

Cinderella's Closet: Dresses into Dreams

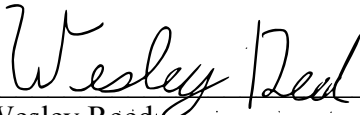
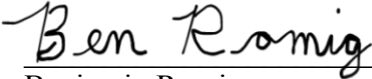

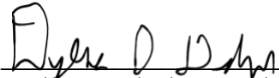
by

Wesley Reed, Benjamin Romig, and Trent McGinnis

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/Cybersecurity

© Copyright 2022 Reed, Romig, McGinnis

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

 _____ Wesley Reed	<u>4/24/2022</u> _____ Date
 _____ Benjamin Romig	<u>4/24/2022</u> _____ Date
 _____ Trent McGinnis	<u>4/24/2022</u> _____ Date
 _____ Dyllon Dekok, Faculty Advisor	<u>4/24/2022</u> _____ Date

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

April 2022

Table of Contents

<i>List of Illustrations</i>	3
Abstract	4
Introduction	5
<i>Project Summary</i>	5
<i>Problem Statement</i>	5
<i>Solution</i>	5
<i>Project Source</i>	6
Discussion	7
<i>Project Objectives/Goals</i>	7
<i>Project Scope</i>	7
<i>Quick Project Timeline</i>	8
<i>Technologies Used</i>	8
<i>Technical Architecture Diagram</i>	9
<i>User Personas</i>	11
<i>Use Cases</i>	12
<i>Use Case Diagram</i>	15
<i>Testing Plan</i>	15
<i>Change Management Plan</i>	17
<i>Budget</i>	18
<i>Problems Encountered and Analysis of Problems Solved</i>	18
Conclusion	19
References	20

List of Illustrations

Figure 1 Server Architecture Diagram 9
Figure 2 Database Diagram..... 10
Figure 3 Use Case Diagram 15

Table 1 User Persona Table 11
Table 2 User Persona Table 11
Table 3 Use Case Table 12
Table 4 Use Case Table 12
Table 5 Use Case Table 13
Table 6 Use Case Table 14
Table 7 Test Logs..... 16
Table 8 Project Budget..... 18

Abstract

Cinderella's Closet's inventory management and event scheduling software is an innovative approach to a complex problem that Cinderella's Closet faces when planning and managing their events. This app is designed to simplify workflows and increase efficiency of managing high school girl's appointments and integrate an inventory management system. It is made accessible through web technology so that there is no need to install software and the app can be accessed through any device with a web browser, including both mobile and desktop devices.

Introduction

Project Summary

Cinderella's Closet Cincinnati is a nonprofit organization in Cincinnati, Ohio that works with junior and senior high school girls (or princesses) who are unable to attend their high school prom and need dresses, shoes, and other items. This project will allow them to give access to high school counselors and local social workers to schedule shopping appointments for princesses at their event(s), assign inventory items (dresses, shoes, jewelry, and accessories) to a princess, and assign a princess's dress to a seamstress for alteration.

Problem Statement

Cinderella's Closet currently uses a system for scheduling princesses that requires developer intervention each year to reset and create new time slots and is not secure (storing passwords in plaintext, etc.). They need a flexible princess scheduling system, to easily schedule future events or appointments, that will also track their donation inventory. Knowing what is in the inventory, will assist them in asking for new donations and track items in use. Due to the nature of the client (high school girls with financial needs), Cinderella's Closet needs the sign up to be confidential and sites like SignUp Genius do not offer this level of confidentiality (SignUp Genius). They also needed a solution that was easy to integrate with an inventory system. While we recognize the prowess of systems like Snipe-IT (Snipe-IT), they needed a solution that would be better suited to fit their needs. By being fully integrated with the princess management and event management side of the application, it will be simpler to operate and cheaper for the client.

