

# Mission Source

by

Blake Hanna, Cooper Kuethe, and Mariya Neshenko

Submitted to  
the Faculty of the School of Information Technology  
in Partial Fulfillment of the Requirements for  
the Degree of Bachelor of Science  
in Information Technology

© Copyright 2021 Blake Hanna, Cooper Kuethe, and Mariya Neshenko

The author grants to the School of Information Technology permission  
to reproduce and distribute copies of this document in whole or in part.



---

Blake Hanna

---

04/20/2021  
Date



---

Cooper Kuethe

---

04/20/2021  
Date



---

Mariya Neshenko

---

04/20/2021  
Date

Abdou Fall  
Abdou Fall, Faculty Advisor

04/20/2021  
Date

University of Cincinnati  
College of  
Education, Criminal Justice, and Human Services

April 2021

## TABLE OF CONTENTS

---

|   |       |
|---|-------|
| Abstract.....   | 1     |
| Introduction  | 2-5   |
| Problem.....  | 2-3   |
| Solution.....   | 3-4   |
| Project Goals.....  | 4-5   |
| Overview.....   | 5     |
| Discussion  | 6-18  |
| Project Concept.....                                      | 6     |
| Design Objectives.....                                    | 6     |
| Methodology.....  | 7     |
| User Profile.....   | 8-9   |
| Use Case Diagram.....                                     | 10    |
| Technical Architecture                                    | 11-12 |
| Testing   | 13-14 |
| Budget.....   | 15    |
| Project Timeline.....                                     | 16-17 |
| Problems Encountered and Analysis of Problems Solved..... | 17-18 |
| Future Recommendations                                    | 18    |
| Conclusion  | 19    |
| Lessons Learned.....                                      | 19    |
| Abilities and Skills Developed During Project.....        | 19    |
| Plans for Continuation                                    | 19    |
| Back Matter   | 20    |
| References.....   | 20    |

LIST OF FIGURES

---

|           |                        |       |
|-----------|------------------------|-------|
| Figure 1. | User Profile – Cameron | 8     |
| Figure 2. | User Profile – Joshua  | 8     |
| Figure 3. | User Profile – Zachary | 9     |
| Figure 4. | Use Case Diagram       | 10    |
| Figure 5. | Budget                 | 15    |
| Table 1.  | Project Timeline       | 16-17 |

## **ABSTRACT**

Many college graduates struggle to find a job after graduation. One reason for this is that it is a daunting task to find like-minded people interested in working on a variety of projects. Even if students are looking for a simple real-world side project to develop their skills, there are few places they can turn to in their search. "Mission Source" is a platform that is aimed at solving this problem. Users are able to go on this platform to find people to collaborate with, seek help with a side project, look for a hackathon group, and other similar activities. The platform is open to all colleges and majors that attend the same university. Having access to such a platform allows students to gain experience, develop new skills, and expand their networking group.

## 1. INTRODUCTION

### 1.1 Problem

According to a study done by the University of Washington, “53 percent of college graduates either are unemployed” (*University of Washington*, 2019). Indeed, an employment engine, is even quoted saying that graduates are currently struggling due to “high competition, little work experience, and little networking” (*Indeed*, 2020). Seeing this as a student really makes them start to think about how they can set themselves apart from the rest of the competition in a job search.

There are not that many resources out there that will put students up against REAL projects.

There are initiatives such as [First Timers Only](#) that allows new developers to contribute to easy open-source projects. [Google Summer Developers](#) is another program students can turn to, if they get in, of course. There is also a subreddit for software development side projects if they are looking to work with anyone who may stumble across the post.

For designers, there are challenges such as [Daily UI](#) to improve skills. Every day, users get a new prompt for a simple design project. [Uplabs challenges](#) is another platform to offer a variety of app/website design challenges.

However, those examples are very niche in their field (design & code). Most of the time, the challenges presented are not tackling real-world problems with real deadlines and requirements. Some students are not even aware that those solutions exist.

Currently, no university offers a platform which students can utilize to develop skills, find like-minded teammates and work on projects they are truly passionate about.

