Santa Clara University
Scholar Commons

Computer Engineering Senior Theses | Engineering Senior Theses

6-4-2019

Promarc: An Online Skills and Projects Marketplace

Connor Carraher

Victor Yu

Follow this and additional works at: https://scholarcommons.scu.edu/cseng_senior

Part of the Computer Engineering Commons

Recommended Citation

Carraher, Connor and Yu, Victor, "Promarc: An Online Skills and Projects Marketplace" (2019). *Computer Engineering Senior Theses*. 147.
https://scholarcommons.scu.edu/cseng_senior/147

This Thesis is brought to you for free and open access by the Engineering Senior Theses at Scholar Commons. It has been accepted for inclusion in Computer Engineering Senior Theses by an authorized administrator of Scholar Commons. For more information, please contact rscroggin@scu.edu.

SANTA CLARA UNIVERSITY
DEPARTMENT OF COMPUTER ENGINEERING

Date: June 7, 2019

I HEREBY RECOMMEND THAT THE THESIS PREPARED UNDER MY SUPERVISION BY

Connor Carraher
Victor Yu

ENTITLED

Promarc: An Online Skills and Projects Marketplace

BE ACCEPTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

BACHELOR OF SCIENCE IN COMPUTER SCIENCE AND ENGINEERING

[illegible]
Thesis Advisor

[illegible]
Department Chair

Promarc: An Online Skills and Projects Marketplace

by

Connor Carraher
Victor Yu

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

Santa Clara, California
June 4, 2019

Promarc: An Online Skills and Projects Marketplace

Connor Carraher
Victor Yu

Department of Computer Engineering
Santa Clara University
June 4, 2019

ABSTRACT

Technical projects can vary greatly in terms of cost, complexity, and time. Project leads spend a lot of valuable time and energy making sure that their teams are organized and on-task. A major part of their responsibilities includes putting together a team with the right skills in order to maximize efficiency. Having a platform where project leads can quickly find team members with the right skills would save them a lot of stress and trouble. The goal of this project is to deliver such a platform, where users can make posts about their projects and the technical skills that they require, and be connected to an entire network of potential viable team members. Our system consists of a web application connected to a database backend, accessible through different interfaces depending on the credentials of the user. This report will also provide an in-depth analysis on the systems requirements specifications, use cases, data flow, involved actors, architecture, testing procedures, risk analysis, development timeline, final results, and societal impact.

Table of Contents

List of Figures

Chapter 1

Introduction

University campuses are often locations for research and innovation. Students, professors, researchers, entrepreneurs, and many others often form groups and teams to do large, intricate projects. However, one of the most difficult obstacles they might encounter is finding team members equipped with the proper skills. It is unreasonable to expect a team leader or project manager to commit the time needed to learn all of the required skills for a complex project with a large scope. Some direct consequences of this issue are that projects may never take off and productivity may be severely limited. Teams that are ill-equipped with the proper skills may immediately abandon their idea, sacrifice quality in order to learn the required skills, or allow their current skill-set to dictate the technologies of the project. Currently, project teams attempt to solve this problem by word of mouth, social media advertisements, and bulletin board postings around campus. Providing such an online platform would enable projects to come to fruition more easily and often, as well as accelerate research and innovation.

Our solution is a web-based forum and marketplace for people on university campuses to advertise their projects and receive responses from potential team members with relevant skills. Three types of people will use our application: posters - users that create a forum post describing their project, and what skills they will require from team members to complete it; providers - users that have a needed skill for a project and can respond to posts when they see a potential match; and moderators - administrators that are responsible for maintaining the integrity of the online community.

Users that are interested in providing their services and learning more about a specific project will be able to communicate with the poster using the web application. Once a provider and poster have deemed they are a good fit, they can exchange contact information for further communication and project development. To add another inherent layer of security and moderation to the online community, prospective users must register and authenticate their account with a .edu e-mail. We strongly believe that a web application dedicated to connecting an innovative and forward-thinking community such as a college campus would lead to an increase of successful projects and academic endeavours. Our system would help to eliminate the postings in Starbucks and stairwells and instead allow teams to focus on what really matters: getting their projects done.

Chapter 2

Requirements

Requirements are specified before the system is implemented in order to establish clear goals for the developers and make sure the client can keep the developers accountable. **Functional** requirements are specific behaviors that must be implemented in order for the system to fulfill its purpose. **Non-functional** requirements are criteria that can be used to judge a system's performance. Finally, constraints are restrictions to which the implementation must adhere.

2.1 Functional

- ESSENTIAL: The system will allow a user to submit a post.
- ESSENTIAL: The system will allow a user to view other posts.
- RECOMMENDED: The system will allow providers and posters to message one another.
- RECOMMENDED: The system will allow moderators to remove inappropriate posts.
- RECOMMENDED: The system will allow users to log in using an SCU email address.

2.2 Non-functional

- RECOMMENDED: The system will be learnable and easy to use.
- RECOMMENDED: The postings list in the home view will be streamlined and intuitive.
- RECOMMENDED: The system will quickly notify a user if they have received a message.

2.3 Design Constraints

- The system will be accessible via Internet browser.
- The system must allow any SCU user to view and submit posts.

Chapter 3

Use Cases

In Promarc, there are three main types of actors. Posters and providers both fall under the category of ”general user.” General users are allowed to view and create posts, and are allowed to message back and forth within the system. Posters and providers are only differentiated by the fact that the former is requesting a skill that the latter has. Moderators are administrators with the power to delete any posts they deem inappropriate, and they are responsible for maintaining integrity and quality in the community.

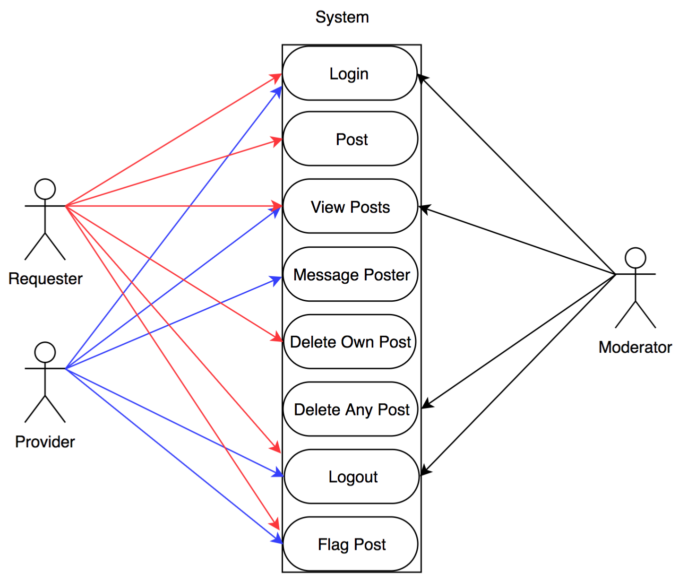


Figure 3.1: Use Case Diagram

Login

- Goal: Login to the system
- Actors: Poster, Provider, Moderator

- Pre-conditions: SCU account exists
- Post-conditions: User is logged in and a session has begun
- Steps:
  - Enter correct username and password
  - Submit information
- Exceptions: Incorrect account information

Post

- Goal: Allow the poster to create a posting which describes the skill necessary to complete the project
- Actors: Poster
- Pre-conditions: Poster is logged in
- Post-conditions: Post is created and entered into database
- Steps:
  - Enter ”create post” view
  - Enter the requested information
  - Submit information
- Exceptions: None

View Posts

- Goal: Allow any user to view the information submitted by the poster
- Actors: Poster, Provider, Moderator
- Pre-conditions: User is logged in
- Post-conditions: UI is showing detailed post view
- Steps:
  - View the home page
  - Select post

- Exceptions: None

Message Poster

- Goal: Allow a provider to contact the poster
- Actors: Provider
- Pre-conditions: Provider is logged in
- Post-conditions: Sent message is in poster's inbox
- Steps:
  - Select post
  - Enter message view
  - Compose message
  - Send message
- Exceptions: None

Delete Personal Post

- Goal: Allow a user to delete a post which they have previously posted
- Actors: Poster
- Pre-conditions: Poster is logged in
- Post-conditions: Post is deleted and removed from database
- Steps:
  - View the home page
  - Select post
  - Delete post
  - Confirm post deletion
- Exceptions: None

Delete Any Post

- Goal: Allow a user to delete any post regardless of who posted it

- Actors: Moderator
- Pre-conditions: Moderator is logged in
- Post-conditions: Post is deleted and removed from database
- Steps:
  - View the home page
  - Select post
  - Delete post
  - Confirm post deletion
- Exceptions: None

Logout

- Goal: Log out of the system
- Actors: Poster, Provider, Moderator
- Pre-conditions: User is logged in
- Post-conditions: User is logged out and the session has ended
- Steps:
  - Select the ”logout” button
- Exceptions: None

Flag Post

- Goal: Flag a post
- Actors: Poster, Provider
- Pre-conditions: User is logged in
- Post-conditions: Post is flagged for review by a Moderator
- Steps:
  - Select the ”Flag as Inappropriate” button
- Exceptions: None

Chapter 4

Activity Diagram

Activity diagrams are a visual flowchart depicting how activities flow from one part of the system to another. Figures 4.1, 4.2, and 4.3 provide an overview of how Promarc operates as a system.

The figures show how each actor will interact with the system, with each diagram being dedicated to a specific actor. The activity flow begins at the top black circle and ends at the bottom black circle.

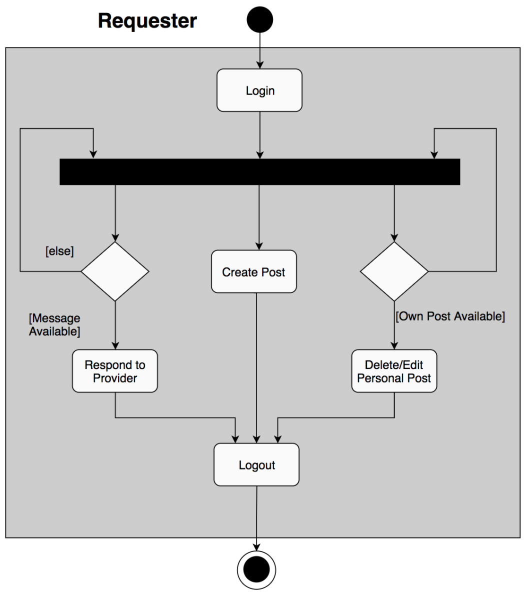


Figure 4.1: Activity Diagram: Poster

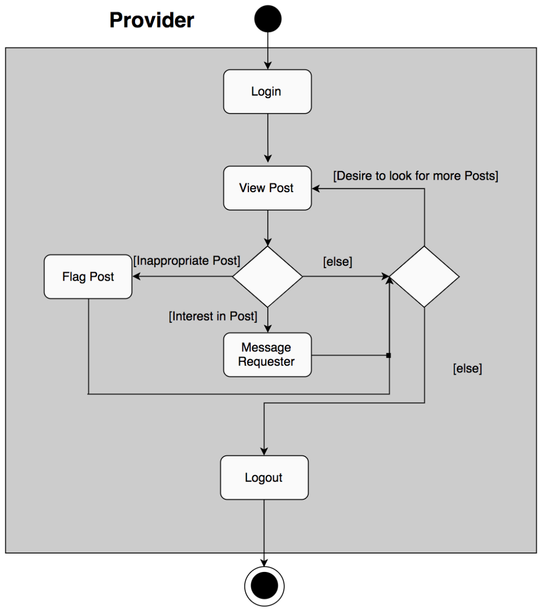


Figure 4.2: Activity Diagram: Provider

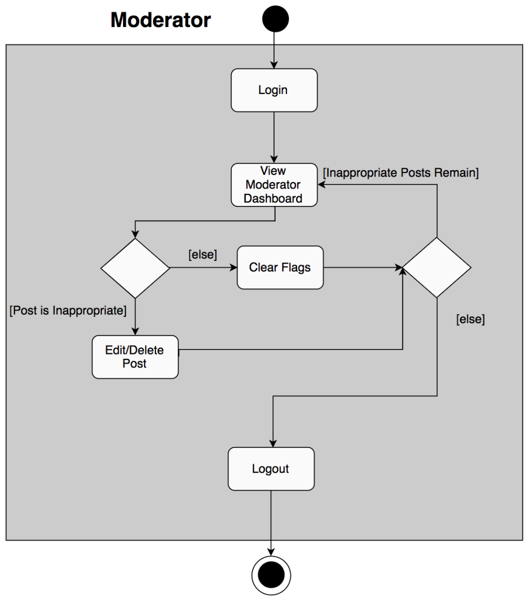


Figure 4.3: Activity Diagram: Moderator

Chapter 5

Conceptual Models

The conceptual model mock-ups for the system are included below. The mock-ups show the user interface views.

