Officium

Sprint 1 Planning Document

Team 2: Noah Alderton, Joseph Barr, Nicholas Raphael, Melissa Ruiz Ruiz, Emma Wrege, Derek Nobbe

Sprint Overview

In this sprint, we hope to initialize the frontend and backend portions of our application. Before we can accomplish anything else, we must first lay down the basic framework, namely Firebase and the corresponding Node.js framework for the backend and the React environment for the frontend. Beyond that, we will implement basic user operations, such as account creation and inputting data. By the end of this sprint, we hope to have at least a minimally functional application.

Scrum Master: Noah Alderton

Meeting Plan:

- Wednesday at 7:30pm with Project Coordinator
- Wednesday at 8:30pm
- Saturday at 11:00am

Risks and Challenges

As this is our first sprint and we do not necessarily have high levels of experience with the frameworks we will be utilizing, our perceptions as to how difficult and time consuming much of this work will be may be quite inaccurate. Some tasks may take much longer than expected, while others might be shorter. This problem should be mitigated, however, by the malleable nature of a project in its infancy. Moving forward into future sprints, out time and difficulty estimation ability should improve.

Current Sprint Detail

User Story #1

As a user, I would like to access Officum from any web browser.

#	Description	Estimated Time	Owner
1	Integrate Firebase JavaScript APK with React	3 hr	Noah
2	Create global styling	4 hr (each)	Nick/Emma
3	Create landing page	3 hr	Emma
4	Navbar/Routing	2 hr	Nick
5	Test to ensure that the landing page works properly with a navbar, and the site does not 404 at any point.	1 hr	Noah

Acceptance Criteria:

- Given the frontend is being hosted in a test environment, when a user navigates to the respective web address, then a landing page for Officium should appear.
- Given the Officium React web app successfully loads, when a user navigates to the appropriate web address, then a navigation bar should appear at the top of the page.
- Given the Officium React web app successfully loads, when a user navigates to the appropriate web address, Officium should have an aesthetically pleasing and consistent user interface design.

User Story #2

As a user, I would like to be able to register for Officium with my email.

#	Description	Estimated Time	Owner
1	Create UI for inputting and confirming email and name	3 hr	Derek
2	Store email address in relation to a unique user	5 hr	Joseph/Melissa
3	Verify email address via a confirmation email	2 hr	Noah
4	Insert and store hashed passwords in the database	4 hr	Joseph/Melissa
5	Verify the security of the password on the front end	5 hr	Nick
6	Display any Firebase issues with account creation in the UI	2 hr	Noah
7	Test to ensure that accounts can be properly created, the database is updated, and any issues are visible to the user.	1 hr	Noah

Acceptance Criteria:

- Given email address storage is implemented correctly, when the user logs in after creating an account, they should see their own account information.
- Given password storage is implemented correctly, when the user attempts to log into their account, only the correct password will allow them to do so.
- Given a user wants to register an account, there are areas to enter account information and submit this data, and then be redirected to a Jobs page.
- Given a user is selecting a password for their account, they will be presented with a password security meter helping the user select a secure password.
- Given that the login user interface is completed and correct, when a user navigates to the Officium login page, they should be presented with dialog boxes to enter their email and password.

User Story #3

As a user, I would like to recover my Officium account.

#	Description	Estimated Time	Owner
1	Create password recovery option on login page	1 hr	Emma
2	Create dialog box for users to insert email address	2 hr	Emma
3	Send password recovery email	2 hr	Joseph
4	Create page to reset password	2 hr	Derek
5	Update password on Firebase	4 hr	Melissa
6	Test to ensure that a user that is recovering an account receives an email with a recovery link and is able to set a new password.	1 hr	Noah

Acceptance Criteria:

- Given that password recovery emails are implemented properly, when the user attempts to recover their account, they will promptly receive an email with a link to do so.
- Given that the password recovery email was successfully sent, when the user clicks on the recovery link, they should be redirected to an Officium web page where they can set their new password.
- Given that password updating is implemented correctly, when the user wants to change their password, they will have a new password for their account.

User Story #4

As a user, I would like to be able to log in with with external accounts.

