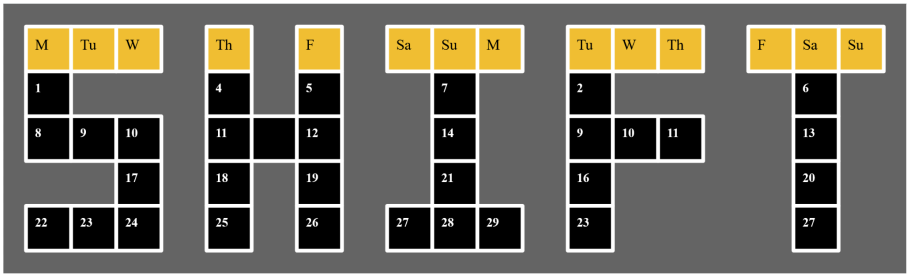

Team 18 Sprint 1 Retrospective

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

PROJECT NAME: Shift

Keywords:

RA: Resident Assistant

REA: Resident Education Assistant

REC: Resident Education Coordinator

Sample

Rubric

What went well?

In general we created the main pages for both resident assistants and Admins (resident education assistants/resident education coordinators) and implemented the basic functionality for creating, reading, updating and deleting events. We also implemented basic user management.

Successfully Completed User Stories:

- **User Story 1: As a user, I would like to be able to create my account.**

1	Create a basic landing page with login and signup buttons.	2 hours	Daniel
2	Link signup functionality with Auth0 API to handle signup.	2 hours	Daniel
3	Implement user onboarding to collect relevant user data and create a new user.	4 hours	Murtuza
4	Create and implement unit tests to verify account creation functionality.	2 hours	Murtuza
5	Implement user input validation i.e, no duplicate emails, strong passwords.	2 hours	Murtuza

Completed:

Signup and login operations are fully operational thanks to the integration of the Auth0 API. Auth0 provided us with a good looking and secure user interface for login operations. It handles duplicate emails, weak passwords, and user authorization ensuring Shift has a secure framework.

- **User Story 2: As a user, I would like to login to my account.**

1	Link login functionality with Auth0 API to handle login.	6 hours	Murtuza
2	Create and implement unit tests to login functionality.	2 hours	Murtuza
3	Implement user friendly error handling i.e., appropriate UI for invalid credentials.	2 hours	Daniel

Completed:

We have successfully implemented login through Auth0 to our application. So now, when a user logs in or registers an account, a user on the Shift application is automatically linked to the user login. Additionally, when a user types a password incorrectly or doesn't meet password requirements, Auth0 handles the errors with built in user interface error handling that relays clear feedback to the user.

- **User Story 3: As a user, I would like to be able to reset my password.**

1	Link verification of a user with Auth0 API.	4 hours	Danielle
2	Create and implement unit tests to login functionality.	2 hours	Danielle
3	Create and maintain criteria for password strength.	2 hours	Danielle
4	Implement a stable reset process.	2 hours	Danielle

Completed:

Resetting a user's password was accomplished through built in functions in the Auth0 API. Our group agreed the criteria provided by the Auth0 was strong itself and ensured password security for all users. Our reset process is simple. Once a user expresses they forget their password, they are prompted to enter their email and if that email is found in the database, they will be sent a change password link. From there, they can change their password and log in normally.

- **User Story 4: As a user, I would like to delete my account and all of its associated data.**

1	Verify the user's identity using Auth0 API.	1 hour	Daniel
2	Search for user's information within the database (schedule, messages, meetings) and safely delete it..	3 hours	Lucas
3	Search for the user's account within the database to delete personal information and remove all associations with other users.	4 hours	Lucas
4	Create and implement unit tests to confirm safe and thorough deletion of user accounts.	3 hours	Lucas

Completed:

To initiate account deletion, user's have to navigate to their profile and select the "Delete Account" button. Upon selecting to delete an account we prompt them with a confirmation in order to go through with the operation. This is important as deleting an account will erase all availability and shifts the user has, so ensuring it is the user's true intention is expected on our end. All the events linked to the user are deleted after the confirmation ensuring optimization of our memory usage.

