WARE

Web Application for Robotics Education

University of Nevada, Reno
Department of Computer Science and Engineering

Software Requirements Specification

Team 17
Zachery Wiles, Ryan Lunt, Sean Griffith, Herman Hira

Instructors: Sergiu Dascalu and Devrin Lee

External Advisor: Ben Gallagher

November 5, 2020

Table of Contents

1. Introduction

This project will focus on the creation of a web application that can be utilized by students engaged in robotics classes. The web application will require that students have accounts that they will be able to log into and use the app. Once logged in, students will be met with a list of robotics environments related to their class. The environments will not only have a description, but an area for students to write code in python and the ability to submit that code to the web server for processing. Students will also be able to view the results of their submission using a 3D model generated through processing their code. The application will compile all the code on the backend of the website. Consequently, particular attention will be focused to ensure that the application is fully functional and runs smoothly so that users will be able to utilize this tool on a variety of devices. The team will use HTML, Bootstrap, JavaScript, and Flask to create an elegant and functional website that allows users to effectively interact with the back-end web server and its functionality. There were no significant changes from the previous paper, but rather more the team has become more focused on what needs to be implemented and what the users will need in order for this web application to be successful. In addition, Ben Gallagher has become an established advisor for the team, and has agreed to aid the team in the development of WARE.

2. Summary of stakeholders' interviews

The following is a summary of the stakeholders that took part in the interviews, the questions given to the interviewees, and the answers received. The team appreciates the time each one of the stakeholders that were interviewed, as their answers will give quality guidance to the team in developing WARE.

2.1 Participants

Three people took part in the interviews conducted by the team. The team's advisor, Ben Gallagher, was chosen to take part in the interview as his experience and knowledge in web development will be invaluable to the team's success. The second person interviewed was Rui Wu, the project's sponsor. He was chosen as he originally came up with the idea of a robotics learning platform and has knowledge of what the team needs to develop with robotics students in mind. In addition, Herman Hira, a member of the team, was chosen to take part in the interview as he has some website development experience.

2.2 Interview Questions

The following questions were given to each interviewee to respond to:

1. In your opinion, what does a good design look like for a website?
2. Think of past online tutorials and interactive learning exercises that you have used. What attributes did these applications have that made them valuable to the end user?
3. When developing the front-end website, should it be a priority to implement user accessibility?
4. With the platforms being used (Bootstrap, JavaScript, Flask, database), what are some ways to ensure that the website has good performance?

5. Thinking about the users of the application (robotics students), what would you say is the most important to them: Performance, Ease of Use, or the Perceived Value of using the application (i.e. learning robotics principles and applying what they have learned)?
6. What tools or techniques can the team use to ensure that the web application is secure and resistant to attacks (like XSS attacks)?
7. How should the team implement a secure sign-in process and manage user credentials?
8. What processes do you use after a web application has been developed to keep it up to date with new content and bug fixes?
9. From personal experience, do you know of any components in Bootstrap that would be better to develop from scratch rather than using what Bootstrap provides?
10. Think of a unique feature that you would want implemented in this web application that would make it stand out from other learning platforms that you have used.

2.3 Ben Gallagher's Responses

The interview questions were submitted to Ben Gallagher by email on November 2, 2020. He replied the same day with his responses. Below is a summary of his thoughts for each question.

1. Users navigating to a new website should be instantly able to intuitively interact with the website with minimal learning curve. Additionally, choosing a color theme and designing buttons and controls based on specific colors corresponding with actions will further help the user in identifying what each element does.
2. I prefer to interact and enter code at each step of the process as if I am doing it live. It would be preferable for the website to give a brief overview of the features that the user can take advantage of, but nothing more. There should be a dedicated space for documentation that the user can refer to if they would like additional information.
3. Proactively design the website with accessibility rather than reactively implementing it later on. Additionally, implementing "Progressive Enhancement" will ensure the website is still usable even if some of the more advanced technologies fail to load.
4. You will overall have much more control of the performance of the web application if most processing is done on the back-end. Use existing Python libraries, understand how browsers cache information, and only use the Bootstrap and JQuery libraries that are necessary for maximum performance.
5. I can imagine that robotics students would want to hit the ground running with such a system. Therefore, ease of use and performance are going to be critical.
6. Keep watch for updates to repositories and remain vigilant of any security patches released. Use a vulnerability scanner to reveal any bugs or security issues. Use CSRF tokens in forms, and sanitize all form inputs.
7. At the least, you would want to utilize a database that stored hashed passwords and unique salts. Use third party libraries that have been tested and verified. Ensure there is some sort of verification such as CAPTCHA during the account creation process.
8. Git has become a key part of my software development process. Using separate branches and having restrictions on direct commits reduces the likelihood of having buggy code enter into production. Use Docker containers to deploy your application to maintain stability. Comment your code and update your documentation frequently.

9. The only Bootstrap element that I always replace is the dropdown component. I use Select2 which is cleaner and very useful. There are other design libraries based on Bootstrap that give a different style, but these always have the potential to add overhead on the system.
10. I would appreciate a learning platform that can serve as a quick reference as well as an in-depth learning experience. That way I could see how the small components fit into the bigger picture. Also, adding a forum and a service to request changes would make the website a one-stop shop.

2.4 Herman Hira's Responses

The interview questions were submitted to Herman Hira by email on November 3, 2020. He replied a day after with his responses. Below is a summary of his thoughts for each question.

1. I think a good design for a website is for it to be accessible and easy for the user to find everything they need. That means obvious tabs for users to utilize and access, font size big enough for people to read, etc. The aesthetics of a website are also important because if users are going to stare at it for a while, the colors should be softer and easier on the eyes like pastels.
2. One attribute I remember in an HTML coding tutorial was that when I made a change, I could see that change in the virtual machine on the right. Being able to not only learn what you are doing, but to test out, manipulate, and discover what other things I can do with the code made the experience that much more memorable. So the user being able to see what they are able to do in real time - whether it be coding or some other learning activity - helps users learn better and it doesn't require downloading any other programs.
3. User accessibility is super important as we want to make sure that everyone will be able to utilize the website. As engineers, we need to be able to accommodate users of all types by utilizing text-to-speech, larger fonts, and making sure everything is structured so that users can utilize our website the best way possible.
4. One thing we can do to increase the website performance is focus on the compilation of code and make sure it runs smoothly. Since there may be multiple users on at the same time, we want to ensure the website won't lag or slow down. To do that, we will focus on the backend and making sure performance is optimal.
5. The most important thing for robotics students is probably learning as well as performance. The robotics students will want an easy to use application that allows them to explore and test their code in an online environment. Focusing on giving the students an application that can compile and perform optimally will allow the students to learn with less frustration and practice at their own pace.
6. I do not have much experience in web applications and security, but I think one way to help with security is built into the design of the program. This means using design patterns that will encapsulate information so that only certain classes can access that information and overall just allow for more security within the code itself.

7. Our team needs to focus on creating a secure database for all this information and the user data that will be stored. This means creating a back-end system utilizing reliable third party libraries to make sure all user data is safe and secure.
8. We can utilize GitHub to push and update our code so that it is up to date. This will allow us to update and remove bugs if we need to and keep the code up to date as the years go on and new features are added.
9. Unfortunately, this is my first time utilizing Bootstrap so I do not have much experience.
10. One unique feature I would want is possibly having a chatroom. That way students can converse and discuss their code and solutions. This will also allow the students to ask professors questions directly through the website rather than sending emails.

2.5 Dr. Rui Wu's Responses

The interview questions were submitted to Dr. Rui Wu by email on November 2, 2020. He submitted his responses to each question on November 4, 2020. The following is a summary of his responses to each question.

