COSC 4P02 - Final Report

Apr 28, 2024

Instructor: Naser Ezzati-Jivan (nezzati@brocku.ca)
TA: Brendan Park (bp18ul@brocku.ca)
HabitConnect's GitHub: https://github.com/kphillippo/habit-tracker

The following report will provide a comprehensive overview of the development of HabitConnect from January to April 2024.

Updated Requirements Document

https://docs.google.com/document/d/1Q2D5-TmcToMA5m-zfrtyOv8wrwMg2XZ6XxQs1W1_U9s/edit?usp=sharing

This document provides a comprehensive overview of what the website is and what it needs to do. It provides an overview of the site and its objectives, as well as a detailed list of user stories, outlining different required functionalities of the site.

Installation Guide

https://docs.google.com/document/d/1pIUTatyo50p0aSejOFXUSQuKknQcqNFVl3Ovi5vCy50/edit?usp=sharing

HabitConnect is not currently publicly hosted. This installation guide was created to lead the user through the collection of the source code from GitHub, installation of all required dependencies, and our recommended method of running the website locally from the user's local system.

User Manual

https://docs.google.com/document/d/11T6s2-Pi2661DUy8lCETxUTxCmdHXfjJZsUflNopark/edit?usp=sharing

This user manual has been created to give a comprehensive overview of the HabitConnect website, for both first time users, and experienced users who have questions about functionalities of the site. The document provides a brief introduction to HabitConnect, including the project objectives, along with a description of how to get started on the site. It continues, providing an outline of the functionalities of pages available to the user, information on account management, a troubleshooting FAQ, and finally information on how to contact the developers regarding updates and feedback for the website.

