

M	Tu	W	Th	F	Sa	Su	M	Tu	W	Th	F	Sa	Su
1			4	5		7		2				6	
8	9	10	11	12		14		9	10	11		13	
		17	18	19		21		16				20	
22	23	24	25	26	27	28	29	23				27	

# Team 18 Sprint 1 Planning Document

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

**PROJECT NAME:** Shift

## Sprint Overview

For this sprint we aim to add the necessary user stories to make our website functional by the end of the first sprint. We would like to make our website usable and have enough functionality so that we can get user feedback and help improve our website accordingly. We aim to deploy our initial program by this sprint and also have the user authorization set-up along with the initial version of our scheduler.

**Scrum Master:** Murtuza Kagalwala

**Meeting Plan:** Monday/Tuesday/Wednesday @ 9:30 p.m.

## Risks and Challenges

One of the biggest challenges for our team to overcome will be the learning curve associated with the technologies and how we will be effectively integrating these technologies in our project. This means that we have to troubleshoot quite a bit when practicing with these new technologies. Next we will all have to set up our dev environments, sign-up for services and download the needed technologies which are important preliminary tasks. Figuring out what each of our teammates want to work on is something that we need to take into consideration too. A big risk that we are trying to take for this sprint is the deployment of the initial website by the end of this sprint on AWS, which has its uncertainty given our unfamiliarity with this service. Despite all these challenges we are committed to trying our best to achieve the sprint goals for this project.

# Current Sprint Detail

## User Story 1

As a user, I would like to be able to create my account.

#	Description	Estimated Time	Owner
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	Murtuza
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

### Acceptance Criteria

- Given that the Auth0 API is integrated for signup correctly, when the user clicks the signup button, they will be redirected to a signup page to input new account information.
- Given that signup input validation is implemented correctly, a user cannot create an account with an email already in use.
- Given that user onboarding is implemented correctly, the user can input relevant user data.

## User Story 2

As a user, I would like to login to my account.

#	Description	Estimated Time	Owner
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

### Acceptance Criteria

- Given that the Auth0 API is implemented for login correctly, when the user clicks the login button, they will be redirected to a login page to be authenticated.
- Given the input verification is implemented correctly, when a user inputs invalid credentials, they will be notified.
- Given that user authentication is implemented correctly, when the user inputs valid credentials, they will be redirected to the main application.

## User Story 3

As a user, I would like to be able to reset my password.

#	Description	Estimated Time	Owner
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

### Acceptance Criteria

- Given that the Auth0 API is implemented for resetting passwords correctly, when the user chooses to reset their password they should receive a secure link through their email to do so.
- Given that password strength criteria are implemented correctly the user should not be able to reset their password to anything outside those parameter.
- Given the password reset process was implemented correctly the user should be prompted for an identification token, and then be prompted to submit an appropriate password twice, and then be directed to the regular login page.

## User Story 4

As a user, I would like to delete my account and all of its associated data.

#	Description	Estimated Time	Owner
1	Verify the users 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

### Acceptance Criteria

- Given that account deletion is implemented correctly, when a user clicks the delete button, the user will be prompted to confirm that they are deleting their account.
- Given that account deletion is implemented correctly, when a user tries to login, they won't be able to authenticate with credentials of a deleted account.
- Given that account deletion is implemented correctly, when a user tries to create an account, they will be able to create a new account with a deleted accounts email.

## 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.

#	Description	Estimated Time	Owner
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

### Acceptance Criteria

- Given that the dashboard is implemented correctly, when the user clicks the dashboard page, they should see a relevant summary of all their data.
- Given that the calendar view is implemented correctly, when the user clicks the today button, they should see their schedule displayed on the calendar.
- Given that the calendar view is implemented correctly, when the user scrolls through the calendar, they should see schedule information from other months and semesters.

