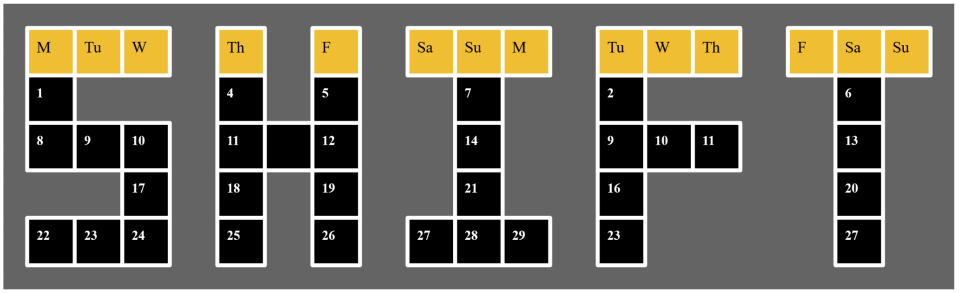

Team 18 Project Backlog

TEAM MEMBERS: Danielle Ejiogu, Daniel Fakunle, Murtuza Kagalwala, Lucas Munteanu, Jenna Rigdon, & Samson Tesfagiorgis

PROJECT NAME: Shift

Problem Statement

University Residences' (UR) current system for assigning duty shifts to Resident Assistants (RA) is archaic, time-consuming, and often leads to asymmetric workloads. For example, at Meredith Hall scheduling duty involves generating a randomized list of the names of the 12 RAs and using this list to assign duties to approximately 70 time slots. Fairly and effectively assigning these slots for a given month can take time, especially when accounting for exceptions. Our goal for this project is to develop a robust, fully-featured website to address these challenges. These features include accommodation for the plurality of duties (holiday, weekday, etc...), and status of duty (lead, secondary, tertiary, desk). From the team's given availability the website will suggest a fair schedule. The specificity of these features and the ability for modification make our solution distinct from other systems on the market all while being free to use (i.e. When2Meet, SignUpGenius, WhenIWork).

Background Information

Targeted Users

The employees at Purdue University Residences (UR) are full-time undergraduate students with hectic schedules, and over the years many aspects of Purdue Student life have been made easier through digitization (e.g Purdue Mobile ID). These student employees of UR work in tight-knit communities, and are generally involved in the wider university community as that is a valued part of their application process. This service is meant to be added to the digital arsenal of modern student life for our RA's and Residential Education Assistants (REA), so that they can easily update their availability and view and maintain their work lives from their own laptops or phones, instead of having to add yet another meeting to their busy lives.

Resident Education Coordinators (RECs) take a more administrative role over both the RAs and REAs, and are not students of the university, but they will reap the same benefits of digitization as the other two target groups.

Similar Platforms

There are similar scheduling platforms like SignUpGenius, When2Meet, and WhenIWork. SignUpGenius allows a user to present several available slots and send them out to others so that they may choose slots on a first-come first-serve basis. When2Meet allows a team to fill out their availability and visualize when members of the team are and aren't available. WhenIWork allows for teams of workplaces to visualize their work schedules in a way similar to what we described, and also has payroll capabilities.

Limitations of Current Platforms

Neither of the aforementioned make any attempt to guarantee an equitable distribution of work for teams, and they additionally lack the specificity that UR would need, such as assigning the different levels of duty (e.g., primary, secondary, tertiary).

Our solution would suggest to the admin (REA or REC) an ideal equitable schedule by analyzing an RA's current time commitments and will assign them a priority (i.e., RA's with more constraints have a higher priority over selecting time slots). Additionally, our solution will keep track of how many times an RA has performed a certain duty (i.e., how many times they have completed a primary, secondary, or tertiary duty) and will assign them the duty accordingly.

WhenIWork is more compatible with workers of hourly positions, while RA's are on a fixed income and are reimbursed primarily through benefits like free room and board, our solution is separate from issues of pay and focuses simply on a straightforward organization of duty. Additionally, they both lack built-in channels of communication to the admin (which would be the REA or REC in this case), like if someone needed to request a change of shift and provided reasoning for that.

Functional Requirements

Management

1. As a user, I would like to be able to create and login to my account.
2. As a user, I would like to be able to reset my password.
3. As a user, I want a dashboard where I can view relevant information, such as my display name, position, and work schedule.
4. As a user, I would like to be able to edit my account details and set up more secure methods of authentication.
5. As a user, I want the ability to edit and update my contact details (phone number, email address) for accurate communication and notification purposes.
6. As a user, I would like to be able to specify the hall I belong to.
7. As a user, I would like to be able to specify my role (RA, REA, REC).
8. As an REA or REC, I would like to be able to create RA accounts as needed and notify new RAs of the new accounts.
9. As an REA or REC, I would like to view at a glance the current status of RAs, e.g, active, inactive, unavailable.