- **User Story 5: As a user, I want a dashboard where I can view relevant information, such as my display name, position, and work schedule.**

1	Create a dashboard page with dummy user data.	3 hours	Daniel
2	Create a basic calendar view with dummy schedule data.	3 hours	Daniel
3	Link a logged in user with a dashboard page.	2 hours	Lucas
4	Create and implement unit tests to verify dashboard page functionality.	2 hours	Daniel, Lucas

Completed:

The dashboard page features a prominently displayed calendar that includes navigation functionality. The dashboard is personalized for each user, including features such as a welcome message that highlights the user's name and role (RA, REA, REC). This personalized display improves user experience with a little personal touch and is an exhibit of how we personalized dashboards for all users. The layout is designed to be neat and organized in order to fulfill user expectations.

- **User Story 6: As a user, I would like to be able to edit my account details and set up more secure methods of authentication like logging in with social provider accounts (eg: Sign in with Google).**

1	Setup social login with Auth0 API.	3 hours	Murtuza
2	Create an account details page to show user data.	2 hours	Daniel
3	Link account details page with an authenticated user.	2 hours	Samson
4	Implement user data editing.	3 hours	Samson
5	Create and implement unit tests to verify social login and account details editing functionality.	2 hours	Samson

Completed:

Shift provides users the ability to log in with their personal google or windows accounts. This is important as our audience is students and staff at Purdue University and everyone at Purdue has an Outlook/Windows email account given to them by the university. This ensures everyone has easy access to an account and makes it easy to manage as the outlook email is most likely their main email they use on campus. User's are also granted the ability to change details of their account such as their name, password, etc., as long as they verify their password beforehand.

- **User Story 7: As an REA or REC, I would like to be able to create accounts for RAs as needed and supply them with a temporary password to access it.**

1	Create RA account creation page.	1 hour	Daniel
2	Using Auth0, implement RA account creation invite.	4 hours	Daniel
3	Implement reset password user flow for invited accounts.	4 hours	Daniel, Lucas
4	Create and implement unit tests to verify RA invite functionality.	1 hour	Lucas

Completed:

REAs and RECs both have access to an RA account creation page where they can enter their RA's email and assign them a temporary password. A link to reset the password will be sent to the RA and also be available to the executive for copy and sending directly. Either option will take the RA to the page where they choose their own personal password. After that, the RA will be granted an account to use.

- **User Story 9: As a user, I would like to add, edit, and delete my personal events.**

1	Creating buttons and editing options.	2 hours	Samson
2	Creating functionality for adding events.	2 hours	Jenna
3	Creating functionality for editing events.	2 hours	Jenna
4	Creating functionality for deleting events.	2 hours	Jenna
5	Debugging & Unit testing.	2 hours	Jenna

Completed:

Users can add, delete, and edit their personal events on the dashboard page. There are two ways to add an event: one by day and the other by a specific time slot within a day. To add an event on a day with no specific time, a user can click on any day in the monthly view and add a title and select create. By going into the day view, a user can slide and drag their mouse to specify the time interval an event is on a given day. Editing an event is simple and

can be done by clicking on the blue event box of a created event. After changing the name of an event, a user selects their changes by once again hitting create. Similar to editing, deleting requires the user to click on a made event and then hit delete. This suite of features ensures users can effortlessly manage their dashboard.

- **User Story 10: As a user, I would like to clear my schedule based on hour, day, week, or month.**

1	Creating UI for clearing schedules based on time.	3 hours	Danielle
2	Implementing functionality in the backend.	4 hours	Jenna
3	Debugging & Unit testing.	2 hours	Lucas

Completed:

In the backend there are options for the user to choose to clear their schedule based on hour, day , week, or month. Utilizing the current date as a reference point, all the events in the selected time frame are cleared from the calendar leaving users with a cleaned out schedule.

- **User Story 11: 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.**

1	Build multiple layouts for the calendar.	5 hours	Daniel
2	Build buttons to switch views and have a smooth transition.	3 hours	Daniel
3	Debugging & Unit testing.	2 hours	Daniel

Completed:

On the dashboard page, in the calendar view, the user can navigate forward and backward, and toggle between the month, week, day and agenda view.

- **User Story 12: As a user, I would like to be able to view future days, weeks, and months of the schedule.**

1	Building a calendar view to toggle between months.	3 hours	Danielle
2	Adding a smooth transition while toggling between months.	2 hours	Danielle
3	Debugging & Unit testing.	2 hours	Danielle

Completed:

Both the availability calendar and dashboard calendar have the ability for users to toggle between the different months and years. Both calendars use a forward and backwards arrow that the user can utilize to move forward in the months of the year as well as backwards. If a user strays too far from the current date, we also added a current date implementation whereupon clicking will send you to the current date.