Solution

To solve this problem, we plan to create a full stack web application using ReactJS, NodeJS, TypeScript, and PostgreSQL that will allow for event scheduling, inventory management, and event progress tracking. An issue with similar scheduling solutions, such as SignUp Genius, reveals sensitive information regarding other attendees to events that the stakeholders would like private. Regarding other aspects of the solution, the usage of multiple tools to fulfill the desired outcome is unfavorable to the stakeholders.

Project Source

Wesley Reed has worked with Cinderella's Closet in the past and saw this problem. He then approached Sara Mosteller to pitch a solution and gather requirements. The team then formed from a post on the Microsoft Teams Senior Design Prep Channel. The team has since gathered additional requirements through meetings with the project sponsors.

Discussion

Project Objectives/Goals

- A scheduling system will be created such that an administrator can set up a new event date and set the times, appointment interval, number of appointments per interval, and users with the scheduler role can invite Princesses to the event
- Each Princess will be assigned an ID number at registration that can be printed from a label printer and used to identify which items are theirs
 - In the future, Cinderella's Closet Cincinnati will build an inventory of items and tag them with barcodes. Item IDs will be assigned to a Princess ID
- Each princess will be assigned a Fairy Godmother in the application that is responsible for tagging each item the Princess chooses to take with them
 - For this first version, Fairy Godmothers need to be able to create an inventory item on the fly to assign it to a princess. Once inventory tagging has been completed, this permission will be revoked
- There will be an alterations section of the application that allows volunteers to mark an assigned dress for alterations, leave notes and take pictures for the alteration, and assign a person to complete the alteration

Project Scope

See GitHub Project Board: <https://github.com/Cinderellas-Closet/Cinderellas-Closet/projects/1>

Quick Project Timeline

Task #	Task Name	Duration	Start Date	End Date
1	Event Scheduling	5 months	09/01/2021	01/24/2022
2	App hosted on server for testing & approval	2 weeks	01/10/2022	1/24/2022
4	Inventory Tracking	4 months	12/01/2021	03/01/2022
5	Final app hosted on staging site for testing & approval	2 weeks	02/08/2022	03/01/2022
6	Cinderella's Closet event (using production site)	6 weeks	03/12/2022	04/23/2022
7	IT EXPO Presentation	1 day	04/12/2022	04/12/2022

Technologies Used

To solve this issue, we are utilizing the following technologies:

Frontend:

- [React.js](#) (JavaScript/TypeScript framework)

Backend:

- [PostgreSQL](#) (Database)
- [Redis](#) (Session Cache Management)
- [Sequelize](#) (ORM)
- [Express](#) (REST API)

Utilities:

- [Docker](#) (Hosting Postgres and Redis)
- [GitHub](#) (Source Version Control)
- [Typescript](#)
- [Lerna Monorepo](#)

Technical Architecture Diagram

Figure 1 is a diagram that represents each aspect of the application when it is hosted on a server and what technologies we are using. The backend is powered by a Postgres database, and Redis for caching. The frontend uses React for the web framework. We are using express to run both our API and Nginx to proxy our frontend and the API.