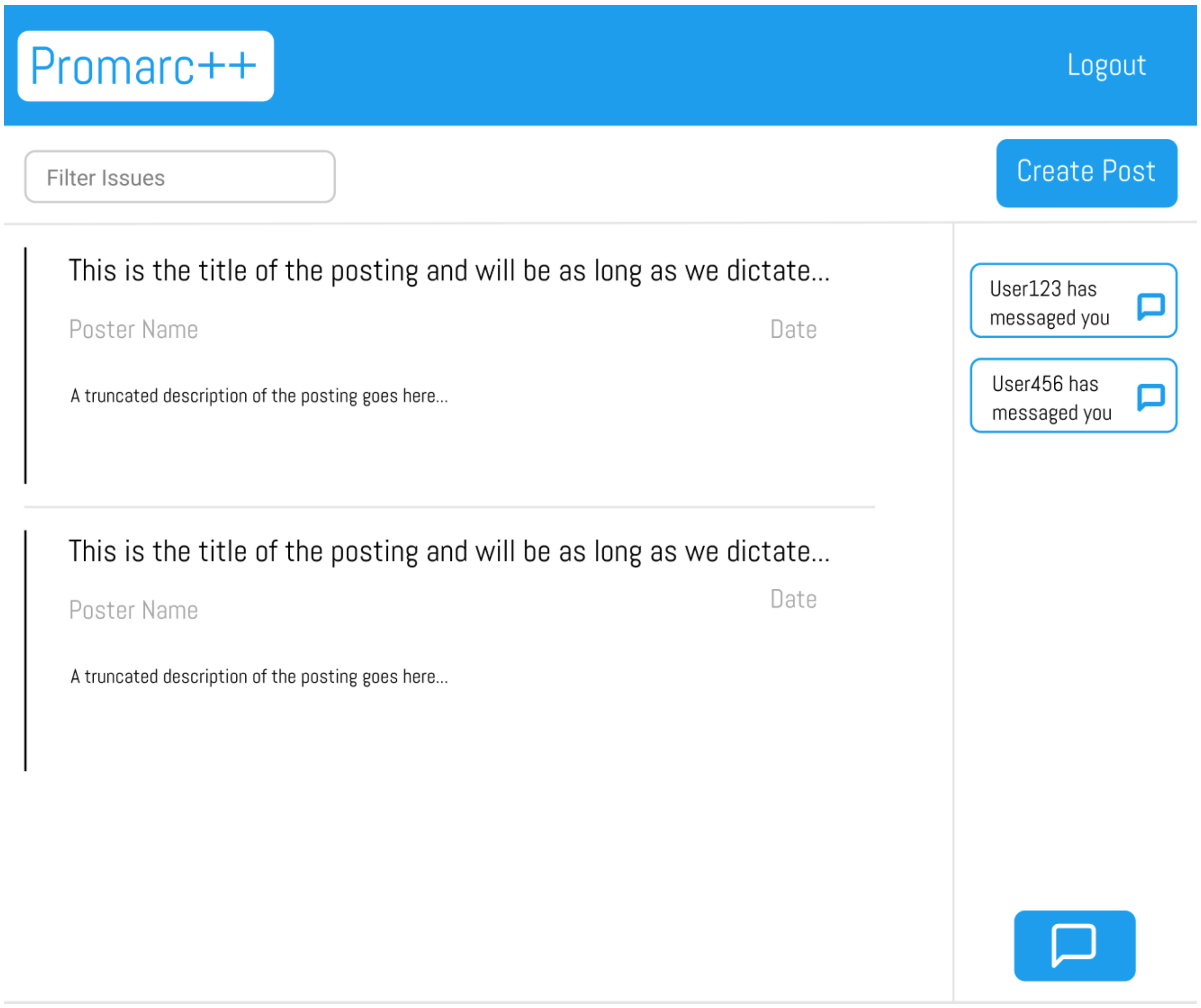

Promarc++
Logout
Filter Issues
Create Post
This is the title of the posting and will be as long as we dictate...
Poster Name
Date
A truncated description of the posting goes here...
This is the title of the posting and will be as long as we dictate...
Poster Name
Date
A truncated description of the posting goes here...
User123 has messaged you
User456 has messaged you