Customization

10. As a user, I would like to be able to select from a variety of themes, allowing me to personalize my visual experience.
11. As a user, I want to be able to set notification preferences, allowing me to choose how and when I receive alerts about shifts, announcements, or other important updates.
12. As a user, I would like to customize the level of detail displayed on my calendar view, allowing me to toggle between daily, weekly, and monthly views based on my preference and workflow.

Scheduling

13. As a user, I would like to be able to submit my availability to the scheduler.
14. As a user I would like to export my schedule for use on other calendar applications/websites such as Google Calendar.
15. As a user, I would like to be able to download the schedule as a PDF.
16. As a user, I would like to be able to schedule meetings with other users.
17. As a user, I would like to be able to change meeting times or cancel after scheduling one.
18. As a user, I would like to receive email notifications to be notified of schedule changes or upcoming events.
19. As an RA, I would like to be able to submit my availability to the scheduler.
20. As an RA, I would like to be able to indicate a preference for certain shifts.
21. As an RA, I would like to be notified when the shift schedule has been finalized.
22. As an RA, I would like to be able to request to drop shifts assigned to me in the face of any personal emergencies.
23. As an RA, I would like to be able to view shifts that are open for taking due to late shift drops.
24. As an RA, I would like to view a detailed schedule containing my shifts in daily, weekly, and monthly formats such that I can have various views of my schedule.
25. As an RA, I'd like to be able to see which RA's are available on any given day (for duty switches, etc.).
26. As an REA or REC, I would like to decline or accept shift drops depending on an RA's given reasoning.
27. As an REA or REC, I would like to create events in bulk (blocking different shifts of certain intervals, e.g., 15,30, 60 minutes, etc., within a certain time range) to make it easier to create events and shifts as they will be back to back.
28. As an REA or REC, I would like to assign time slots randomly to my RAs based on their priority such that the shifts can be distributed fairly while also accommodating for RAs with more time constraints.
29. As an REA or REC, I would like to be able to manually edit the schedule as needed (e.g., changing the times of shifts, etc.).

Messaging

30. As a user, I would like to chat with other users privately.
31. As a user, I would like to have access to a public chat of all users in the same residence hall.
32. As a user, I would like to have the option to receive chat notifications.
33. As a user, I would like to be able to report inappropriate messages in the chat.
34. As a user, I would like the ability to delete messages.
35. As a user I would like to pin messages, so that they won't be deleted after a week.

Non-Functional Requirements

Architecture

The website will be split into 2 layers: front-end and back-end. The front-end will be built using React, to give users a simple and easy to navigate interface. The back-end will be implemented with Java using the Spring Boot framework. We will be using SQL as our database to store information. We will be using REST APIs to connect our back-end with our front-end. For better performance, users should be able to store and access their data for at least a 6 month time period. Users will only be able to access messages from the past 7 days, unless older messages are pinned.

Security

We will be utilizing Auth0 to handle user authentication and authorization. This will provide a more secure and streamlined way for users to create and login to their accounts. We will also be using a framework (Spring Boot) that prevents SQL injections. To preventDoS/ DDoS we will be limiting the number of messages that a user can send in a conversation to 500 messages a day, the number of button clicks will be limited to 1 every 5 second during times of high traffic. We would also like to fix bugs within a 48-hour time period from when the bug was discovered, depending on the criticality of the bug.

Usability

To ensure usability, our front-end will adhere to essential design principles such as visual hierarchy, consistency, spacing and appropriate use of color. We will also make it more accessible to a wide range of users, with the website being able to operate 24/7 as well as being able to handle at least 1000 users at a time, to allow for high traffic. Furthermore, we will make sure the website is still user-friendly while using it from a mobile device. If time permits, there is a scalable option to expand to more universities. Additionally to increase accessibility, our solution will include the option for high-contrast color schemes and accommodations for color-blind individuals.

Deployment/Hosting

We would like to host the website on Netlify which will simplify our deployment and will mainly be hosting our front-end and will help connect it with our back-end. We will host our back-end on Heroku. (We will try asking Purdue's hosting service to make things easier.) If time permits, we will be using Docker and Jenkins to build a CI/CD pipeline.