Introduction to HabitConnect Youtube Video

https://www.youtube.com/watch?v=jobu9mN4Yjo

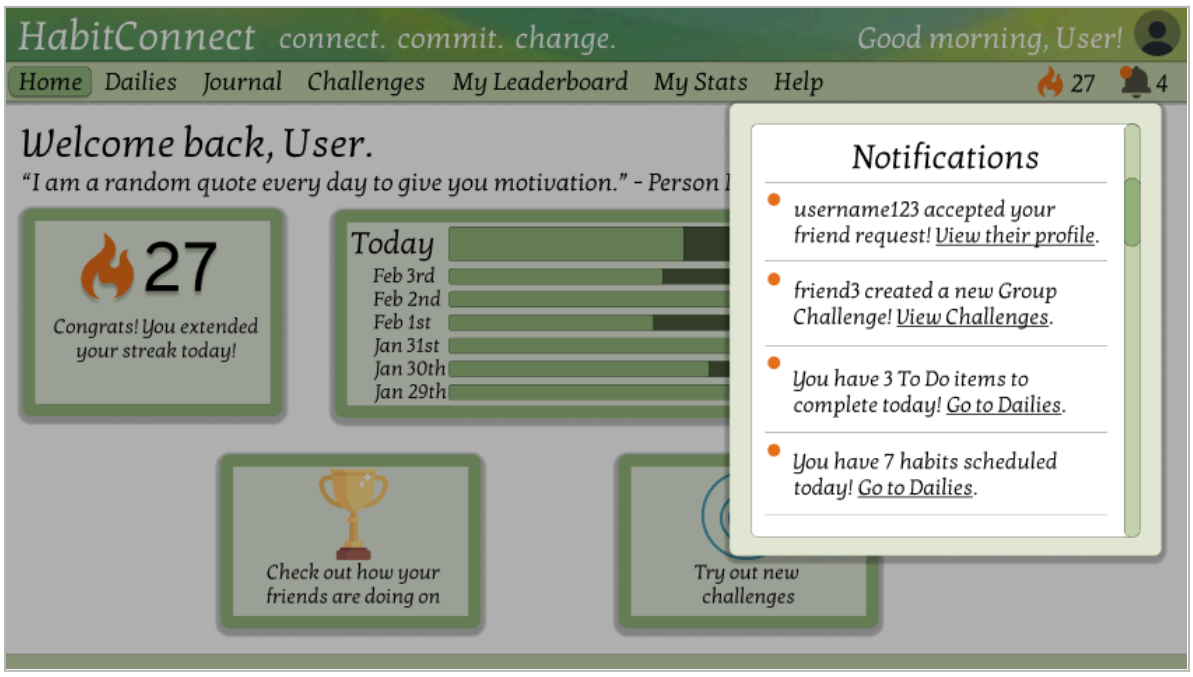
Development Process

Front-End

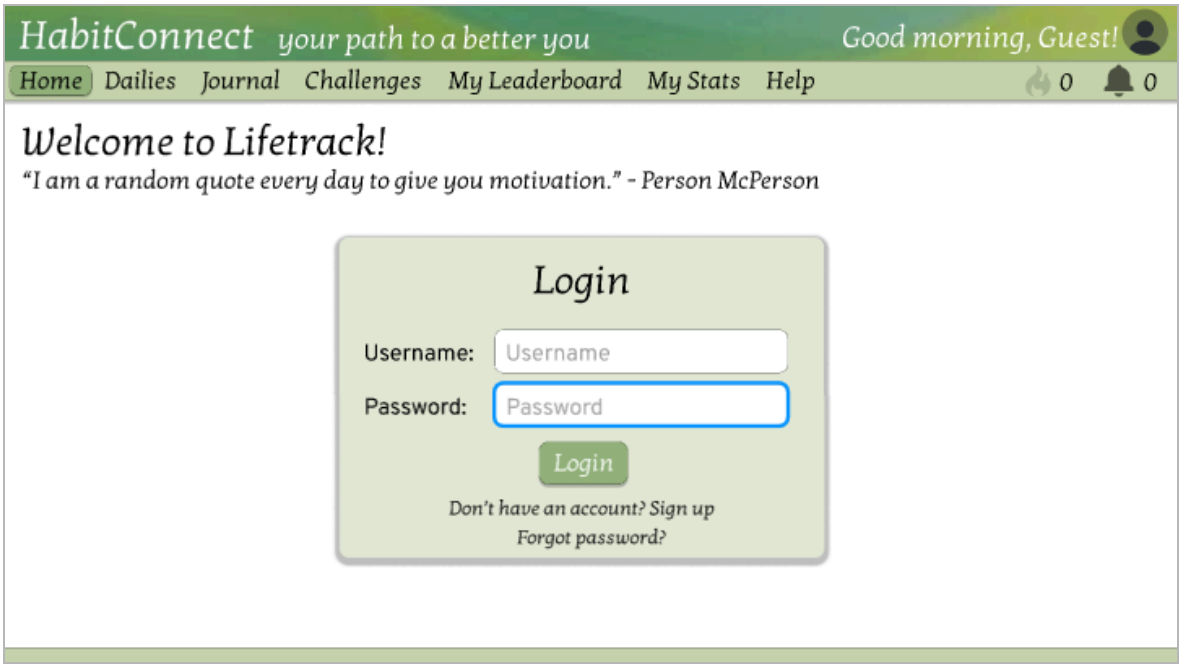
The main focus of the front-end development team was to create a solid foundation for the design of the website in the beginning stages of this project. We knew that implementation would be made easier when there were strong designs in place to work from. This is why we made extensive use of Figma and its design features, to help us plan the implementation of all desired features on the site, and to guarantee consistent styling across different pages. Our original designs were modified in many aspects once the plans were finalized, but we found the designs to be a crucial starting point to ease implementation. Below are the finalized designs created for our website pages, with all designs created by Kacie and Elya:

Guest user homepage design.

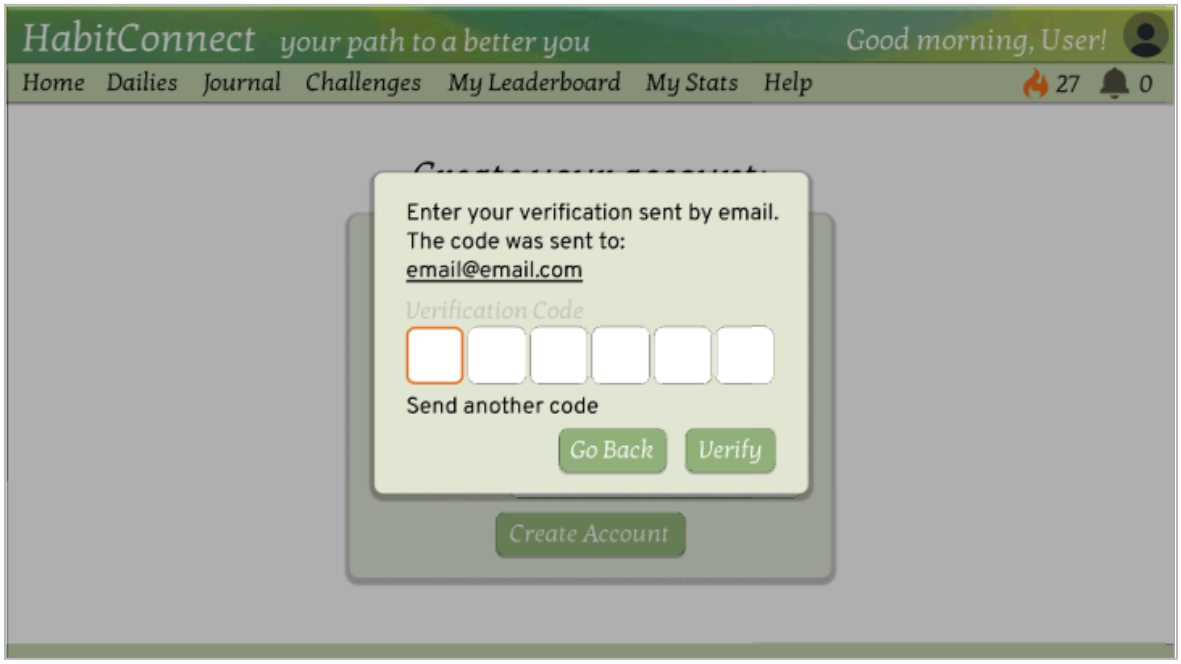
User homepage design.