## **1.2 Solution**

Our solution is aimed at tackling problems such as: lack of real-world projects, group formation and motivation. Mission Source is a platform that will serve as a hub for students. They can go on this platform to find people to collaborate with, seek help with a side project, look for a hackathon group, and other similar activities.

This platform is different from other solutions in a few ways.

First, the user pool is limited to students attending the same university, encouraging collaborations between colleges and majors. Students are no longer limited to their network created through classes and clubs. They may find someone who has similar ideas outside of their college, which they may not have met otherwise.

Second, Mission Source utilizes gamification to encourage participation. The Mission Source platform uses achievements, badges, and ratings to keep students honest and engaged.

Achievements and badges are the motivators to participate in projects. Students can show them off on their profile page, which encourages healthy competition. For example, GitHub shows users activity on their profile page. Users are motivated to contribute 'pull requests' to keep the bar green. Achievements have a similar effect.

Missions Source utilizes ratings as an honor system. After working on a project, a user can rate the rest of the team and leave a review. Rating is helpful when users are deciding whether they want to work with someone or not.

As described, this platform is different because it allows for collaboration in multiple fields simultaneously. It encourages users to participate through gamification and they gain experience by tackling real problems with deadlines and requirements.

### **1.3 Project Goals**

Our primary goal is to allow students to create an account and look for potential groups. The user should be able to create a post describing their idea and chat with potential teammates on the platform. The user should be able to filter posts based on their interest, e.g. Python development.

Once a user is in the group, there is an option to attach supporting documents/links in the project overview, in case a user wants to share it in their portfolio.

#### **Core Features:**

- Log In / Sign Up
- Create missions with specific requirements (tags, skills, roles, timeline, description)
- Ability to view all the posts made by users.
- View specific post.
- Rate a user based on their performance.

- Leave a user review based on their performance.
- Ability to view achievements, reviews and ratings for a specific user.
- Message/email a user about their project.

**Added Features:**

- Filter based on a tag.
- Robust achievement system with titles, tracking etc.
- Resources on various topics that may be beneficial.

**1.4 Overview**

The remainder of this final report outlines in detail how the project was completed. The report includes the following sections: design objectives, methodology, budget, timeline, problems encountered, and future recommendations.

## 2. DISCUSSION

### 2.1 Project Concept

The idea behind this project has been brewing for a few years. It began by knowing students who wanted to build an application but were lacking design skills. Their app never saw the light.

Some students missed out on hackathons because they did not know where to go to form a team.

Multiple students did not get co-op offers due to a lack of "out-of-class" projects.

It is a problem that has affected a lot of us but has always been brushed off. The last straw was having a difficult time finding a senior design group. Sorting through excel sheets and discussion boards can get overwhelming quickly. Students deserved a platform where they base their connection on the idea rather than desperation.

### 2.2 Design Objectives

Our team objectives have become clearer to us during this project. We knew what problem we wanted to solve and developed our goals from there. We wanted our web application to be a well-designed, safe, accessible, and gamified site where our users have a place to go not just to find listings on a page, but to find the people they want to connect and work on projects with.

We decided to go with the space theme, henceforth, *Mission Source*.

This is a product that will hopefully be sold and used at colleges across the United States. We wanted to start with the University of Cincinnati and allow for customization and expansion as needed in the future with growth.

