



# GoatConnect

## *Major Qualifying Project*

A Major Qualifying Project (MQP) Report  
Submitted to the Faculty of  
WORCESTER POLYTECHNIC INSTITUTE  
in partial fulfillment of the requirements  
for the Degree of Bachelor of Science in

Computer Science

By:

Harrison Rubin  
Isabella Pabón  
John Mezzo  
Owen Lehane

Project Advisors:

Professor Rodica Neamtu

Date:

March 1, 2024

*This report represents work of WPI undergraduate students submitted to the faculty as evidence of a degree requirement. WPI routinely publishes these reports on its website without editorial or peer review. For more information about the projects program at WPI, see <http://www.wpi.edu/Academics/Projects>.*

## Abstract

GoatConnect is a live website created by a former MQP group with the goal of having past and present student athletes form meaningful relationships and gain job opportunities. The past MQP developed an effective base application that allowed alumni to post jobs and for athletes to apply to them. However, it was missing important features that would effectively help students to build their professional network, such as an area for messaging. Along with adding additional elements to GoatConnect, we aim to refactor the current structure and migrate the application to a permanent location. Lastly, we wish to generalize this website so that other WPI athletic organizations can adopt it in the future. Ultimately, the goal of this MQP is design, implement, and test new features for GoatConnect to complete the initial objective of creating an application to help current student athletes connect with alumni and overall strengthen the football community.

## **Acknowledgements**

This project could not have been completed without the expert guidance from both Professor Rodica Neamtu and Coach Chris Robertson. We would also like to thank Alex Gutierrez and Professor Ermal Toto for facilitating meetings and assisting with the migration of the application.

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Project Objectives . . . . .	2
<b>2</b>	<b>Background</b>	<b>3</b>
2.1	Direct Messaging . . . . .	3
2.2	Accessibility . . . . .	4
<b>3</b>	<b>Methodology</b>	<b>6</b>
3.1	Project Management . . . . .	6
3.2	User Beta Test . . . . .	7
<b>4</b>	<b>Implementation</b>	<b>9</b>
4.1	Incremental Development . . . . .	9
4.2	Front-End Development for Direct Messaging . . . . .	9
4.3	Back-End Development for Direct Messaging . . . . .	11
4.4	Development for User Profiles . . . . .	11
4.5	Application Migration to WPI Server . . . . .	12
4.6	Maintenance . . . . .	13
4.7	Application Updates Due to Alumni Email Changes . . . . .	13
4.8	Documentation . . . . .	14
<b>5</b>	<b>User Test Results and Analysis</b>	<b>14</b>
<b>6</b>	<b>Conclusion</b>	<b>20</b>
	<b>Appendices</b>	<b>23</b>
<b>A</b>	<b>Evaluation Consent Form</b>	<b>23</b>
<b>B</b>	<b>Evaluation Protocol</b>	<b>25</b>
	<b>References</b>	<b>27</b>
<b>7</b>	<b>References</b>	<b>27</b>

## List of Figures

1	GoatConnect Improved Accessibility (Left) and Initial Accessibility (Right) . . . . .	5
2	Jira for Agile Methodology . . . . .	6
3	Beta Test in Unity Hall on Campus . . . . .	8
4	Mock Up of User Interface for Direct Messaging Feature . . . . .	10
5	Final User Interface for Direct Messaging Feature . . . . .	10
6	How intuitive the application was with both sessions . . . . .	15
7	How satisfied the users were with the application . . . . .	16
8	How intuitive the application was with just the second session . . . . .	16
9	How satisfied the users were with the application with just the second session . . . . .	17
10	How difficult was it to create a conversation results . . . . .	17
11	How difficult was it to send messages results . . . . .	18
12	How difficult was it to use the add user button results . . . . .	19
13	Additional feedback responses . . . . .	20
14	How difficult was it to create a conversation from a job posting results . . . . .	21
15	How difficult was it to delete a conversation results . . . . .	22

# 1 Introduction

In college, students will face numerous difficulties, including securing a job after graduation. College can be especially challenging for student athletes, as they regularly juggle academic and athletic excellence, where performance in one can potentially impact the other. This often creates an increased amount of stress due to the limited time and resources a student athlete has to build their professional networks and explore career options. In 2022, a group of three students football players, Loren DiLoreto, Jacob Feiss and Harrison Kyriacou created GoatConnect, a website aimed at helping current and former football players connect and explore career options directly. (Diloreto, Feiss, et al.).

Prior to GoatConnect, these relationships were facilitated by the team's coaches. This became challenging for the coaches, due to the increasing alumni network and their being solely responsible for exchanging important information between alumni and the team. Gaining inspiration from other platforms such as LinkedIn, GoatConnect offers student athletes the ability to create a profile to display their major, skills and graduation year. They can then access current job listings posted by alumni, complete with job descriptions and direct links to the job application and company website.

The previous team established a solid foundation for this application, but faced a limited time frame. Thus they were unable to incorporate all the desired features to GoatConnect. As a result, a key goal of this MQP is to incorporate new features into the website. Such as adding a direct messaging feature and enhancing profiles with pictures and resumes. Incorporating both of these features will additionally allow the application to achieve its objective of serving as a designated platform for alumni and students to engage in network building, and for students to explore an array of job opportunities.

Along with implementing new features to GoatConnect our team is prioritizing making the application accessible. While the web is fundamentally designed for universal enjoyment, certain websites are constructed in a manner that hinders usability for some users. Currently, GoatConnect requires improvement to be accessible. Adding this aspect to the application will make GoatConnect inclusive to all students and alumni.

Lastly, GoatConnect is currently tailored to the WPI football team, but this application could benefit all sports teams. The final goal of this MQP is to format the website to be easily adjustable to any sport, allowing all teams the opportunity to connect with alumni through GoatConnect.

By incorporating new features into GoatConnect and ensuring the website's accessibility to all users, GoatConnect will establish itself as a central platform for current student athletes to connect with

alumni. This will effectively provide students with a centralized hub to discover job opportunities and will ultimately increase community growth overall.

## **1.1 Project Objectives**

Brief Overview of all objectives in order of Implementation

Task 1: Direct Messaging

Implementation of a direct message chat feature so that users may communicate on the website.

Task 2: Design

Improve the overall design of GoatConnect, as the graphics and overall UI is not as intuitive as our team desires.

Task 3: Accessibility

Improve the overall level of accessibility of GoatConnect to ensure that users with disabilities can effectively use the website.

Task 4: Profile Pictures

Adding photos to users profiles.

Task 5: Creating Templates

Creating templates of the application as a whole, so that other sports at WPI can implement GoatConnect, and it may expand beyond just WPI Football.