- **User Story 13: As an RA, I would like to be able to select a preference for when to schedule certain shifts.**

1	Implement interactive calendar where select shift preferences.	6 hours	Samson
2	Implement functionality that verifies user selects minimum amount of preferences	2 hours	Jenna
3	Create and implement unit tests to verify preferences selection functionality.	4 hours	Jenna

Completed:

There is a calendar display that allows a user to select dates. The calendar is only shown in a monthly view. When selecting dates for shifts on the calendar as an RA, there is a counter in the bottom right hand corner indicating how many shifts are necessary to take. If the minimum number of shifts (as determined on the end of an REA/REC) is not selected by the RA, the RA is prompted with a warning notification. This gives them the option to either select more dates or submit anyway.

- **User Story 15: As an RA, I would like to be able to view shifts that are available due to late shift drops.**

2	Find and show RA who selected that shift as available to fill the shift if necessary, update number of shifts taken.	3 hours	Lucas
3	Implement functionality that keeps track of all upcoming available shifts.	3 hours	Lucas
4	Unit Testing	2 hours	Danielle

Completed:

The backend for this user story is complete and does exactly what the user story says. The shifts are stored in the REA/REC's schedule and they can make it visible so that the RAs can see the available shifts.

- **User Story 16: As an REA or REC, I would like to set a minimum/required number of shifts each RA has to select.**

1	Create field for REAs and RECs to illustrate minimum number of shifts for their RA's	1 hour	Samson
2	Notify RAs if not meeting threshold	2 hours	Samson
3	Notify REA if RA submits schedule with less shifts than required	2 hours	Samson
4	Give REA a list of their RAs with under the minimum shifts required	2 hours	Samson
5	Provide Feedback to RAs of how many more shifts are needed while building their schedule	2 hours	Samson
6	Unit Testing	1 hour	Samson

Completed:

In the executive page, an REA or REC can specify the minimum number of days that each RA should have specified in their availability. The number will be reflected in the availability page when an RA is selecting their shifts.

- **User Story 17: As an REA or REC, I would like to approve of/confirm all RA schedules as well as accept or decline shift drops, depending on an RA's given reasoning.**

2	Implement functionality of approval or denial	3 hours	Danielle
3	Unit testing	2 hours	Danielle

Completed

This part has been completed in the backend as checks have been made to prevent RAs from delete shifts, only allowing REAs and RECs to delete shifts when requested. However, we are still working on the front end part for this which will display a UI for the REAs and RECs to approve schedules and shift drops.

- **User Story 18: As an REA or REC, I would like to edit RA schedules (for last minute shift drops, etc.).**

1	Implement manual deleting/inserting/assigning shifts.	3 hours	Lucas
2	Implement automatically updating the count of all RA's shifts to reflect the manual change	2 hours	Lucas
3	Implement notifying members of a change in schedule	2 hours	Lucas
4	Unit Testing	1 hour	Lucas

Completed:

Using the user accounts, an REA or REC can edit an RA schedule that has been submitted. After the edit has been made, the REA or REC can confirm the changes and they will be reflected when the RA logs into their account.

What did not go well?

In general, our group was unable to connect the functionalities implemented in the front end with the associated ones in the back end. Additionally, one of our team members was absent during the last week of the sprint and we were unable to complete two user stories as a result.

Incomplete User Stories:

- **User Story 4: As a user, I would like to delete my account and all of its associated data.**

Not Completed:

Functionality for this user story has been implemented on both the front end and the back end. All we need to do is to connect the front end with the back end in order to properly execute this story.

- **User Story 8: As a user, I would like to be able to submit my availability to the scheduler.**

#	Description	Estimated Time	Owner
1	Create a submission button from within the users calendar page.	1 hour	Jenna
2	Create functionality for the submission button.	1 hour	Jenna
3	Prevent any race conditions when submitting a shift.	4 hours	Lucas
4	Unit Testing for valid submission and race conditions.	3 hours	Jenna

Not Completed:

Due to the absence of the developer primarily in charge of this story, this user story has been approved by the Project Coordinator to be completed in Sprint 2.