Figure 5.1: Home View

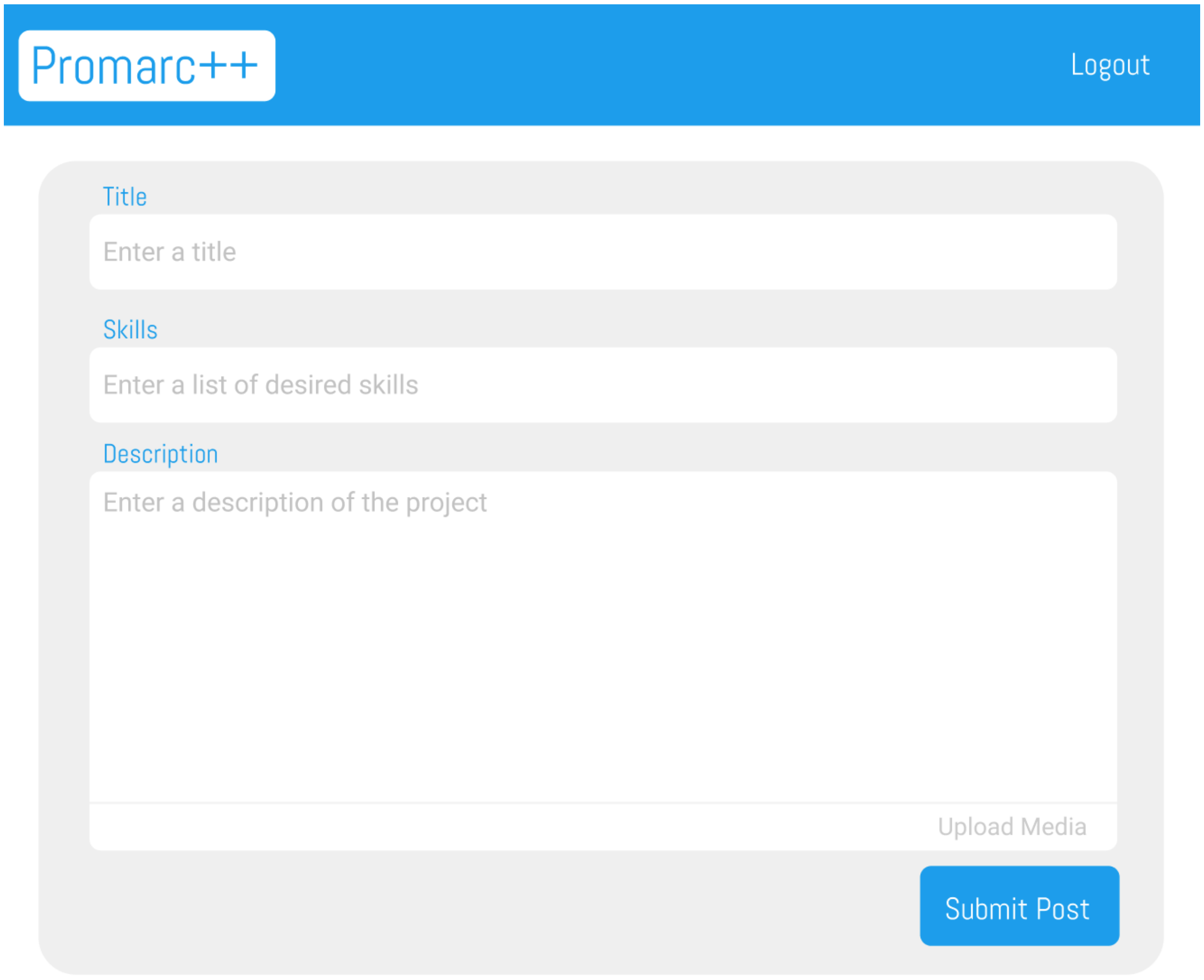


Figure 5.2: Create Post View

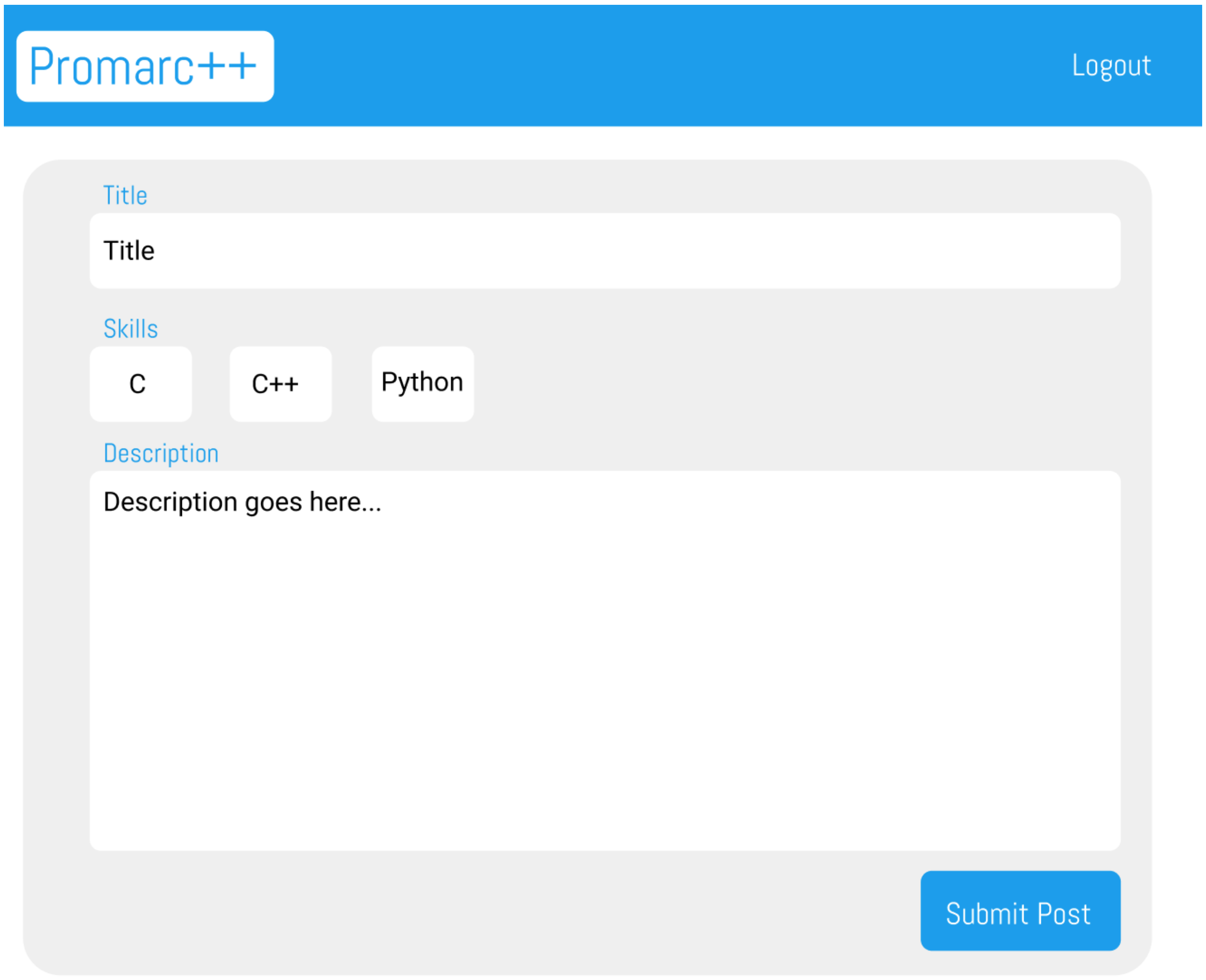
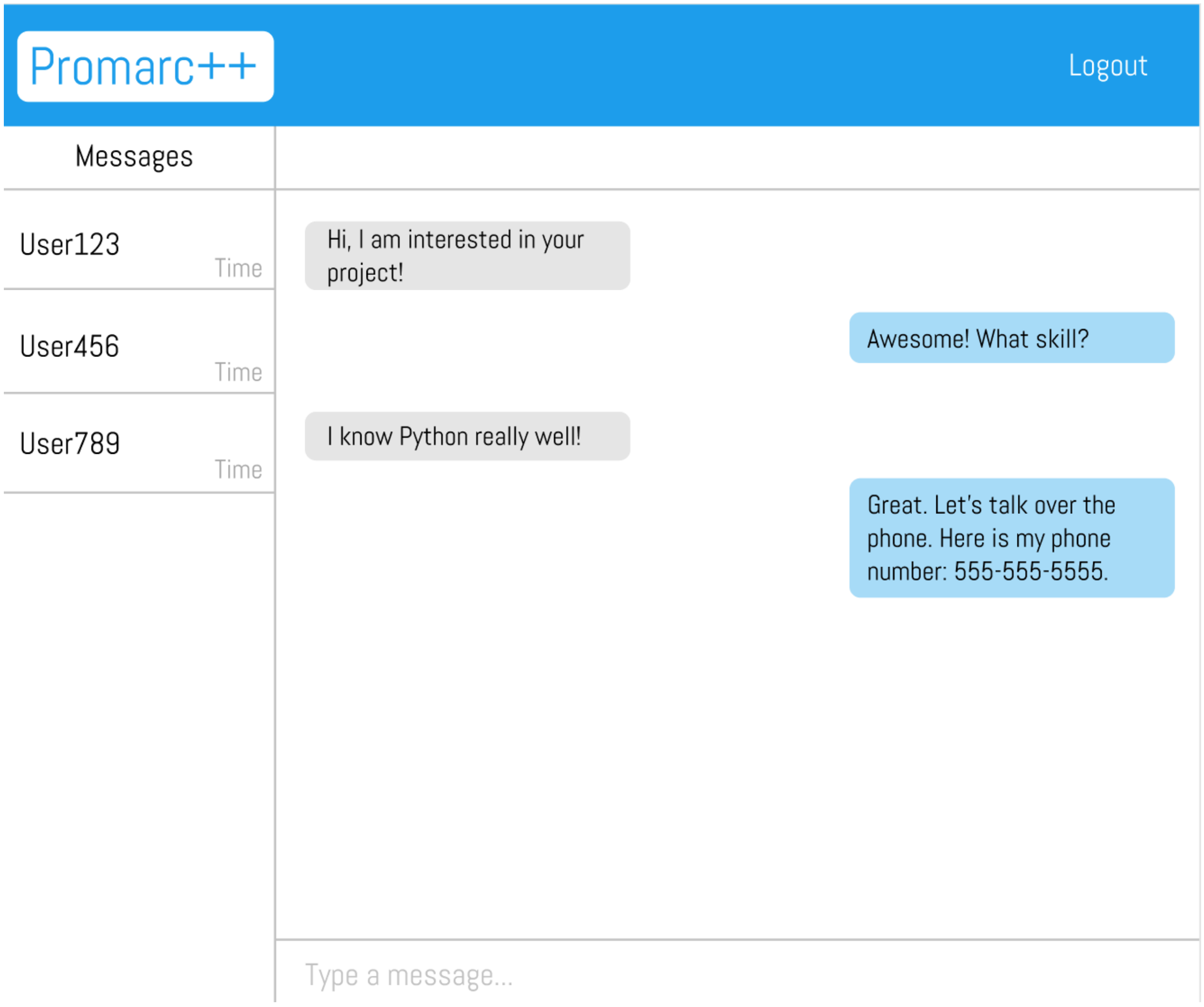


Figure 5.3: Post View



Hi, I am interested in your project!

Awesome! What skill?

I know Python really well!

Great. Let's talk over the phone. Here is my phone number: 555-555-5555.

Figure 5.4: Message View

Chapter 6

Technologies Used

We have decided on the following technologies that we will use to build our web application, and here are our justifications for why we chose each one.

MongoDB [10]

MongoDB is a NoSQL document-oriented database, with a flexible schema and a JSON-based query language. The JSON documents correspond nicely with the concept of creating and storing posts. The schema-less design will allow for easy DB configuration and quick iterations if database keys change throughout the development process. Finally, MongoDB's language is also JavaScript, making it compatible with the rest of our tech stack.

Express.js [7]

Node.js is a runtime environment that can run JavaScript. Express.js is a framework that helps to simplify the task of writing server code. Express.js will specifically offer an easy way to define routes and parse request URLs.

React [6]

React is the defining component of our technology stack. React is an open-source JavaScript library maintained by Facebook that can be used for creating views rendered in HTML. Simply put, React is a library and thus does not dictate a framework pattern. Specifically, we will be leveraging Reacts component-based capabilities that will maintain each components state and rendering.

Node.js [1]

Node.js accompanies React nicely. Whereas React is JavaScript in the browser, Node.js is JavaScript outside of the browser. Node.js has an asynchronous, event-driven, non-blocking input/output (I/O) model which will allow for users to quickly create posts and communicate with each other. In addition, Node.js leverages npm and modules. This will remove much of the development burden, as we will be able to utilize these packages and libraries.

Heroku [2]

Heroku is a cloud-based platform for serverless web development. Applications hosted on Heroku can be quickly deployed, are easily scalable, and require minimal configuration. Using Heroku for our hosting service will allow us to spend less time focusing on hosting and deploying Promarc, and more time on developing features and additions to make our product even better.

Chapter 7

Architecture Diagram

The following figures describe our system architecture. We use a data-centric architecture to describe the way our clients, databases, and servers interact. We will be leveraging the blackboard variant [3]. This implies that changes in data trigger communication with clients. This will allow us to have a single centralized communication system. For drawbacks, we recognize the challenge of having a single point-of-failure [9].

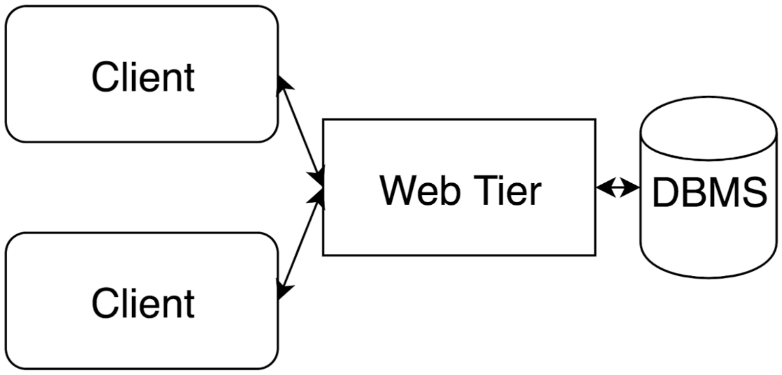


Figure 7.1: Client(s) interaction described by a Data-Centric architecture

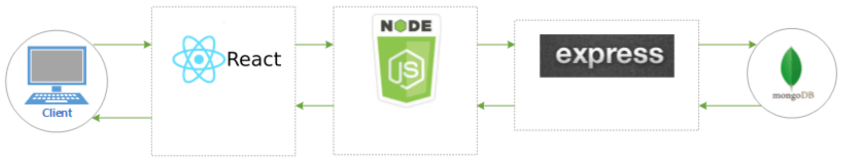


Figure 7.2: Technology Stack Diagram

Chapter 8

Design Rationale

We decided to limit Promarc's user base to individuals with "@scu.edu" email addresses. We wanted to ensure that our system had an inherently safe and productive community in the early proof-of-concept stages. Furthermore, we envisioned Promarc to be especially useful to environments similar to and including college campuses, and Santa Clara University was the perfect platform to test it on.

The two most important aspects of our system are being able to view posts and connecting with other people, and this is reflected in the way our user interface is arranged. The navigation bar at the top allows users to quickly switch back and forth between the Home View, for viewing posts, and the Messaging View, for messaging other users. These features are only available once a user has logged in with an authenticated account; this is to maintain an inherent level of security and accountability within Promarcs online community. There are two types of Promarc users:

General user: a regular member of the community with no special permissions. General users are allowed to view and create posts, as well as communicate with other general users through the messaging system. They can also flag existing posts as inappropriate, for review by a moderator.

Moderator: a community administrator. Moderators have the ability to review flagged posts and delete or edit any posts that they deem to be inappropriate.

Other views accessible to authenticated users include the Modal View and the Create Post View. Users are taken to the Modal View after clicking on a post in the Home View, and the Modal View offers an expanded, more detailed version of the chosen post. The Create Post View allows users to create a post and specify a title, required skills, and a short project description. Finally, as mentioned above, the Messaging View shows the user their inbox and allows them to communicate with other Promarc users. We felt this organization of application views felt the most simple and intuitive to use, and new Promarc users should have no issues familiarizing themselves with the system.

We chose MERN (MongoDB, Express.js, React.js, Node.js) for our tech stack because all these technologies are modern, relevant, and representative of current web development trends. We also chose Heroku for our hosting service because of its simplicity and ease of use [8]. More details about our tech stack can be found in the Technologies Used section.

Chapter 9

Test Plan

We utilized a test-driven development process to expedite any final testing of our system, developing strictly for specific test cases due to our very short development deadline. A test-driven process allowed us to ensure functionality and limit any surprise bugs once the project was completed.

Once an initial build of the system was completed, we created accounts for members of our own team, created test posts, and followed those posts through the system. We tested the system against known edge cases to verify that the system was behaving as expected.

Once we were confident that we achieved basic functionality and a complete system, we began black box testing by granting other peers unfamiliar with our project accounts and access to our system. We let them test the system in order to gather information regarding the effectiveness and simplicity of the interface we developed, as well as identify any unforeseen edge cases.

