CAPSTONE PORTFOLIO Intranet Web App for HRDC

ESOF 423 - Spring 2025

Developers

Talia Clarke Mark Kwapisz Sierrah Paul Ryan Plusch Ryley Sanden Henry Weston

Designer

Kira Hopkins

Section 1: Program

In the meeting at the beginning of the semester, Krista Dicomitis, a member of the HRDC team, introduced to the class an issue pressing the organization: Creating a portal for volunteers and employees to access important files, forms, and payment information, to create a more user friendly application as an alternative to the multiple Google drives the organization has. Krista was adamant on having a simple and streamlined application that allows for easy navigation, with clear and easy-to-find links to important sites, such as Paychex (used by the HRDC for timesheets), new hire information, and easy access to their Google Drive.

The current system the HRDC is using, strictly Google Drive, works; however, there is confusion among the volunteers on how to find the important information. To address their concerns, our team started brainstorming some ideas for developing a simple, easy-to-follow site design. Our design has been deliberately created for ease of use for any volunteer, administrative worker, or staff member, specifically focused on ensuring the elderly and anyone with visual troubles can easily navigate the site. With these design aspects as our main focus, our application not only simplifies the navigation but also empowers HRDC administrators and staff to spread important information to those within the organization, with the ultimate goal to be enhancing overall operational efficiency.

Link to our Github: https://github.com/423S25/repo5 Link to our Project: https://repo5-rho.vercel.app/

Section 2: Teamwork

Our team collaborated effectively to complete this project, placing a strong emphasis on clear and consistent communication to maximize efficiency. We maintained open lines of communication both in and outside of class, notifying one another in advance of any absences and providing updates on asynchronous tasks. During class sessions, we held regular check-ins to discuss current progress, identify blockers, and coordinate next steps.

We utilized a task-sharing approach, where each team member reported on what was completed, what was in progress, and any challenges encountered. When unanticipated issues arose, other team members proactively offered support, contributing to problem-solving efforts and maintaining project momentum.

Throughout the sprint, we remained committed to our goals by clearly defining task priorities and ensuring all individual responsibilities were completed on schedule. Our proactive planning and mutual accountability allowed us to stay on track and consistently meet our milestones.

Developer Contributions

Team member One:

At the beginning of the project, I focused on creating preliminary designs and preliminary mockups in Figma. I conducted research on our target users to ensure the site would be highly usable and empathetic to their needs. Based on this research, I compiled recommendations for features that would best serve our users. I also contributed to both user and developer documentation.

Initially, I began implementing a calendar feature but encountered packaging issues, so I shifted my focus to researching testing procedures and compliance requirements. I contributed to writing our project portfolio. I did additional research and worked on which design patterns we were using or considering for future implementation. I conducted user testing, including multiple inperson interviews with individuals who represent our target market. I evaluated the site for WCAG and ADA compliance and documented areas that required improvement.

Additionally, I explored different methods for transferring the site to HRDC and included those findings in the developer documentation. Throughout the project, I continued to work on our developer and user documentation, retrospectives, some burndown charts and the portfolio.

Team Member Two:

Throughout this project, my primary focus was on building the functionality of the web application, ensuring that all features worked seamlessly and were easy to use. I was responsible for setting up and integrating Firebase into the project. This included configuring the database and handling user authentication. By doing this, I made it so that only authorized users (admins) could manage content on the site, such as adding, editing, or deleting announcements and links.

I also worked on several frontend features to improve the user experience, like implementing search and enabling admins to manage content easily through the site. This involved writing the code for the search page, help page, and admin pages (announcement management page and the links page), where admins could dynamically add, edit, and delete content. These pages were directly tied to the Firebase database, ensuring that all updates were reflected in real-time across the site.

Additionally, I was responsible for setting up Vercel, making sure that the project was properly deployed and accessible online. I also kept the documentation up to date with version updates. I also participated in testing and debugging to ensure the app functioned smoothly.

Team Member Three:

Throughout the project, I mainly worked on implementation of UX for the site. I was responsible for the entire implementation of all design factors onto the website. This included taking the figma code that was based on our UX/UI designs and implementing, I also adjusted the code to be able to accurately fit all content within the site.

I also worked on attempting to integrate either Builder.io or Strapi into the site, had this been successful the HRDC would have been able to drag and drop to change the site. Both interactions had their problems. With Builder.io there was an issue on their side with how their integration worked which did not allow us to use the software, with Strapi there were several issues within the process of integration that caused issues within our project so we decided to scrap the idea of using external software. We instead decided to use the editing set up as described within Team Member Two's writeup.