## **2.3 Methodology**

### **Design Requirements**

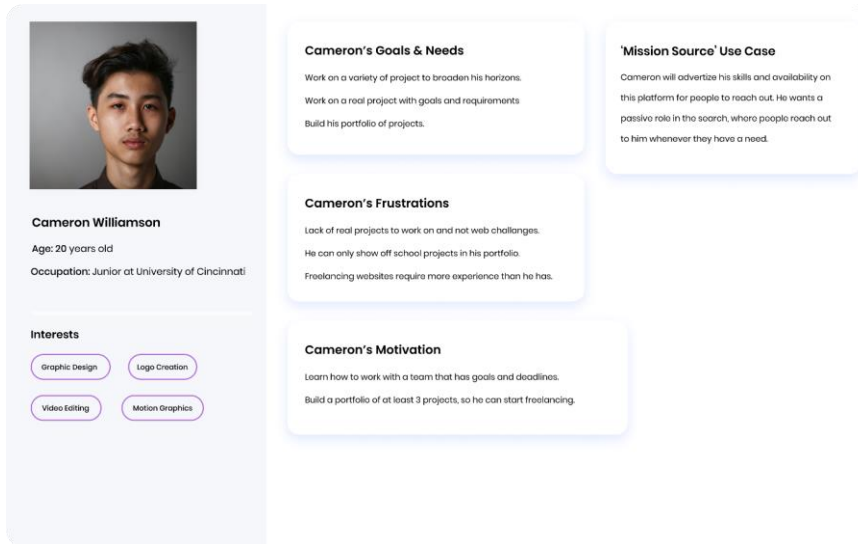
At Missions Source, our design requirements methodology consisted of the team working together at the beginning of the project to come up with a rough concept of the project we wanted to create. We brainstormed the different features we would like to see in our program and brought those potential features to a focus group. With the feedback received from the focus groups, we were able to identify features that were needed to be prioritized.

### **Procedures**

We used an Agile methodology within our team. We meet weekly to discuss the work completed from the prior week and discuss what work we wanted to include for the next week. Planning our work for the week and having the option of pulling in or taking out work based on availability for our team was crucial for our team's success. With all our team members having very busy schedules including undergraduate school, graduate school, employment, etc., an agile methodology allowed us to change our velocity week by week.

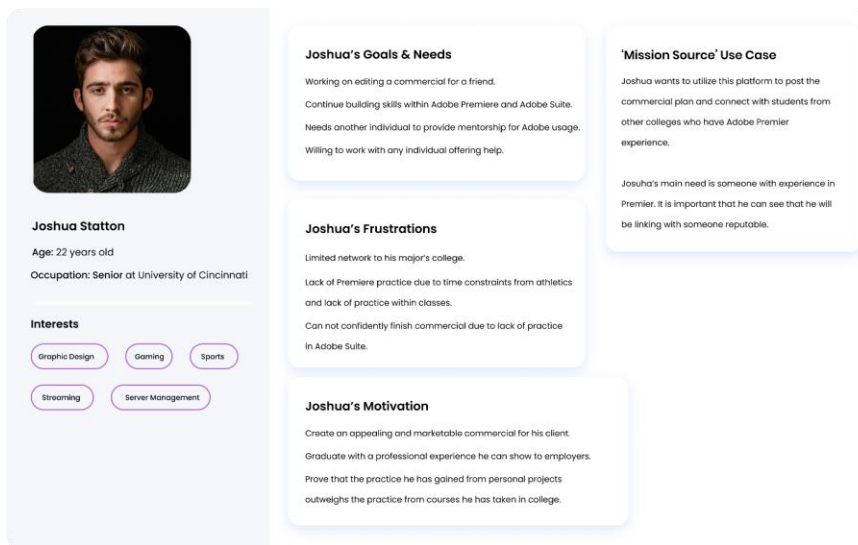
## 2.4 User Profiles

**Figure 1: User Profile – Cameron** The following diagram, Figure 1, shows an example of a user profile for Mission Source.