## 2 Background

Last year's team had the goal of "creating a production ready and fully capable application for current student athletes and alumni to connect and find jobs or internships." Throughout their time with this project they made significant progress and set the project up for future success. As outlined in their report, there was still a lot of future work to be done for the app to improve the capabilities of the platform. From a front-end perspective, the User Interface Design was not intuitive. The profiles were still in the early stages of development, consisting of only the name of the player and a text box with basic information. Also, The current accessibility level average for the entire website was approximately 76 percent according to Google Lighthouse. Additionally, feedback from last year's beta test indicated a high interest in a direct messaging feature. One of the major suggestions was the implementation of direct messaging. Previously, there was no chat feature on the application to allow for contact between anyone. If two users wanted to discuss job opportunities, they would need to email each other separately outside the application. Thus, this year's team focused on adding a chat feature to allow for one-on-one communication directly on the GoatConnect

### 2.1 Direct Messaging

Building relationships and networking are vital when searching for a job. With the further development of the technical world, in person interactions tend to happen less frequently than they previously have. However, the integration of direct messaging features on job sites has significantly enhanced job opportunities and facilitated more streamlined professional connections by fostering direct and immediate communication between users. Thus our team decided to make this our main priority for GoatConnect. One of the most important and useful features on GoatConnect was the posting feature. After posting however, there is no way for a current athlete or alumni to communicate within the application. Instead, the only available options were an application link to the company's website and an email conversation. Both this year's team and the previous team intend for an application where previous and current student athletes can communicate directly with each other.

Many other web applications like Handshake, LinkedIn, and Facebook have a direct messaging feature. LinkedIn allows for direct communication between two or more users. The primary elements of their chat feature include, creating filtering and deleting conversations, along with receiving notifications for new messages.(LinkedIn.com). Facebook began as a social media platform, and has evolved into an application



for many different things, one of which is job searching. Facebook also has similar direct messaging features, but also includes the ability to send voice recordings and pictures or videos. Although these are not features that our team intends on implementing, it is important to explore features on other job search applications (Facebook.com).

Both LinkedIn and Facebook are great examples of how direct messaging features have evolved to cater to a wide range of communication needs, from personal connections to professional networking and business transactions. The availability and accessibility of direct messaging in these platforms outline the impact they have on shaping the way interactions occur in the digital age.

Direct messaging in applications, often referred to as instant messaging or chat functionality, has undergone a remarkable evolution over the past few decades. This communication paradigm has highly improved traditional forms of correspondence, revolutionizing the way businesses and potential employees interact in more modern times. The ubiquity of smartphones and the internet, in addition to the need for real-time and convenient communication, has driven the widespread adoption of direct messaging across various platforms and applications (Thompson et al.).

## **2.2 Accessibility**

Website accessibility, often abbreviated as "web accessibility," has become a fundamental aspect of web design and development in recent years. It refers to the inclusive practice of designing and building websites that can be used and navigated effectively by individuals with disabilities, ensuring equitable access to digital information and services (Gower). The concept of web accessibility evolved alongside the rapid expansion of the internet. In the early days of the web, accessibility was often overlooked, resulting in websites that were challenging or impossible for people with disabilities to use. These disabilities encompass a wide range of conditions, including visual, auditory, motor, and cognitive impairments. Examples of such impairments include blindness, low vision, deafness, hearing loss, limited mobility, and cognitive disorders (Gower).

Employment has become heavily reliant on the internet, and to ensure that everyone can use the internet, it is important that websites have a high level of accessibility. In addition, websites that are better developed than others, with more graphics and higher accessibility create a higher chance that someone looking for employment would use the specific website (Thompson et al.).

All websites are required to have a level of accessibility outlined by the Web Content Accessibility Guidelines (WCAG). The WCAG says that every website should be developed in such a way that they are

Pages	Accessibility (Fixed)
Applied Jobs	90
Jobs View - P	90
Edit Account	92
Feed	97
Player View	90
Jobs View - C	97
Alumni View	97

Pages	Accessibility (Before)
Applied Jobs	78
Jobs View - P	78
Edit Account	78
Feed	82
Player View	73
Jobs View - C	70
Alumni View	73

Figure 1: GoatConnect Improved Accessibility (Left) and Initial Accessibility (Right)

perceivable, operable, understandable, and robust (WCAG). Using LinkedIn for example, has adopted these guidelines and made many changes. Some of the most recent changes include immersive reader, alternative text descriptions for pictures, text scaling, bigger touch targets, and an emphasis on contrast for readability. LinkedIn constantly uses different software tools to evaluate their level of accessibility so that LinkedIn can be used by all.

To find out how accessible GoatConnect actually was, this year’s team used Google Lighthouse which is an open source software tool that looks to improve the overall quality of websites, including the accessibility of websites. As previously mentioned, GoatConnect originally had an accessibility rating of 76 out of 100, with scores for individual features ranging from 70-82 out of 100.

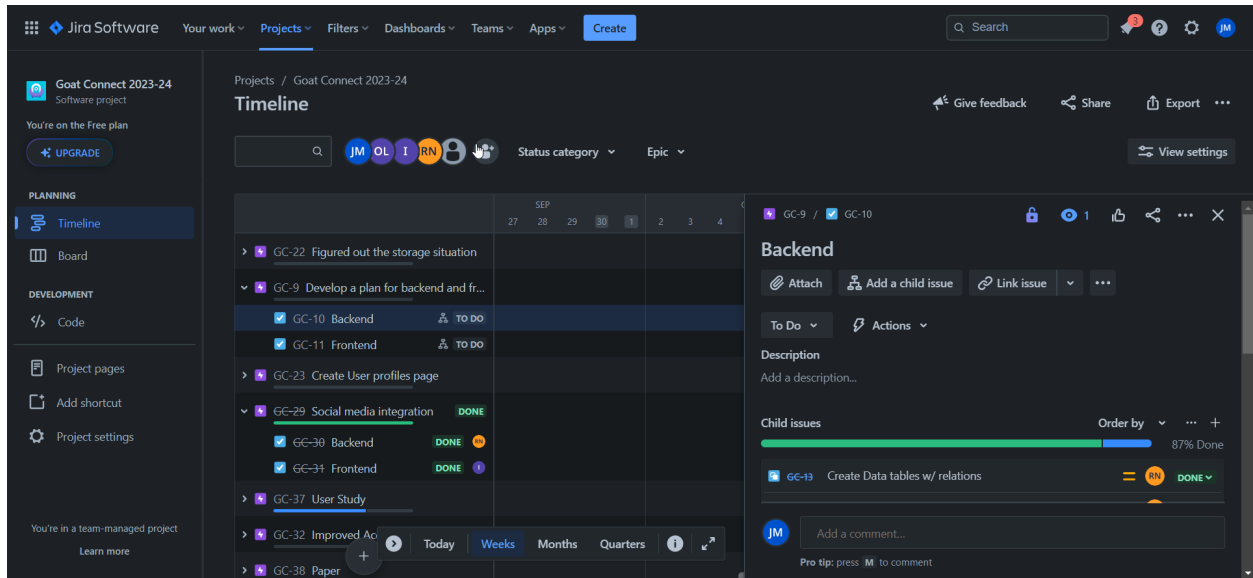


Figure 2: Jira for Agile Methodology

## 3 Methodology

### 3.1 Project Management

In all projects, teams need an organized method to complete tasks in a timely and efficient manner to produce results. It is also helpful to break large tasks into smaller objectives and use a scheduler to assign ownership and track progress. Our team chose to use Agile methodology to organize our workflow. Agile is a flexible way to plan and complete tasks in an iterative manner. Agile consists of sprints which are weekly meetings, where the major tasks and subtasks are decided by the team, and then assigned to each team member based on their skills. It also consists of scrums, which are short daily meetings for each team member to give updates on their progress and what they plan to complete for the next scrum. Jira is a project tracking software that allows team members to collaborate and post tasks to efficiently carry out the Agile methodology. Because our team had experience with Jira in previous courses, we decided to use it for this project. [2].