Chapter 10

Risk Analysis

Table 10.1 depicts an analysis of potential problems we could have run into during development, including consequences, relative probability of occurrence, severity, and ways we avoided them. These probability numbers are not specifically calculated, but are generic representations of how likely we believed these problems were to occur. The impact column values are calculated by a simple multiplication of the probability and the severity values, to give a normalized impact rating on a scale from 1 to 10.

Table 10.1: Risk Analysis

Risk	**Consequences**	**P**	**S**	**I**	**Mitigation**
Software bugs	Application does not function as intended; fixes must be deployed, contributing to development schedule delays.	1.0	8	8.0	Perform frequent and effective tests to reduce the chances of software bugs.
Unfamiliar technology	Team members must spend time familiarizing themselves with tech stack, contributing to development schedule delays.	1.0	7	7.0	Conduct research before implementation schedule to get a head start, and divvy up tasks according to team members strengths.
Schedule delays	Insufficient time to fully implement application, possibly resulting in lower implementation quality and unfulfilled requirements.	0.8	7	5.6	Adhere as closely as possible to development timeline, and collaborate on individual components if necessary to keep up with pace.
Missing requirements	Final application does not fulfill all specified requirements, leading to client dissatisfaction.	0.4	7	2.8	Consistently refer to requirements specification throughout development to maintain focus, and place greater emphasis on implementing essential functional requirements first.
Scope creep	Too much scope creep in our design might result in not fulfilling all of the specified requirements.	0.1	2	0.2	Maintain focus on the specified requirements.
File loss	Sudden device, version control system, or memory failure; application must restart development procedure, or schedule is significantly delayed.	0.01	10	0.1	Keep back-ups of all application files in addition to using version control so that file loss does not significantly impact the development timeline.

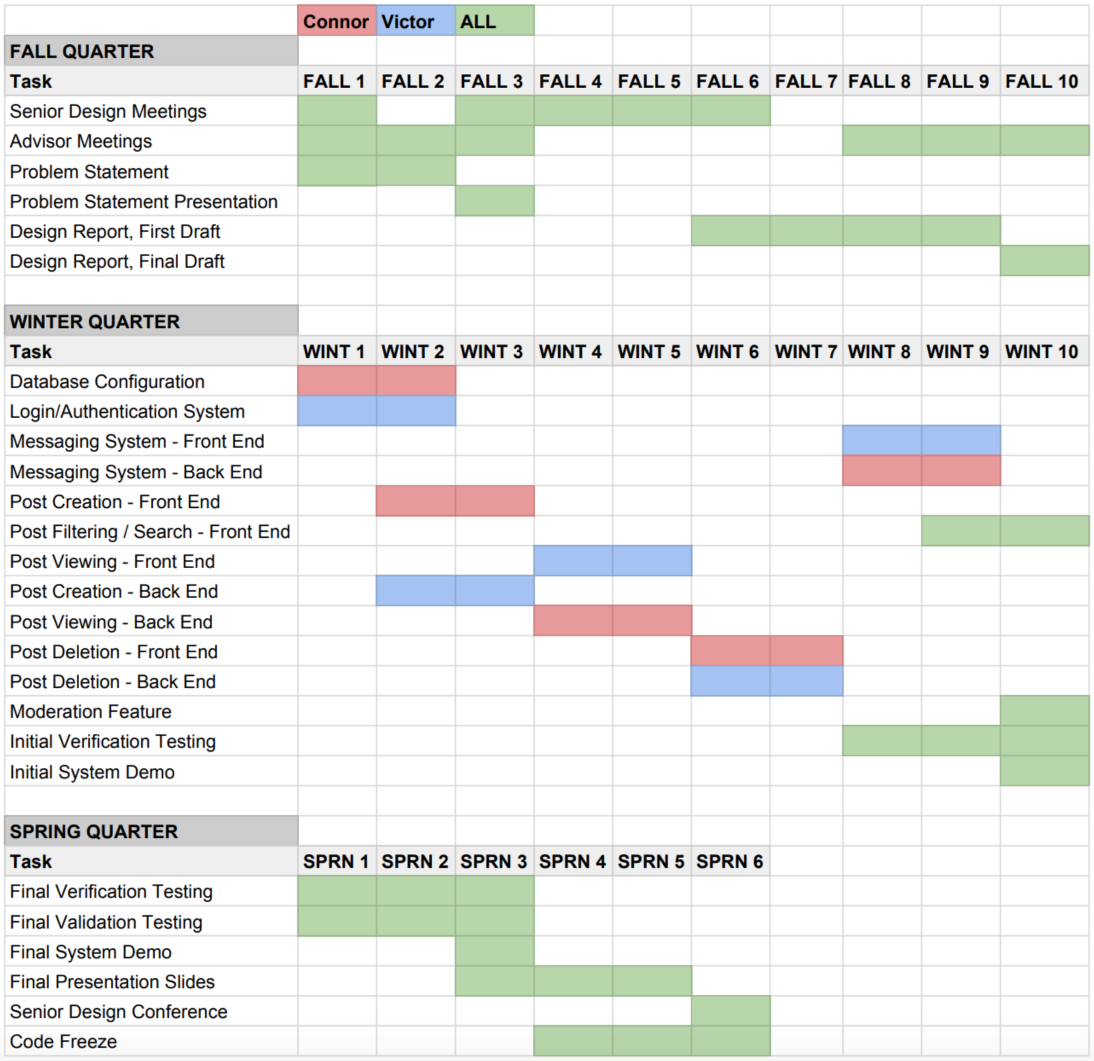
P = Probability, S = Severity, I = Impact

Chapter 11

Development Timeline

Figure 11.1 is a Gantt chart depicting our development timeline for our system. In general, soft deliverables like design documents and presentations involved both team members, while implementation tasks and testing were divvied up by category. Both team members had an even distribution of frontend and backend responsibilities. However, all implementation tasks and testing had a collaborative element to them, in order to ensure the cohesion of our individual modules in the final system.

To remain on schedule, we made sure to have an initial working system to demo by the end of winter quarter, and this system was thoroughly tested from the end of winter quarter through the beginning of spring quarter to be ready for the final demo.



	🟥 Connor	🟦 Victor	🟩 ALL							
FALL QUARTER										
Task	**FALL 1**	**FALL 2**	**FALL 3**	**FALL 4**	**FALL 5**	**FALL 6**	**FALL 7**	**FALL 8**	**FALL 9**	**FALL 10**
Senior Design Meetings	🟩		🟩	🟩	🟩	🟩				
Advisor Meetings	🟩	🟩	🟩					🟩	🟩	🟩
Problem Statement	🟩	🟩								
Problem Statement Presentation			🟩							
Design Report, First Draft						🟩	🟩	🟩	🟩	
Design Report, Final Draft										🟩
WINTER QUARTER										
Task	**WINT 1**	**WINT 2**	**WINT 3**	**WINT 4**	**WINT 5**	**WINT 6**	**WINT 7**	**WINT 8**	**WINT 9**	**WINT 10**
Database Configuration	🟥	🟥								
Login/Authentication System	🟦	🟦								
Messaging System - Front End								🟦	🟦	
Messaging System - Back End								🟥	🟥	
Post Creation - Front End		🟥	🟥							
Post Filtering / Search - Front End									🟩	🟩
Post Viewing - Front End				🟦	🟦					
Post Creation - Back End		🟦	🟦							
Post Viewing - Back End				🟥	🟥					
Post Deletion - Front End						🟥	🟥			
Post Deletion - Back End						🟦	🟦			
Moderation Feature										🟩
Initial Verification Testing								🟩	🟩	🟩
Initial System Demo										🟩
SPRING QUARTER										
Task	**SPRN 1**	**SPRN 2**	**SPRN 3**	**SPRN 4**	**SPRN 5**	**SPRN 6**				
Final Verification Testing	🟩	🟩	🟩							
Final Validation Testing	🟩	🟩	🟩							
Final System Demo			🟩							
Final Presentation Slides			🟩	🟩	🟩					
Senior Design Conference						🟩				
Code Freeze				🟩	🟩	🟩				

Figure 11.1: Development Timeline

Chapter 12

Societal Issues

It is important to discuss any engineering project's relevant societal issues in order to make sure that the developers have the right intentions, and that the project's societal impact is properly understood. This chapter highlights the societal issues that are relevant to Promarc. We believe that Promarc does not have direct impact on political issues, manufacturability, sustainability, environmental issues, and health and safety issues. However, we recognize that the teams which are created using our software may, in turn, create projects which have significant impact on these categories.

12.1 Ethical

One of the main goals of our service is to ensure that the teams which are being created are working on ethical projects. We do not want our platform to be a vehicle for people to act unethically and we recognize that our software could be used to create teams with nefarious motives. To combat this, we included a flagging system which allows users to flag inappropriate posts. The flagged posts will later be reviewed by moderators who will either edit or remove them. It is for this reason that we must ensure the selected moderators for Promarc are dedicated to ensuring the integrity of our online community.

12.2 Social

Promarc is a platform which enables people in a larger community to form teams, and in many ways, Promarc itself is a community. For this first iteration of Promarc, we have striven to confine this community to Santa Clara University students and faculty members as a proof of concept. We hope that Promarc will have an overall positive social impact; SCU students, teachers, and alumni, who might never have interacted without Promarc, could band together and effect positive social change. However, we also recognize that picking and choosing team members will naturally exclude some individuals. This could potentially cause an excluded Promarc user to feel inferior and harbor negative feelings towards the original poster, which goes against our goal of maintaining a healthy and collaborative community.

12.3 Economic

Promarc is currently configured and deployed on a free server, database instance and domain name. If we were to scale our solution for many users, these technologies would be inadequate. Consequently, we would need to subscribe to a monthly database and server plan. In addition, we would need to purchase a domain that will need to be renewed each year. The cost of these services would directly correlate with the size to which we want to scale.

12.4 Environmental Impact

Overall, Promarc has a very limited impact on the environment. We do recognize, however, that if Promarc were to scale upwards, we would need to deploy multiple servers and databases. These will require resources and energy to run and should be something we consider when selecting providers.

12.5 Usability

One of the main design constraints of Promarc was to ensure a streamlined and simple user experience. In order to do this, we tried to eliminate any overhead and extraneous features in order to lower the learning curve. In addition, we attempted to add a level of comfort and familiarity by creating a user interface which is consistent with other prominent software. For example, Promarc's Inbox view closely resembles the interface of popular instant messaging applications like iMessage and Facebook Messenger.

12.6 Lifelong Learning

We envisioned Promarc to be a platform focused on learning and collaboration, and we certainly hope that our system encourages users to engage in lifelong learning. Participation in a project created on Promarc is a valuable opportunity to learn new technologies and new skills.

In terms of our own lifelong learning, one of the biggest lessons we learned from creating Promarc was the importance of lifelong learning. We learned that web technologies are constantly being updated, modified, and deprecated. We quickly found that any online resources and documentation we used to learn Promarc's technology stack needed to be created or updated within the last year for them to still be considered best practice. This lesson will be invaluable for our careers, as we will have to work to stay relevant and learn the most recent technologies, methodologies, and best practices.