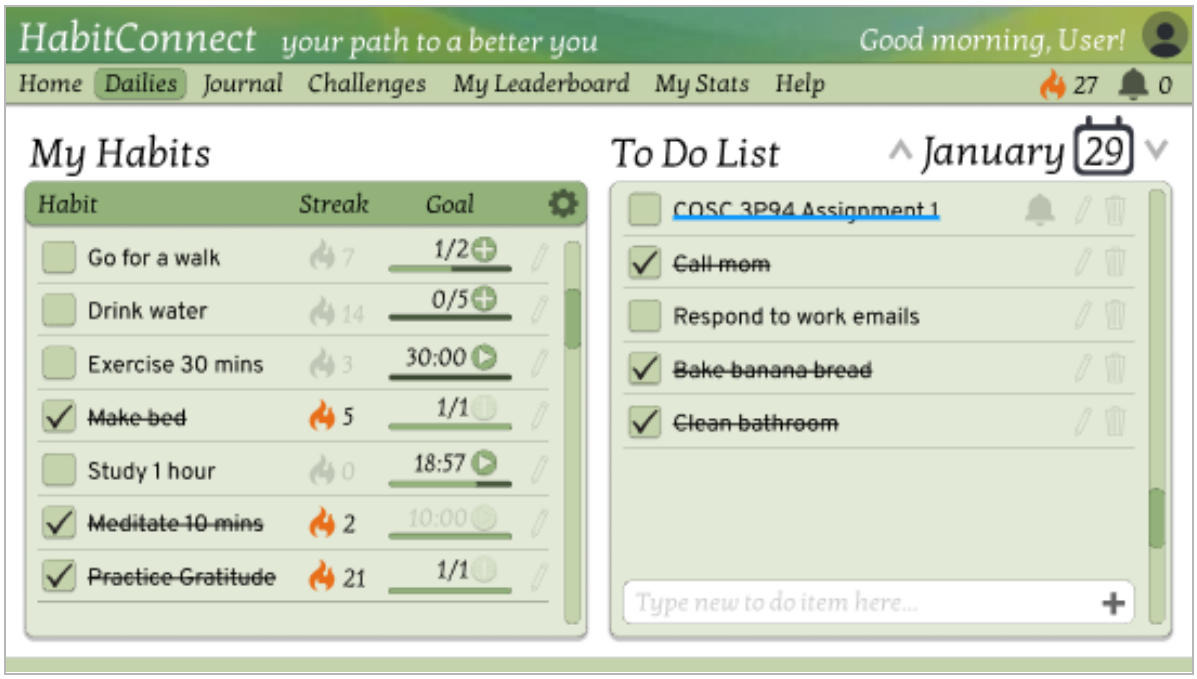
Notifications pop-up design.



Login page design.

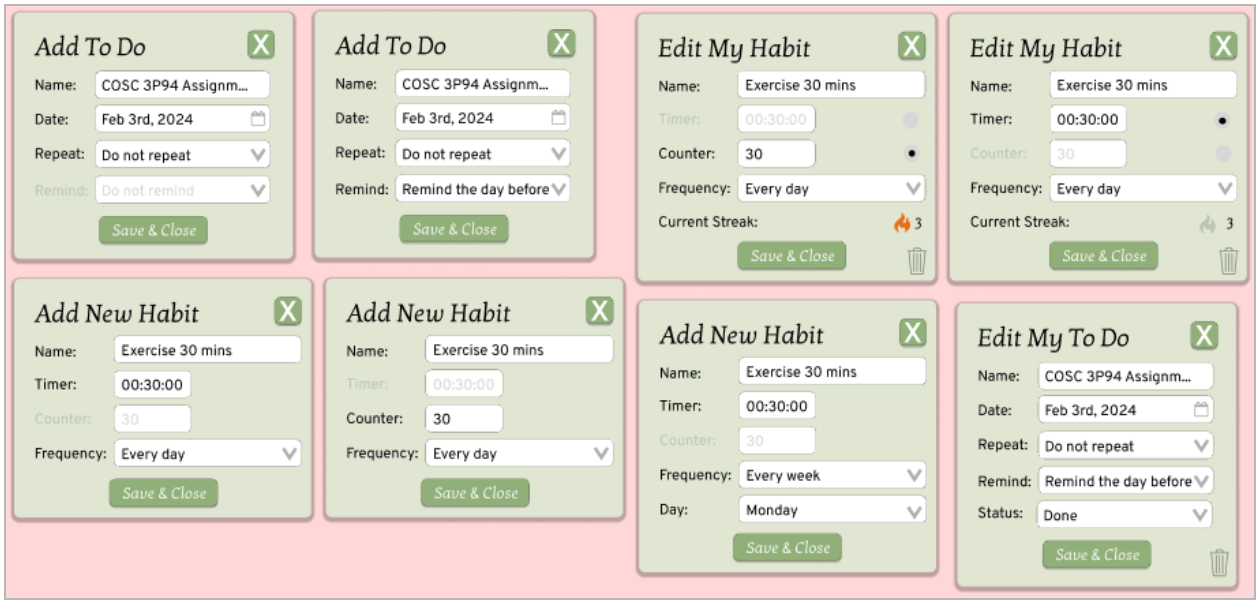


Verification email design.



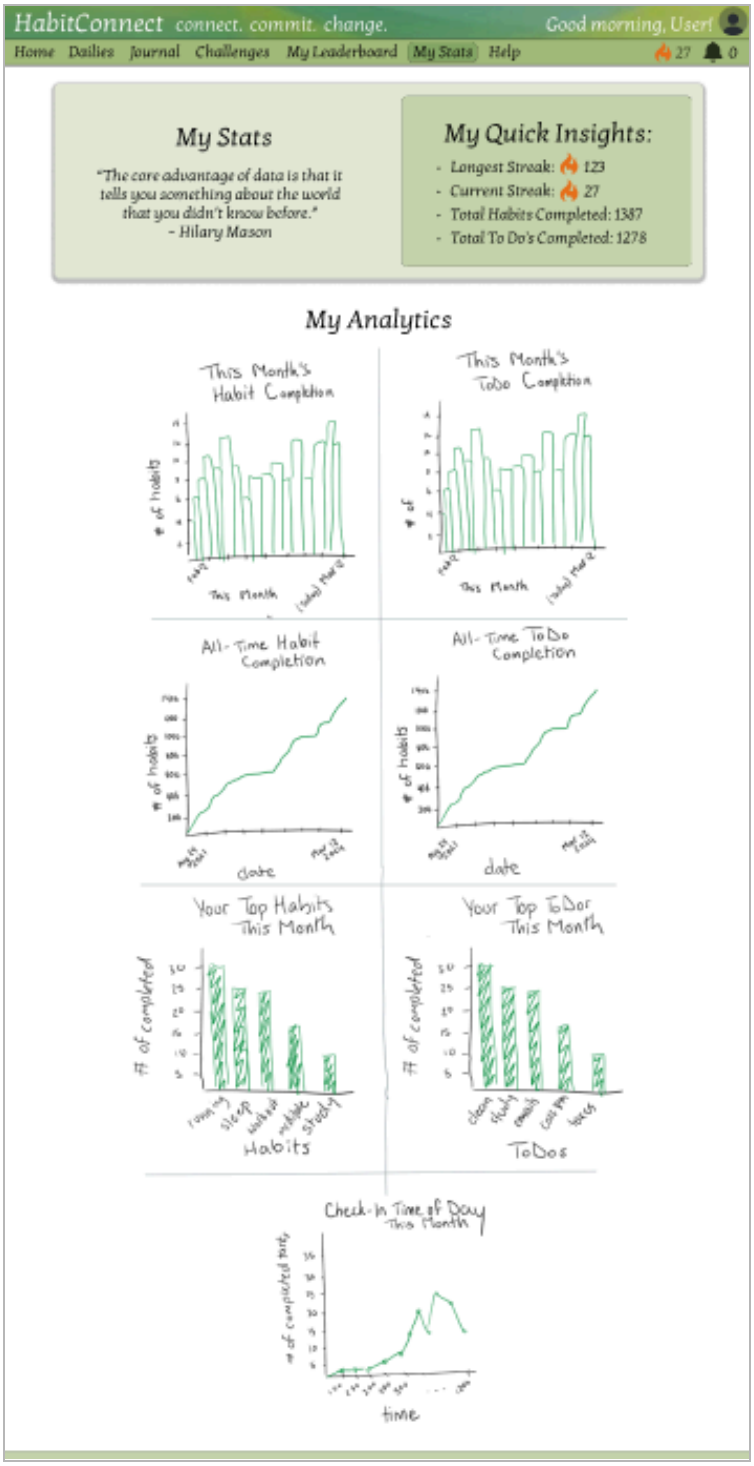
Dailies page design.