- **User Story 9: As a user, I would like to add, edit, and delete my personal events.**

Not Completed:

Functionality for this user story has been implemented on both the front end and the back end. All we need to do is to connect the front end with the back end in order to properly execute this story.

- **User Story 10: As a user, I would like to clear my schedule based on hour, day, week, or month.**

Not Completed:

Functionality for this user story has been implemented on both the front end and the back end. All we need to do is to connect the front end with the back end in order to properly execute this story.

- **User Story 13: As an RA, I would like to be able to select a preference for when to schedule certain shifts.**

Not Completed:

Functionality for this user story has been implemented on both the front end and the back end. All we need to do is to connect the front end with the back end in order to properly execute this story.

- **User Story 14: As an RA, I would like to be able to request to shift drops assigned to me in the face of any personal emergencies.**

#	Description	Estimated Time	Owner
1	Create an option for requesting a shift drop when viewing the assigned shift.	2 hours	Samson
2	Implement sending requests to REA/REC's account, possibly via message notification or home screen notification.	3 hours	Jenna
3	Implement functionality to deny or approve a request, with or without a message.	1 hour	Jenna
4	Implement functionality for the RA who dropped a shift to select a shift of the person who picked up theirs (a duty switch rather than just a drop)	2 hours	Jenna
5	Debugging & Unit testing.	1 hour	Jenna

Not Completed:

Due to the absence of the developer primarily in charge of this story, this user story has been approved by the Project Coordinator to be completed in Sprint 2.

- **User Story 15: As an RA, I would like to be able to view shifts that are available due to late shift drops.**

1	Implement a UI, like a board, that displays available shifts.	3 hours	Lucas

Not Completed:

Most of the functionality for this user story has only been implemented in the back end. Our group still needs to implement the UI on the front end in order for the RAs to interact with the shifts displayed. Additionally, we would then need to connect the front end with the back end in order to properly execute this story.

- **User Story 17: As an REA or REC, I would like to approve of/confirm all RA schedules as well as accept or decline shift drops, depending on an RA's given reasoning.**

1	UI for view of schedule requests and drops	4 hours	Danielle

Not Completed:

Most of the functionality for this user story has only been implemented in the back end. Our group still needs to implement the UI on the front end in order for the REAs and RECs to interact with a view of the schedule requests and drops. Additionally, we would then need to connect the front end with the back end in order to properly execute this story.

- **User Story 18: As an REA or REC, I would like to edit RA schedules (for last minute shift drops, etc.).**

Not Completed:

Functionality for this user story has been implemented on both the front end and the back end. All we need to do is to connect the front end with the back end in order to properly execute this story.

How should you improve?

Reflecting on our team's performance for the first sprint, we as a team did a pretty good job and we were able to finish most of our tasks completely. Although some of our tasks were partially completed (implemented on both the frontend and the backend, but not connected with each other). Overall, we were able to meet up to discuss what we have been doing and as teammates we were helpful to each other.

The very first thing that we could work on is trying to figure out and divide tasks evenly when we are planning for our sprints. Initially when we first divided tasks we had a fairly even split for our teammates handling frontend and backend, at least according to our sprint planning document. When we actually started implementation, some of us were able to get our tasks done quickly while others were not able to complete their tasks as easily, and it turned out that we had miscalculated our time estimates for some tasks.

One way that we could fix this is that we can spend some time before dividing our work to make sure that everyone has equal amounts of work or if a user story could be split between two people to help speed up our work on that particular user story. We also did not consider the time that we would end up taking to set-up our code base during this sprint which resulted in some uneven work for some of our team members. This also caused us to slow down on a few user stories that we had to leave partially completed. Due to this we will have some catching up to do for sprint 2.

Another issue that we faced during this sprint is that we ended up mixing some of our user stories with other team members' user stories. This not only caused some confusion in the team, but it also ended up causing imbalances in the division of tasks that were originally planned out.

To combat this issue, we will try our best to be more selective when dividing tasks while planning out the sprint and we will try our best to ensure that we comply with the document to the best of our abilities. Another way to ensure that we won't mix up tasks is that we as a team will mention what we are working on in our group chat, before we start for the day, so that other team members can cross-check and correct them if they might have any discrepancies in their work.

Despite these challenges we as a team are optimistic in achieving our goals, with better communication and planning we will be able to resolve our issues quickly and foster a productive environment for the team as a whole.