1. I prefer the "flat design" which is where unessential details are not added - such as shadows or 3D affects. This allows users to be able to focus on what they are supposed to be learning, as well as making tasks less overwhelming and more simple. This will especially help make the application more mobile friendly.
2. When it comes down to things, users don't care what's on the backend of an application as they only experience the front end side of things. However with that being said, users will want the good traits that a good back end can provide, such as responsive design.
3. User accessibility is no doubt one of the most important parts and should be prioritized. We need to make sure this application is easy to use for as many people as we possibly can.
4. I'm not the best one to answer this, but I believe if you utilize some resources that are already optimized, such as using libraries, that would be one way to ensure good performance.
5. I desire ease of use and value it as one of the most important. Fast response is also important for this kind of application. There will be a lot of user actions throughout the various labs - we need to make it as friendly as possible.
6. For a robotics education application, I don't believe security is going to be a high end issue for us to have to deal with. There are some basic defensive measures you should put into place such as using the JWT token for the login field as well as encrypting user data in the database.
7. I would suggest using Flask-login to accomplish that. There are many good resources and examples online to be able to implement it.
8. I will read and listen to what users have to say and determine what may need to potentially change or polish based on their feedback. I will also talk to stakeholders just as you're doing with me.
9. Sorry to say but I wouldn't be able to give you an answer to this with confidence. I have little to none experience in that area. I have a feeling once you get a bit more experience

with it throughout experimentation and implementation, you'd be able to determine that on your own or through researching it.

10. I would like to see a robotics lab result with visualization in the finished product. Most other training apps only have terminal results. By doing this, the user will likely get a quick summary of the content and what they have effectively just completed, ideally cementing the concepts in their heads.

3. High Level Business Requirements

These requirements briefly describe the main goals of the project, and highlight the key design components that will help achieve these goals.

3.1 Fully responsive platform

The front-end website should be designed to accommodate a range of screen sizes, including smartphones and tablets. Users that primarily use mobile devices for learning should be able to use the application the same way in which a user uses a desktop or laptop computer with minimal differences.

3.2 Interactive learning experience

Most, if not all learning environments implemented should utilize a basic form to accept user code and an output section that shows results of the user's submission. If the user is able to generate the desired output, the database should store that the user completed the environment and is eligible to continue to the next environment.

3.3 Fast and efficient operation

The front-end website should exhibit minimal load and run time. Interactive environments in which students submit code and the output is rendered should be highly responsive, indicating to the user when the code is being compiled and displaying a timing component such as a progress bar.

3.4 Logging of user progress

A database should store user information and their progress throughout each environment. An account management system should be set up for users to create accounts and use them to track their progress. Access to user information should be made available to instructors whose students are utilizing the platform for grading purposes.

4. Technical Requirements Specification

The technical requirements given below will guide the team in deciding which components to build first as well as aid in the gradual development of more advanced features. Each requirement is given a priority level.

- Level 1 - Planned for implementation by the end of spring semester
- Level 2 - Planned for possible implementation by the end of spring semester
- Level 3 - Implementation will not occur until after the spring semester

4.1 Functional Requirements

FR1. [1] WARE shall allow the user to create an account.

FR2. [1] WARE shall allow the user to log into their account.

FR3. [1] WARE shall allow the user to log out of their account.

FR4. [1] WARE shall allow the user to submit their code to the web server for processing.

FR5. [1] WARE shall process code submitted by a user to produce an output.

FR6. [1] WARE shall display to the user textual results of code submitted to the web server.

FR7. [1] WARE shall display to the user graphical results of code submitted to the web server.

FR8. [1] WARE shall display to the user errors that result from code submitted to the web server.

FR9. [1] WARE shall track user progress in each robotics environment.

FR10. [1] WARE shall allow the user to view their progress in each robotics environment.

FR11. [1] WARE shall allow the user to select a robotics environment.

FR12. [1] WARE shall display a detailed description of each robotics environment to the user.

FR13. [1] WARE shall display examples of working input for each robotics environment to the user.

FR14. [1] WARE shall display to the user a graphical preview for each robotics environment.

FR15. [1] WARE shall allow the user to input their code through the use of a text box.

FR16. [1] WARE shall allow the user to input their code through the use of a file upload.

FR17. [1] WARE shall allow the user to view environments which they have made progress in from their homepage.

FR18. [1] WARE shall allow instructors to choose a list of environments that their students should have access to.

FR19. [2] WARE shall check user input for syntax errors.

FR20. [2] WARE shall check user input for formatting mistakes.

FR21. [2] WARE shall log user sessions and associated interactions over time.

FR22. [2] WARE shall allow the user to save code for later use.

FR23. [2] WARE shall use Python syntax highlighting for code written within the textbox on the website.

FR24. [2] WARE shall allow users to make feature requests and report errors.

FR25. [2] WARE shall allow instructors to view and run their student's last code submission in an environment.

FR26. [3] WARE shall implement a representational state transfer application programming interface for advanced user interaction.

FR27. [3] WARE shall allow users to communicate with other students and their instructors within each robotics environment.

Table 4.1.1: Functional Requirements for the Web Application for Robotics Education (WARE) project.

4.2 Non-Functional Requirements

NFR1.	[1]	WARE will utilize a database to store user information and robotics environments.
NFR2.	[1]	WARE will minimize CPU usage and perform intensive computations remotely.
NFR3.	[1]	WARE will provide an intuitive and elegant front-end website.
NFR4.	[1]	WARE will be compatible with Chromium-based web browsers.
NRF5.	[1]	WARE will utilize the Bootstrap, JavaScript, and Flask platforms.
NRF6.	[1]	WARE will be fully responsive to accommodate a wide range of devices.
NRF7.	[2]	WARE will utilize fully encrypted communications for each user session.
NRF8.	[2]	WARE will be compatible with Chromium-based and Firefox web browsers.
NRF9.	[3]	WARE will implement accessibility standards to accommodate screen readers and users with sight impairments.
NRF10.	[3]	WARE will be compatible with Chromium-based, Firefox, and Internet Explorer web browsers.

Table 4.2.1: Non-Functional Requirements for the Web Application for Robotics Education (WARE) project.

5. Use Case Modeling

Drafting use cases and creating use case diagrams will explicitly define how the core functions of the application should perform from the perspective of the user and the system.