I also went through the Accessibility Checker to see if our website was up to US Accessibility Laws. Originally our site was at 76%, so I went through and updated button names, so that screen readers can easily access the site along with updating contrast throughout the site for easier readability. The site is now at 95% accessible according to the Accessibility Checker which is the highest that the Checker gives sites, so our website is now fully in compliance with US Accessibility Laws.

Additionally, I was in charge of recording all standup meetings, updating some of the documentation as needed, and participated in recruitment of users for testing of the site debugging to ensure the site runs smoothly.

Team Member Four:

During our initial meeting with Krista, I had a few ideas for ways I would contribute; however, once we got into our groups and started, my focus changed. For the first couple of weeks working on the project, most of my effort went into researching file management, because Krista was speaking to us, one of the ideas she was interested in was a way to help the organization's file management of their Google Drives. While I was able to find a third-party Google add-on that could alleviate some of their troubles, ultimately, this was scrapped, and our group's focus went to the Intranet and its functions. While we didn't end up including this in the project, it can still be a tool that the organization can look into and decide if they would like to implement it. Once we knew what features we wanted to have in our project, I was able to change my focus to documentation, keeping consistent contact with Krista, going to their offices for walkthroughs, and speaking to others in the organization to ensure we could create the best product for their

needs. This weekly communication helped ensure we kept the scope of the project to just the customer's needs.

One of the features the HRDC was looking for was an integrated Google calendar so employees could keep track of important days and events within the organization. I worked on this in part with others in the group and was able to get the organization's calendar integrated into the site.

Team Member Five:

My contributions to this project were much more focused at the start of the project compared to the finishing touches. I am very familiar with NextJS, and introduced that as a possible framework for our website. This made sense to me since it has a lot of plugins and libraries available to put on the website if needed. The structure of a NextJS project is also pretty intuitive and I had a lot of knowledge from a previous class.

My main focus in the first two sprints was creating the skeleton for our website. We decided what tabs we wanted to have, and the general structure our website should have, and then I built the skeleton based on that. I made a new NextJS project, created the general structure for what pages we should have and then created a responsive navigation bar. The responsive navigation bar allows for our website to be compatible with computers, phones, and tablets.

In the beginning of the project I also researched some ways that we as a team could better utilize GitHub as some of us were new to it. I worked with a friend outside the class to learn about GitKraken and then brought that information to the group and especially the designer. This visual representation of GitHub, helped some of us that were new to GitHub. Throughout the middle of the project I just worked on adding content to the site. Towards the end of the project I created a Google Forms page that would allow us to conduct User Testing with people outside of the class. The feedback we got from user testing helped us identify the final changes we needed to make to our website.

Team Member Six:

This project has been quite the learning experience in both working as a group and doing applicable web development. At the beginning I started out working on authentication however after a week or so we deemed it non-essential to the project. I tried to implement it using builder.io however integrating it caused some deprecation on my computer that forced me to delegate quite a bit of time resolving those. Since then we pivoted and Mark was able to implement an authentication using google firebase.

In all honesty I wasn't able to give this project as much attention as I would have liked due to a very heavy credit count while working a part time job. That being said I took on a more holistic

role of working on a broad range of topics relating to our app. Namely, dev docs, code commenting, running some user testing and touching up some of our js pages.

Designer Contributions

Designer:

At the beginning of this project, my job was to work with Talia in defining the UX/UI wireframes within Figma, and integrating the HRDC brand guidelines as we received them. Integration included creating a shared folder of assets, setting stylistic standards within our Figma document, and learning more about the functionality needs as we met more with Krista.

At this point in the project, I had the opportunity to spend time with guest speaker Taekyeom Lee, a graphic design professor who specializes in visual disability, and how graphic design and type accommodate those disabilities. After I attended his workshops and his lecture, I was able to bring that research back to Talia and Sierrah to better integrate the research into our work. This was vital to better understand color contrast and make standardized type decisions.

Once more of the backend was built and we had clarified the functionality needs from Krista, I was able to rebuild a template on Figma, which included color combinations, type settings with the type families used, and overall button functionality of the webpage through the prototype settings. I was then able to pay for three (dev) seats, which allowed for Talia and Sierrah to "copy" necessary portions of my Figma document into usable HTML to be applied in GitHub.

Lastly, I assisted in creating a pitch deck to present to the class and HRDC, communicating the amount of collaboration, time, effort, and care we put into the project as a group.

Section 3: Design Pattern

Our project does not adhere to a single design pattern; rather, it incorporates multiple design patterns and principles to meet the specific functional requirements defined during our initial meetings. Our development approach was guided by the project scope and timeline, focusing on delivering prioritized features that aligned with user needs and technical feasibility.