12.7 Compassion

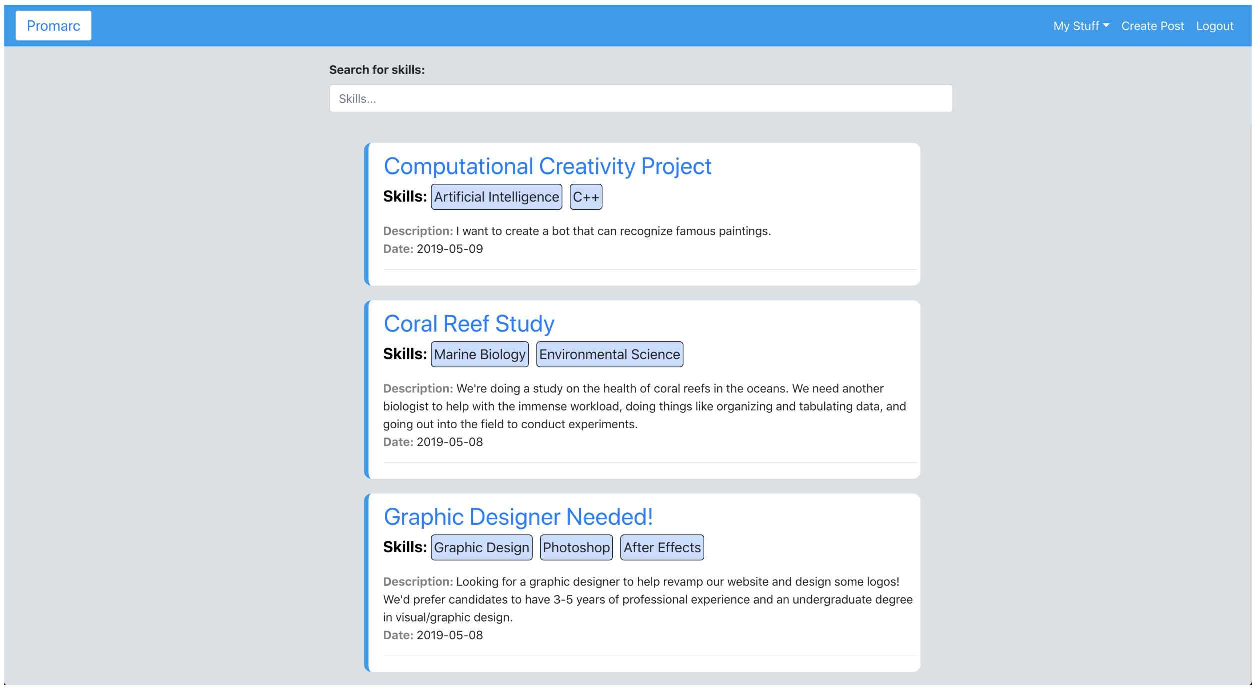
A big part of why we developed Promarc was the compassion we felt towards our fellow students that have had wonderful ideas for projects, but didn't have the means to complete them. College is the perfect opportunity for students to pursue projects because they are surrounded by a diverse set of wonderful minds and ideas. We hope that Promarc will be a huge help in connecting great minds and reducing the number of lost and forgotten ideas. By building a platform which connects users and enables the completion of projects, we hope to improve every student's overall experience at Santa Clara University.

Chapter 13

Final Results

Our system's final implementation is shown and discussed below, with screenshots included to illustrate the different aspects of the user interface. Differences between the conceptual models and final results will be highlighted and rationalized. All the functionality will not be covered, as this is meant to be a general overview of the scope of the project.

13.1 Home View



Computational Creativity Project
Skills: Artificial Intelligence | C++
Description: I want to create a bot that can recognize famous paintings.
Date: 2019-05-09

Coral Reef Study
Skills: Marine Biology | Environmental Science
Description: We're doing a study on the health of coral reefs in the oceans. We need another biologist to help with the immense workload, doing things like organizing and tabulating data, and going out into the field to conduct experiments.
Date: 2019-05-08

Graphic Designer Needed!
Skills: Graphic Design | Photoshop | After Effects
Description: Looking for a graphic designer to help revamp our website and design some logos! We'd prefer candidates to have 3-5 years of professional experience and an undergraduate degree in visual/graphic design.
Date: 2019-05-08

Figure 13.1: Home View

The Home View is Promarc's flagship feature. It serves as the main hub where users can see posts that other users in the community have submitted. New content is pushed to the top of the feed, and users are able to filter posts by specific skills using the search bar. Individual posts can be clicked to bring up a modal view that allows for deeper interaction, like starting conversations or flagging as inappropriate.

The Home View has some major differences from the initial conceptual model. The most notable change is the removal of the notification section. We made this decision because we wanted to focus on Promarc being as simple and usable as possible. The notification system cluttered the Home View and brought attention away from the posts, which is the main purpose of the page. In the final implementation, the navigation bar replaces the Create Post button and the Message button from the conceptual model, which we feel helps to improve the organization of the page.

13.2 Create Post View

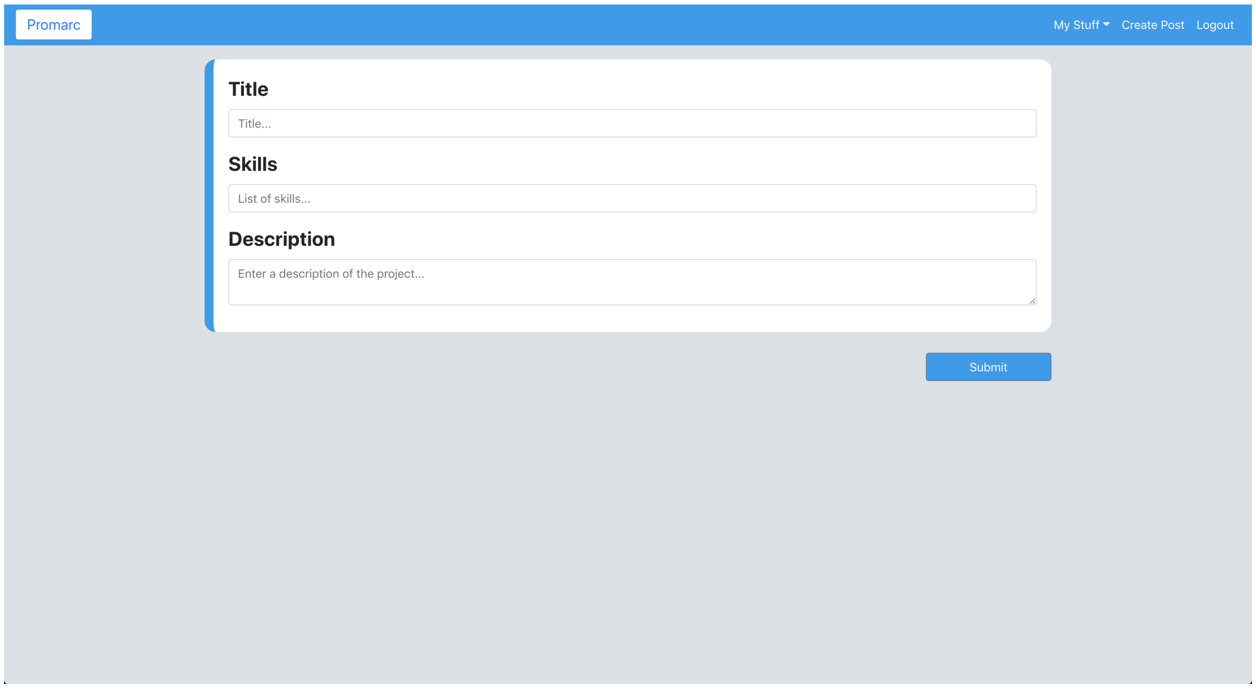
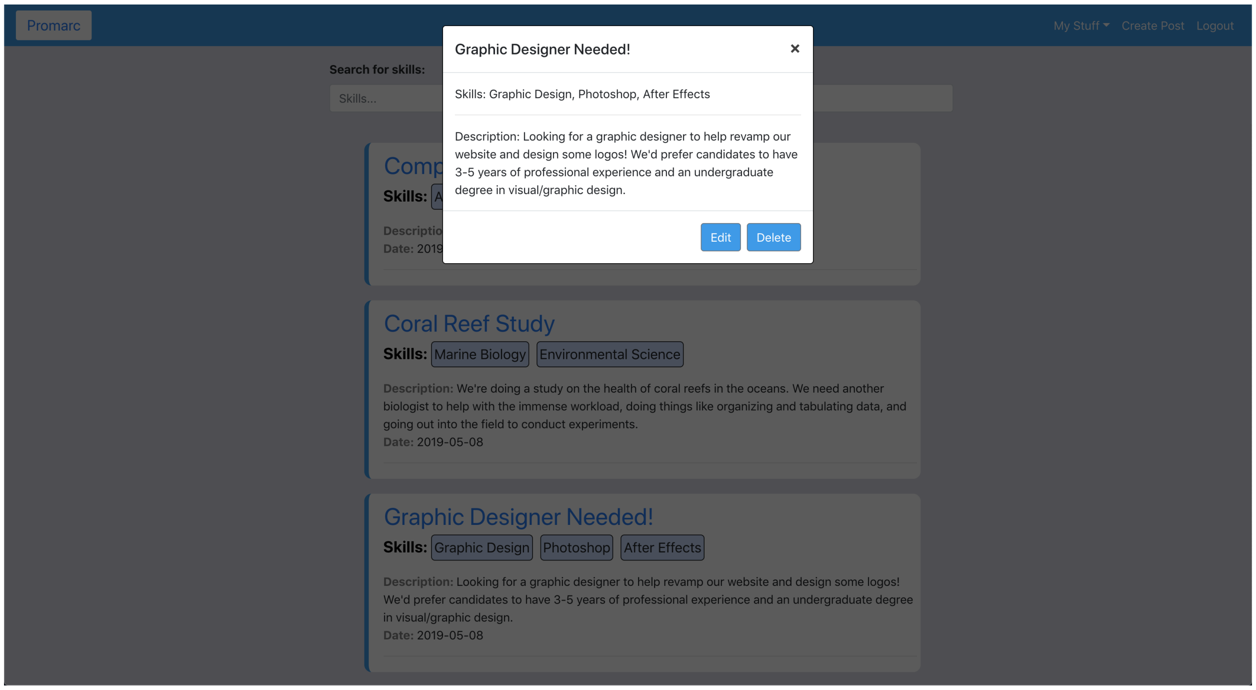


Figure 13.2: Create Post View

The Create Post View is where users go to fill out the title, skills, and description for a new post. Clicking the Submit button in this view will push the new post to the top of the Home View, where it will be available for the community to view and interact with. Other than minor visual tweaks, the final Create Post View remained almost identical to the conceptual model in terms of functionality.

13.3 Modal View



Graphic Designer Needed!

Skills: Graphic Design, Photoshop, After Effects

Description: Looking for a graphic designer to help revamp our website and design some logos! We'd prefer candidates to have 3-5 years of professional experience and an undergraduate degree in visual/graphic design.

Comp

Skills: A

Descriptio

Date: 2019

Coral Reef Study

Skills: Marine Biology Environmental Science

Description: We're doing a study on the health of coral reefs in the oceans. We need another biologist to help with the immense workload, doing things like organizing and tabulating data, and going out into the field to conduct experiments.

Date: 2019-05-08

Graphic Designer Needed!

