CAPSTONE PORTFOLIO

Web App For HRDC Warming Shelter

ESOF 423 - Spring 2024

Shane Costello, Cody Fingerson, Sage James, and Jonathan Neuman

ESOF 423 Group 3 Portfolio

Section 1: Program

At the beginning of the semester, Jenna Huey, a member of the HRDC, brought to light a pressing issue faced by the organization: the laborious task of sifting through many Google Spreadsheets every night to manage check-ins, bunk and locker assignments, and guest statuses. She underscored the inefficiency and frustration this process caused, particularly the lack of visual indicators to easily identify guests who were not allowed on the premises. Jenna emphasized the urgent need for a streamlined solution that could consolidate these scattered spreadsheets and provide intuitive visual cues for identifying trespassers. This challenge not only hindered nightly operations but also posed security risks due to the inability to promptly recognize unauthorized guests. Jenna presented this critical problem to the team, emphasizing the imperative to develop a web application that not only addressed the current inefficiencies but also prioritized visual clarity and security in managing HRDC operations.

To tackle these pressing concerns head-on, our team embarked on the development of a comprehensive web application. This innovative solution was meticulously crafted to streamline the check-in processes, harnessing the power of real-time data updates. By integrating cutting-edge technology, we aimed to revolutionize the check-in experience for both the HRDC team and volunteers, ensuring unparalleled efficiency and effectiveness in every step of the process. With a user-friendly interface and seamless functionality, our web application not only simplifies the check-in process but also empowers HRDC staff to make informed decisions swiftly, ultimately enhancing overall operational efficiency.

Here is a link to our project
And to our GitHub

Section 2: Teamwork

Team Member One

Team Member One's contributions to the team have been encompassed by the development of two distinct features. The first feature is the back-end database model and controllers for handling guest data. The database model has attributes for all information the client wishes to store and have access to as it pertains to guests of the warming shelter. The controller for the model has multiple methods that allow the database to be seamlessly integrated with the front end. These methods include PULL, PUT, and PUSH requests which facilitate the viewing and modification of data entries. The second feature is the guest profile. The guest profile is accessible in multiple locations of the application and allows the user to view all data being stored for an individual guest. Further, the user can update any of this information and save the changes to the database via the aforementioned controller methods. A high priority while developing this feature was the ease of use and highly abstracted database integration.

As the guest model, and its controller, are used in nearly every feature of the application, Team Member One has become the point person for any relevant questions. They aid in the implementation of their code in other features and develop new controller methods as needed.

Team Member Two

Team Member Two has been an active participant in the development cycle of the project, consistently fulfilling the tasks set during SCRUM meetings. They took the initiative to set up the backend API directory on GitHub and crafted the developer documentation. In the meetings, they proposed the use of Express with MongoDB for the backend and volunteered to implement user authentication, which the team agreed to. They also took charge of state management by implementing Vuex after evaluating its suitability for the project.

They successfully integrated new features into both user and developer documentation and contributed to resolving challenges related to state management. More recently, they concentrated on adding more robust authorization checks and overcoming hosting hurdles. Following several discussions with the school's server admin, they managed to host the API on the school's server. While the Vue.js app is now hosted on Vercel, they encountered and resolved HTTPS communication issues with help from the admin. Although there's a persisting challenge with data transfer between the frontend and backend, the hosting setup is almost finalized.

Team Member Three

Throughout the project's development cycle, Team Member Three has actively engaged in fulfilling tasks outlined during Scrum meetings, playing a pivotal role in shaping both the design and technical aspects of the project. One of their key contributions has been the establishment of the initial Figma UI prototype, aligning it with the project's objectives and iterating upon it to ensure optimal user experience. Additionally, they've dedicated time to gaining insights into Vue.js, Tailwind CSS, and user empathy, all of which have informed the project's direction and implementation strategies.