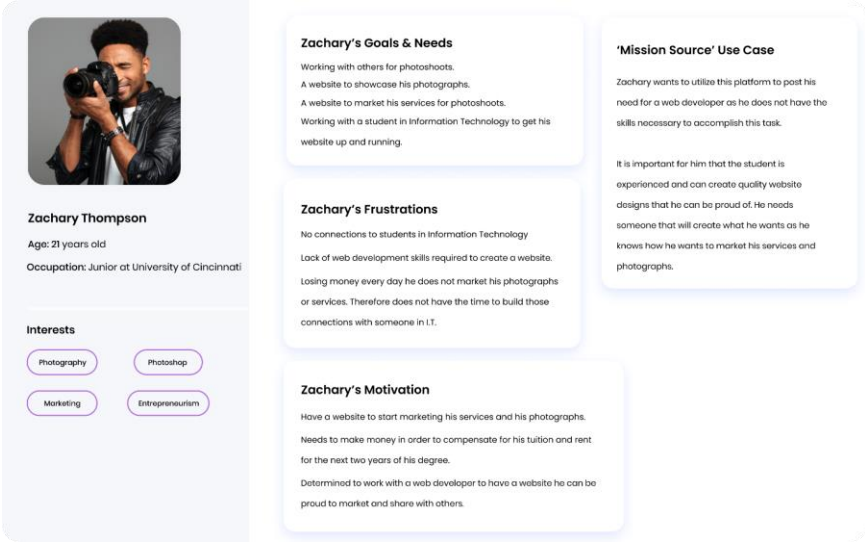
**Figure 1: User Profile – Cameron**

**Figure 2: User Profile – Joshua** The following diagram, Figure 2, shows an example of a user profile for Mission Source.



**Figure 2: User Profile – Joshua**

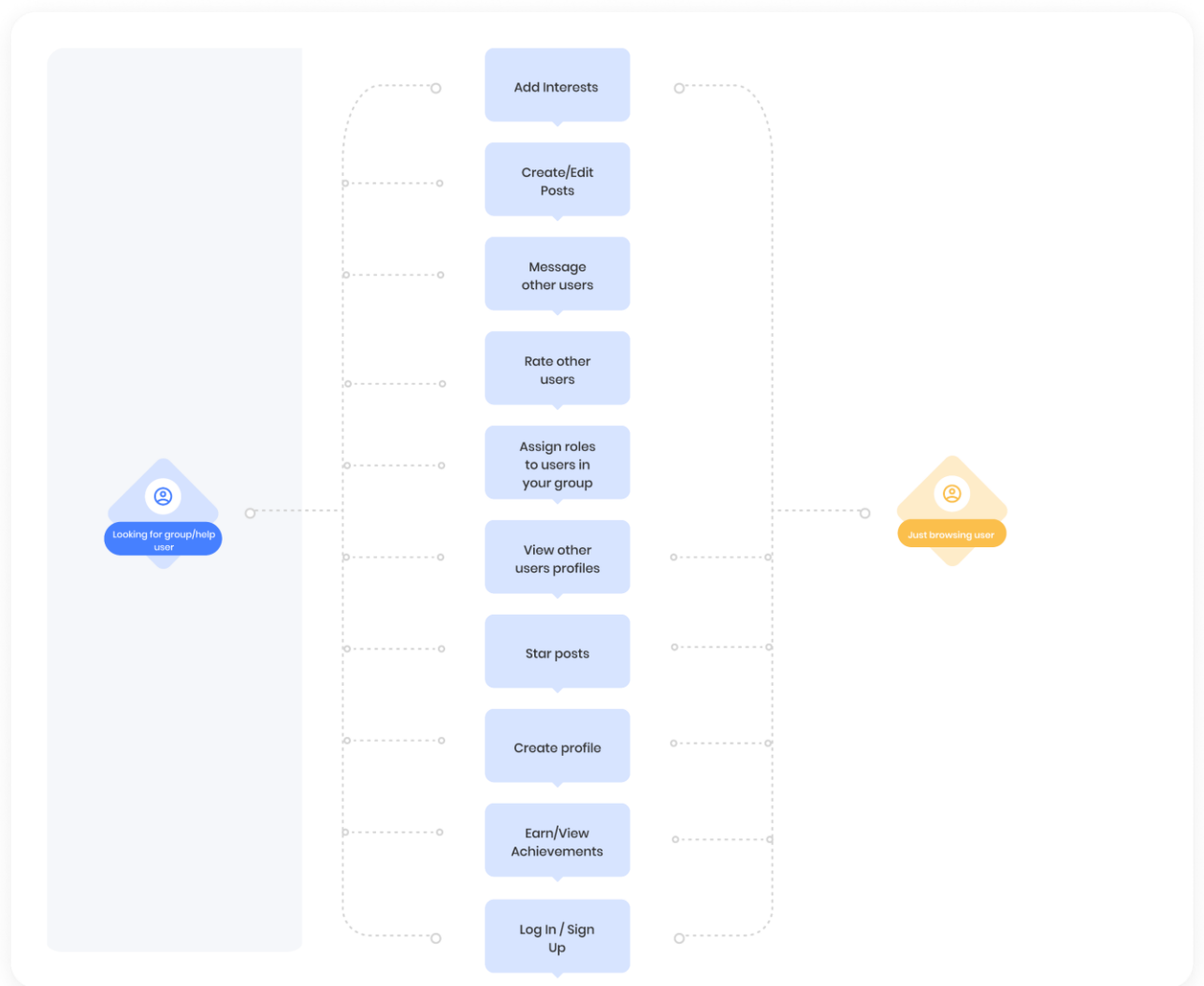
*Figure 3: User Profile – Zachary The following diagram, Figure 3, shows an example of a user profile for Mission Source.*