My Habit Manager pop-up design.

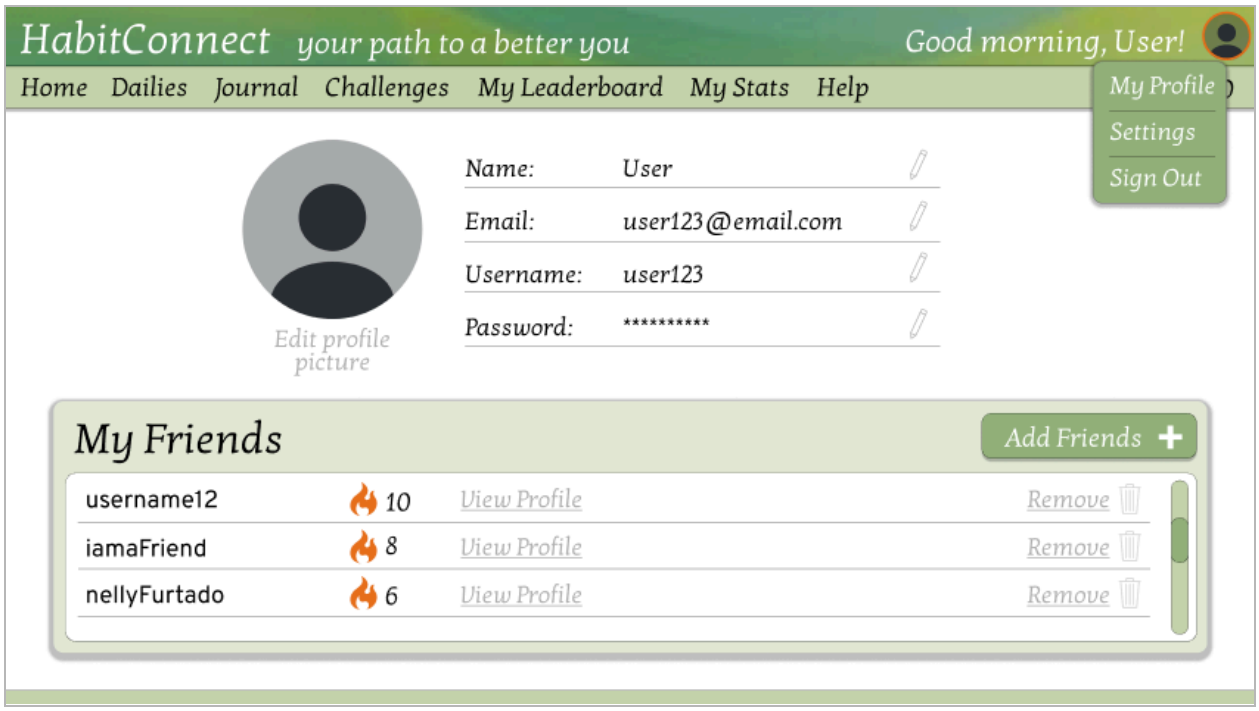


Dailies page pop-up designs.