## 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).

#	Description	Estimated Time	Owner
1	Setup social login with Auth0 API.	3 hours	Daniel
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

### Acceptance Criteria

- Given that social login is implemented correctly a user should be able to login with their accounts from social providers like Apple, FaceBook, and Google.
- Given that the account details page is implemented correctly the user should be able to view the details they specified during account creation, with sensitive information being appropriately hidden from plain view.
- Given that the data editing functionality is implemented correctly the user should be able to change data fields such as their name, email, gender, and id, whilst other fields remain unchangeable such as their work position.

## 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.

#	Description	Estimated Time	Owner
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

### Acceptance Criteria

- Given that an REA/REC accesses the system they should be able to access the RA account creation page.
- Given that the REA/REC creates a new RA account, AuthO sends a temporary password to the email inputted with the RA account creation.
- Given the RA receives the email from AuthO, they should be able to create a new password, and reset if necessary.

## 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

### Acceptance Criteria

- Given that the UI is implemented correctly the user should be able to submit their availability from the calendar and have that data sent to the database to be stored and used for other parts of the program.
- Given that an availability exists, when multiple users attempt to accept the available event, then only one user is given the event while an error appears for the other, indicating that the event requested is no longer available.
- Given that an availability can no longer be selected due to time expiration or that it is claimed, when the user tries to view the event, then it should not be seen by any users and should not be available to claim.

## User Story 9

As a user, I would like to add, edit, and delete my personal events.

#	Description	Estimated Time	Owner
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

### Acceptance Criteria

- Given that all the necessary information has been filled for an event, when the add button is clicked/the event is submitted, then the event should be added to the user's account in the database and changes should be visible on the user's schedule.
- Given that all the necessary information has been edited for an event after it has been selected for change, when the edit button is clicked/the changes are saved, then the changes made to the event should be made to the user's account in the database and should be visible on the user's schedule.
- Given that a valid event has been selected, when the event is marked for deletion, then the event should be removed from the user's account in the database and the deletion should be visible on the user's schedule.

## User Story 10

As a user, I would like to clear my schedule based on hour, day, week, or month.

#	Description	Estimated Time	Owner
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

### Acceptance Criteria

- Given that the UI is implemented correctly, the user should be able to select events within a certain specified time frame and have them deleted from view.
- Given that the implementation is correct, if an RA user has a shift selected as a part of their events to be deleted they will be given a visual warning about how they must request permission to drop shifts, which are distinct from other personal events.
- Given that backend implementation is correct, when the events are deleted, then the changes will be reflected in the database as well.



## 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.

#	Description	Estimated Time	Owner
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

### Acceptance Criteria

- Given that a user accesses the calendar view, they should be able to select between multiple layouts for the calendar.
- Given that a user selects to switch views, they should be able to toggle between different viewing options.
- Given that the functionality is implemented, unit testing should be performed to debug and validate the calendar layouts and toggle buttons.

## User Story 12

As a user, I would like to be able to view future days, weeks, and months of the schedule.

#	Description	Estimated Time	Owner
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

### Acceptance Criteria

- Given all users have access to a functional calendar UI, they will be able to click a view button that displays button options for upcoming months.
- Given that the first functionality is implemented, upon interaction with the different buttons, the application will appropriately change the calendar to the calendar view selected.
- Given that the switching implementation is implemented correctly, future events will be showcased on the newly toggled calendar.

## User Story 13

As an RA, I would like to be able to indicate a preference for when to schedule certain shifts.

#	Description	Estimated Time	Owner
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

### Acceptance Criteria

- Given that the schedule preferences functionality is implemented correctly, when the user clicks the submit button, the app prompts if the submission was a success or failure.
- Given the schedule preferences functionality is implemented correctly, when the user tries to submit without filling all their preferences, the app shows that all schedule preferences haven't been selected.
- Given that the schedule preferences functionality is implemented correctly, when the user clicks an available shift on the calendar, the app shows that a schedule preferences has been selected.