For our research paper, our team decided to use Overleaf because of the simplicity in formatting and familiarity our team has with this application. Overleaf is a scientific project specific software that uses LaTeX software. LaTeX software allows for the specific formatting used in scientific research papers, in comparison to the formatted text found in other software, such as Microsoft Word.

To complete this project, our team initially decided to split into pairs, with two working on the front

end, and two working on the back-end. As the project changed and the major features were implemented, one person in each pair began to work primarily on bug fixes and finishing elements of the website, while the other two teammates would focus on other important aspects to our project, such as the user study and writing our research paper.

Every Wednesday our team would meet with our advisor Professor Rodica Neamtu. During these meetings our team would present to Professor Neamtu what we did over the course of the last week, any current blocks in our process, our future plans, and any overall questions. We would only ask our sponsor, Coach Robertson to attend when we had pressing questions or needed to demonstrate a new iteration of the application.

Our team met almost daily with online scrums that lasted about 15 minutes. In addition, we would fit in one in-person meeting each week because we all agreed that adding in-person communication and collaboration would be more beneficial to the team.

## **3.2 User Beta Test**

To test the functionality, usability, and performance of the software, we conducted User Beta tests with the WPI football team where the players were able to use our application. We split the team into two forty-five minute sessions and had the offensive players attend the first session and the defensive players attend the second. While using the application, the subjects were given directions [Appendix A] to perform specific tasks. As the athletes used the application, they were asked to fill out a survey through a software called Qualtrics. They were asked to rate the difficulty of each element of the feature and of the overall intuitiveness of direct messaging in Goat Connect [Appendix B]. Throughout the study, we allowed for open collaboration between the subjects and walked around the room answering questions. The WPI Football players are the ones who will be using the application the most once deployed, so we wanted to make sure we accommodated all their ideas and requests. These User Beta tests were successfully approved by the IRB (Institutional Review Board).



Figure 3: Beta Test in Unity Hall on Campus

## 4 Implementation

### 4.1 Incremental Development

Along with following an Agile methodology, we also added incremental development to our implementation plan. Each iteration of our application was a usable version, and each piece added new user-visible functionality. Incremental programming was beneficial to our project because we were building off of a already created foundation. The past MQP had created most of the large technical features needed for the application to run, such as the database schema and server location, thus our team didn't need to focus on those items for each iteration. Instead, our features had new usable features and design elements each week.

Additionally, as the final iterations before the user test were completed, we asked small groups of users to comment on the current state of the application. As the creators of each feature, it was difficult to decipher how intuitive our designs were, thus this feedback was crucial in finalizing each feature. For example, the input we received resulted in additional user-friendly elements to the direct messaging design. These elements included: smaller buttons with icons instead of text, having the first conversation automatically highlighted instead of waiting for the user to click on a conversation, and text that notified the user that there were no conversations available instead of having a blank screen.

### 4.2 Front-End Development for Direct Messaging

For the front-end development of the direct messaging feature, the team began by designing a user-interface mock-up using Google Slides [3]. Creating a mock-up allows programmers to map out how each component will be implemented. We decided that the most important features to add would be creating and deleting conversations with multiple users, and finding conversations based on the users first and last names. We imagined the user's conversation history would appear on the left-hand side of the page, with an open conversation appearing on the right-hand side. Each conversation would display the recipient's profile picture and their first and last names. Before any implementation began, we made sure that this initial design was approved by Coach Christopher Robertson and Professor Rodica Neamtu [4].

Our first step in creating the direct messaging feature was to develop the two main components: the user's conversation history and a window for messaging. First, we implemented a basic UI such that users could view their conversation history and converse with a recipient. We then built upon this initial functionality by adding the ability to create and delete conversations. A user creates a conversation by