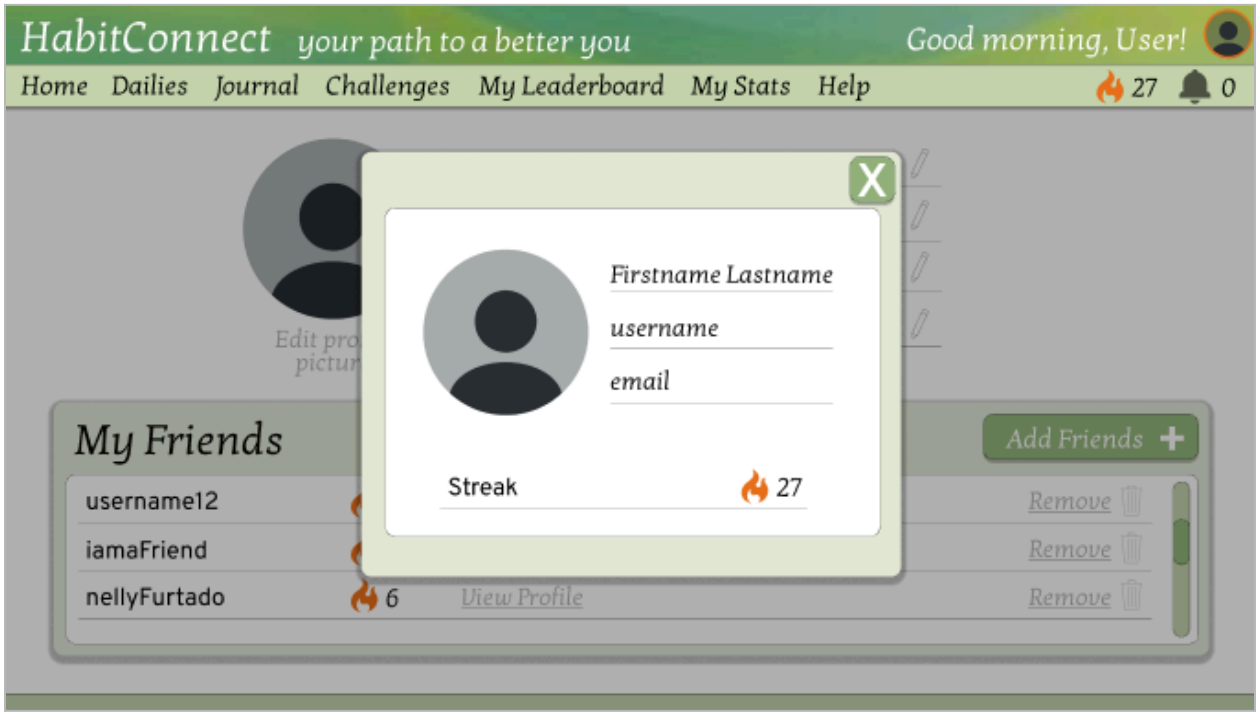
My Leaderboard page design.



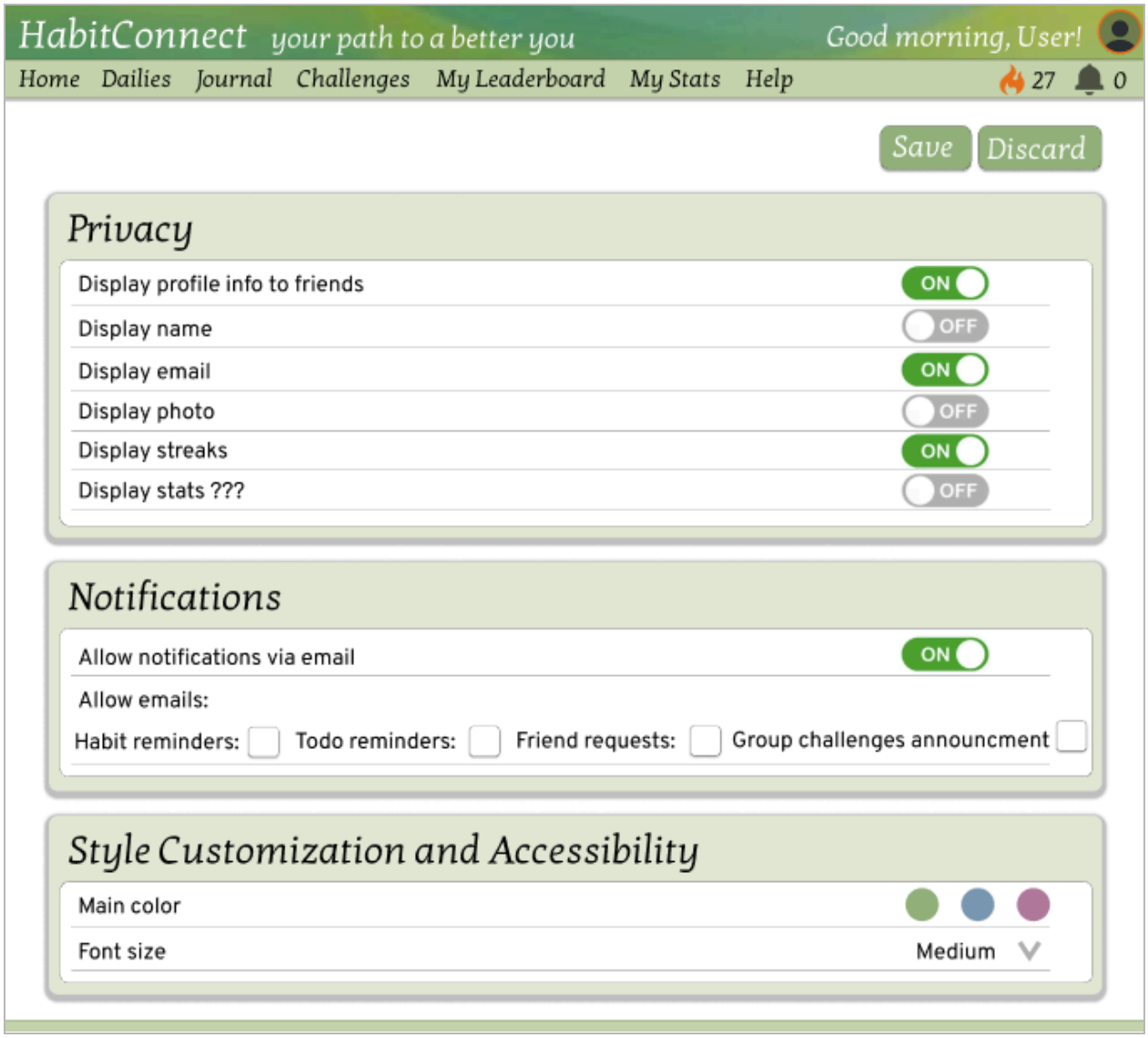
My Stats page design.