## 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

### Acceptance Criteria

- Given that an RA views their assigned shift, they should be able to press a button to request an override from an REA/REC to drop/switch that shift.
- Given that an RA requests an override, notify the REA/REC.
- Given that an REA/REC receives the notification of a drop request, they should have the ability to deny or approve the request.
- Given that an RA has gotten their shift drop approved, they should select from the person who's picking up their shift's schedule so that they are trading duty and not one RA is taking more duty than another.

## User Story 15

As an RA, I would like to be able to view shifts that are available due to late shift drops.

#	Description	Estimated Time	Owner
1	Implement a UI, like a board, that displays available shifts.	3 hours	Lucas
2	Find and show RA's 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

### Acceptance Criteria

- Given that the UI is implemented correctly, when the user clicks the dropped shifts tab, they will be shown all the newly available shifts.
- Given that the shift drop viewing functionality is implemented correctly, when the user is on the dropped shifts page, the total newly available shifts are updated in real time.
- Given that the UI is implemented correctly, when the user clicks a newly available shift, they should see more information on that shift.

## User Story 16

As an REA or REC, I would like to set a minimum/required number of shifts each RA has to select.

#	Description	Estimated Time	Owner
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

### Acceptance Criteria

- Given the REA/REC have a functional admin page, they can establish the minimum hours to be applied to all of their RAs.
- Given that the process of indicating shift availability is implemented correctly, the RA's calendar will display an indicator of them not meeting the threshold if their total shifts don't meet the standard.
- Given that the submission of an RA's schedule is handled properly, REAs will receive a notification through email when an RA submits their personal schedule that is under the standard.
- Given the REAs and RECs have a dedicated page for admin features, they will be able to look through a dropdown of users with less than the required shifts.
- Given all RAs have a functional calendar page, they will have a reminder text field in the top of their calendar informing them if they need more shifts to reach the goal set by REA/REC.

## 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.

#	Description	Estimated Time	Owner
1	UI for view of schedule requests and drops	4 hours	Danielle
2	Implement functionality of approval or denial	3 hours	Danielle
3	Unit testing	2 hours	Danielle

### Acceptance Criteria

- Given that an REA/REC accesses the system to manage RA schedules, they should be able read requests from the RA's for schedule changes or drops.
- Given that the REA/REC can read the request from the RA, they can choose to either approve the request or decline it.
- Given the approval or the denial, the RA is notified of the response from the REA/REC.

## User Story 18

As an REA or REC, I would like to edit RA schedules (for last minute shift drops, etc.).

#	Description	Estimated Time	Owner
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

### Acceptance Criteria

- Given that an REA/REC accesses the system to manage RA schedules, they should be able insert/delete/assign shifts as desired.
- Given a manual change is made to the RA schedule, the system should update the counts of how many shifts each RA has and store those values.
- Given a change is made, users are notified of the schedule change.

## **Non-functional work**

We are planning on Deploying our work on AWS. We will be using the Amplify service to host our front end, and we will be using Beanstalk to deploy our backend. We will also be setting up the database in AWS, for which we are using the AWS RDS service. We estimate that to familiarize ourselves with AWS we will need 3 hours, implementing all the 3 services should take about 10 hours and unit testing will be another 2 hours. In total we will be requiring 15 hours on the minimum to set-up and connect the AWS services with each other.

Estimated time: 15 hours

Owners: Murtuza (12 hours) + Samson (3 hours)

### **Acceptance Criteria**

- Given that our front end is set-up, we will be hosting it on AWS Amplify
- Given that our backend is set-up, we will be deploying it on our backend
- Given that we need database access, we will be connecting our backend service with the database.

# Remaining Backlog

Note: Current user stories are highlighted

## Functional Requirements

### Account Management

As a user,

1. I would like to be able to create my account
2. I would like to login to my account.
3. I would like to be able to reset my password.
4. I would like to delete my account and all of its associated data.
5. I want a dashboard where I can view relevant information, such as my display name, position, and work schedule.
6. 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).
7. I would like to be able to specify my hall, such that I can search for other users in my hall.

