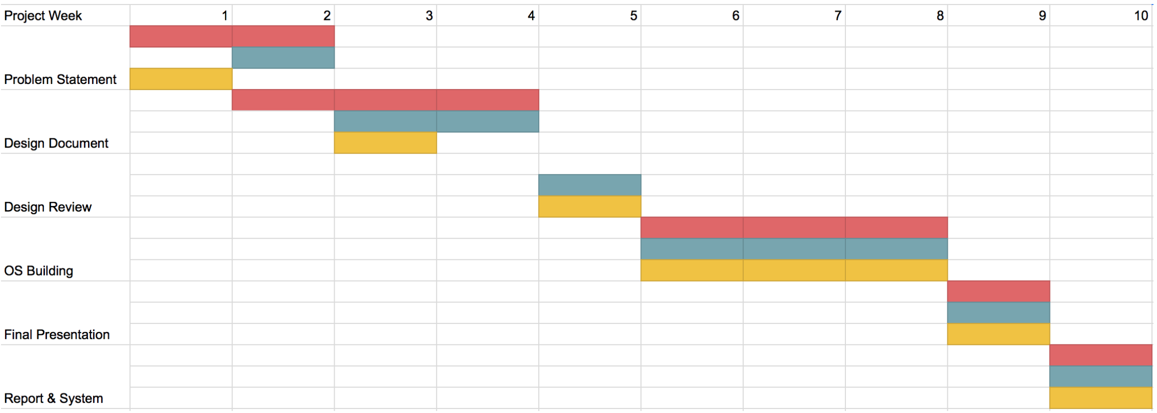
Table 9.1: Risk table

Risk	Consequences	Probability	Severity	Impact	Mitigation
Message fails to send to available TAs successfully	User can not inform others	0.3	10	3	Implement a high response system
Long response time in searching	User can not get list of available TAs	0.2	9	1.8	Make the database system easy to use and quick to respond
Failure to inform the professor when no available TA is found	There will be no TA in the lab	0.1	10	1	Have an acknowledgement to confirm that the professor has been notified
Long response time in sending messages	The communication process will be longer	0.4	2	0.8	Implement the system with quicker response time

Chapter 10

Development Timeline

The Development Timeline shows the project timeline in 10 weeks. Red is Ruiwen Li's contribution. Blue is Zheqing Li's contribution, and yellow is Stephen Chuang's contribution.



Project Week	1	2	3	4	5	6	7	8	9	10
	🟥	🟥								
		🟦								
Problem Statement	🟨									
		🟥	🟥	🟥						
			🟦	🟦						
Design Document			🟨							
					🟦					
Design Review					🟨					
						🟥	🟥	🟥		
						🟦	🟦	🟦		
OS Building						🟨	🟨	🟨		
									🟥	
									🟦	
Final Presentation									🟨	
										🟥
										🟦
Report & System										🟨

Figure 10.1: Development Timeline