5.1 Use Case Diagram

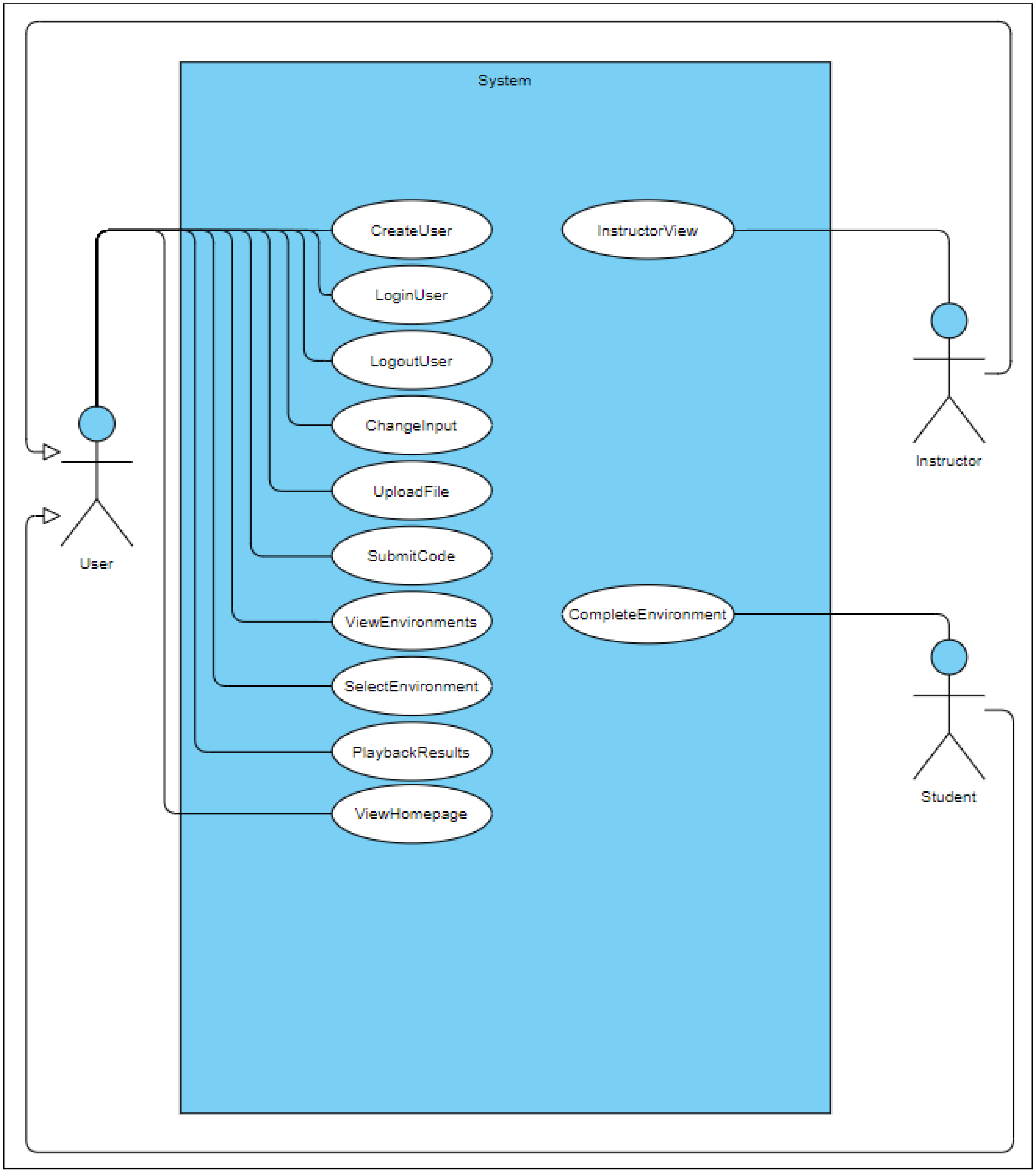


Figure 5.1.1: Use Case Diagram showing the generalized relationship between the Instructor, Student, and User actors as well as their relationships with the various use cases.

5.2 Detailed Use Case Descriptions

ID	Use Case	Description
UC01	**CreateUser**	Upon accessing the website homepage, the user has the option to create a new account in order to utilize the website's robotics environment labs.
UC02	**LoginUser**	Upon accessing the website homepage, the user has the option to log in to an existing user account. This login queries the webserver for the validity of user credentials and either grants the user access or denies login.
UC03	**LogoutUser**	The user may log out of their account and return to the website's homepage at any time, using a "Logout" button.
UC04	**ChangeInput**	Upon accessing a specific robotics environment, the user may change the provided sample code to fit their experimentation needs and develop a unique solution to the environment with which they are engaged.
UC05	**UploadFile**	Upon accessing a specific robotics environment, the user may utilize a Python file upload to overwrite the provided sample code with code contained in the uploaded file.
UC06	**SubmitCode**	Upon accessing a specific robotics environment, the user may submit code contained in the code input area to the server for processing.
UC07	**ViewEnvironments**	The user is able to view a list of all available robotics environments and their progress in each of them. This list includes brief descriptions of each environment.
UC08	**SelectEnvironment**	Upon accessing the list of available environments, the user may select a specific environment, and the user is then redirected to the page for that environment.

UC09	**PlaybackResults**	Upon receipt of results for code submitted by a user, the user is able to view the state of the robotics environment throughout the execution of their code.
UC10	**InstructorView**	Instructors are able to access a unique view in each environment page through which they will be able to access their students' code and progress.
UC11	**ViewHomepage**	Students will be able to navigate to the homepage of the website, which will include information on environments engaged in by the user if logged in.
UC12	**CompleteEnvironment**	Students will be able to mark a robotics environment as completed once experimentation is completed. This option will be available from both the user's homepage and the environment list for the user.

Table 5.2.1: Detailed use case descriptions.

5.3 Detailed Use Case Templates

Use case: CreateUser	
Use Case ID (UCID)	UC01
Actor(s)	User
Precondition(s)	1. The user has loaded the website’s homepage in their web browser.
Flow of Events	1. Upon selecting the “Register” button, the user is redirected to an account creation webpage. 2. The user is prompted with multiple fields to fill out for account creation a. Email b. Password c. Full Name d. Class ID e. Account Type i. Student ii. Instructor 3. Upon submission of account information, the data is sent to the web server for verification. 4. The web server checks received account info for validity and ensures uniqueness. a. If no conflict occurs, the account is added to the user database and can be logged into. 5. If an error occurred during account creation, the user is notified of the problem and asked to resubmit after making the required changes.
Postcondition(s)	1. If no errors occurred, the user has a functional account on the website.

Table 5.3.1: Template for the *CreateUser* use case.

Use case: LoginUser	
Use Case ID (UCID)	UC02
Actor(s)	User
Precondition(s)	1. The user has loaded the website’s homepage in their web browser.
Flow of Events	1. The user fills out their account’s credentials in the login form. a. Email b. Password 2. Upon submission of account credentials, the data is sent to the web server for verification. a. If credentials are invalid, the user is blocked from logging in to the account. b. If credentials are valid, the user is granted access to the website’s functionality and redirected to their home page. 3. If an error occurred during verification, the user is notified of the issue and asked to retry.
Postcondition(s)	1. If no errors occurred, the user has been redirected to their homepage and has access to website functionality.

Table 5.3.2: Template for the *LoginUser* use case.