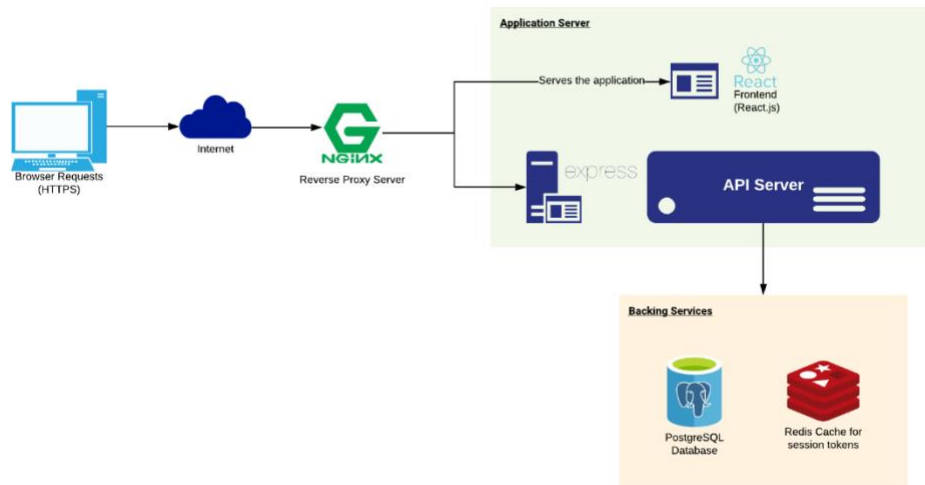


Figure 1 Server Architecture Diagram

Figure 2 is a database diagram representing each schema, table, and relationship we have in the database. We have five schemas in our database: Users, Events, Princesses, Inventory, and Shared. We have eight tables in the Users schema, three tables in the Events schema, five tables in the Princesses schema, seven tables in the Inventory schema, and two tables in the Shared schema. These tables represent all the data we will need to operate each aspect of the application. Anywhere you see a column marked FK that is called created_by, updated_by, or deleted_by, those columns are foreign key references to users.id but the relationship lines have been omitted from the diagram to make it more readable.