In terms of technical responsibilities, they've taken ownership of researching hosting platforms and delving into Vue.js documentation to solidify the frontend architecture. Furthermore, they've actively participated in team development research, drawing inspiration from the Agile method and "ShapeUp" by Basecamp to enhance collaborative dynamics and project management strategies. Their commitment to effective communication and coordination is evident in their efforts to facilitate team discussions and organize activities such as visits to the warming shelter, which was pivotal in understanding user needs. Moving forward, they aim to continue contributing to both the design and technical aspects of the project, ensuring its success through proactive engagement and continuous improvement.

Team Member Four

Team Member Four's primary area of focus during their involvement in the project was the front-end development of the web application. A significant portion of their efforts was directed towards the creation and refinement of the "Enter Guests" page, as it is the primary replacement for the Google Sheets page used by HRDC for tracking and entering incoming guests into their system. They dedicated a substantial portion of their time towards ensuring its functionality and ease of use. They also implemented other features such as the site-wide header, the “Assign Spaces” page, and CSS across the app following the designs laid out in the Figma UI prototype.

Section 3: Design Pattern

In our approach to developing our capstone project, our team didn't strictly follow one particular design pattern. Instead, we took a practical approach influenced by what works well in the industry and what our project needed. Our decisions were based on various factors, like making sure our project could be easily changed, could grow if needed, and wouldn't be hard to maintain. We also considered the skills and preferences of each team member.

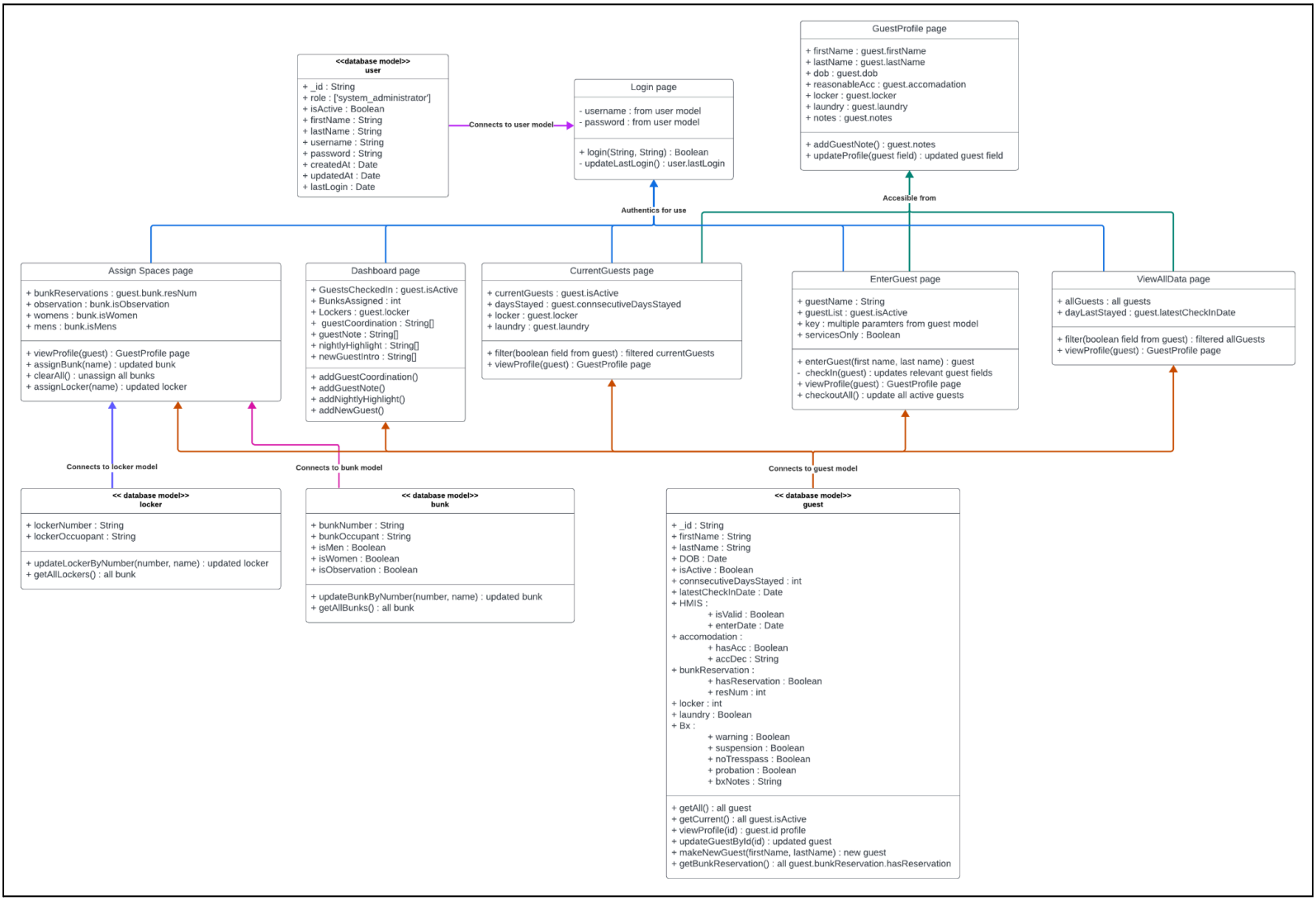
Even though we didn't stick to a specific design plan, our project naturally ended up resembling the Model-View-Controller (MVC) architecture. This means we separated the way our project looked and felt (the "View") from how it handled data and worked behind the scenes (the "Model" and "Controller"). We used Vue.js for the part users interact with, and Node.js with Express for managing data and how the app works. This setup helped us work well together, make things efficient, and keep our code organized.