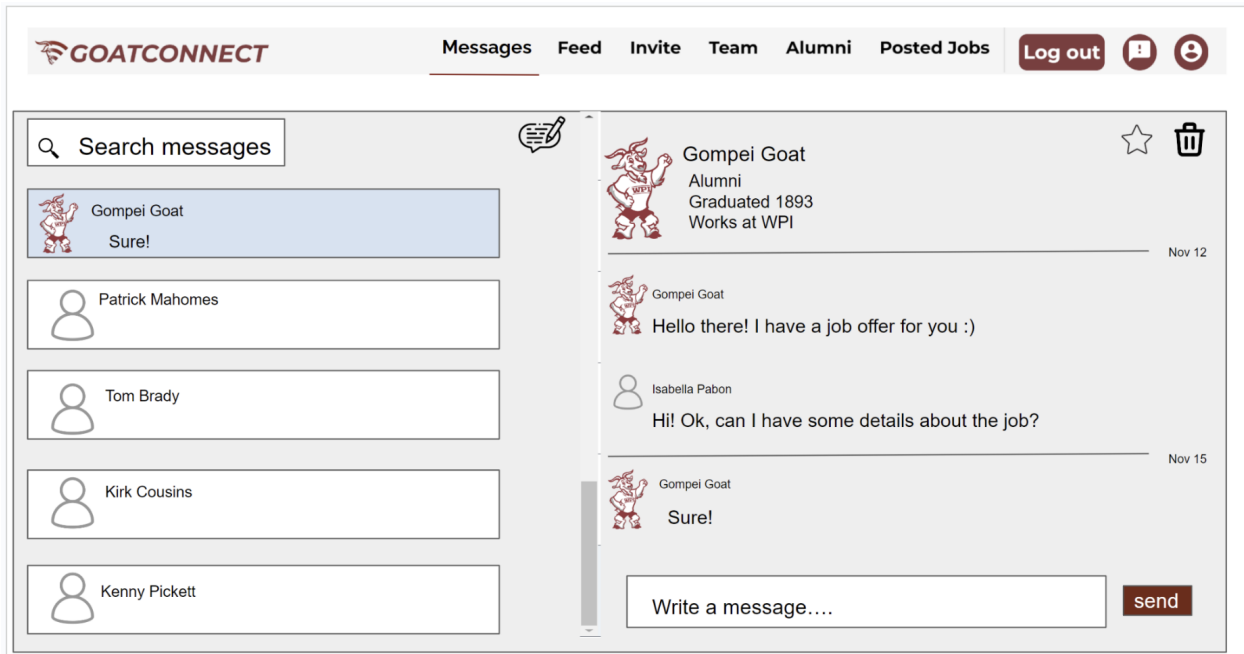


Figure 4: Mock Up of User Interface for Direct Messaging Feature

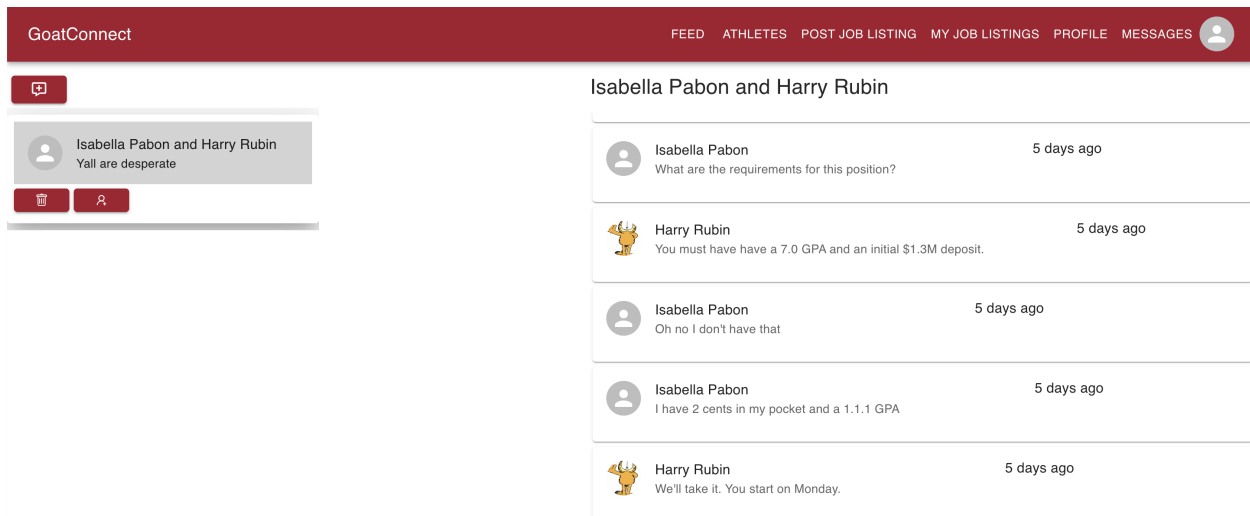


Figure 5: Final User Interface for Direct Messaging Feature

inputting the recipient's emails in a text box and our application automatically adds the conversation to the user's conversation history and displays the updated conversation list. To delete a conversation the user clicks a trash can icon that appears on each conversation to remove it from the list. Lastly, a user searches for a conversation by entering in the first or last name of one of the recipients in the desired conversation in a search bar at the top of the page. Our application will then search the current active conversations for the one including the specified recipients and return it to the user.

For the styling of the direct messaging page, we used the React Component Library MUI to create a grid based system for placing components on the page. We also used MUI's button and text components. We continued using the color palette from the previous MQP team's iteration. At each weekly meeting with Professor Neamtu, we displayed our updated progress on the feature and incorporated her feedback in the next iteration.

### **4.3 Back-End Development for Direct Messaging**

The implementation of direct messaging is straightforward in terms of database design. Put simply, using Prisma (the Object-Relational-Mapping tool) we created a Conversation object and a Message object. There is a many-to-many relationship between conversations and users, such that a single conversation can have multiple users attached, and each user can be assigned to many different conversations. Furthermore, the contents of the message are character-limited strings so that the size of the database is reasonable.

The back-end of this implementation is split across the following endpoints: `/createConversation`, `/createMessage`, `/getConversations`, `/getMessageHistory`, and `/deleteConversation`. See documentation for further details. We have made a conscious effort to avoid performance and network bandwidth issues with this design, most importantly for `/getMessageHistory`, a single conversation may have a large amount of messages, and it is crucial to avoid sending every message every time the user views a conversation. To avoid this, the `/getMessageHistory` endpoint is limited to 10 messages and whenever the user scrolls up, the front end will automatically request more message history. The design of this method assumes one major detail: all message IDs are incremental such that we can determine the relative post time.

### **4.4 Development for User Profiles**

The implementation of profile images was minimal, but effective. Beforehand, only coaches were able to see all profiles on the application. Current student athletes could only access their own profile and alumni profiles. We decided that it could be beneficial for students to be able to access their team's profiles



as well. To do this we added the page "Team" to the top of navigation bar for player.

We also added new features to all user profiles. When creating or editing their account, users will now be able to add a profile image allowing them to further personalize their accounts. Additionally, they can now add social media links, for their other work related websites and profiles. For the front end of these features, we added text boxes to the edit account forms that handled social media links and image uploads. In the back end, we added an attribute called socials, which was composed of a list of social media links, and an attribute called profile image, to each users table in the database.

## 4.5 Application Migration to WPI Server

When creating this application, the past MQP team decided to host GoatConnect on a server in Fuller laboratories. They also kept all file uploads in a S3 bucket with an AWS free trial. Although this was a fine temporary storage plan, using an AWS free trial is not a long term solution. If GoatConnect exceeded the storage limitations set by the free trial, then the previous MQP team would be charged monthly. Our team decided that the best place for GoatConnect to be hosted would be on the server that hosted all football related content for WPI. This will allow users to easily find our application and for it to be readily managed. To achieve this, we met with the assistant sports information director, several times to discuss a plan for migrating GoatConnect. Through these meeting we discovered that WPI currently uses SIDEARM sports, a popular web application provider for sports related media, to host the WPI sports websites. We engaged in discussions with a SIDEARM sports representative to discuss the details of possible migration. Mainly, we needed to discover if the SIDEARM sports server had enough space for a constantly growing application, like GoatConnect.

Although moving GoatConnect to the SIDEARM sports server would be ideal, we knew that it may not be a possibility for this MQP project. Thus, we created a secondary plan to move uploading profile pictures and resumes from AWS to the server in Fuller laboratories, where the rest of the application is hosted.

Unfortunately after many discussions with SIDEARM and WPI football administration, it became obvious that we had to pursue our secondary plan and migrate onto a school-owned virtual private server. To prevent a major rewrite of existing code, we decided to create a local S3 instance for bulk data storage. Our solution uses MinIO, which is a high performance, open source object store. It is licensed under GPLv3, which allows us to use it privately, with no cost. For our relational storage, we are still using PostgreSQL under a local instance. With both solutions in place, no outside data store is used.

We also migrated the way outgoing emails are sent. Originally the application used AWS to send their emails, which we had to change. After communication with WPI IT Services, we were able to acquire the goatconnect@wpi.edu address along with credentials to send emails through the school's Microsoft365 email server.

## 4.6 Maintenance

To ensure the longevity of the project, our team has implemented some automation and tools to make sure that the web-server is always online, and up to date. To achieve this we used pm2, a process manager for Node.js that will automatically start and make sure that the process is online. Pm2 is an advanced application that can handle automatic restarts, monitoring, and notifications when an unexpected error occurs.

Additionally, we have implemented certain automatic scripts such as SSL Certificate automation. An SSL certificate is a security measure to prevent against man-in-the-middle attacks. Prior to our User Study, our SSL certificate expired. To prevent this from occurring again, we have implemented a cron-job that will automatically renew and deploy the website certificate.

## 4.7 Application Updates Due to Alumni Email Changes

Early in January 2024, WPI released a article that stated that all WPI alumni email addresses will be converted into forwarding addresses as of July 12, 2024. This change had several unfortunate implications for GoatConnect. With a forwarding email instead of keeping their current WPI email addresses, alumni will no longer be able to access any WPI applications. Prior to this announcement, all GoatConnect users had created accounts using their WPI email addresses and the application initially included a security check to verify the WPI emails. Additionally, users also had the option to use WPI's SSO authentication system to both create an account in GoatConnect and login. Fortunately for the team, we were able to find a work around for this issue while maintaining the security of the website.

To quickly resolve this issue, our team decided to test if a user could create an account with their personal email addresses. We invited a user with a @gmail.com email address and a @chrysalix.org email address. Thankfully, both users were able to successfully create accounts on GoatConnect without needing WPI email addresses. After we found this solution we decided to delete the previous accounts, because there was a minimal number of users in the database and GoatConnect does not have a 'change email' feature. To have all WPI football team members sign up to the application Coach Chris Robertson sent a excel sheet to

the football team. The athletes were asked to input their personal email addresses and confirm they created their accounts successfully. Once all current WPI Football players had been invited, we were able to open the website to the Alumni and have them sign up.