One of the primary design patterns we implemented is Role-Based Access Control (RBAC). RBAC is a control methodology that assigns permissions to users based on their roles within an organization. This pattern was chosen in direct response to the project requirement that administrators and staff have distinct access levels. Specifically, administrators were to be granted privileges to post announcements, update the calendar, and manage backend content such as resource links, while staff members were restricted to read-only access.

To implement RBAC effectively, we created two distinct user roles, *Admin* and *Staff*, each with associated permission sets. We implemented backend logic to create the different permission levels. We integrated features to allow administrators to make back end edits of contents such as updating links. By centralizing permissions based on roles and not granting individual permissions, RBAC simplifies authorization logic. It also can also be used to mitigate cybersecurity risks, protect sensitive data by leveraging the principle of least privilege (Tunggal).

Additionally, our application follows the Model-View-Controller (MVC) pattern, which divides the application into three interconnected components:

- **Model**: Manages the core data structures and logic, including the different types of permissions granted to the staff and administrator account.
- View: Handles the presentation layer, displaying content tailored to staff and administrators through a responsive and accessible UI.
- **Controller**: The controller handles user input, enforcing role-based permissions, and executing actions such as posting announcements or creating calendar events.



This diagram shows the flow of data when using the MVC pattern. In our program when posting an announcement data is stored in the backend of the model such as the name of the author. When the announcement is posted the view aspect comes into play. Here the UI works ensures that the announcement posted is ADA compliant and renders correctly. The controller is the logic that connects the frontend to the backend. In this example it would be the logic that allows a user to delete the announcement. The main goal of implementing MVC is to solve the problem of users controlling a large and complex data set by breaking it down (GeeksforGeeks).

Section 4: Technical Writing

Link to Developer Documentation: https://github.com/423S25/repo5/blob/main/docs/dev-docs.html

Link to User Documentation: https://github.com/423S25/repo5/blob/main/docs/user-docs.html

Section 5: UML







Section 6: Design Trade-Offs

Design Tradeoffs and Decision-Making Process

Throughout the course of our project, our team encountered numerous design tradeoffs that required careful decision. In evaluating which features and designs to implement or exclude, we consistently prioritized usability and the core functionalities requested by the HRDC.

Simplicity vs. Feature Richness

One of the most significant trade-offs involved balancing simplicity with feature completeness. We had to choose between building a streamlined, fully functional platform and incorporating a broader range of features that risked being incomplete or unstable due to time constraints. Ultimately, we chose to focus on delivering a dependable and intuitive experience.

To maintain this focus, we removed certain features originally planned in our development roadmap. For instance, we decided against integrating Strapi for backend editing and eliminated the Google sign-in feature, both of which would have added unnecessary complexity. We also simplified user account roles from three levels, admin, staff, and volunteer, to two roles, admin and staff. After assessing user needs, we determined that staff members only required read-only access, making individual logins for them redundant and inefficient to implement.

UI/UX Considerations

Another major area of tradeoff was in user interface (UI) and user experience (UX) design. During our initial meeting with HRDC, we learned that the primary users of this platform would be elderly individuals who may not be highly proficient with technology. As a result, accessibility and ease of use became central to our design approach.

We deviated from standard design conventions to better meet our users' needs. This included using larger, bolder, and more legible fonts, as well as maintaining high contrast across the interface to improve visibility. These accessibility enhancements were continuously evaluated through multiple rounds of user testing to ensure the final product met accessibility standards.

Security vs. Ease of Use

We also considered the tradeoff between security and user-friendliness. One key decision in this area was our implementation of Role-Based Access Control (RBAC). This architecture added an essential layer of security by restricting critical content modification capabilities to admin users only. While this made the system more complex to build and slightly less intuitive to use, we determined the enhanced security justified the additional effort.

We also evaluated the possibility of integrating Google authentication for login purposes. However, we concluded that the marginal gains in usability did not outweigh the added complexity and potential security concerns, especially given our user base and project scope.

Section 7: Software Development Life Cycle Model

Our team followed the Agile development model, specifically utilizing the Scrum methodology. For each two-week sprint, a team member was designated as the Scrum Master, responsible for leading stand-up meetings, facilitating sprint activities, and laying out any additional work that needed to be completed.

Each sprint began with a planning meeting during which we reviewed the product backlog and prioritized the features to be developed. Here we set our sprint goal and put to words what we hoped to complete this sprint as a group. Throughout the sprint, we held Scrum meetings to communicate progress, identify blockers, and coordinate on resolving any issues that emerged.