#	Description	Estimated Time	Owner
1	Add “Login with Google” button the login UI	1 hr	Joseph
2	Add Google account login functionality to the backend	2 hr	Joseph
3	Add “Login with Facebook” button the login UI	1 hr	Joseph
4	Add Facebook account login functionality to the backend	2 hr	Joseph
5	Add “Login with Twitter” button the login UI	1 hr	Joseph
6	Add Twitter account login functionality to the backend	2 hr	Joseph
7	Add “Login with GitHub” button the login UI	1 hr	Joseph
8	Add GitHub account login functionality to the backend	2 hr	Joseph
9	Test to ensure that users can log into Officum using Google, Facebook, Twitter, and GitHub.	1 hr	Noah

Acceptance Criteria:

- Given that the login user interface is complete, when the user clicks on the “Login with Google” button, then the user should be redirected to login with their Google account.
- Given that logging in with Google is implemented properly, when the user logs in with Google, their account will be populated with the requisite information and they will not have to log in when they visit next.
- Given that the login user interface is complete, when the user clicks on the “Login with Facebook” button, then the user should be redirected to login with their Facebook account.
- Given that logging in with Facebook is implemented properly, when the user logs in with Facebook account, their account will be populated with the requisite information and they will not have to log in when they visit next.
- Given that the login user interface is complete, when the user clicks on the “Login with Twitter” button, then the user should be redirected to login with their Twitter account.
- Given that logging in with Twitter is implemented properly, when the user logs in with Twitter account, their account will be populated with the requisite information and they will not have to log in when they visit next.
- Given that the login user interface is complete, when the user clicks on the “Login with Github” button, then the user should be redirected to login with their Github account.
- Given that logging in with Github is implemented properly, when the user logs in with Github account, their account will be populated with the requisite information and they will not have to log in when they visit next.

User Story #5

As a user, I would like to be able to insert a job application.

#	Description	Estimated Time	Owner
1	Assign unique identifier and database space for applications	4 hr	Melissa
2	Implement Location and Contact classes	5 hr	Joseph
3	Create a user interface for job application wizard	15 hr	Nick/Derek/Emma
4	Store company name, job title, location, recruiter information, how applied, application status, if interviewed, references, and whether received offer	8 hr	Noah
5	Store optimism and enthusiasm	2 hr	Melissa
6	Implement document storage for resumes and cover letters	5 hr	Melissa
7	Store application status, portal link, business card picture, distance, and cost of living	5 hr	Melissa
8	Test to ensure that a new application can be created and each of the fields can store data.	1 hr	Noah

Acceptance Criteria:

- Given that the assignment of a unique identifier for database space is properly implemented, when a user wants to import company details, the data will be stored in the database and will be retrievable.
- Given that storing application status, portal link, business card picture, distance, and cost of living is implemented correctly, when the user wants to edit these options, the database will return the preexisting data, and save any changes on the database.
- Given that the document storage has been successfully implemented on Firebase, when a user uploads a resume or cover letter, then Firebase should store it.
- Given that the document storage has been successfully implemented on Firebase, when a user attempted to retrieve a resume or cover letter, then Firebase should return it.
- Given that the new application button has been implemented on the applications list page, when the user clicks on the new application button, and taken to a new job wizard to insert the information.
- Given that the redirects between the new application page and applications list page are working properly, when the user clicks the “Save” button they should be redirected to the applications list page with the job application displaying in the list and the database information should be updated.

User Story #6

As a user, I would like to be able to view details about my job application.

#	Description	Estimated Time	Owner
1	Retrieve data from Firebase	3 hrs	Derek
2	Show company name and other fields from data	5 hrs	Emma
3	Test that selecting an application will show the appropriate details in the user interface	1 hr	Noah

Acceptance Criteria:

- Given that the assignment of a unique identifier for database space is properly implemented, when a user wants to import company details, the data will be stored in the database and will be retrievable.
- Given that the frontend is able to retrieve data from the backend, when the user attempts to view details about a specific job application, then they should be able to get to a detail page from the list of their active applications that displays all this data.
- Given that the frontend is able to retrieve data from the backend, when the user is on the application list page, then the user will see an aesthetically pleasing and easy to read page that lists all of their saved companies and information about them

User Story #7

As a user, I would like to be able to modify and add additional company information.

#	Description	Estimated Time	Owner
1	Create button for modifying already saved company info on job detail page	1 hr	Derek
2	Update Job Wizard to handle editing of existing applications	5 hr	Nick
3	Routing back to detail page on save/cancel	1 hr	Nick
4	Update the Firebase database with modified information	5 hr	Nick
5	Display a message saying the changes were successfully made	3 hr	Nick
6	Test to ensure that each of the fields in an application are saved upon modification.	1 hr	Noah

Acceptance Criteria:

- Given the user already has some information saved for a given company, when they click on the button to modify the company information, a wizard should pop up with the previously saved information in boxes that they will be able to edit and then save the updated info.
- Given that Firebase SDK has been implemented into React, when the user has hit the “Save Changes” button in the wizard, Firebase will be updated with the newly changed information.
- Given the user made changes, when they hit “Save changes” in the wizard, a message will display that the changes have successfully been stored, and the user will be redirected to the job detail page.