**Figure 3: User Profile - Zachary**

## 2.5 Use Case Diagram

**Figure 4: Use Case Diagram** The following diagram, Figure 4, demonstrates the use case for Mission Source. The diagram below shows the potential users and how they could use Mission Source.



**Figure 4: Use Case Diagram**

## 2.6 Technical Architecture

### User Interface Design

For User Interface Design, Mission Source utilized [Figma](#). Figma was used to create high-res mockups and a prototype. Allocating time for UI design beforehand, it was easy to move from one feature to another swiftly with minimal effort, as developers were aware of how the feature should look like. Moreover, creating a theme beforehand, allows for less development time spend on re-write SCSS theme, as new designs emerge.

### Client-Side Development

To develop the client-side for this application, the [Vue.js](#) framework was used. This is a JavaScript based framework that allows modular development and reusability. Vue.js supports SPA development, leading to no page reloads as the user navigates the application. The framework also utilizes “Vue Store” which allows the application to fetch the data once and store it within our app. Resulting in the ability to call specific data variables, rather than making the same API request repeatedly.

[Vue Bootstrap](#) and [Element UI](#) libraries were also leveraged to help build the UI faster. Those libraries provide basic components such as cards, buttons, etc. Meaning that those elements did not have to be built from scratch, saving development time. Moreover, using [SCSS](#), allows for overriding the library styles to fit the theme in a standardized way. SCSS makes it easy to create global style variables that we can re-use throughout the application. The benefit of doing

this is avoiding repetitions in the code, and if the value needed to be changed, it only needed to be edited in one place.

## **Server-Side Development**

[Laravel](#) was used for the server-side development. It is a PHP based framework that provided everything that was needed. Laravel was used to write up API routes and functions, create needed database tables, models etc. This framework provides many benefits, such as [Resources](#), allowing the definition of exactly what is needed to return in the JSON and creating smart links to other models. [Collections](#) allow for inheritance of all the methods that are related to a model, making route writing a lot more efficient and less cluttered.

For the chat system, [Pusher Channels](#) was used. It allows for creating a chat system between users utilizing their pre-made methods, resulting in spending more time on feature development, rather than re-inventing the chat system.

## **Database**

For the database, [SQLite](#), which is a relational database, was used. It comes as one of the pre-shipped databases with Laravel. At this time MySQL was decided against because it is not anticipated that the app will have a massive database that will have to process thousands of entries at a time. Another reason for that is that the lead developer's system is Mac OS and MySQL is not something that is able to be run on their system.

## **2.7 Testing**

### **Testing Methodology**

Testing the functionality of our web application will be done through unit and system testing. We chose unit testing to ensure that our web application renders the appropriate content, when called. Unit tests are written with every feature development to ensure that the feature behaves appropriately based on the given conditions.

We also utilized system testing for our end-to-end scenarios. This testing methodology allows us to see how the application behaves when a user is interacting with it. While unit testing ensures that the software works, system testing ensures appropriate user interaction and consistent experience throughout the application.

### **Scope of Testing**

The use cases and features we will be testing against are listed below:

- Account creation
- Browsing for missions
- Interactions with missions (liking and responding to needs)
- Mission creation
- Viewing past missions
- Ratings system
- Viewing achievements

By testing these features, we will gain an understanding of how functional the web application we built is and how effective it will be in fulfilling the needs set forth by our use cases and user personas created from the research we performed prior to starting our project.