As an REA or REC,

8. I would like to be able to create accounts for RAs as needed and supply them with a temporary password to access it.
9. I would like to view at a glance the current status of RAs, e.g., active, inactive, unavailable.

As an REA,

10. I would like to be able to temporarily disable or delete accounts of the RAs.

As an REC,

11. I would like to be able to temporarily disable or delete accounts of the RAs and REAs.

### Customization

As a user,

12. I would like to be able to select from a variety of themes, allowing me to personalize my visual experience.
13. 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.



14. I would like to be able to reorient the layout of the widgets (e.g., calendar, tabs, taskbar, etc.) on my account.
15. I would like to be able to create a quick view of any key messages, shifts, or events I would like to see after I login.

As an REA or REC,

16. I would like to have the option of creating a welcome page that outlines the rules of scheduling and a brief overview of the website, when I add an RA who is just starting their new job with us.

## **Scheduling**

As a user,

- 17. I would like to be able to submit my availability to the scheduler.**
- 18. I would like to add, edit, and delete my personal events.**
- 19. I would like to clear my schedule based on hour, day, week, or month.**
20. I would like to export my schedule for use on other calendar applications/websites such as Google Calendar.
21. I would like to be able to synchronize my schedule with an external calendar service such as Google Calendar or Outlook Calendar.
22. I would like to be able to download the schedule as a PDF.
23. I would like to be able to schedule, change, and cancel meetings with other users.
24. I would like to receive email notifications to be notified of schedule changes or upcoming events.
25. I would like to see whether or not an RA has clocked in for their shift.
26. I would like to search for important events in my hall or other halls.
- 27. 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.**
- 28. I would like to be able to view future days, weeks, and months of the schedule.**

As an RA,

- 29. I would like to be able to indicate a preference for when to schedule certain shifts.**
30. I would like to be able to select whether or not I can view and select conflicting shifts (i.e., I can access shifts which may overlap with my other events because I am unable to attend any other available ones).

**31. I would like to be able to request to shift drops assigned to me in the face of any personal emergencies.**

**32. I would like to be able to view shifts that are available due to late shift drops.**

33. I'd like to be able to see which RA's are available on any given day (for duty switches, etc.).

As an REA or REC,

**34. I would like to set a minimum/required number of shifts each RA has to select.**

**35. 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.**

36. I would like to create, edit, and delete 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.

37. 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.

38. I would like to be able to manually assign time slots to RAs.

39. I would like to be able to add special rules to shifts (e.g., certain halls, such as coeds, must have RAs of the opposite gender while performing certain duties).

40. I would like to view the number as well as the types of shifts completed by an RA.

**41. I would like to edit RA schedules (for last minute shift drops, etc.).**

42. I would like the option to generate a daily, weekly, or monthly report detailing the shifts completed by RAs (e.g., hours worked, duties completed, absences, etc.).

## **Messaging**

As a user,

43. I would like to chat with other users privately.

44. I would like to have access to a public chat of all users in the same residence hall.

45. I would like to be able to send hyperlinks or documents.

46. I would like my messages to be encrypted.

47. I would like to be able to see the most recent chat field at the top of the list.

48. I would like to be able to report inappropriate messages in the chat.

49. I would like the ability to delete individual messages from a conversation and to delete the message group itself.

50. I would like the ability to block other users from messaging in private or group conversation.

51. I would like to pin messages that I would like to save for future reference.

52. I would like to schedule messages to be sent in the future.

As an REA or REC,

53. I would like to set rules for spamming in group chats.

54. I would like to have an emergency response chat that will be visible on any page of the website when the emergency is active.

## **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 AWS which will simplify our deployment and will mainly be hosting our front-end through the AWS service called amplify and we will be hosting the backend on the AWS service called Beanstalk. The database will be hosted using an AWS service called RDS. (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.