User profile page design.



Friend profile pop-up design.



Settings page design.

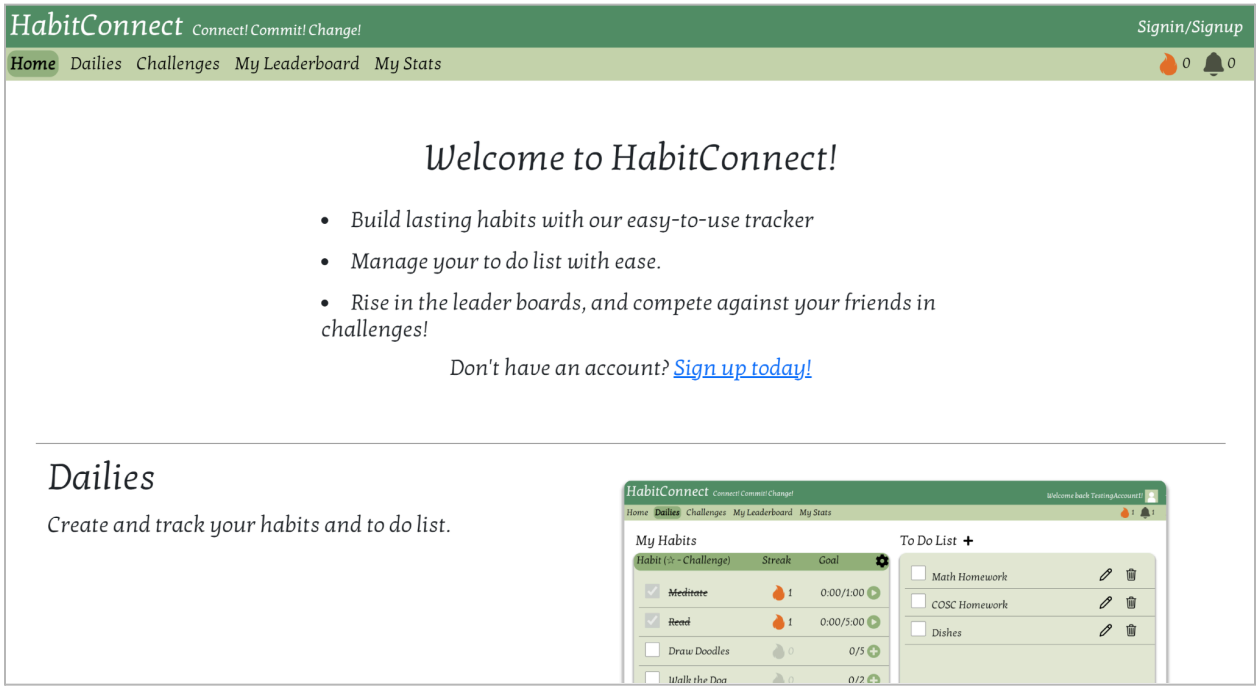
With these detailed designs, implementation of all pages in the front-end became much easier. The front-end team worked tirelessly, implementing the designs to the best of our ability and modifying the plan as the situation required.

It is important to note some known technical issues with the front-end of our website.

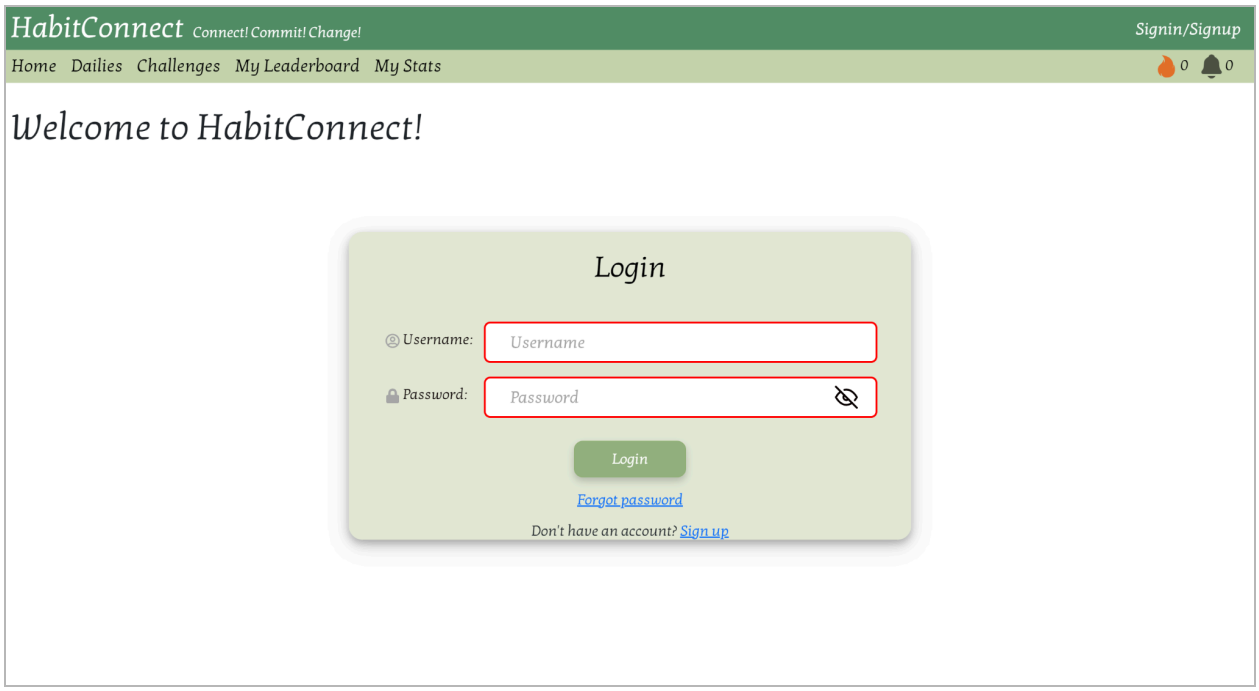
- The Dailies page has some bugs regarding tracking the user's daily streak and sending the correct values to be stored in the back-end.
- The Dailies page to do list does not properly send check-in information to the back end, so completion is not stored properly.
- Our original plan for the Dailies page would have been to have a date picker in the top right of the screen, so to-do's and habits could be added by the day, not just to every single day. This unfortunately did not get implemented in time.