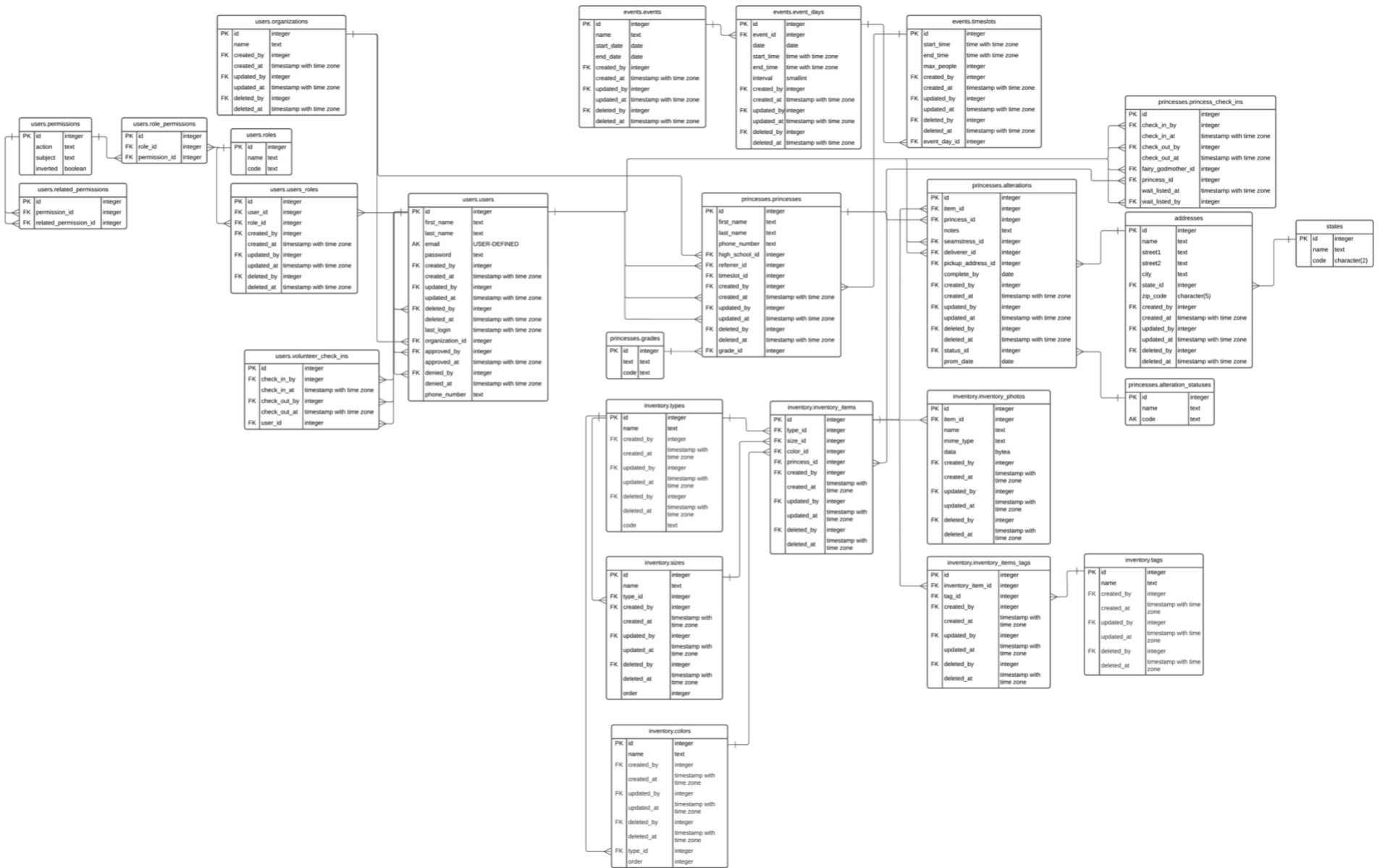


Figure 2 Database Diagram

User Personas

This section contains two user personas that represent typical users of our application. One is a volunteer for the non-profit who is responsible for maintaining the events and inventory, and the other is a high school guidance counselor who is responsible for assigning high school girls to event appointments.

Table 1 User Persona Table



User Persona: 1	
	Title: Cinderella's Closet Volunteer
	Name: Jacob Jones
	Age: 24
	Gender: Male
Behavior	<ul style="list-style-type: none"> Likes to watch football on the weekends Enjoys spending time with family Likes helping others
Pain	<ul style="list-style-type: none"> Has trouble keeping himself organized
Needs & Goals	<ol style="list-style-type: none"> To get married and have kids Start his own business

Table 2 User Persona Table

User Persona: 2	
	Title: High School Counselor
	Name: Sarah Smith
	Age: 30
	Gender: Female
Behavior	<ol style="list-style-type: none"> Enjoys coming home to her family after work Enjoys reading Likes to bake
Pain	<ul style="list-style-type: none"> Has trouble keeping up with her kids
Needs & Goals	<ul style="list-style-type: none"> Wants to eventually climb the administration chain Wants to see her kids succeed after graduating high school

Use Cases

This section contains several tables that each describe the flow of one feature or use case for our application. This list is not all-encompassing of the features of the application but describes the four critical features of the application.

Table 3 Use Case Table

Use Case ID	1
Use Case Name	Princess registration
End Objective	A high school girl is registered for an event appointment
User/Actor	High school counselor/social worker
Trigger	Actor sees high school girl had a need to go to prom
Frequency of Use	Many times, through January and February before the March event
Preconditions	The actor has an account with the scheduler role; that account has been approved by an admin
Basic Flow	<ol style="list-style-type: none"> 1. Actor logs in 2. Actor chooses an open time slot 3. Actor enters high school girl information
Alternate Flow	<ol style="list-style-type: none"> 1. Actor is already logged in 2. Actor views girls they have registered 3. Actor changes a girl's time slot
Postconditions	High school girl is registered for the appropriate time slot; actor can download an appointment reminder certificate

Table 4 Use Case Table

Use Case ID	2
Use Case Name	Receiving Inventory
End Objective	The system has a record of the inventory item, and a tag has been printed
User/Actor	Cinderella's Closet Volunteer
Trigger	An item is donated to or purchased by Cinderella's Closet
Frequency of Use	At any point during the year (repeatedly)
Preconditions	The actor has an account with the inventory manager role; that account has been approved by an admin
Basic Flow	<ol style="list-style-type: none"> 1. Actor logs in