## 4.8 Documentation

Another crucial part to creating a maintainable codebase is the creation of documentation. Because this project has spanned multiple years and many developers have touched it, it is very unlikely someone will immediately understand every part of how this application functions. Therefore documentation is useful to give both a high-level outline of how it works, as well as how it is implemented.

For every endpoint, there is a summary of what it does, including both the expected inputs and outputs. This allows a front end developer to use these endpoints easily and predictably, without having to reverse engineer them.

We've also created documentation outlining how to maintain the application from a system administrator standpoint. This includes restarting the application, making sure the SSL certificates are up-to-date, and that both the relational and bulk data storage are running.

<https://www.overleaf.com/project/64f664698ac318d2b2e3a591>

## 5 User Test Results and Analysis

Our User Study for GoatConnect was conducted on November 2023 and was comprised of two sessions. The first held approximately fifteen offensive players and the second session had approximately thirty defensive players. Both of these sessions allowed our group to view users interacting with the direct messaging feature and collect feedback on its performance.

In the first session, the participants noticed an error in the application that prevented them from using the create conversation button. During the user study, our team found the issue causing this error and fixed it before the second session, but the results show that the bug greatly influenced the first groups view of our feature.

Overall, we received feedback that most users felt the feature was either extremely intuitive or somewhat intuitive [5]. Three users felt that the design was somewhat confusing and one user found it extremely confusing [6]. We also noted that most users felt satisfied with their experience on the website,

Q2 - Overall, how intuitive was completing the tasks requested during this activity?

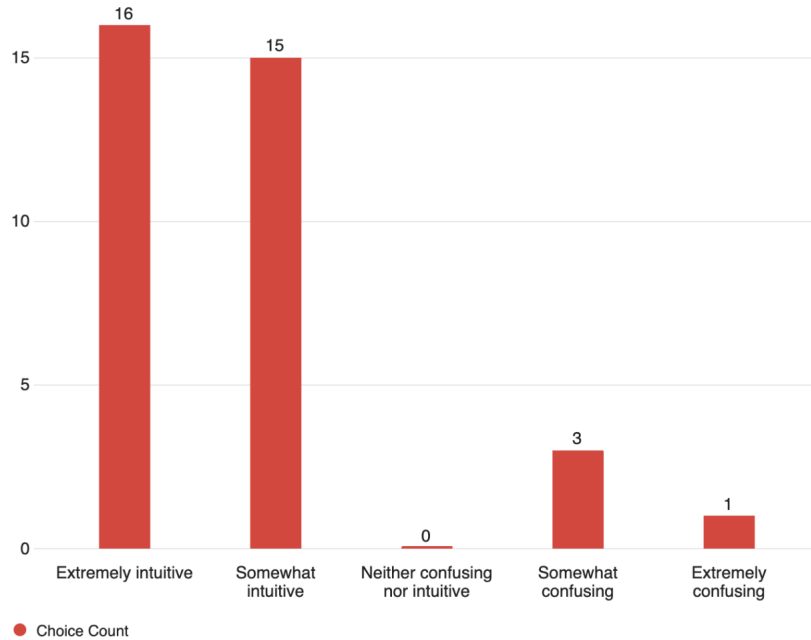


Figure 6: How intuitive the application was with both sessions

with only two users choosing somewhat satisfied and two users choosing extremely satisfied.

To see how the bug affected the results of this question, we created two charts with just the data from the second session. In the first chart, only one person found direct messaging to be somewhat confusing and most found it to be extremely intuitive [7]. In the second chart, only one person found the entire experience to be somewhat confusing and no users found it to be extremely confusing [8].

After reviewing the users overall feedback on the application, we analyzed how each user felt about each specific element of the direct messaging feature. The users started the survey by using the 'create conversation' button to create a conversation with the first recipient, Harrison Rubin. We decided that the users would converse with us during the study instead of each other to allow them to complete the survey at their own paces. The majority of users found this feature extremely easy or somewhat easy to use [9].

However, one user reported that it was somewhat confusing and three found it extremely confusing. We believe that this outcome was most likely because this button was not working for the first session.

Next, the users were instructed to message Harrison back and forth until they have sent three messages in total. This was to test the 'send' button and how quickly messages were sent and received. Again, most of the users found this extremely easy, with two users finding it neither easy nor difficult [10].

Q1 - Overall, how satisfied or dissatisfied were you with your experience on our website today?