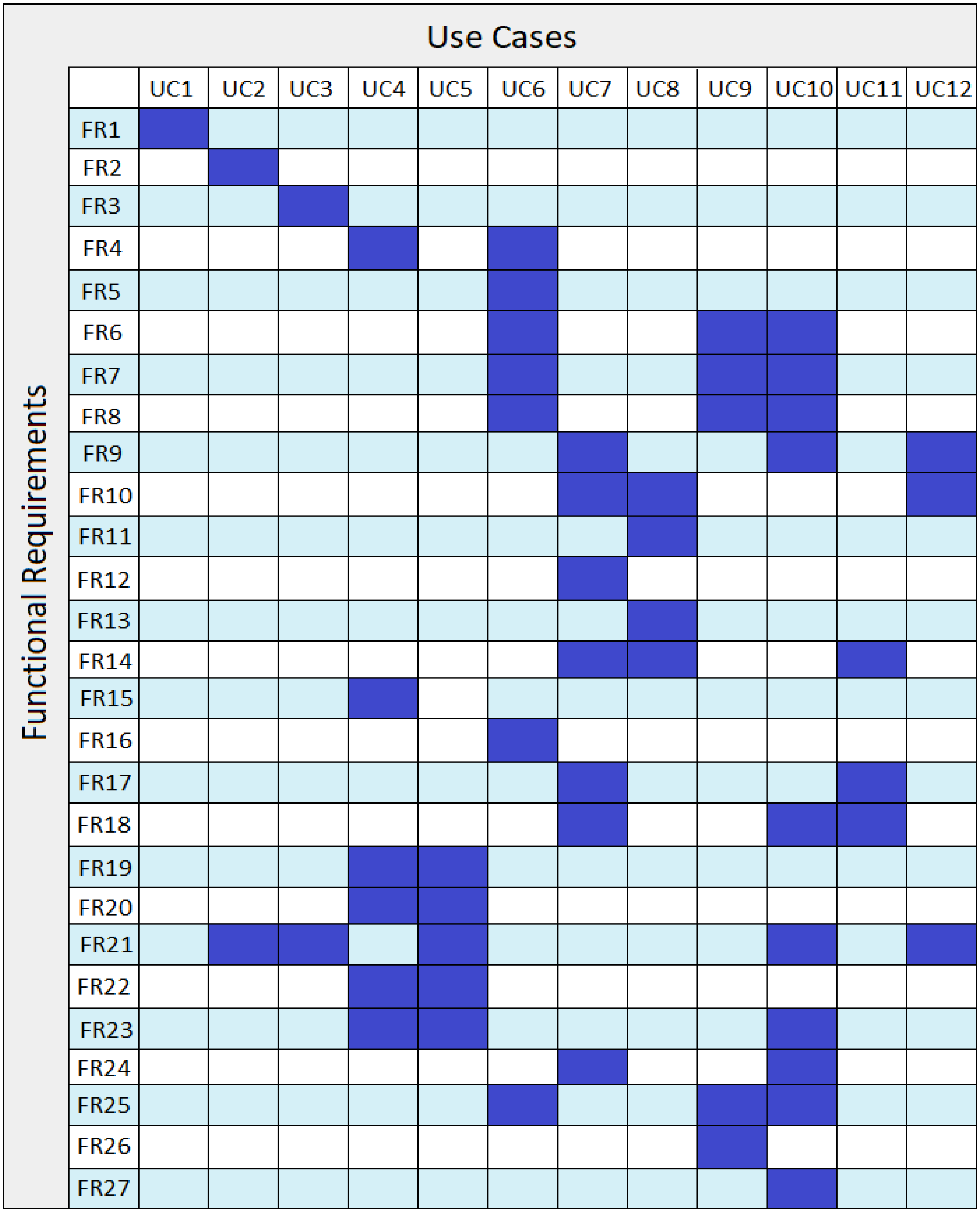
Use case: SubmitCode	
Use Case ID (UCID)	UC06
Actor(s)	User
Precondition(s)	1. The user has logged into their account. 2. The user has selected an environment for experimentation.
Flow of Events	1. The use case is initialized when the user selects “Submit” from an environment webpage. 2. Code within the submission text area is sent to the web server. 3. Submitted code is received by the webserver and verified. 4. Code received by the web server is executed and changes to the environment throughout execution, as well as console output and errors are recorded. 5. Results of the submission are formatted and sent back to the user for display. a. Video / GIF recording of the environment over time. b. Console and Error text output.
Postcondition(s)	1. The user has received the results from the submission and they are now contained within the environment webpage for viewing. 2. User progress in the environment is updated.

Table 5.3.3: Template for the *SubmitCode* use case.

Use case: ViewEnvironments	
Use Case ID (UCID)	UC08
Actor(s)	User
Precondition(s)	1. The user has logged into their account.
Flow of Events	1. The user selects the “Robotics Environments” link from their home page. 2. The web server finds all available robotics environments associated with the user’s account. 3. The web server fetches the user’s progress within each available environment. 4. Details for each available robotics environment and the associated progress is formatted and sent to the user for viewing.
Postcondition(s)	1. The user is able to select an environment for experimentation. 2. The user is able to view their progress in each environment.

Table 5.3.4: Template for the *ViewEnvironments* use case.

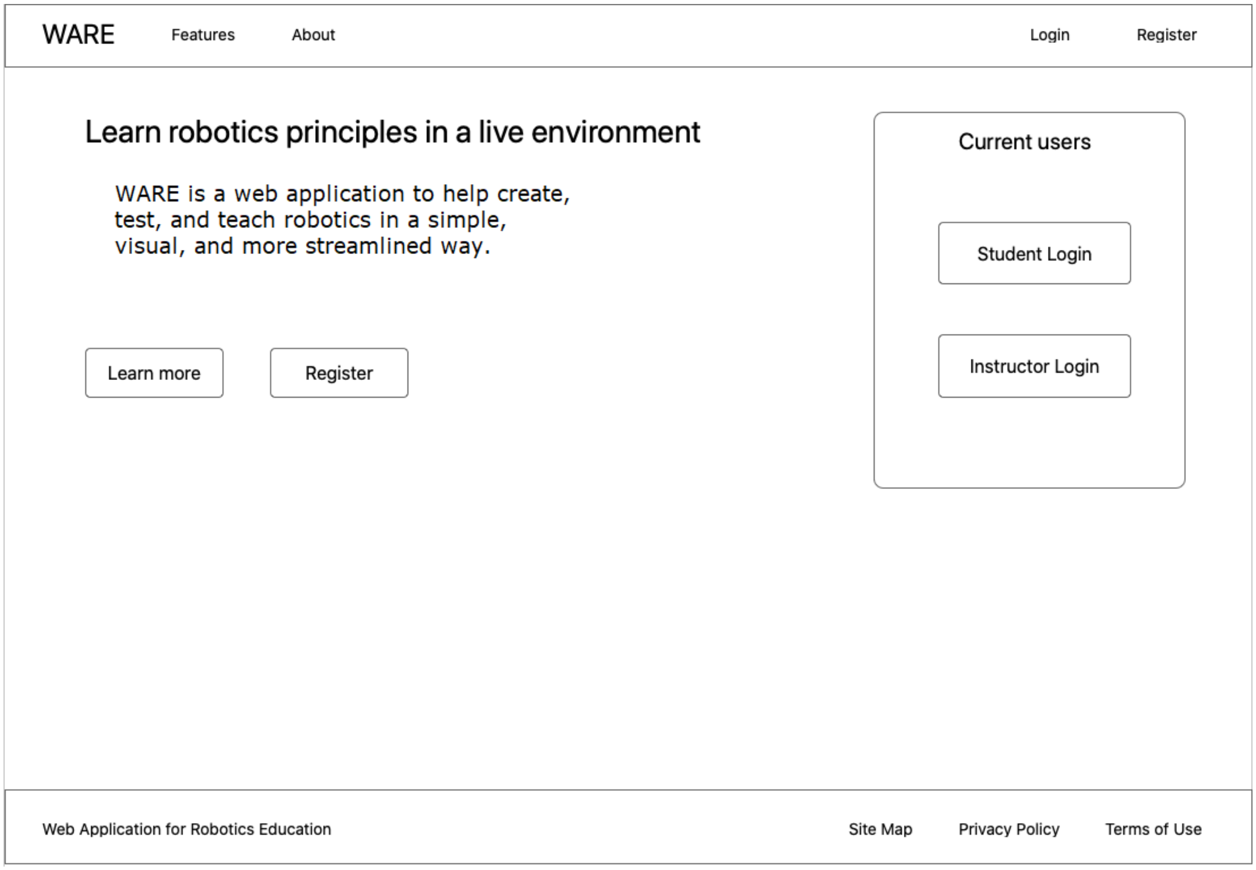
6. Requirement Traceability Matrix


Use Cases
Functional Requirements

	UC1	UC2	UC3	UC4	UC5	UC6	UC7	UC8	UC9	UC10	UC11	UC12
FR1	■											
FR2		■										
FR3			■									
FR4				■		■						
FR5						■						
FR6						■			■	■		
FR7						■			■	■		
FR8						■			■	■		
FR9							■			■		■
FR10							■	■				■
FR11								■				
FR12							■					
FR13								■				
FR14							■	■			■	
FR15				■								
FR16						■						
FR17							■				■	
FR18							■			■	■	
FR19				■	■							
FR20				■	■							
FR21		■	■		■					■		■
FR22				■	■							
FR23				■	■					■		
FR24							■			■		
FR25						■			■	■		
FR26									■			
FR27										■		

Figure 6.0.1: Requirement Traceability Matrix displaying the relationships between use cases and functional requirements.