Adopting Agile with Scrum proved effective for our project's development over the course of the semester. This iterative approach enabled us to remain organized, flexible, and aligned as a team. Although standard Scrum practices recommend daily stand-ups, our team was only able to meet in person three times per week due to class scheduling. This occasionally made it difficult to stay fully updated on individual progress. To mitigate this, we maintained detailed Sprint Notes, allowing team members to document issues, blockers, and ongoing work. We communicated through text and email to support the group's understanding of what was being worked on by whom, what needed to be done, and any other important information. This documentation ensures transparency and enables team members to offer support when needed.

HRDC Intranet Software

Developer Documentation v1.0.0

> Talia Clarke Mark Kwapisz Sierrah Paul Ryan Plusch Ryley Sanden Henry Weston

Motivation

The primary motivation for the development of this software arose from the needs of the HRDC to increase ease of use for managing employee's timesheets, benefits, accessing the employee handbook, filling out forms from the administrators, and an ability to search documents within their Google Drive. Prior to the introduction to our software, the HRDC relied on using multiple Google Drives to store information, which included 3 separate drives, an "Executive drive", a "managers drive", and an "everyone drive". This method was not only confusing but prone to issues, leading to errors in data management and gathering.

Purpose

Our software solution has been created to reduce inefficiency and errors in data collection while improving ease of use for the elderly volunteers and workers that the HRDC employs. It is designed to be as simple as possible so as not to confuse any users of the web app, this is emphasized using larger fonts, utilizing a minimalistic page design, and using as few pages as possible. The overall outcome of our product is to reduce the time needed to find and complete important forms, timesheets, or handbooks, while providing a simple, easy-to-follow design, specifically for those who may not have a comprehensive background using the HRDC's Google Drive or any online applications.

Directory Structure

Backend:

Because this is a static site, we do not have a separate backend structure that contains any controllers or servers. Our program consists of TypeScript built on Node.js using Figma to compile CSS for the site design. For user authentication, we used Firebase to allow users to sign in with their Google Accounts. HRDC staff and administration use Google for all their internal communications.

Frontend:

The frontend is built on Node.js, and authentication is using Firebase.

| -public -src |
|-------------------|
| нарр |
| announcements |
| components |
| help |
| links |
| login |
| l ——search |
| ↓staff |
| |
| |
| |
| |
| tirebase tirebase |

- ./public Static assets
- ./src The heart of the web application
 - ./app Folders contain the different aspects of the webapp
 - ./announcements Code for announcements page
 - ./components Navigation bar
 - ./help Code for Help page
 - ./links Code for links page
 - ./login Code for login page
 - ./search Code for search page
 - ./staff Code for the staff page
 - ./_tests_ Tests for the project
 - ./utils Firebase Configuration

Builds

- \rightarrow Zero Feature Release
- →Alpha Release
- →Beta Release
- \rightarrow Feature Complete Release
- \rightarrow Release Candidate
- →Final Release

Testing

Testing was done using Selenium User testing was done using Google Forms. Links have been sent out to friends, family, coworkers, and other students for testing.

USER TESTING FORM

Releases

- \rightarrow Zero Feature Release \rightarrow February 7, 2025
- \rightarrow Alpha Release \rightarrow February 21, 2025
- \rightarrow Beta Release \rightarrow March 7, 2025
- \rightarrow Feature Complete Release \rightarrow March 28, 2025
- \rightarrow Release Candidate \rightarrow April 11, 2025
- \rightarrow Final Release \rightarrow May 2, 2025

Errors and Bugs

For any issues, please submit a ticket here: https://github.com/423S25/repo5/issues

Code Format and Documentation

We included comprehensive, well-structured comments throughout the codebase to ensure future developers can easily understand and maintain the project. These comments range from single-line summaries to multi-sentence explanations, depending on the complexity of the code section. Each comment provides a high-level overview of the feature's purpose, how it functions, and any important implementation details or design considerations.

For example:

// main function that shows the Links.

This comment precedes approximately 20 lines of code responsible for implementing the logic and layout of the Links page.

HRDC Intranet Software

User Documentation

Talia Clarke Mark Kwapisz Sierrah Paul Ryan Plusch Ryley Sanden Henry Weston

Description

This software solution is designed to increase usability and productivity to ensure smooth operating procedures of the HRDC's employee operations. It aims to assist new and continuing employees and reduce the operational burden to its administrators, spreading pertinent information to its employees and volunteers accurately and promptly.

How to obtain and install software

This is a web app, there is no requirement to install any software

How to run software

This is a webapp. To use the app go to the webpage here: https://repo5-rho.vercel.app/

How to use software

We will discuss with our HRDC contact on how to best integrate our design with the system they are currently using.

How to report a bug

To submit a bug, go to the <u>GitHub issues page</u> and click the "New Issue" button. <u>https://github.com/423S25/repo5/issues</u>