- Some of the pages have warnings which appear in the console which have not been resolved.
- Some of the pages are not compatible with zooming (using the zoom feature within the browser does not change the size of the font or other elements on the screen).
- The timer field within the Habit Manager only allows for times to be input in seconds. Our original idea was to give the user the ability to enter the time in either seconds, minutes or hours.
- Some of the pages are not responsive to different screen widths/heights, and would be used with difficulty on devices like tablets or phones.
- Our Testing Master Spreadsheet holds other bugs found in the front-end.

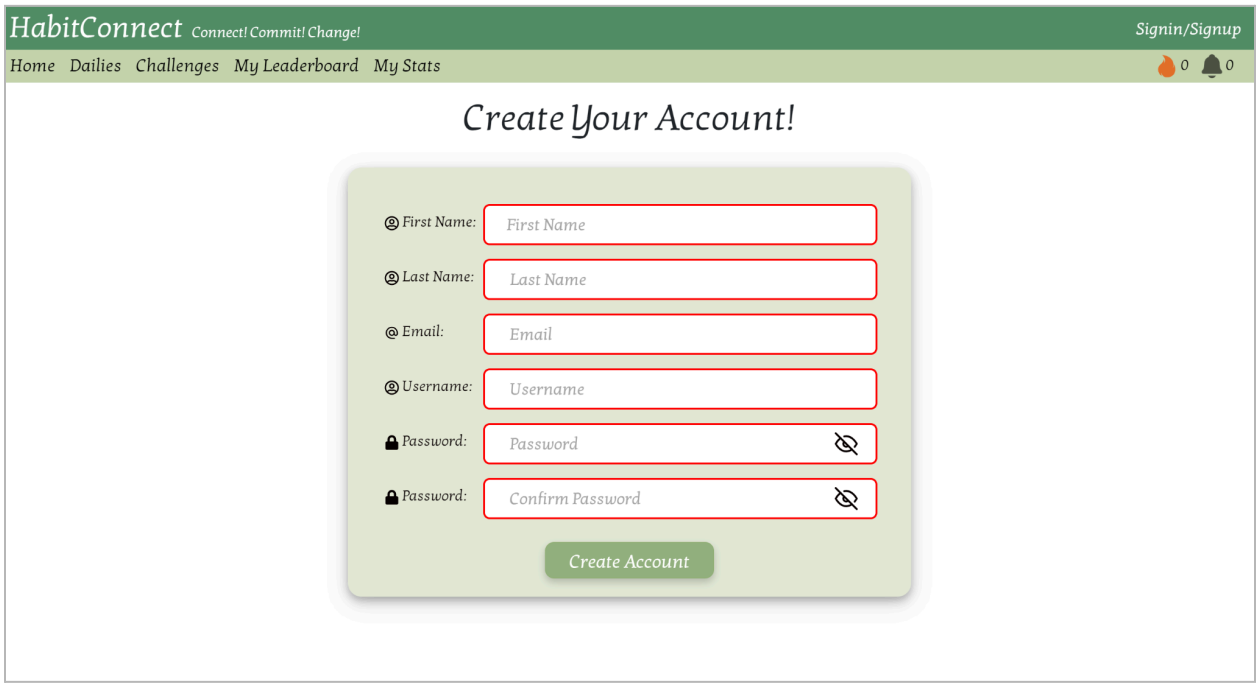
These issues were not resolved due to a lack of time at the end of the project. A full rundown of some of the tasks which were knowingly incomplete at the end of our final sprint can be found in our Sprint4Tasks.xlsx file (also available on GitHub). Below are some screenshots of all the pages implemented in the front-end:



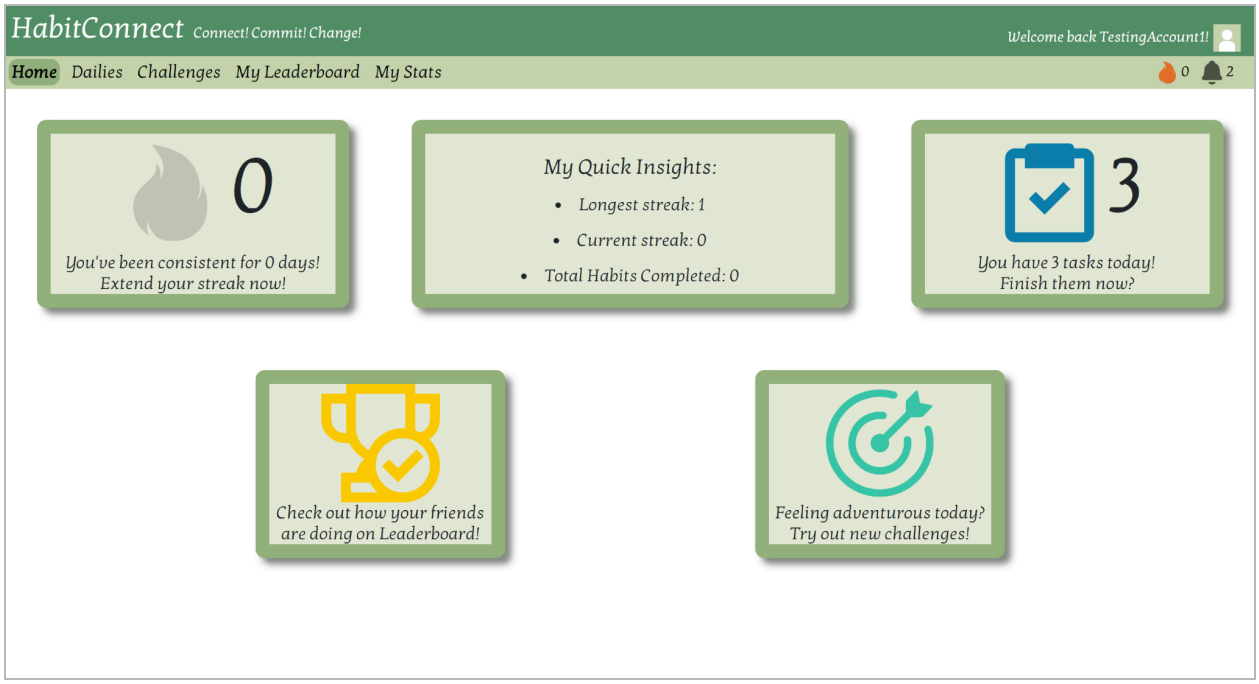
Guest user homepage.