7. Initial snapshots of the User Interface



Learn robotics principles in a live environment

WARE is a web application to help create,
test, and teach robotics in a simple,
visual, and more streamlined way.

Figure 7.0.1: Potential format of the WARE homepage - viewed as a user who has visited for the first time and is not logged in.

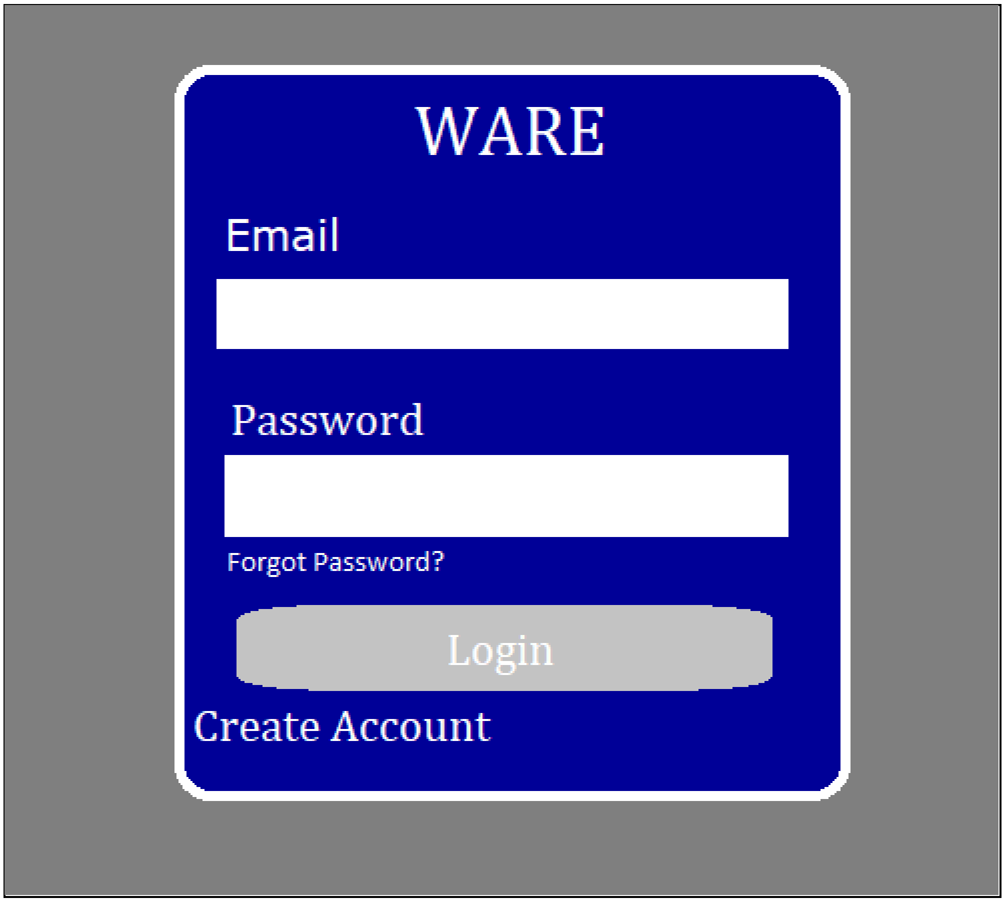
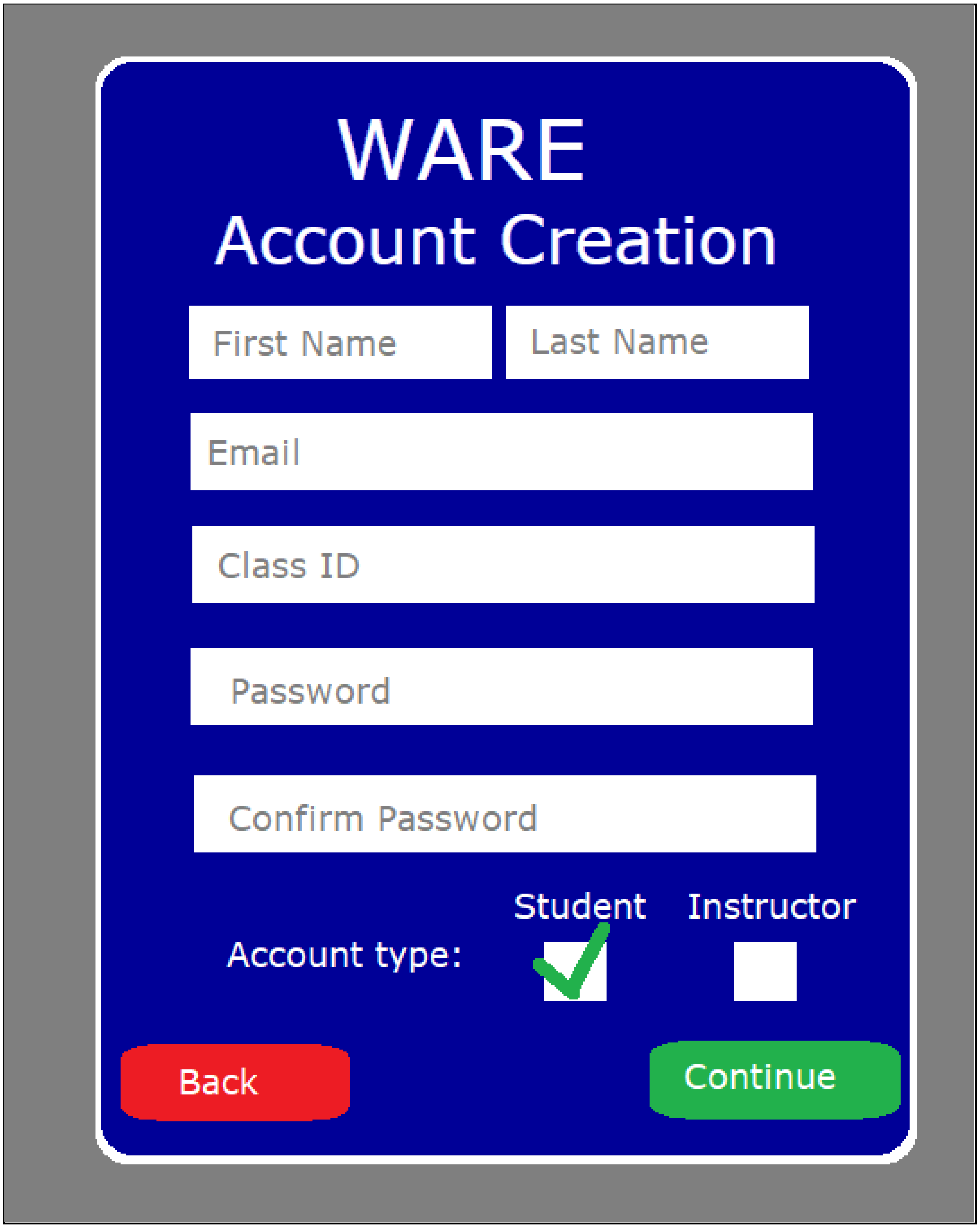


Figure 7.0.2: Potential login screen of the WARE application. If the user doesn't have an account they can begin the process with the create account button at the bottom.