Skills: Graphic Design Photoshop After Effects

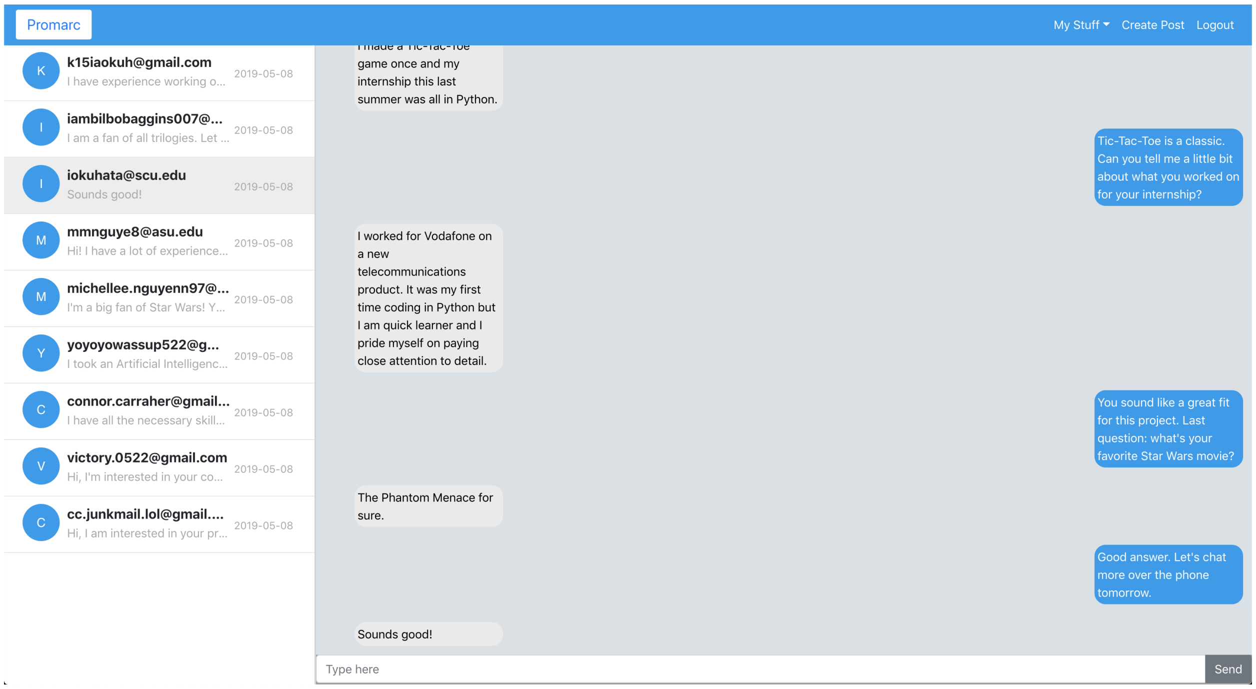
Description: Looking for a graphic designer to help revamp our website and design some logos! We'd prefer candidates to have 3-5 years of professional experience and an undergraduate degree in visual/graphic design.

Date: 2019-05-08

Figure 13.3: Modal View

The Modal View appears as on overlay on the Home View whenever a post is clicked. It dims the rest of the Home View to further emphasize the content of the individual post, and it reveals buttons with options like ”Edit,” ”Delete,” ”Contact,” and ”Flag as Inappropriate,” which will change depending who the user is and the user’s relationship to the post. The Modal View was not included as part of the initial conceptual models.

13.4 Message View



k15iaokuh@gmail.com — I have experience working o... — 2019-05-08
iambilbobaggins007@... — I am a fan of all trilogies. Let ... — 2019-05-08
iokuhata@scu.edu — Sounds good! — 2019-05-08
mmnguye8@asu.edu — Hi! I have a lot of experience... — 2019-05-08
michellee.nguyenn97@... — I'm a big fan of Star Wars! Y... — 2019-05-08
yoyoyowassup522@g... — I took an Artificial Intelligenc... — 2019-05-08
connor.carraher@gmail... — I have all the necessary skill... — 2019-05-08
victory.0522@gmail.com — Hi, I'm interested in your co... — 2019-05-08
cc.junkmail.lol@gmail.... — Hi, I am interested in your pr... — 2019-05-08

I made a Tic-Tac-Toe game once and my internship this last summer was all in Python.

Tic-Tac-Toe is a classic. Can you tell me a little bit about what you worked on for your internship?

I worked for Vodafone on a new telecommunications product. It was my first time coding in Python but I am quick learner and I pride myself on paying close attention to detail.

You sound like a great fit for this project. Last question: what's your favorite Star Wars movie?

The Phantom Menace for sure.

Good answer. Let's chat more over the phone tomorrow.

Sounds good!

Figure 13.4: Message View

We tried to make sure that the Message View's interface looked familiar to users from other popular applications like iMessage and Facebook Messenger, in order to make Promarc's in-house messaging feature as intuitive to use as possible. The final implementation of the messaging feature ended up looking very similar to our initial conceptualization.

13.5 Login View

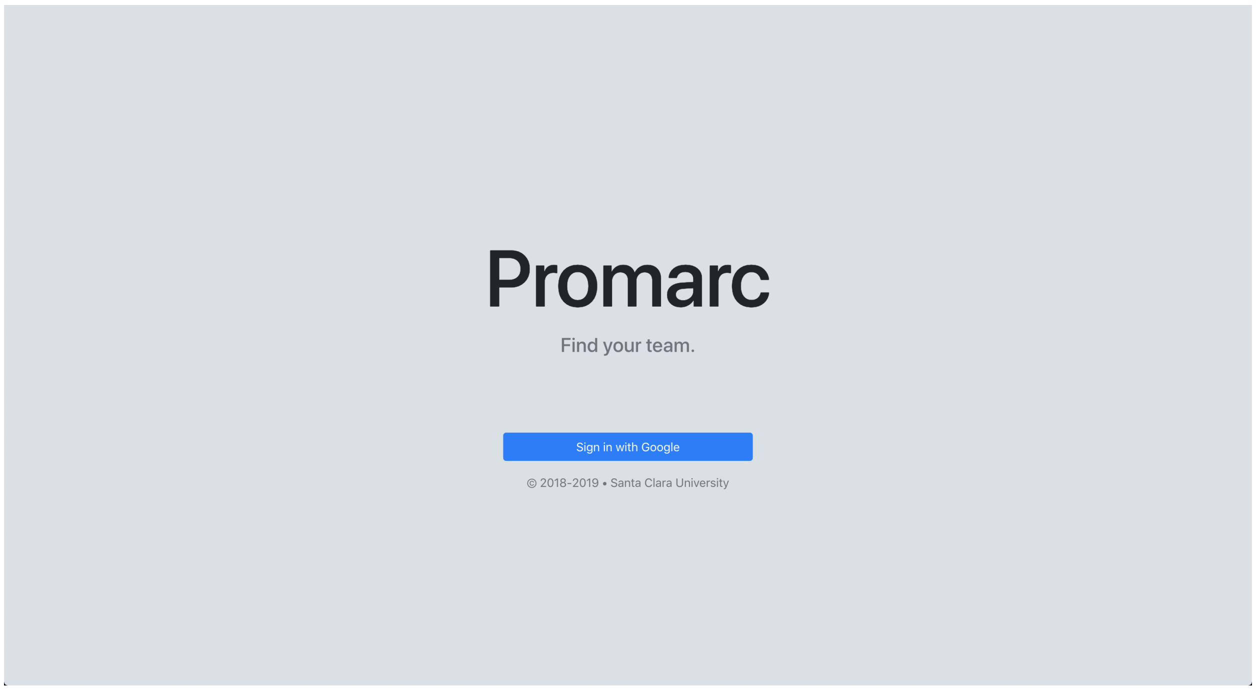


Figure 13.5: Login View

The Login View was added as a landing page for Promarc, and was not included in our initial conceptualization. This page is meant to be simple and quickly let the user know whether they are currently logged into the system or not. Interaction with the landing page is purely linear, as users can only use it to log in into the system. The Login View is also a major contributor to establishing Promarc's brand theme and design aesthetic.

Chapter 14

Conclusion

This last section reflects upon the final implementation of our system and the development process. We will discuss the valuable lessons we learned throughout the project, as well as an analysis of our project's positive and negative aspects. Finally, we will talk about our future plans for Promarc and the work we still have left to do.

14.1 Lessons Learned

We learned a great deal throughout the process of development of Promarc. First, we greatly underestimated how intricate messaging applications are under the hood. Promarc's Inbox feature took, by far, the most time and effort to complete, and it instilled in us a great respect for how well-designed popular messaging applications are nowadays. Next, we found that making even small UI/UX tweaks often had profound effects on how Promarc was perceived by users. For instance, we found adding the simple landing page for users to log in with lent Promarc a sense of enhanced security and cohesion. Finally, and perhaps most importantly, we learned that web technologies are constantly evolving and that we need to always be learning in order to keep up with current standards and best practices. One of our main objectives with Promarc was to learn an entirely new tech stack, and we feel we did a decent job of this.

14.2 Analysis of Project

This section will go into greater detail about the advantages and disadvantages of Promarc.

Advantages

One of the biggest advantages of our system is that we designed it to be exceptionally simple and intuitive to use. We tried to eliminate any extraneous features and unnecessary visual elements so that users can focus on their goal of finding the people they need. The Inbox feature was implemented to simplify the process of actually connecting with people. Another advantage of Promarc in its current iteration is that we limited the user base to only Santa Clara students, faculty, and alumni. This sort of natural filter helps to ensure that Promarc's community is responsible, productive, and collaborative.

Disadvantages

We have also identified several of Promarc's disadvantages, which need to be addressed in future iterations. While Promarc's simplicity will be a positive aspect for many of its users, there are also many other people who expect vast suites of functionality from the applications they use. In that sense, Promarc is rather limited in the scope of its features. In addition, the way the system is currently implemented introduces many scalability issues, especially with the Inbox feature. Using the free versions of Heroku and MongoDB coupled with our database schema means that if Promarc wouldn't be able to handle a large number of users and requests.

14.3 Future Work

Notifications

When a user receives a message, there is no way to notify them if they are not on the Inbox page. The only way to determine if a new message has been received is to navigate to the Inbox page and check. A non-intrusive notification on the other pages would help the user notice and respond to messages faster.

Database Refactoring

The current database for Promarc is limited to 16 MB. If we wanted to scale Promarc to support a greater number of users, the size of the database would need to increase in order to support the increase of data. The database also stores some redundant or unnecessary data, which could be eliminated for greater efficiency.

Web Sockets

Promarc's current messaging client is not configured for messages to be pushed. Instead, the client attempts to pull any new messages from the database. Ideally, messages should be pushed from the server to the client. This would reduce the number of HTTP requests Promarc must make when viewing the messaging client.

UI Overhaul

We believe that while Promarc's current user interface is usable and intuitive, it can absolutely be further improved. The current iteration of our UI is a product of us learning proper UI design while creating the platform. In any system, the UI has the greatest impact on their perception of the software, improving and modernizing the UI is an essential task for us.

Chapter 15

References

[1] Basarat Syed. 2014. Beginning Node, Berkeley, CA: Apress.

[2] Documentation. Retrieved May 30, 2019 from https://devcenter.heroku.com/categories/reference Node.js.

[3] Frank Tsui. Orlando Karam. Barbara Bernal. 2014. Essentials of Software Engineering, 3rd Edition, Jones and Bartlett Learning.

[4] Foundation. Docs. Retrieved May 30, 2019 from https://nodejs.org/en/docs/.

[5] Getting Started React. Retrieved May 30, 2019 from https://reactjs.org/docs/getting-started.html.

[6] Mark Tielens Thomas. 2018. React in action, Shelter Island, NY: Manning Publications.

[7] Routing. Retrieved May 30, 2019 from https://expressjs.com/en/guide/routing.html.

[8] Simon Holmes. 2016. Getting MEAN: with Mongo, Express, Angular, and Node, Shelter Island: Manning.