Additionally, we chose tools like Vuex for managing the app's state and MongoDB for handling the database based on what would make our project run smoothly and perform well with our technology stack. We focused on keeping things simple and practical, choosing solutions that fit well with what we were already using and met our project's specific needs.

Although our approach didn't strictly follow one set design pattern, it combined industry best practices tailored to our project's unique requirements. This flexibility allowed us to adapt quickly to changes and ultimately contributed to the success of our project.

Section 4: Technical Writing
User documentation can be found here.
Developer documentation can be found here.

Section 5: UML



[Link for expanded view]

Section 6: Design trade-offs

One significant design trade-off we faced was in choosing the appropriate deployment strategy for our backend API, written in TypeScript, using Express.js as the server.

Initially, we considered using the school-provided server to deploy both our frontend and backend. However, we encountered difficulties connecting to our MongoDB Atlas database due to firewall protocols implemented after last spring's cyberattack.

In an attempt to find an alternative solution, we explored platforms like Netlify and Vercel, both of which successfully hosted the frontend but lacked support for the backend. These platforms

are optimized for static site hosting and client-side rendering frameworks, presenting significant limitations for our backend requirements.

Given the challenges posed by these hosting solutions, we deliberated on the trade-offs involved. Ultimately, we decided to deploy our application locally. This decision required us to weigh the benefits of high availability and scalability against the control and cost-effectiveness of local deployment. Our justification for this decision includes:

- Control and configuration: Local deployment grants us complete control over the server environment, enabling precise configuration tailored to our project's requirements.
- Cost-effectiveness: By deploying locally, we eliminate additional costs for both our team and any external hosting service provider.

This trade-off underscored the importance of aligning technical decisions with project goals, available resources, and constraints. It provided valuable insights into the necessity of balancing practical implementations with the project's overarching objectives.

Section 7: Software development life cycle model

The software development model we used for this project was the agile model with the scrum methodology. For each sprint, one member of the team was designated the scrum master who then facilitated the scrum meetings for the duration of the sprint. During each 2-week sprint, we had one sprint planning meeting, where we would consult the product backlog to plan what features would be added during the current sprint, and three scrum meetings, during which we would check in on our progress during the sprint, and discuss any problems and or progression blockers members of the team encountered since the previous meeting.

The agile model was a good fit for the development of our project due to the limited development time we had over the span of the semester. The scrum methodology was helpful in keeping each member of the team on the same page in regards to the current state of the development. The downside of the agile scrum approach to development was the fact that the team was only able to meet four times each sprint. Scrum is usually employed in environments where daily meetings are possible. With our team’s busy schedules, and only 3 actual scrum meetings per sprint, it was challenging to keep up to date on what other members were working on, as well as helping each other with any issues that came up.