	<ol style="list-style-type: none"> 2. Actor chooses “receive inventory item” option 3. Actor fills out required fields describing the inventory item (type, color, size, etc.) 4. Actor takes a photo of the item 5. System prompts the actor to print a QR tag for the item
Alternate Flow	<ol style="list-style-type: none"> 1. Actor is already logged in 2. Actor chooses “receive inventory item” option 3. Actor fills out required fields describing the inventory item (type, color, size, etc.) 4. Actor takes a photo of the item 5. Actor marks that they are receiving multiple identical items 6. System prompts the actor to print a QR tag for each item
Postconditions	Inventory items have been recorded, photographed, and tagged

Table 5 Use Case Table

Use Case ID	3
Use Case Name	Disposing Inventory
End Objective	The system has matched the inventory item to a registered princess
User/Actor	Cinderella’s Closet Volunteer
Trigger	A princess arrives at her registered time slot and chooses to take an inventory item home
Frequency of Use	During the event in the month of March (repeatedly in each time slot)
Preconditions	The actor has an account with the inventory manager role; that account has been approved by an admin; the inventory item has been received into the system and tagged
Basic Flow	<ol style="list-style-type: none"> 1. Actor logs in 2. Actor chooses a princess 3. Actor chooses to assign inventory item to princess

	<ol style="list-style-type: none"> 4. System prompts actor to scan QR code of item 5. After scan, system shows the image and description so the actor can verify the right item was found
Alternate Flow	<ol style="list-style-type: none"> 1. Actor is already logged in 2. Actor chooses a princess 3. Actor chooses to assign inventory item to princess 4. Actor searches for inventory items by type, size, color, etc. 5. Actor uses picture and description from search results to assign item
Postconditions	Inventory items have been assigned to princess and are no longer able to be assigned to others

Table 6 Use Case Table

Use Case ID	4
Use Case Name	Event Management
End Objective	The system has created an event that will have timeslots available for signup
User/Actor	Cinderella's Closet Volunteer
Trigger	A volunteer fills out the event creation form
Frequency of Use	Before March to start sign up process
Preconditions	The actor has an account with the event manager role; that account has been approved by an admin
Basic Flow	<ol style="list-style-type: none"> 1. Actor logs in 2. Actor goes to the event panel 3. Actor fills out the event creation form 4. System creates timeslots for the event.
Alternate Flow	<ol style="list-style-type: none"> 1. Actor is already logged in 2. Actor edits the dates on the event 3. Timeslots are updated
Postconditions	Event does not overlap another event and timeslots are available for use

Use Case Diagram

Figure 3 is a use case diagram containing the actions that the two actors, a high school counselor and a volunteer, can take within the system using the above use cases.



Figure 3 Use Case Diagram

Testing Plan

Overview

The testing plan contains sections on testing methodology, scope, objectives, logs and procedures, a review, and a change management plan for the project. These sections will detail how testing is conducted, what is expected to be tested, how testing will be recorded, and how changes to the project will be managed.

Methodology

All testing will be done by QA on a remote server that is automatically updated to production ready. Stakeholders will be able to access this server to provide feedback and report bugs. After any major implementation on an issue, testing occurs to report on functionality, bugs, or enhancements. The approach for a production server was taken to centralize a ready

environment for testing. Testing after major implementations provides feedback early to minimize functionality issues and bugs.

Scope

Features/use cases being tested:

- Create, register, approve, delete, deny, get, login, restore, and update a user
- Create and move a student/princess appointment
- Create, delete, and update an inventory item
- Create, delete, and update a tag for inventory item
- Create, delete, and update an event
- Creating an appointment certificate
- Checking a user’s permissions
- Requesting a password reset, completing a password reset, and strength testing a password
- Sending an email

The above features/use cases are being tested as they provide the core functionality to the project demonstrated by the use case diagram.