[9] Steve McConnell. 2004. Code complete: a practical handbook of software construction, Redmond, Wash: Microsoft Press.

[10] Welcome to the MongoDB Docs. Retrieved May 30, 2019 from https://docs.mongodb.com/.

Chapter 16

Appendix

In this section, we will provide step-by-step instructions for the inexperienced user on how to use Promarc's main features. Screenshots are provided for clarity.

16.1 Logging In

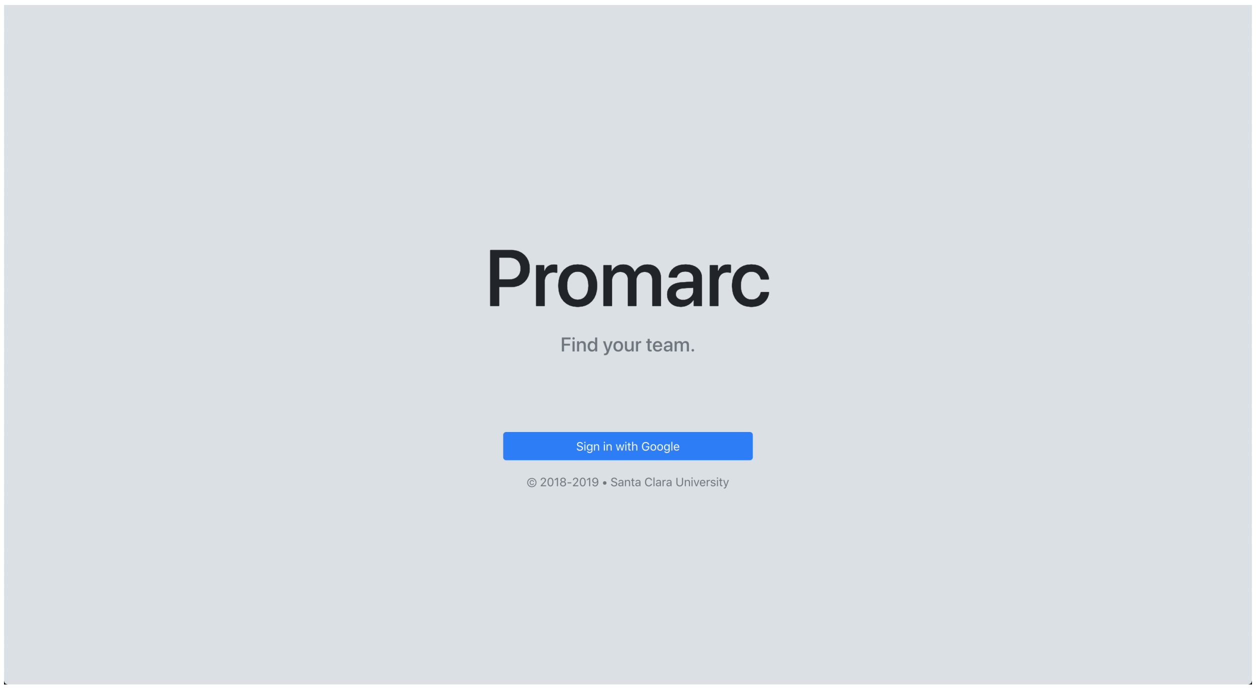


Figure 16.1: Login View

The landing page is the first page users will see after loading the URL. All remaining appendix entries will assume the user has already logged into the system. Follow the steps below in order to log in:

1. Click the blue ”Sign in with Google” button. The page will refresh and take you to a Google login page.

2. Enter your Google account information. The page will refresh again and take you to Promarc's Home View.

16.2 Creating a Post

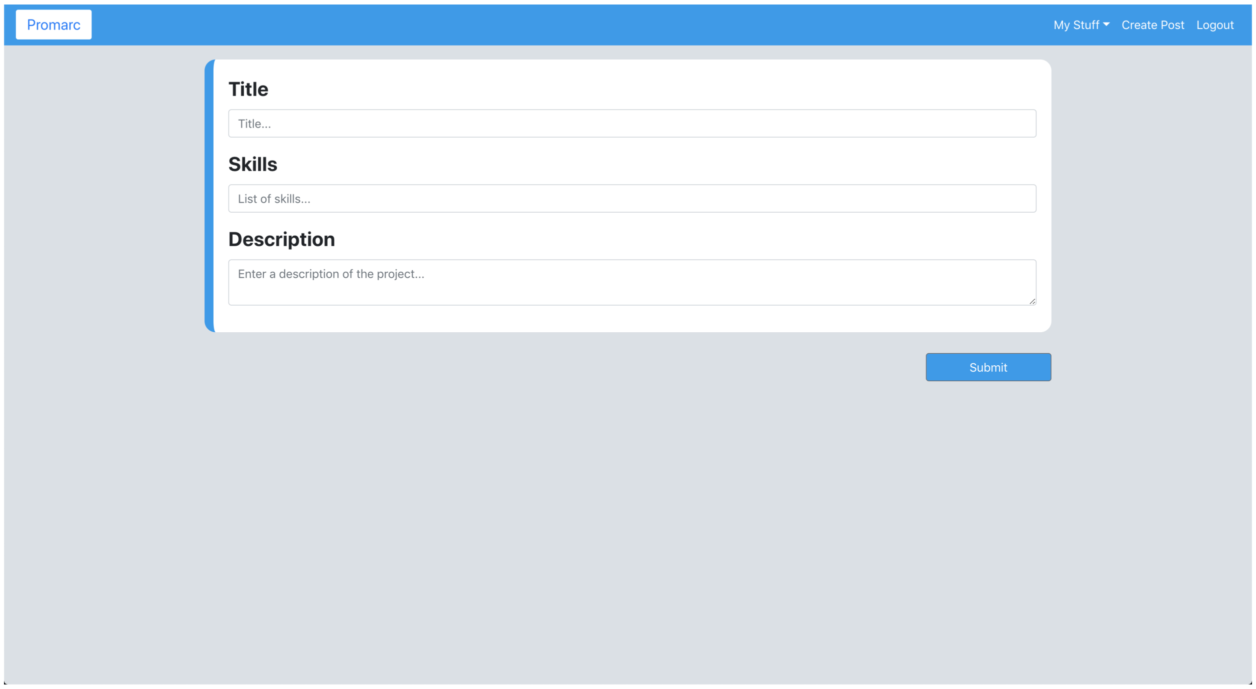
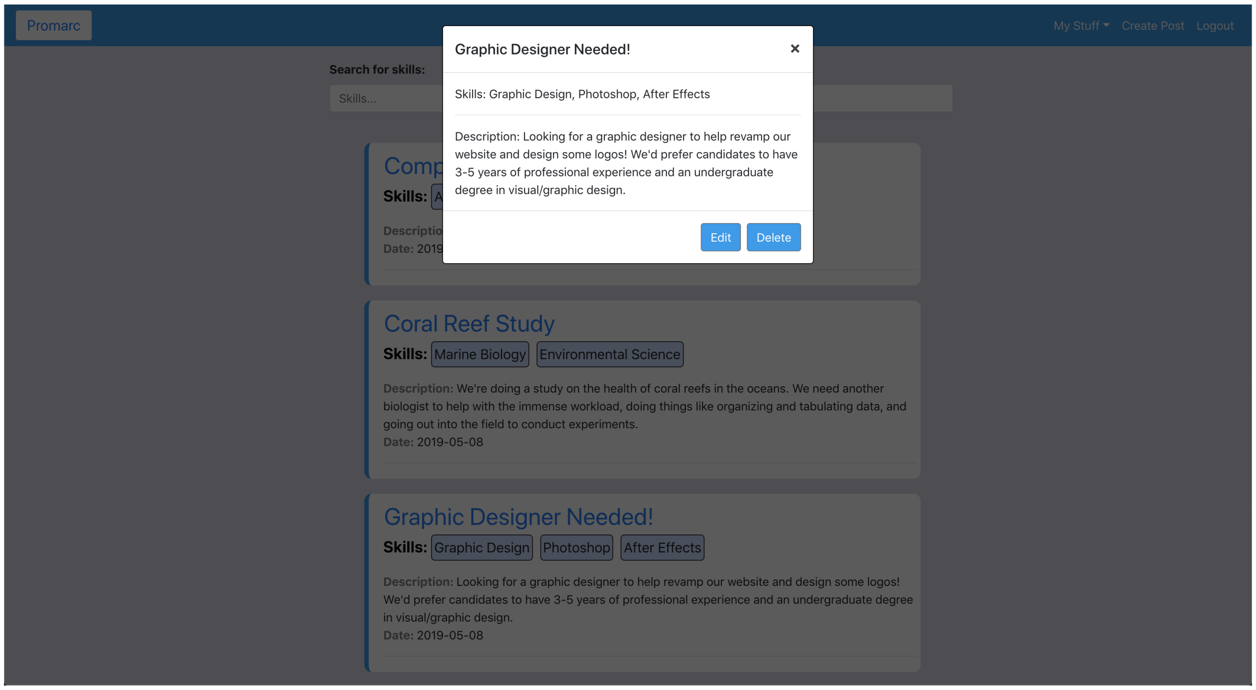


Figure 16.2: Create Post View

Users can submit a new post for other users to view and interact with through the Create Post View. Following the steps below in order to create a new post:

1. From any page, click the "Create Post" button in the navbar on the top right. The page will refresh and take you to the Create Post View, as shown in Figure 16.2.
2. Fill out the empty fields.
3. Click on the "Submit" button. The page will refresh and take you to the Home View, where your newly submitted post will be at or near the top of the feed.

16.3 Editing or Deleting a Post



Graphic Designer Needed!

Skills: Graphic Design, Photoshop, After Effects

Description: Looking for a graphic designer to help revamp our website and design some logos! We'd prefer candidates to have 3-5 years of professional experience and an undergraduate degree in visual/graphic design.

Coral Reef Study

Skills: Marine Biology Environmental Science

Description: We're doing a study on the health of coral reefs in the oceans. We need another biologist to help with the immense workload, doing things like organizing and tabulating data, and going out into the field to conduct experiments.
Date: 2019-05-08

Graphic Designer Needed!