## **Objectives**

We hope, through the testing we perform, we verify the functionality of our web application and ensure that it solves the problem we identified that led to us creating *Mission Source*. We will measure the success of the testing through the following criteria:

- a. All major features and use cases need to be accounted for and function properly
- b. The web application should perform all core functionality components
- c. No unfinished work/features should be shown during the demo at the IT Expo
- d. All bugs should be resolved before the IT Expo

## **Review**

When doing our testing, we wanted to make sure that all functions and features are properly working. Through continuous unit testing we are able to catch potential bugs and failures to resolve them early on. We edit the component and its functionality based on the unit test results, rather than the other way around. This ensures consistency. If a new feature breaks our current components, we were able to catch that locally rather than in production.

## 2.8 Budget

*Figure 5: Project Budget The figure below, Figure 5, shows the estimated costs of development and production of Mission Source in real world applications.*

| Project Asset Type  |                                 |   |                    |                | Funding Source (if applicable)  |                  |  |  |  |
|---|---------------------------------|---|--------------------|----------------|---|------------------|--|--|--|
| Revenue Generating  |                                 |   |                    |                | Self  |                  |  |  |  |
| Comments:   |                                 |   |                    |                | Comments: Funding for any hardware, and software has come from us. We have tried to use as much free and open source software as possible. Along with Open Source software, we have used student levels of software as well.  |                  |  |  |  |
| Risk Identification (See Risk Types table)                      |                                 |   |                    |                | Project Stakeholder(s)  |                  |  |  |  |
|   | Risk Rating*<br>1-5 (1 is High) | Comments  | Weight             | Score          | Blake Hanna, Cooper Kuesche, Mariya Nesterenko,<br>University of Cincinnati Students<br>University of Cincinnati Student Life Center<br>We are a new company creating this product. Our stakeholders are our customers, employees, and anyone that may purchase this service. |                  |  |  |  |
| Work Effort (days)  | 4                               | This was ranked as a medium to high in the score. We are a brand new company with a custom platform using software that is currently existing for parts of our project such as API, Database, and Web Servers | 40%                | 1.60           |   |                  |  |  |  |
| Complexity  | 3                               |   | 60%                | 1.80           |   |                  |  |  |  |
| Project Risk Score:   |                                 |   |                    | 3.40           |   |                  |  |  |  |
| Estimate of Benefits  |                                 |   |                    |                |   |                  |  |  |  |
| If project will generate revenue, estimate 1 year here:         |                                 |   |                    |                |   |                  |  |  |  |
| Select other benefits the project may bring a customer or user: |                                 |   |                    |                |   |                  |  |  |  |
| Risk Avoidance:   |                                 |   |                    |                |   |                  |  |  |  |
| Improved customer satisfaction                                  |                                 |   |                    |                |   |                  |  |  |  |
| Increased system availability                                   |                                 |   |                    |                |   |                  |  |  |  |
| Productivity or process   |                                 |   |                    |                |   |                  |  |  |  |
| Reduced costs   |                                 |   |                    |                |   |                  |  |  |  |
| Estimated Cost Rough Order of Magnitude:                        |                                 |   |                    |                |   |                  |  |  |  |
|   | Rate Per/Hr                     | Work Effort (Hours)   | 1 X Costs          | Ongoing Annual |   |                  | Comments: We have several options for external software including Azure, AWS, Google Cloud, Digital Ocean, Linode, and Heroku App. Google Cloud has a free tier that will be sufficient for our service for the time being and until growth is achieved. |  |  |
|   |                                 |   |                    | Rate Per/Hr    | Work Effort (Hours)   | 1 X Support Cost |  |  |  |
| Labor - IT  | 20                              | 640   | \$ 12,800.00       | 20             |   | \$ -             |  |  |  |
| Labor - External  | 20                              | 0   | \$ -               |                |   | \$ -             |  |  |  |
| Software - External   | 1                               | 0   | \$ -               |                |   | \$ -             |  |  |  |
| Hardware - External   | 1                               | \$225   | \$ 225.00          | 1              | \$225   | \$ 225.00        |  |  |  |
| Misc.   |                                 |   |                    |                |   |                  |  |  |  |
| TOTAL   |                                 |   | \$ 13,025.00       |                |   | \$ 225.00        |  |  |  |
| 5-Year ROI Analysis   |                                 |   |                    |                |   |                  |  |  |  |
| Description   | 5- Year Expected                |   | Conservative (1.5) |                |   |                  |  |  |  |
| Total Costs   | \$ 14,150.00                    |   | \$ 21,225.00       |                |   |                  |  |  |  |
| Total Benefit   | \$ -                            |   | \$ 50              |                |   |                  |  |  |  |
| Total Costs/Benefit Differential                                | \$ (14,150.00)                  |   |                    |                |   |                  |  |  |  |
| Conservative Costs/Benefit Differential                         | \$ (21,225.00)                  |   |                    |                |   |                  |  |  |  |