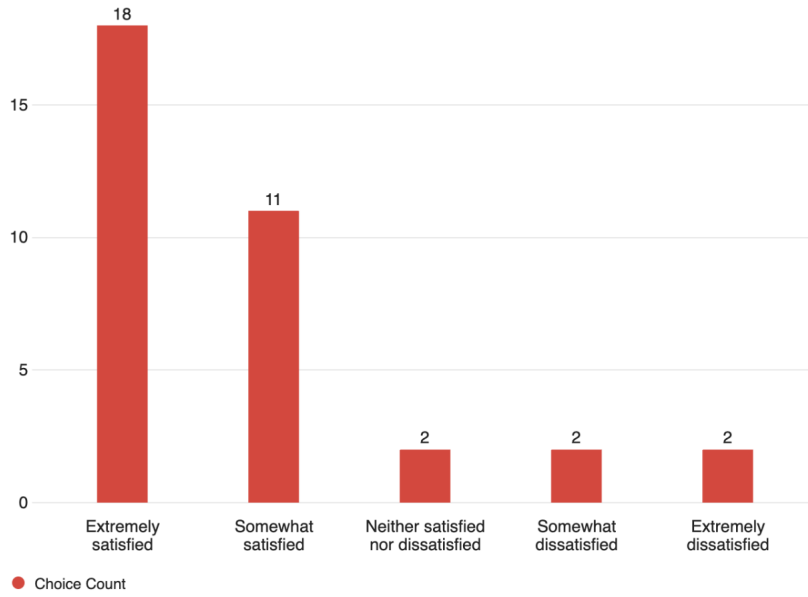


Figure 7: How satisfied the users were with the application

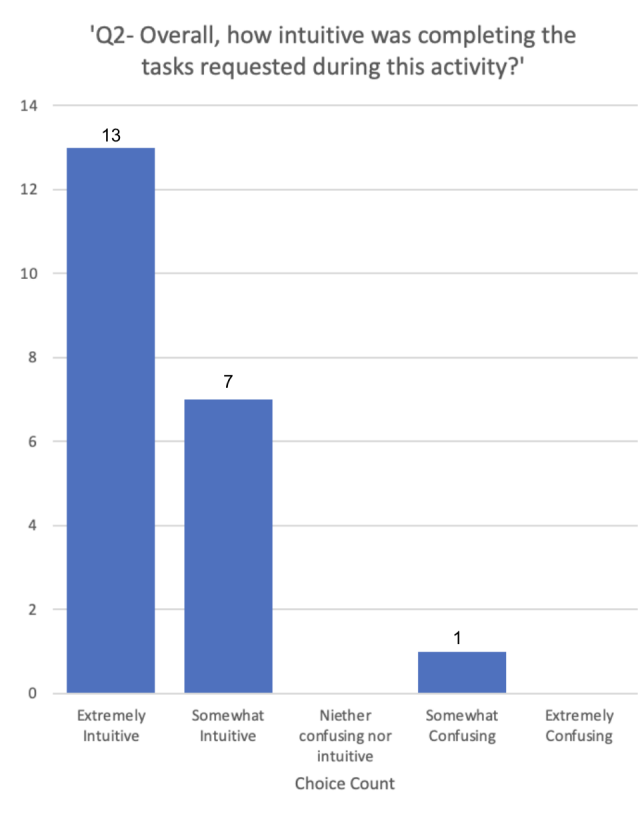


Figure 8: How intuitive the application was with just the second session

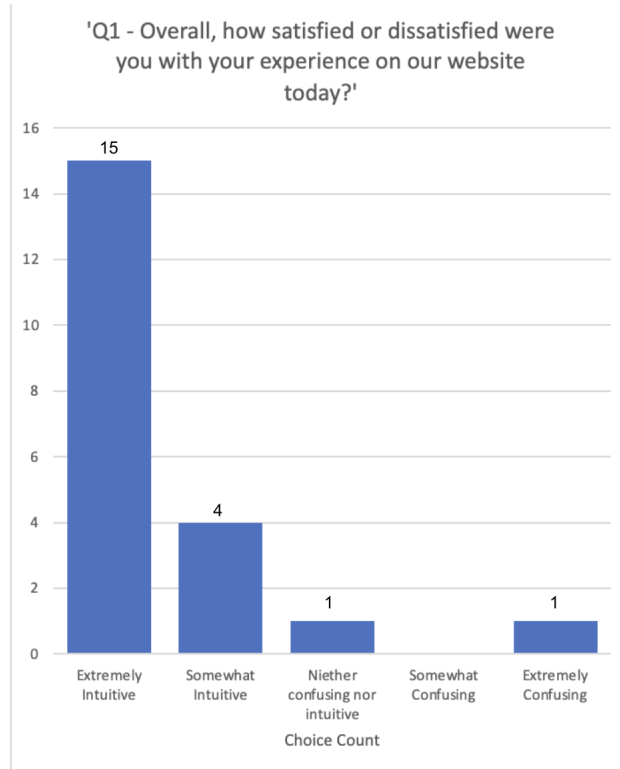


Figure 9: How satisfied the users were with the application with just the second session

Q2 - To begin, please log in to GoatConnect with the email and password combination provided for you. Once you have logged in, navigate to the Messages tab on the navigation bar and click on it. Once on the page, click the create conversations button. After clicking, enter the email labeled "first user" on the screen and click submit. After clicking submit, please indicate how difficult it was to create a conversation on the scale 1-5, with 5 being extremely easy.

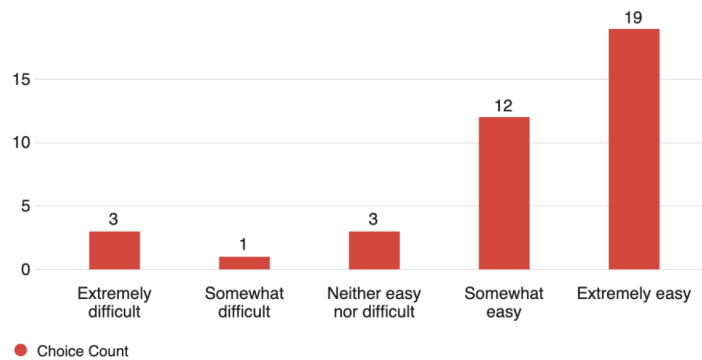


Figure 10: How difficult was it to create a conversation results

Q3 - Next, type a message in the conversation you created and hit "Send." Please text back and forth until you have sent three messages. After finishing the conversation, please indicate how difficult it was to send messages, with 5 being extremely easy.

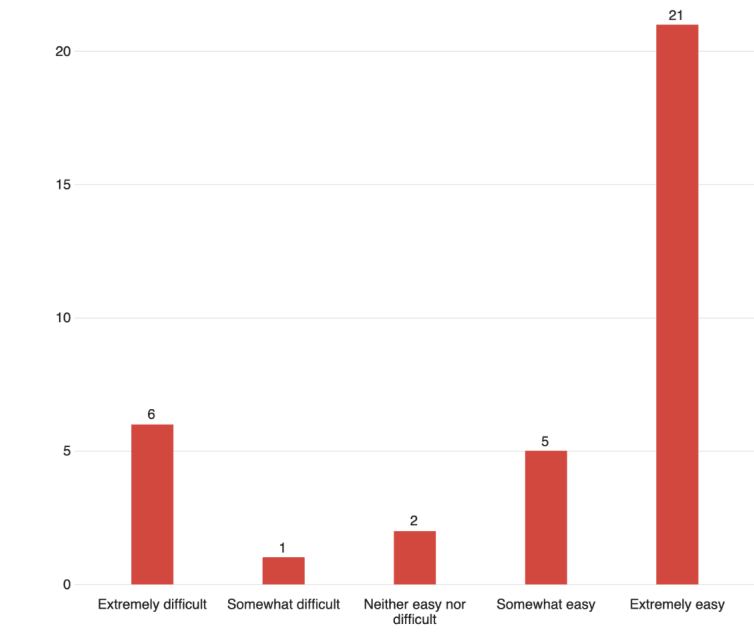


Figure 11: How difficult was it to send messages results

Six users did find this feature to be extremely difficult and one user found it to be somewhat difficult. We theorize that this could be because some users voiced that they were not receiving a quick enough response to their messages.

The users were then asked to create a new conversation with the second recipient, test user. Isabella Pabon was using the test user account to respond to the users messages. After creating the second conversation, they were then instructed to use the 'add user' button to invite another user to this conversation. The third user was test coach, with John Mezzo using this account. This feature received the most variety in answers [11]. Most users did find it extremely easy, but more than usual found it to be somewhat easy. Four users found it to be somewhat confusing and four users responded that it was extremely confusing. Based on the users open response answers were they were asked if they had any additional feedback, there were a few problems with the 'invite user' button [12]. One user stated that the feature was a little "buggy" and another stated that the feature is "glitchy" and doesn't notify the user when the additional recipient was added. Another user stated that if they didn't type the users name fast enough it would disappear from the bar. All of the bugs discovered from the study were fixed and a message now appears when a user is successfully added to a conversation.

Q4 - Click on the create conversation button again and type in the email labeled "second user." Once the conversation has been created, click on the "Add User" button. In the text box, type the email labeled "third user" and click submit. In this new conversation, text back and forth until you have sent three messages.

When finished, please indicate below how easy it was to use the invite user button, with 5 being extremely easy.