User Story #8

As a user, I would like to be able to logout.

#	Description	Estimated Time	Owner
1	UI logout button on all pages with reroute to login page	2 hr	Derek
2	Implement sign out functionality with Firebase Authentication	3 hr	Melissa
3	Test to ensure that upon signing out, the user's authentication isn't maintained and they have to re-enter their credentials to log in again	1 hr	Noah

Acceptance Criteria:

- Given that the implementation of logging them out is correct, when the user wants to log out of their account, the user will no longer have access to Officium's features until they login again.
- Given that the logout button is properly implemented, when the user clicks the logout button, they will be redirected to the login page.
- Given that the implementation of a logout button is correct, when the user is on any Officium page, the logout button should appear in the navigation bar.

User Story #9

As a user, I would like to be able to see all the applications I've submitted.

#	Description	Estimated Time	Owner
1	Show applications for all applied companies	2 hr	Derek
2	Sort companies alphabetically	2 hr	Derek
3	Sort companies by my level of optimism	2 hr	Emma
4	Sort companies by my level of enthusiasm	2 hr	Emma
5	Sort by due date	2 hr	Derek
6	Manually organize by preference	3 hr	Melissa
7	Store preference order in database	3 hr	Joseph
8	Test to ensure that all applications can be viewed and sorted via the appropriate sorting mechanism	1 hr	Noah

Acceptance Criteria:

- Given that manually organizing by preference is implemented properly, when the user wants to sort companies in their own format, the application will make it simple to create a custom list.
- Given that preference order is stored properly in the database, when the user is greeted with the application list page, the applications will be sorted in preference order, unless they choose to modify it.
- Given that UI implementation is correct, when the user accesses the application list page, then the user sees a list of their current job applications and sorting options.

User Story #10

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

#	Description	Estimated Time	Owner
1	Create account information page	6 hr	Derek
2	Have change password option on account page	4 hr	Emma
3	Call API reset password in UI	2 hr	Noah
4	Test to ensure that the user's password can be successfully changed and the old password no longer works.	1 hr	Noah

Acceptance Criteria:

- Given that the user has an account, when they click on their account, the user will see a profile page that has all of their personal information listed and a button that will allow them to change their password.
- Given the account page is implemented, when the user is accessing their account page, the site will display an aesthetically pleasing page that lists the users information and companies they have save applications for.
- Given that the password hash on the database is updated, when the user utilizes the option to change their password, their old password will no longer function, and the new password will be the only functional password when logging in in the future.

User Story #11

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

#	Description	Estimated Time	Owner
1	Delete user from database	4 hr	Joseph
2	Create UI for account deletion	2 hr	Emma
3	Account deletion email confirmation	2 hr	Noah
4	Test to ensure that a user can delete their Officium account and their corresponding data is purged from the database	1 hr	Noah

Acceptance Criteria:

- Given that user deletion is implemented properly, when the user deletes their account, there will be no remaining record of the user in the system and a confirmation.
- Given that the interaction between the Firebase database and Google Cloud Functions are properly implemented, when the user selects to delete their account, an email will be sent to the previously stored email address confirming the deletion of their account.
- Given that the account page is implemented, when the user navigates to their account page, then there will be an option on their profile page to delete their account, and when selected, will bring up a confirmation dialog.

Remaining Backlog

Functional Requirements

1. Users can register for Officium with any current account or make a new one
   - ~~As a user, I would like to be able to register for Officium with my email, Google, Facebook, Twitter, or GitHub~~
   - ~~As a user, I would like to be able to logout.~~
   - ~~As a user, I would like to be able to change my password.~~
   - ~~As a user, I would like to receive a confirmation email when an account was created.~~
   - ~~As a user, I would like to be able to recover my account after forgetting my password.~~