**Figure 5: Project Budget**

## 2.9 Project Timeline

*Table 1: Project Timeline* The following diagram, Table 1, shows the Project Timeline for Missions Source. The Project Timeline shows the project schedule for fall and spring semester.

| Task #            | Task Name*                         | Duration    | Start Date | End Date |
|-------------------|------------------------------------|-------------|------------|----------|
| <b>Task # 0</b>   | Team Contract                      | ~ 2.5 weeks | 08-31      | 09-18    |
| Task # 0.1        | Contract Revisions                 | ~ 1 week    | 10-5       | 10-12    |
| <b>Task # 1</b>   | UI Design                          | ~ 3.5 weeks | 09-21      | 10-5     |
| Task 1.1          | Choose Color Scheme                | ~0.5 week   | 9-25       | 10-14    |
| Task 1.2          | Create Wireframes                  | ~ 1 week    | 9-29       | 10-5     |
| Task 1.3          | Decide which user pages to mockup  | ~1 day      | 10-5       | 10-5     |
| Task 1.4          | Main Mockups                       | ~1.5 weeks  | 10-5       | 10-14    |
| <i>Task 1.4.1</i> | Home Page                          | ~1 week     | 10-5       | 10-12    |
| <i>Task 1.4.2</i> | Create a Mission Page              | ~1 week     | 10-5       | 10-12    |
| <i>Task 1.4.3</i> | Mission Info Page                  | ~1 week     | 10-5       | 10-12    |
| <b>Task # 2</b>   | Create prototype                   | ~ 1.5 month | 09-28      | 11-14    |
| Task 2.1          | Initialize the BitBucket           | ~.5 week    | 09-28      | 10-1     |
| Task 2.2          | Create landing page                | ~ 2 weeks   | 10-5       | 10-19    |
| Task 2.3          | Create login/signup component      | ~1 week     | 10-5       | 10-12    |
| Task 2.4          | Create Mockup Pages into Vue Pages | ~2 weeks    | 10-19      | 11-14    |
| <i>Task 2.4.1</i> | Create side bar component          | ~1 week     | 10-25      | 11-1     |
| <i>Task 2.4.2</i> | Create “Create a Mission” process  | ~1 week     | 10-28      | 11-1     |

|                 |                                      |            |       |       |
|-----------------|--------------------------------------|------------|-------|-------|
| <b>Task # 3</b> | Create API Pipelines and Routes      | ~ 7 weeks  | 10-5  | 11-21 |
| Task 3.1        | CRUD Operations for Post             | ~2 weeks   | 10-12 | 10-26 |
| Task 3.2        | CRUD Operations for User             | ~1 week    | 10-26 | 11-4  |
| Task 3.3        | Connect Mission Creation to Database | ~2 weeks   | 11-4  | 11-18 |
| <b>Task # 4</b> | Budget Creation                      | ~1 week    | 10-26 | 11-9  |
| <b>Task # 5</b> | User Access                          | ~3 weeks   | 11-1  | 11-22 |
| Task 4.1        | User Authentication                  | ~2.5 weeks | 11-1  | 11-17 |
| Task 4.2        | Tie Missions to Users                | ~.5 week   | 11-17 | 11-22 |
| <b>Task # 6</b> | Basic User Functionality             | ~2 months  | 1-10  | 3-15  |
| Task 6.1        | Mission Favoriting with UI & API     | ~2 weeks   | 1-10  | 1-24  |
| Task 6.2        | Grab Mission Details from API        | ~1 week    | 1-17  | 1-24  |
| Task 6.3        | Profile Page with UI & API           | ~1 week    | 1-24  | 1-31  |
| Task 6.4        | Tags                                 | ~2.5 week  | 1-24  | 2-10  |
| Task 6.5        | Ratings System                       | ~4.5 weeks | 2-10  | 3-15  |
| Task 6.7        | User Registration                    | ~1 week    | 2-17  | 2-24  |
| <b>Task # 7</b> | Security Assessment                  | ~1.2 month | 1-17  | 2-24  |
| <b>Task # 8</b> | Logo Creation                        | ~2 months  | 1-10  | 3-10  |