WARE

Account Creation

First Name

Last Name

Email

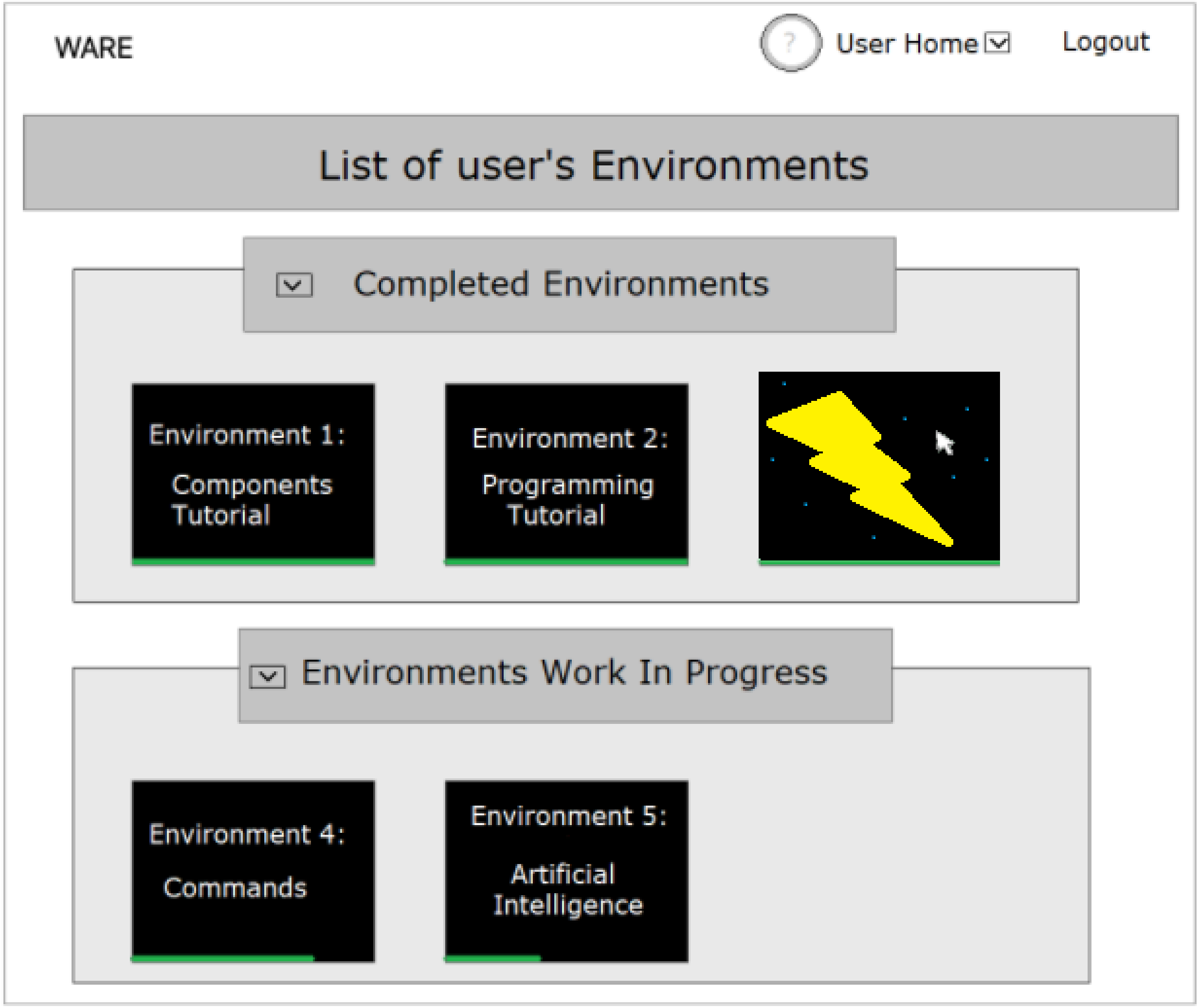
Class ID

Password

Confirm Password

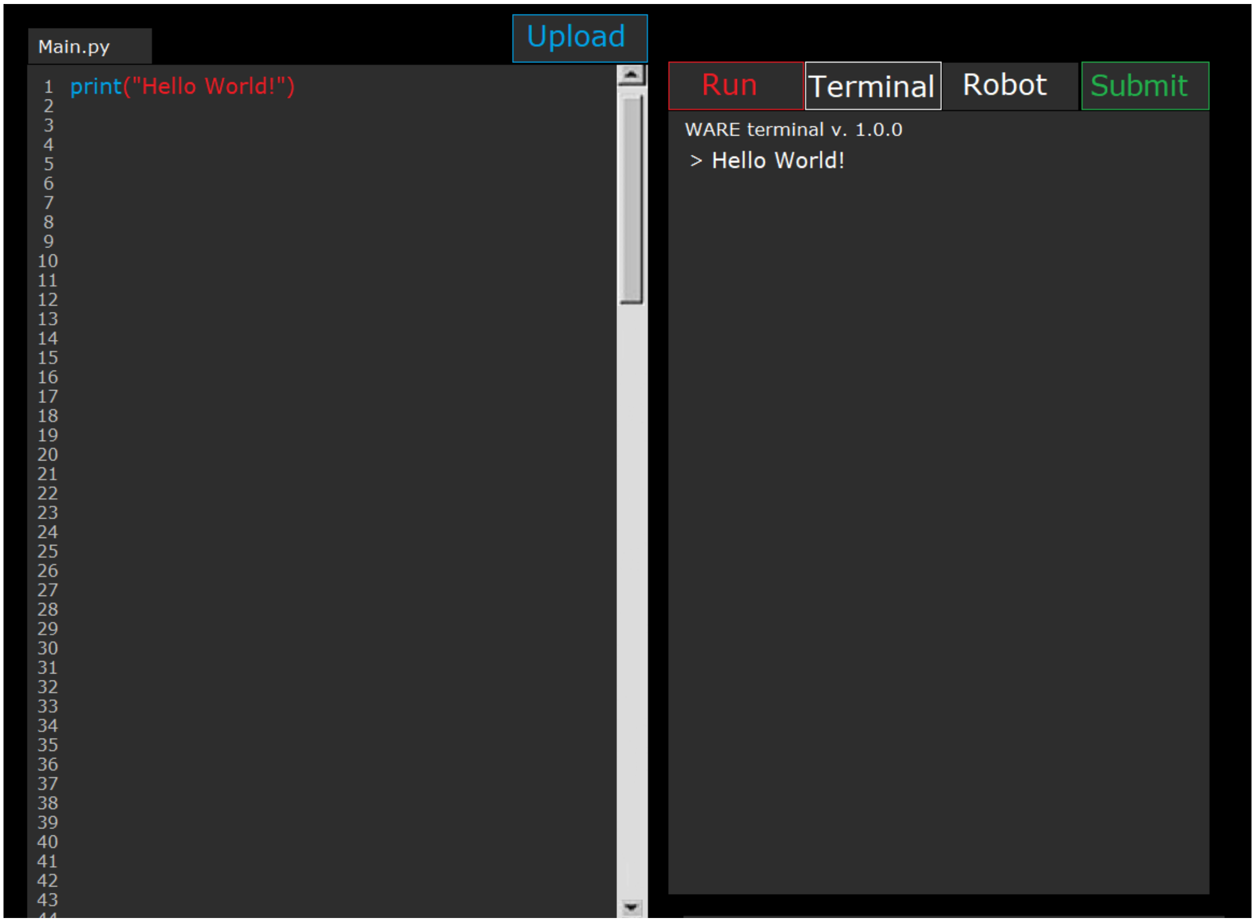
Account type: Student ☑ Instructor □

Figure 7.0.3: Account registration screen which allows the professor to easily identify the account owner. The account owner will give themselves certain permissions by selecting whether they are a student or instructor.