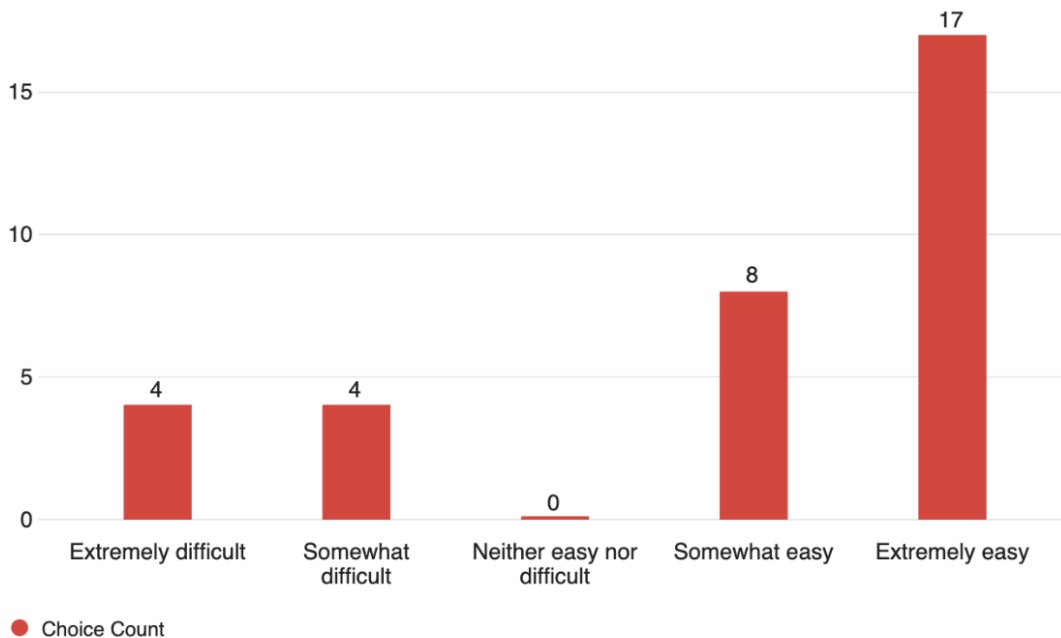


Figure 12: How difficult was it to use the add user button results



Complementary
Great job
The software is very straight forwards, making it very easy for the user which is good
Very easy to manage, simple and allows you to create conversations very quickly.
i really like the idea and look forward to using it in the future.
I like the simplicity of everything
I really enjoyed using the app and the interface is easy to use.
I like it guys this is good. Easy to use, very intuitive.
I think the website is very easy to use
Worked great, easy to use, and definitely looks effective.
Constructive
Kept running into the error that the conversation already exists
Would not allow me to create conversations
Email was not working for making the conversations
The messages tab allows me to put in the email but a conversation doesn't show up
Was unable to complete first portion, the rest was easy to follow
For the job opportunities though, the layout could be a little better, as it could be a little more clean and easier for the user.
When trying to make the conversation with the second and third user it wasn't working and if I didn't type fast enough then it would clear the conversation
The invite user to conversation was little buggy for me
When creating a conversation from the job opportunities page, may want to have that conversation labeled with the job/company
The add user to a conversation chain is kind of glitchy and doesn't tell us if the user has accepted the invite
I would love if the website had a few photos of alumni and students playing football.
The title on the job "software engineer internship was overlapping
Netural
nothing
Conversations moved in order.

Figure 13: Additional feedback responses

They were then asked to navigate to the 'job opportunities page' from the navigation bar and find a job posting. On the job posting, we instructed them to click the 'create conversation' button to create a conversation with the alumni that posted the job. The majority of the users found this to be extremely easy, with seven finding it somewhat easy [13]. Only one user found it neither difficult nor easy and two users found it to be extremely difficult. We theorize that these users may have found the layout of the page confusing. In the open response section, a few users commented that the job opportunity page "could be a little more clean" and another stated that the title was overlapping [12]. To incorporate this feedback, we changed the font and fixed the errors with the design so the title no longer flows off the page.

Their last task was to use the 'delete conversation' to delete all of their conversations. Almost all of the users found the delete button to be extremely easy to use, with three finding it somewhat easy and only one user finding it extremely difficult [14].

## 6 Conclusion

The initial goals of this project were to implement a direct messaging feature, enhance the overall design, improve the accessibility, add icons to use profiles, and generalize the application. By the end of

Q5 - Now, go to the navigation bar and go to the page "job opportunities." Find a job posting and click on the "create conversation" button. After the conversation is created, please indicate below how easy it was to use the create a conversation using the button on a job posting, with 5 being extremely easy.

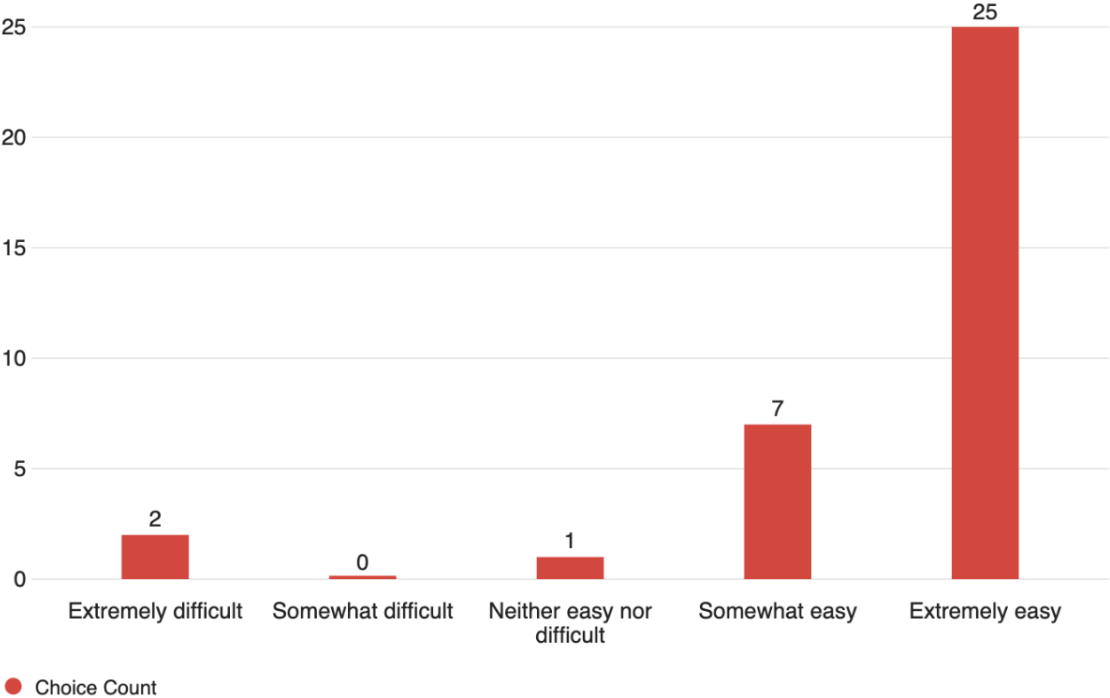


Figure 14: How difficult was it to create a conversation from a job posting results

Q6 - Now, please delete all of your conversations using the delete button. After you have finished, please indicate how easy it was to delete a conversation below, with 5 being extremely easy.