## 2.10 Problems Encountered and Analysis of Problems Solved

As a team with very busy schedules, coming up with times to meet all together was quite challenging. This is due to full course schedules, co-ops, athletics, and other extenuating circumstances. As a team, however, we have done a very good job of working around those issues. We stayed in communication as much as possible to ensure that everyone knows what needs to be done and what has been done. We also found times that, while unconventional

(Sunday's at 9 PM), work for our team to get work done together to meet some of the deadlines for the course. Overall, there have been times where scheduling has made things tough for the team, but we have all done a great job of pushing through and doing what we can to get in communication with the team.

### **2.11 Future Recommendations**

In the future moving to MySQL rather than SQLite would be beneficial as the userbase begins to grow. Mission Source currently uses SQLite, which is a for smaller projects and at the time there was no need for the higher performance of MySQL.

Another recommendation would be to implement a more robust achievement system. That includes achievement tracking, titles, upcoming achievements etc. That would create an even more interactive experience for users going forward.

Another recommendation would be to add a more robust editor for mission descriptions. Our team was considering adding a CKEditor to give users an ability to create very long and carefully formatted descriptions.

### **3. CONCLUSION**

#### **3.1 Lessons Learned**

Throughout the course of this project, our team has learned quite a lot about the intricacies of creating a project from start to finish with a technology and business mindset. Through this process we have learned to make decisions that will impact us positively in both scenarios as that is what is needed to create a successful project. We have also learned how to make technology decisions with not only the best fit for our project, but also keeping in mind budgeting as well.

#### **3.2 Abilities and Skills Developed Throughout Project**

Budgeting is an important skill that we have developed in this project. While making decisions about which technologies users want to use, it can also be very important to make sure they are making decisions economically that fit with their scale. It can be very easy to want the biggest, fastest, and best product, but many times that is not what is needed to fit the scope of the project.

Each team member in our group has also been involved in new areas because of this project.

Some team members have focused on business and other have focused on IT work and this project has blended those skillsets back together, allowing us to become more well-rounded as individuals and as a team.

#### **3.3 Plans for Continuation**

Current no plans have been made regarding sponsorship for Mission Source. We would love for the platform to be put to use by a university as it is a valuable asset to students along their journey to finding a career.

## REFERENCES

### 4.1 References

1. 7 Reasons Why it's So Hard to Get a Job After College (With Helpful Tips). (2020, April 14). Retrieved October 12, 2020, from <https://www.indeed.com/career-advice/finding-a-job/why-is-it-so-hard-to-get-a-job-after-college> (Indeed.com)
2. N/a, N. (2019, April 30). What can students do to improve their chances of finding employment after college? Retrieved October 13, 2020, from <https://www.washington.edu/doit/what-can-students-do-improve-their-chances-finding-employment-after-college>
3. Lerdorf, R., 1995. *PHP*. The PHP Development Team, Zend Technologies. <https://www.php.net/>
4. Otwell, T., 2011. *Laravel*. Laravel Repository. <https://laravel.com/>
5. You, E., 2014. *Vue.js*. <https://vuejs.org/>
6. Parsa, P., Regan, A., Muller, J., Mosin, V. and Morehouse, T., 2016. *Bootstrap Vue*. <https://bootstrap-vue.org/>
7. Catlin, H., Weizenbaum, N. and Epstein, C., 2006. *Syntactically awesome style sheets*. <https://sass-lang.com/>
8. 2016. *Element UI*. <https://element.ele.me/>