Skills: Graphic Design Photoshop After Effects

Description: Looking for a graphic designer to help revamp our website and design some logos! We'd prefer candidates to have 3-5 years of professional experience and an undergraduate degree in visual/graphic design.
Date: 2019-05-08

Figure 16.3: Edit and Delete Buttons in Modal View

Users may only edit or delete posts that they themselves have submitted. These options appear in the Modal View upon clicking the post in the Home View. Follow the steps below in order to edit or delete a post:

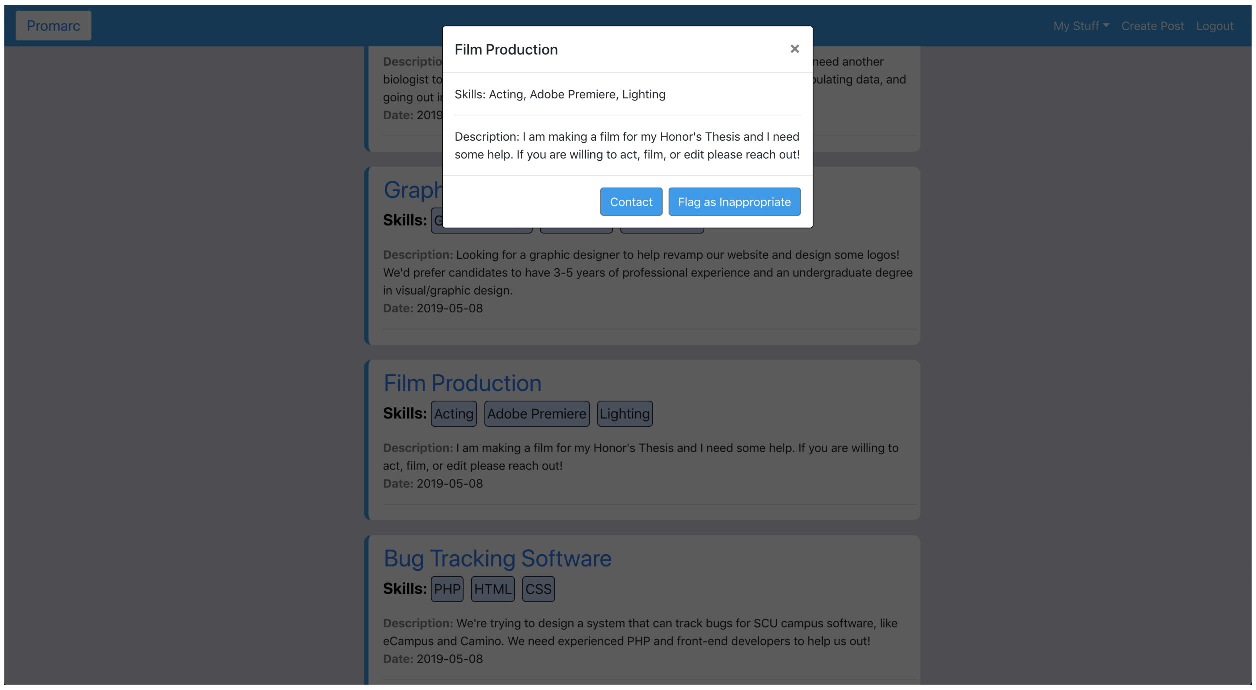
1. In the Home View, click on the title of the post to be edited or deleted. The Modal View for that post will appear overlaid on the screen.

2. Click on the ”Edit” or ”Delete” button as appropriate.

   (a) If the ”Edit” button is clicked, the page will refresh and take you to a page similar to the Create Post View, with the fields pre-filled in. Clicking on ”Submit” will reload the Home View, with the edits now reflected in the post.

   (b) If the ”Delete” button is clicked, the Modal View will disappear and the Home View will refresh. The deleted post will no longer appear.

16.4 Creating a Conversation



Film Production ×

Skills: Acting, Adobe Premiere, Lighting

Description: I am making a film for my Honor's Thesis and I need some help. If you are willing to act, film, or edit please reach out!

Contact Flag as Inappropriate

Descriptio... need another
biologist to ... pulating data, and
going out i...
Date: 2019

Graph...
Skills: ...

Description: Looking for a graphic designer to help revamp our website and design some logos! We'd prefer candidates to have 3-5 years of professional experience and an undergraduate degree in visual/graphic design.
Date: 2019-05-08

Film Production
Skills: Acting Adobe Premiere Lighting

Description: I am making a film for my Honor's Thesis and I need some help. If you are willing to act, film, or edit please reach out!
Date: 2019-05-08

Bug Tracking Software
Skills: PHP HTML CSS

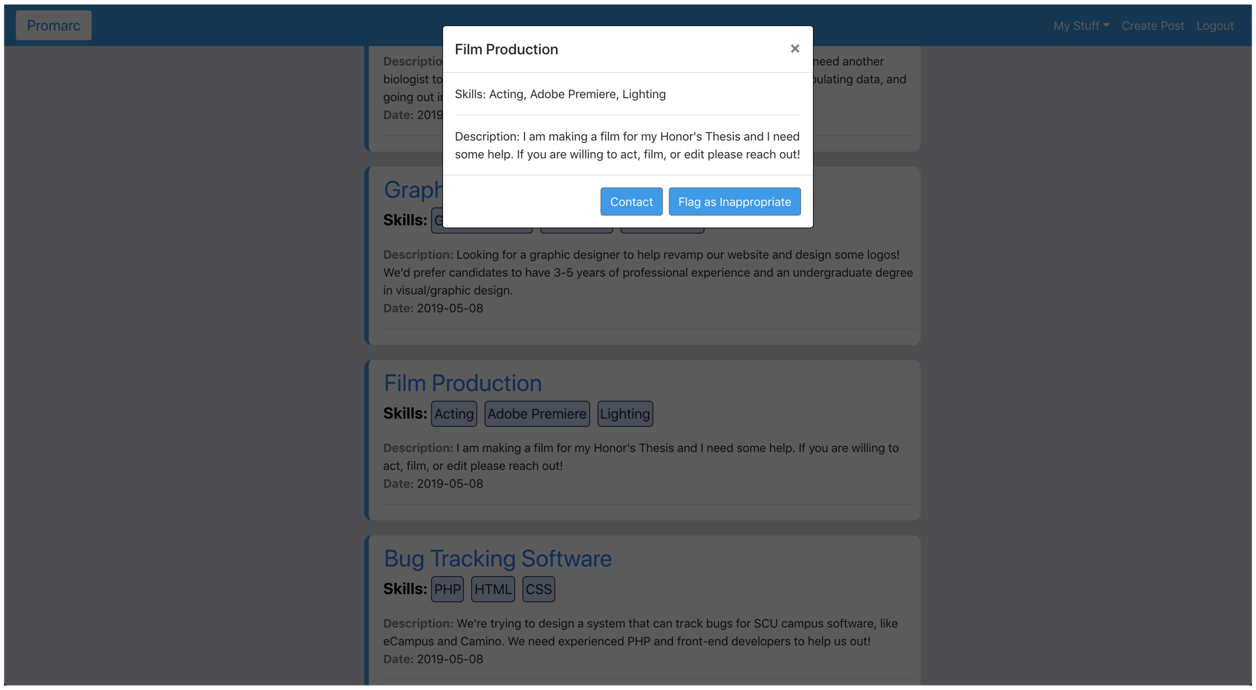
Description: We're trying to design a system that can track bugs for SCU campus software, like eCampus and Camino. We need experienced PHP and front-end developers to help us out!
Date: 2019-05-08

Figure 16.4: Contact Button in Modal View

Users may create conversations with other users to discuss the contents of a post. Follow the steps below to create a new conversation:

1. In the Home View, click on the title of the post whose poster you want to contact. The Modal View will appear.
2. Click the ”Contact” button. The page will refresh and take you to the Messaging View, where you can see a list of your conversations in your Inbox.
3. The new conversation will be open, and you can begin conversing with the other user.

16.5 Flagging a Post as Inappropriate



Film Production ×

Skills: Acting, Adobe Premiere, Lighting

Description: I am making a film for my Honor's Thesis and I need some help. If you are willing to act, film, or edit please reach out!

Descriptio… need another biologist to… …ulating data, and going out i…
Date: 2019…

Graph…
Skills:

Description: Looking for a graphic designer to help revamp our website and design some logos! We'd prefer candidates to have 3-5 years of professional experience and an undergraduate degree in visual/graphic design.
Date: 2019-05-08

Film Production
Skills: Acting Adobe Premiere Lighting

Description: I am making a film for my Honor's Thesis and I need some help. If you are willing to act, film, or edit please reach out!
Date: 2019-05-08

Bug Tracking Software
Skills: PHP HTML CSS

Description: We're trying to design a system that can track bugs for SCU campus software, like eCampus and Camino. We need experienced PHP and front-end developers to help us out!
Date: 2019-05-08

Figure 16.5: Flag as Inappropriate Button in Modal View

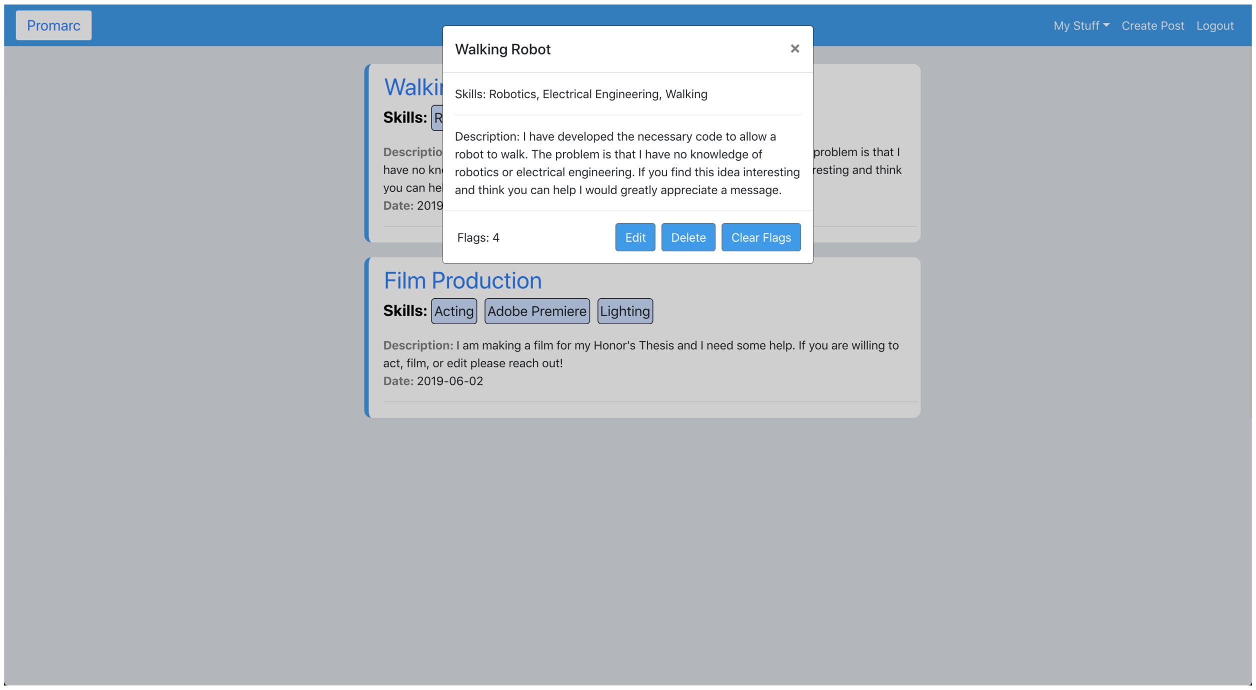
While browsing posts in the Home View, users may come across posts that they feel are inappropriate or not in accordance with community standards. In these cases, users may flag these posts, which will send them to a moderator for review. Follow the steps below to flag a post as inappropriate:

1. In the Home View, click on the title of the inappropriate post. The Modal View will appear.

2. Click the ”Flag as Inappropriate” button. The Home View will refresh, and the post will have been sent to the moderators for review.

16.6 Moderator Features

Moderators have access to several elevated features. They have the ability to edit or delete any post they wish to, using the Modal View, and they also have access to the Moderator Dashboard. After selecting a post in the Moderator



Skills: Robotics, Electrical Engineering, Walking

Description: I have developed the necessary code to allow a robot to walk. The problem is that I have no knowledge of robotics or electrical engineering. If you find this idea interesting and think you can help I would greatly appreciate a message.

Description: I am making a film for my Honor's Thesis and I need some help. If you are willing to act, film, or edit please reach out!
Date: 2019-06-02

Figure 16.6: Moderator Flagging Feature

Dashboard, the moderator will be able to review the post. The Modal View will contain all the information found on the Home View but will also display the number of times the post has been flagged. The moderator will be able to edit, delete or clear the flags for any post found in the Moderator Dashboard.