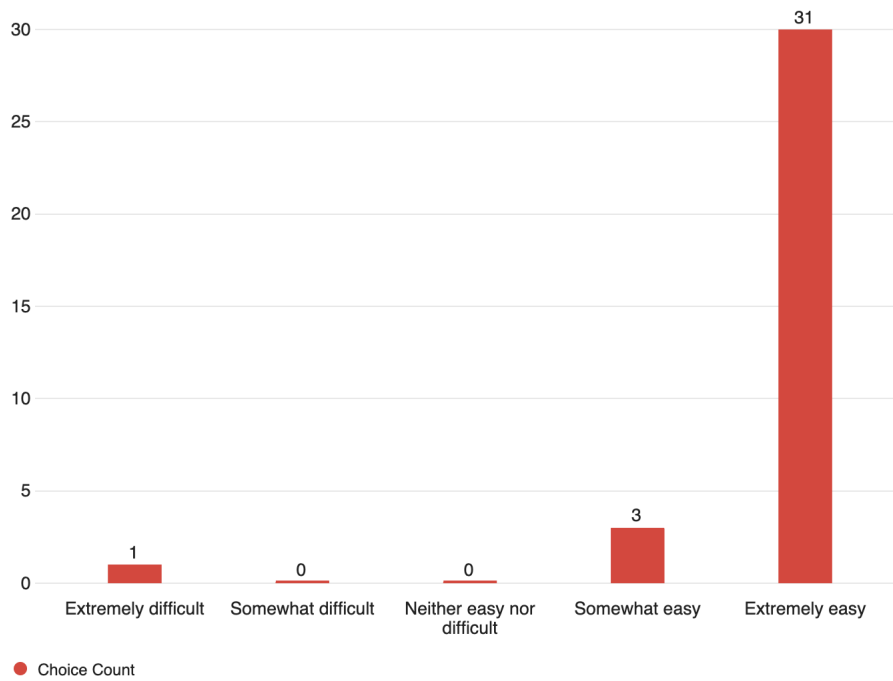


Figure 15: How difficult was it to delete a conversation results

C term, the application was officially live, and the new direct messaging feature was available to all users. Additionally, users can now upload profile icons and resumes to their personal accounts, and the accessibility has been marginally improved. Lastly, we migrated Goat Connect to a permanent location on a server in Fuller Labs. Thus, the application will have enhanced storage, security, and durability for the foreseeable future.

Although our team accomplished the goals set at the beginning of the project, there are additional features and items that we have designated as future work, potentially for the next possible MQP team. First, we suggest implementing a posting feature. This would allow current athletes to create posts when they receive a job offer or the coach to share general news updates about the football team. Another possible improvement we could make to GoatConnect, is the integration of a mobile app. Having a dedicated mobile app can enhance recognition and establish a stronger presence for the software. It reflects a commitment to delivering a quality user experience, with the potential to increase user activity from the athletes.

## Appendices

### A Evaluation Consent Form

GoatConnect: User Study Consent Form

Investigators: John Mezzo (jcmezzo@wpi.edu)

About GoatConnect: GoatConnect is currently an active website and resource for members of the WPI Football team to look for internship and job opportunities, however, it is missing key features that would allow these athletes to better utilize this tool.

Purpose: The purpose of this study is to obtain feedback on our current application to refine its current state to be able to deploy for use.

User Study Details: This study will consist of a list of directions to follow that take you through the steps of our features. After you finish with the app instructions, a link will take you to a Qualtrics Survey where you will answer a series of questions about your experience. You will then be able to give written feedback about any possible improvements for each feature. The study should take about 15-20 minutes.

How your data will be used: The data we collect from your survey will be saved in a record to be analyzed. This data will be anonymous and sensitive information will not be included in the record.

By checking off “Yes” you consent to participate in this study. By checking off “No” you do not consent to participate in this study.

## B Evaluation Protocol

### User Study Protocol

The purpose of this user study is to assess the effectiveness of the recently integrated direct messaging feature into the prototype website GoatConnect intended to be used by current and former student-athletes. Recruited participants will receive an email including a link to the website, a link to a Qualtrics survey, temporary login credentials, and three other participant emails that they will be corresponding with. Within the survey, participants will be asked to perform predefined tasks and respond to questions. Following the execution of all of the tasks, the participants will be asked questions regarding their experience using the application in general. The survey responses will be recorded through Qualtrics. Additionally, the participants' interactions with the app will be logged to assess timing and application performance. The following are potential tasks that each participant will be asked to do:

Steps: Log in to the application with the username and password provided to you Navigate to the Messages tab in the navigation bar at the top of the page and click on it Click the "Create a Conversation" button Enter the email titled: "First Correspondent" in the email provided to you and click "Submit" Click on the conversation you created, type a message, and click "Send" Wait for a response After receiving a response, type another message and click "Send" Have a short conversation (around 30 seconds) Click the "Jump to the last message" button Click the "Create a Conversation" button Enter the email titled: "Second Correspondent" in the email provided to you and click "Submit" Repeat step 5-8 with this conversation On this conversation, click on the "Invite User" button and type the username titled "Third Correspondent" in the email provided to you and click "Submit" Repeat steps 5-8 Navigate to the Job Listings page from the navigation bar Locate a job listing and click the "Create Conversation" button Repeat steps 5-8 Navigate to the search bar and type in one of the recipients in your conversations Delete all three of your conversations using the "Delete conversation" button

Questions: Please answer the following questions on a scale of 1-5 with 5 being the most difficult: How difficult was it to send and receive messages? How difficult was it to create and delete messages? How difficult was it to search for a conversation? Generally, how difficult was it to use the direct messaging feature? Please answer the following questions on a scale of 1-5 with 5 being a long time How long did it take to create a conversation? How long did it take to delete a conversation? How long did it take to send a message? How long did it take to receive a message? How long did it take to add another user to an existing conversation? How long did it take to create a message using the "create conversation" button in

a job posting? Please answer the following questions in a few words. Are there any additional features or functionalities you would like to see integrated into the direct messaging feature? Did you find the design of the website appealing? Is there anything you would like to change about the design? Please provide a brief summary of your experience and any questions or comments you may have.

## 7 References

Diloreto, Loren, et al. “Goatconnect.” Goatconnect.wpi.edu, goatconnect.wpi.edu/. Accessed 10 Oct. 2023.

—. Goatconnect Major Qualifying Project. Edited by Rodica Neamtu, 1 Mar. 2023. Accessed 10 Oct. 2023.

Gower, Michael. “Reframing Accessibility for the Web.” A List Apart, 3 Feb. 2015, alistapart.com/article/reframing-accessibility-for-the-web/. Accessed 10 Oct. 2023.

Hong, Yili, et al. “Just DM Me (Politely): Direct Messaging, Politeness, and Hiring Outcomes in Online Labor Markets.” Information Systems Research, June 2021, <https://doi.org/10.1287/isre.2021.1003>. Accessed 7 June 2021.

“Intuitive Programming.” Sensetalk.com, [sensetalk.com/intuitive/intuitive/intuitive.html](https://sensetalk.com/intuitive/intuitive/intuitive.html): :text=(chiefly

“Overview.” Chrome Developers, 27 Sept. 2016, [developer.chrome.com/docs/lighthouse/overview/](https://developer.chrome.com/docs/lighthouse/overview/).

Selcher, Marissa. “Our Journey to Make LinkedIn More Inclusive and Accessible.” Www.linkedin.com, 28 Apr. 2021, [www.linkedin.com/pulse/our-journey-make-linkedin-more-inclusive-accessible-melissa-selcher/](https://www.linkedin.com/pulse/our-journey-make-linkedin-more-inclusive-accessible-melissa-selcher/). Accessed 11 Oct. 2023.

Thompson, Lori Foster, et al. “E-Recruitment and the Benefits of Organizational Web Appeal.” Computers in Human Behavior, vol. 24, no. 5, Sept. 2008, pp. 2384–98.

W3C. “Web Content Accessibility Guidelines (WCAG) Overview.” Web Accessibility Initiative (WAI), 2018, [www.w3.org/WAI/standards-guidelines/wcag/](https://www.w3.org/WAI/standards-guidelines/wcag/).