WARE
User Home
Logout
List of user's Environments
Completed Environments
Environment 1: Components Tutorial
Environment 2: Programming Tutorial
Environments Work In Progress
Environment 4: Commands
Environment 5: Artificial Intelligence

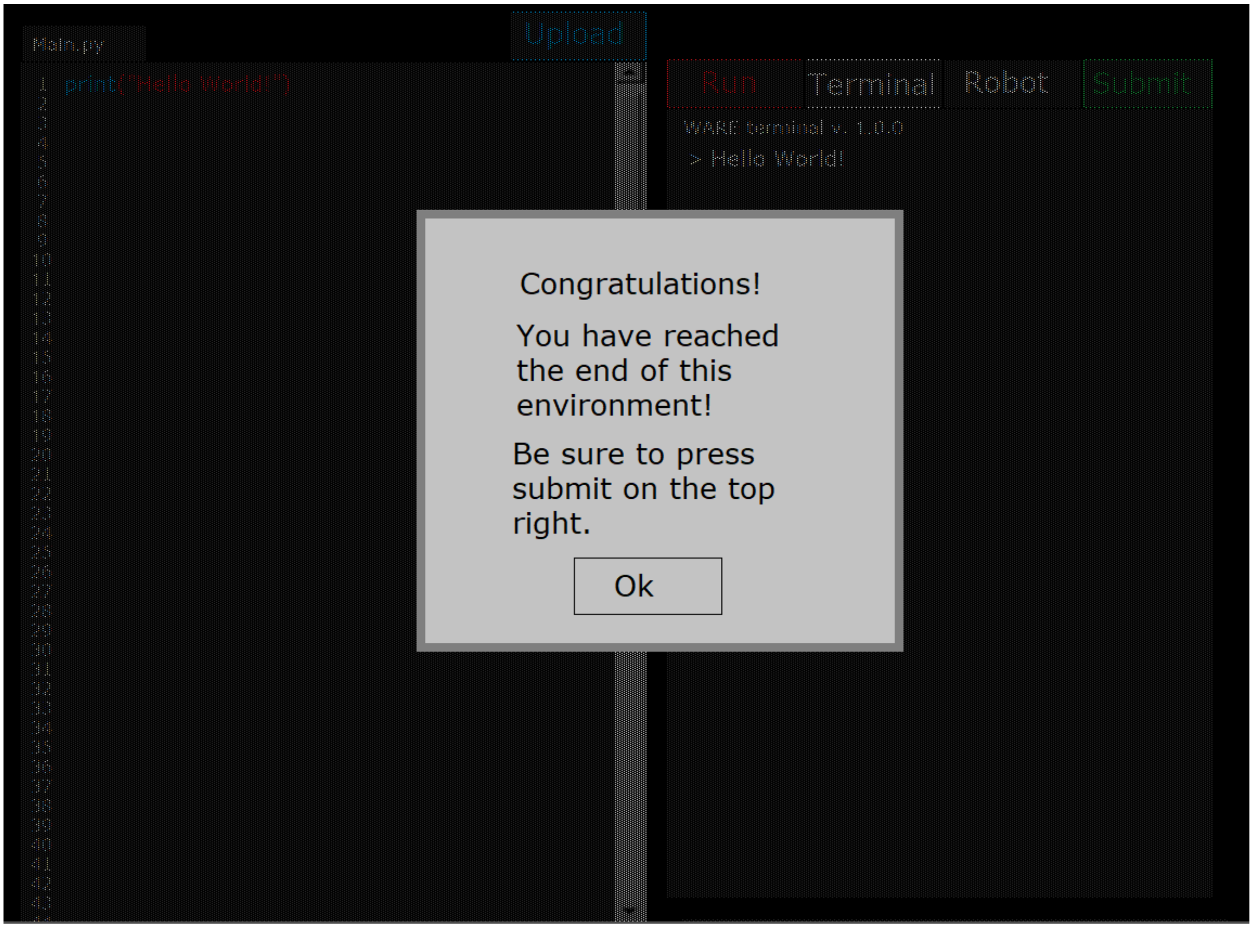
Figure 7.0.4: Screenshot of what the Environments page may look like. Drop down buttons are included to hide either category of the environments in case the user doesn't want to see them. As seen in 'Environment 3' if you hover over the environment name you will get a graphical preview (that will be soon implemented). The green progress bar is present at the bottom of each environment.



```
print("Hello World!")
```

```
WARE terminal v. 1.0.0
 > Hello World!
```

Figure 7.0.5: Inside the environment where you can program. There is a terminal tab that displays outputs, a robot tab to view the robot visuals, and buttons to upload, run, and submit code.



```
print("Hello World!")
```

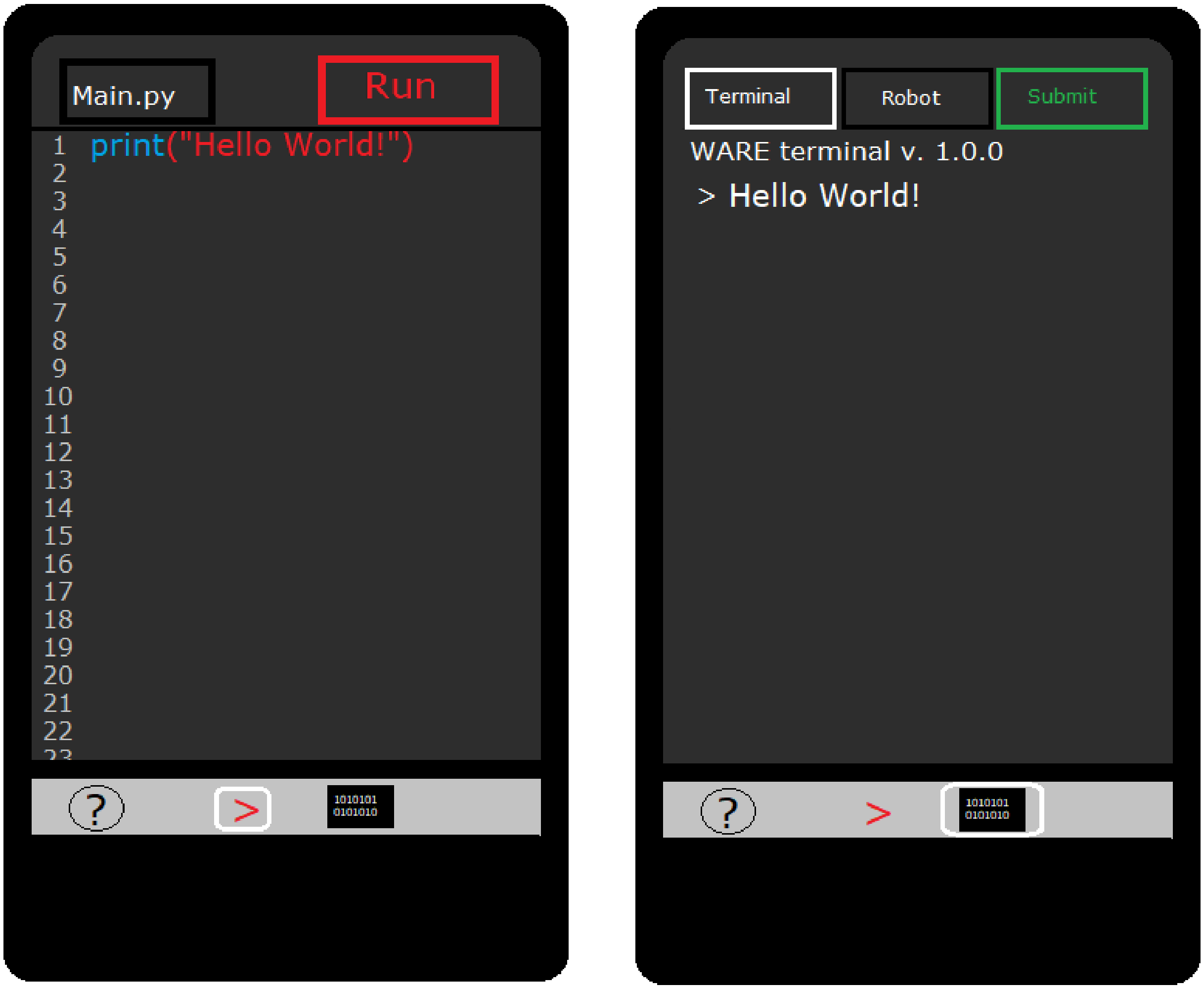
```
WARE terminal v. 1.0.0
 > Hello World!
```

Congratulations!