Warming Hut Software

Developer Documentation

v1.0.0

Shane Costello
Cody Fingerson
Sage James
Jonathan Neuman

Table of contents

Motivation **3**
Purpose **3**
Directory structure **3**
Backend 3
Frontend 4
Builds **4**
Testing **4**
Automations **4**
Releases **5**
Errors and Bugs **5**
Code Format and Documentation **5**

Motivation

The primary motivation behind the development of this software stems from the need to streamline and enhance the efficiency of managing guest data at the HRDC's homeless warming center. Prior to the introduction of this software, the center relied heavily on manual data entry processes, utilizing Google Sheets to input and manage information across two separate databases: one internal (CaseWorthy) and one statewide (HMIS). This method was not only time-consuming but also prone to errors, leading to inefficiencies in data management and reporting.

Purpose

This custom software solution is designed to enhance the operation of the HRDC's year-round homeless warming center in Bozeman, Montana. It aims to assist front desk staff in managing incoming and outgoing guests efficiently. This software aims to minimize the learning curve and reduce the operation burden on front desk staff.

Directory structure

Backend

The backend consists of a JavaScript REST API built on Node.js using Express.js as the application framework.

```
warming-center-api/
├── README.md
├── config/
├── controllers/
├── middleware/
├── models/
└── routes/
└── server.js
```

- `config/` - Application configuration settings (i.e. database operations, API keys, connection settings).

- `controllers/` - Logic of how user requests are handled and how the response is formulated.
- `middleware/` - Custom middleware functions that are used to perform custom ./operations like logging, authentication, authorization, and error handling.
- `models/` - Representation of MongoDB collections.
- `routes/` - The endpoints of the web application.

Frontend

The frontend consists of a Vuejs project built on Node.js and Vite as the development server.

```
warming-center/
├── public/
├── src/
|    ├── assets/
|    ├── components/
|    ├── views/
|    ├── App.vue
|    ├── Router.js
|    ├── main.js
|    ├── style.css
├── README.md
├── index.html
├── models/
└── routes/
└── server.js
```

- `./public` - Static assets that should not be bundled by the transpiler.
- `./src` - The “heart” of the application.
    - `/assets` - Assets used by components. (i.e. SASS, CSS, images, etc…)
    - `/components` - Reusable Vue instances that define a part of the U.I.
    - `/views` - The pages of the application associated with the application HTTP routes.

Builds

➔ Zero Feature Release

- ➔ Beta Release
- ➔ Feature Complete Release
- ➔ Release Candidate
- ➔ Final Release

Testing

Testing was completed using Selenium for the front-end…….

Releases

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Aliquet bibendum enim facilisis gravida neque convallis. Placerat duis ultricies lacus sed turpis tincidunt id aliquet. Volutpat est velit egestas dui id ornare arcu.

Errors and Bugs

To submit a bug, go to https://github.com/ShaneCost/esof423-warming_shelter/issues and click the “New Issue” button.

Code Format and Documentation

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Aliquet bibendum enim facilisis gravida neque convallis. Placerat duis ultricies lacus sed turpis tincidunt id aliquet. Volutpat est velit egestas dui id ornare arcu.

Warming Hut Software

User Documentation

v1.0.0

Shane Costello
Cody Fingerson
Sage James
Jonathan Neuman

Description

This custom software solution is designed to enhance the operation of the HRDC's year-round homeless warming center in Bozeman, Montana. It aims to assist front desk staff in managing incoming and outgoing guests efficiently. This software aims to minimize the learning curve and reduce the operation burden on front desk staff.

How to obtain and install software

N/A the software is a web app.

How to run software

To run the application, all that is needed is to visit http://localhost:5173/ and enter your username and password.

How to use software

Still waiting to visit Warming Shelter to determine how to best design our user interface to fit their workflow.

How to report a bug

To submit a bug, go to https://github.com/ShaneCost/esof423-warming_shelter/issues and click the “New Issue” button.