Objectives

- a. All major features and use cases need to be accounted for
- b. All use cases must account for all the user roles
- c. All bugs need to be resolved before the stakeholder’s deadline for the feature

Test Logs and Procedures

Table 7 Test Logs

Item #	Test Case	User	Role	Expected Output	Actual Output	Pass/Fail	Reason
1	Password Manager correctly denies a password for being too weak (input !aBc4567&9)	Stakeholder	Counselor	true	true	pass	Password is secure
2	Marking a user as approved the UserRepo update method should be called once	Stakeholder	Admin	true	true	pass	Update is called once
3	Marking a user as approved there should be an email sent to welcome them	Stakeholder	Admin	true	true	pass	Email is generated and sent

4	Creating a user, the User Repo create method should be called	Stakeholder	Counselor	true	true	pass	Method is called
5	Creating a user, the welcome email is sent	Stakeholder	Counselor	true	true	pass	Email is generated and sent
6	Creating a user that already exists will not work and then the create method on repo will not be called	Stakeholder	Counselor	true	true	pass	Repo calls exist method and returns true

Testing Review

We learned that to provide the best code and the best solution to the problem, we need to test many aspects of the project. The testing will include mocking data to test that the backend is performing as expected and testing the front end to make sure that the pages are loading as anticipated.

Change Management Plan

When changes are requested to the application by stakeholders, one of our developers or project advisor, they will be added to the project board on GitHub as issues. We will discuss with stakeholders to determine if the change is high priority, normal priority, low priority, or a nice-to-have and attach a label to it in the backlog.

To determine the priority of the change, a meeting will take place between the developers and the stakeholders to discuss the impact of the change on the existing backlog and how important the change is. Based on the assessed priority of the change in context of other requested changes, the team will decide when and if the change will be completed.

Budget

The budget presented below includes an estimated cost of labor for two academic semesters based on the average weekly work effort by the team. A one-time cost of external hardware is included for printing hardware, and an annual cost of external software is included for hosting services for the application.

Table 8 Project Budget

	Rate Per/Hr	Work Effort (Hours)	1 X Costs	Ongoing Annual		
				Rate Per/Hr	Work Effort (Hours)	1 X Support Cost
Labor - IT	20	1,350	\$ 27,000.00	0	0	\$ -
Labor - External		0	\$ -		0	\$ -
Software - External			\$ 0			\$ 150
Hardware - External			\$ 300			\$ 0
Misc.			\$ 0			\$ 0
TOTAL			\$ 27,300.00			\$ 150

Problems Encountered and Analysis of Problems Solved

- More information was often needed from the stakeholders to align project requirements with their vision of the solution. Meetings and communication with the stakeholders occurred throughout the development cycle to show progress, ask questions, and propose solutions to issues.
- Debugging new technology and code proved to be a challenge during the development cycle. Setting up a repository of resources for the technology and libraries being used helped to apply those in our project.
- Complying with Lerna Monorepo was an early issue when committing changes to code branches. Discussion occurred throughout team meetings on how to resolve errors, issues, and comply with our code linting and validation tools to successfully commit to the code base.

Conclusion

Throughout the course of the project, our team utilized GitHub Project Boards to set sprints and manage our work for the semester. Since the project deliverables are for a customer, our team has worked closely with them to build an application that meets their requirements. Our ability to communicate and work with them has given us applicable experience of working with customers throughout the development lifecycle.

We have also learned through working with the technologies present how to better develop a web application. Through continuous discussion about improvement and better practices, we have bettered ourselves as web application developers. This was manifested in our use of Docker at the beginning of our project on our local machines, and eventually moving the database and Redis caching to a remote server to improve developmental efficiency.

Starting in February of 2022, the application went into production for event scheduling. Continuing into March of 2022, we released inventory management and event check-in features. From there, we have been making small enhancements at the request of the stakeholders and fixing bugs as they come up. We plan to continue supporting this application moving forward by adding new features, such as volunteer registration and enhanced reporting.

References

SignUpGenius. (n.d.). *SignUpGenius Pricing*. <https://www.signupgenius.com/pricing>

Snipe-IT. (n.d.). *SnipeIT Description*. <https://medevel.com/snipe-it-asset-warehouse>