You have reached the end of this environment!

Be sure to press submit on the top right.

Figure 7.0.6: Possible congratulation screen after successfully getting the correct end result and thereby finishing the lab and reminding the user to submit their work. The background screen is dimmed when the notification comes up.



```
print("Hello World!")
```

```
WARE terminal v. 1.0.0
> Hello World!
```

Figure 7.0.7: A potential display of the mobile display of the console and terminal. Users can swipe to the right and to the left to access both. In the terminal you can switch to the robot tab to see the robot visualization. This design would keep it as functional as it is on laptops and desktops - minus the upload file option.

8. Glossary of Terms

Bootstrap - A website development library that utilizes HTML, CSS, and JavaScript to deliver components with some standard styling that can be inserted into a website with minimal modification

Browser - An application that transmits and receives HTTP web page requests and renders the content received by such requests. Some examples of browsers are Mozilla Firefox, Google Chrome, and Internet Explorer

Chromium-based - An application that utilizes the Chromium Blink engine to render web content

CSS (Cascading Style Sheets) - this type of programming is utilized by a web browser to style HTML components and interpret how a web page is displayed to users.

Database - An organized collection of information stored electronically that can be accessed, modified, and controlled.

Environment - Refers to a specific robotics learning exercise in which the user can input code and see the results from compilation of their code.

Flask - A back-end framework that utilizes Python. Gives an interface for content creators to insert content into web pages, while the framework invisibly serves the actual web page and design displayed to the user.

HTML (HyperText Markup Language) - The widely adopted programming language used to develop web pages

IDE (Integrated Development Environment) - An application a user can download or access remotely to create code, compile the input, and display the output

JavaScript - A programming language that has an object oriented approach. Commonly used in web pages to create interactive elements and design effects

Login - The process that a user undergoes to access their account information on the server. Usually involves entering a username and password.

Mobile - Term used to describe cordless computing devices that are smaller than general computers, more power-efficient, and utilize a touch screen for user input. Examples include smartphones and tablets.

MySQL - a relational database management system that is commonly utilized to keep store of website data

OpenAI Gym - An open source library for creating testing environments in which learning algorithms can be tested and verified.

Python - A popular, interpretive programming language that is often used for increased readability and minimal learning curve

WARE (Web Application for Robotics Education) - The project’s name. A web-based learning platform that offers robotics learning environments for higher education students.

9. List of references

9.1 Problem Domain Book

Khine, M. S. (2017). Robotics in STEM education redesigning the learning experience. Cham: Springer. doi:10.1007/978-3-319-57786-9

Education regarding robotics is highly important - and unfortunately not many seek it out due to lack of resources or lack of interest. This book discusses robotics in education and how to best not only teach such concepts - but also strategies on how to get more students interested in the topic which is what we hope to achieve. Khine also helps illustrate the importance of robots - and how much of an impact they have on our future.

9.2 Reference Articles

L. Wu, G. Liang, S. Kui and Q. Wang, "CEclipse: An Online IDE for Programing in the Cloud," 2011 IEEE World Congress on Services, Washington, DC, 2011, pp. 45-52, doi: 10.1109/SERVICES.2011.74.

While Online IDEs are very versatile and convenient to developers, they still suffer from issues that hold them back from being used at their full potential. These issues range from security concerns to the implementation of functions in programs. The article attempts to provide solutions to these issues.

L. Wu, G. Liang and Q. Wang, "Program Behavior Analysis and Control for Online IDE," 2012 IEEE 36th Annual Computer Software and Applications Conference Workshops, Izmir, 2012, pp. 182-187, doi: 10.1109/COMPSACW.2012.42.

Online IDEs are gaining more and more researcher's focus because of how there is no complicated installation required and that they are adaptable and can be used in many different locations and hardware. Although this technology is very exciting it suffers from a lot of unique challenges such as wrong file operations, excessive resource consumption, and banned method calling.

S. España-Boquera, D. Guerrero-López, A. Hermida-Pérez, J. Silva and J. V. Benlloch-Dualde, "Analyzing the learning process (in Programming) by using data collected from an online IDE," 2017 16th International Conference on Information Technology Based Higher Education and Training (ITHET), Ohrid, 2017, pp. 1-4, doi: 10.1109/ITHET.2017.8067822.

This article discusses how CS courses require a lot of time to learn and practice. Unfortunately some students may find it difficult to install IDEs on their local machines at home and only use computer labs instead limiting their time to practice coding. This article goes into depth about the experiences of using an online IDE called Codeboard for a CS class and an analysis of its effects on students and their learning.

Q. Wang, W. Li, and T. Xie, "Educational programming systems for learning at scale." 2014 In Proceedings of the first ACM conference on Learning @ scal conference (L@S '14). Association for Computing Machinery, New York, 2014, pp. 177-178, doi:
https://doi.org//11.1145/2556325.2567868

In this article the discussion focuses on two IDEs developed by Peking University for their beginners CS courses and artificial intelligence courses. These IDEs allow students to use a multitude of languages as well as other features such as code auto-completion, code selection. The article also discusses how this type of versatile system has positively impacted student learning.

9.3 Websites

https://dev.mysql.com/doc/

Creating a functional and secure database will likely be one of the bigger and challenging portions of our project. We need to take special precautions and care when it comes to this section due to the cyber security risks. Having a full grasp of SQL and the platform we use to create the database is crucial in being able to deliver a safe, respectable, and ethically good project. This website has a lot of information we need to learn and understand.

https://flask.palletsprojects.com/en/1.1.x/

Flask is going to be one of our most used platforms - if not the most used. This project relies on strong and clean website design which is user friendly, and the many tasks and examples offered will teach us good design and habits. This especially holds true when we need to make this app cross platform.

https://openai.com/blog/ingredients-for-robotics-research/

This website came highly recommended by the head of the project - Dr. Wu. This website and the corresponding githubs and resources that contribute to it will prove to be an incredible resource for this project as it provides the environments necessary for users to implement their experimentations. Essentially this is the most important resource we will have in implementing this project.

10. Contributions of team members

The team's contributions are summarized below. The name of each team member, the time each member allocated for this assignment, and the tasks that were completed are given.

Sean Griffith - 8 hours
Designed the cover page and table of contents, defined the functional and non-functional requirements, created the use case diagram, defined the use cases and their descriptions, created the use case templates and defined the set of chosen use cases with the templates, edited the introduction section.

Zachery Wiles - 7.75 hours
Wrote the interview section introduction paragraphs, revised some of the interview questions, interviewed Ben Gallagher and summarized his responses, defined the high level business requirements, edited some of the glossary definitions, designed the document with appropriate formatting.

Ryan Lunt - 7.5 hours
Drafted some of the interview questions, interviewed Dr. Rui Wu and summarized his responses, created the requirements traceability matrix, created and designed prototype snapshots and captions with various software for each, located and listed applicable references and documentation.

Herman Hira - 6.75 hours
Wrote the introduction, drafted some of the interview questions, contributed his responses to the interview questions, chose and defined terms for the glossary. Also helped create snapshots for the initial snapshots section, and found articles/conferences and wrote descriptions for those in the references.