2. Users can enter in the specifics of the internships/jobs they applied for
   - ~~As a user, I would like to be able to store the company name.~~
   - ~~As a user, I would like to be able to store the job title of the application.~~
   - ~~As a user, I would like to be able to store the location of the job/internship.~~
   - ~~As a user, I would like to be able to store recruiter contact information.~~
   - ~~As a user, I would like to be able to store how I applied to the job/internship.~~
   - ~~As a user, I would like to be able to store the status of my application.~~
   - ~~As a user, I would like to be able to store if I interviewed for the position.~~
   - ~~As a user, I would like to be able to store any references used for a position.~~
   - ~~As a user, I would like to be able to store if I received an offer.~~
   - ~~As a user, I would like to be able to store the status of my applications.~~
   - ~~As a user, I would like to be able to store the link to the employers' portal.~~
   - ~~As a user, I would like to be able to delete a particular application.~~
   - ~~As a user, I would like to be able to store my credentials for the application portal.~~
   - ~~As a user, I would like to be able to modify company information.~~

3. Users can sort their applications in several ways
   - ~~As a user, I would like to be able to sort companies alphabetically.~~
   - ~~As a user, I would like to be able to sort companies by my level of optimism.~~
   - ~~As a user, I would like to be able to sort companies by my level of enthusiasm.~~
   - ~~As a user, I would like to be able to sort by due date.~~
   - ~~As a user, I would like to be able to sort by geographic proximity.~~
   - ~~As a user, I would like to be able to sort by cost of living.~~
   - ~~As a user, I would like to be able to rank companies by preference.~~
   - As a user, I would like to be able to sort by estimated pay rate from the Glassdoor API.

4. Users can see and store personal specifications about their applications
    - ~~As a user, I would like to be able to rate my optimism for a particular application.~~
    - ~~As a user, I would like to be able to rate my enthusiasm for a particular application.~~
    - ~~As a user, I would like to be able to see the physical distance to a particular location.~~
    - As a user, I would like to be able to see the cost of living for a particular location.
    - ~~As a user, I would like to be able to weigh my priorities for how to select a job.~~
    - As a user, I would like to be able to insert notes for particular companies.
    - As a user, I would like to be able to modify notes.
    - As a user, I would like to be able to view all my contacts.

5. Users can store documents in their Officium account
    - As a user, I would like to be able to upload my resume.
    - As a user, I would like to be able to update my resume on file.
    - As a user, I would like to be able to remove my resume on file.
    - As a user, I would like to be able to upload my cover letter.
    - As a user, I would like to be able to update my cover letter on file.
    - As a user, I would like to be able to remove my cover letter on file.
    - ~~As a user, I would like to be able to store pictures of business cards.~~

6. Users can report any issues with the web application
    - As a user, I would like to be guided through the user interface.
    - As a user, I would like to be able to get help for the user interface.
    - As a user, I would like to be able to report bugs with my experience.

7. Users will be able to utilize various mediums to see different information about their applications
    - As a user, I would like to be able to generate a Sankey diagram for my applications.
    - As a user, I would like to be able to see Glassdoor reviews of a particular employer.
    - As a user, I would like to be able to see estimated pay rates for a particular employer from Glassdoor.
    - As a user, I would like to be able to see a map of the company locations using Google Maps.
    - As a user, I would like to be able to export due dates into a CSV file.
    - As a user, I would like to be able to export due dates into Google Calendar.
    - As a user, I would like to be able to export contacts as a vCard.

8. Users will receive email updates about their applications
    - As a user, I would like to receive timely emails when an application deadline is approaching.
    - As a user, I would like to receive emails when I have not heard from a company in a long time.

Non-functional Requirements

1. Performance
    - As a developer, I would like Officium to be responsive so I can use it on various devices.
    - As a developer, I would like the site to run smoothly with no bugs.
    - As a developer, I would like the server to be able to handle at least 100 clients at a time.
    - As a developer, I want the site to be able to handle errors gracefully and not crash

2. Server
    - As a developer, I want the server to support the client and carry information to and from the interface in a timely manner.
    - As a developer, I want the server to be able to handle multiple client requests with ease.
    - As a developer, I would like the server to store information in an organized and easy-to-access way.
    - As a developer, I would like the server to handle requests to an API in an efficient manner.

3. Appearance
    - As a developer, I would like the appearance of the site to reflect a business aesthetic and be pleasing to the eye.

4. Security
    - As a developer, I want any data collected by the user to be secure.
    - As a developer, I would like the store user credentials safely in the server.

5. Ease of Use
    - As a developer, I want the site the be user friendly and self-explanatory.
    - As a developer, I want the site to be able to be run on a web-browser.