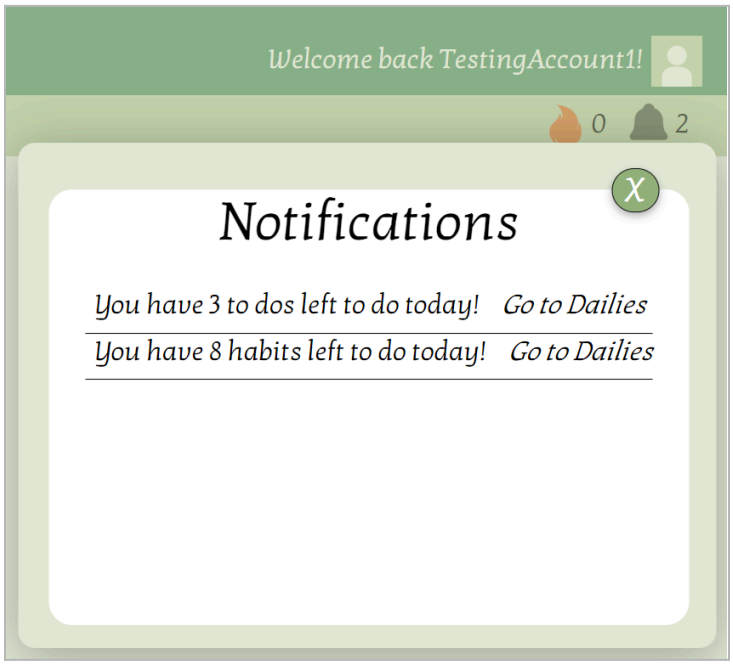
Sign in page.



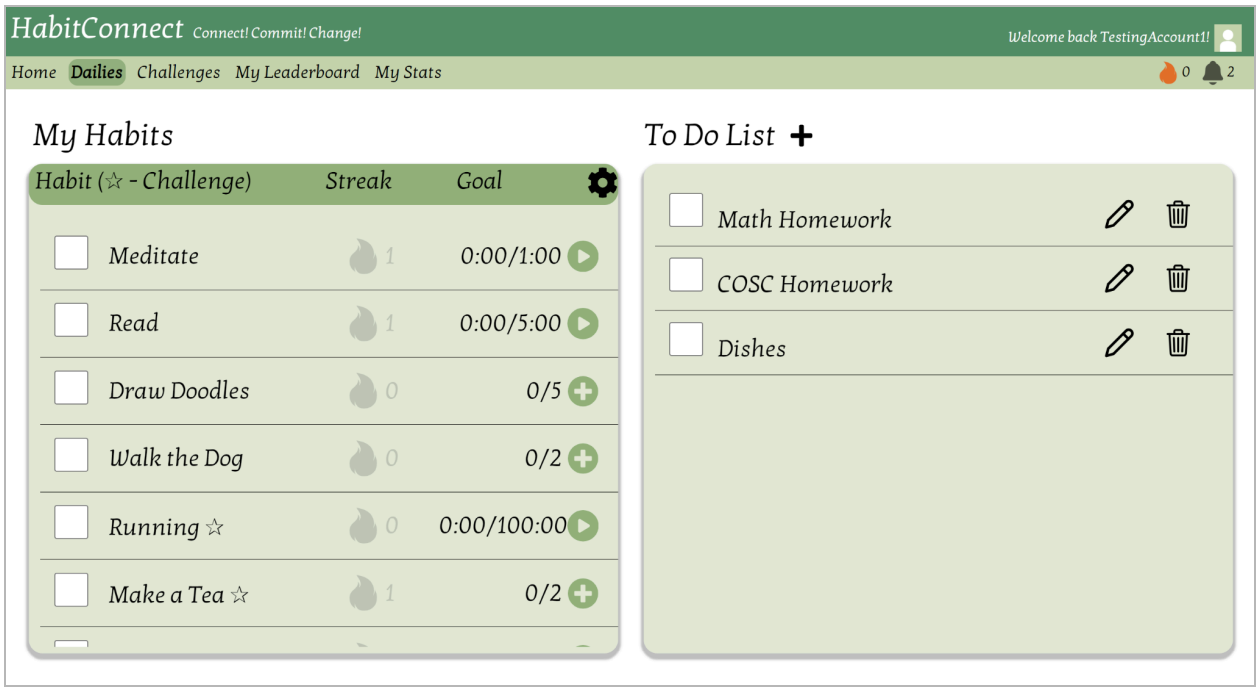
Sign up page.



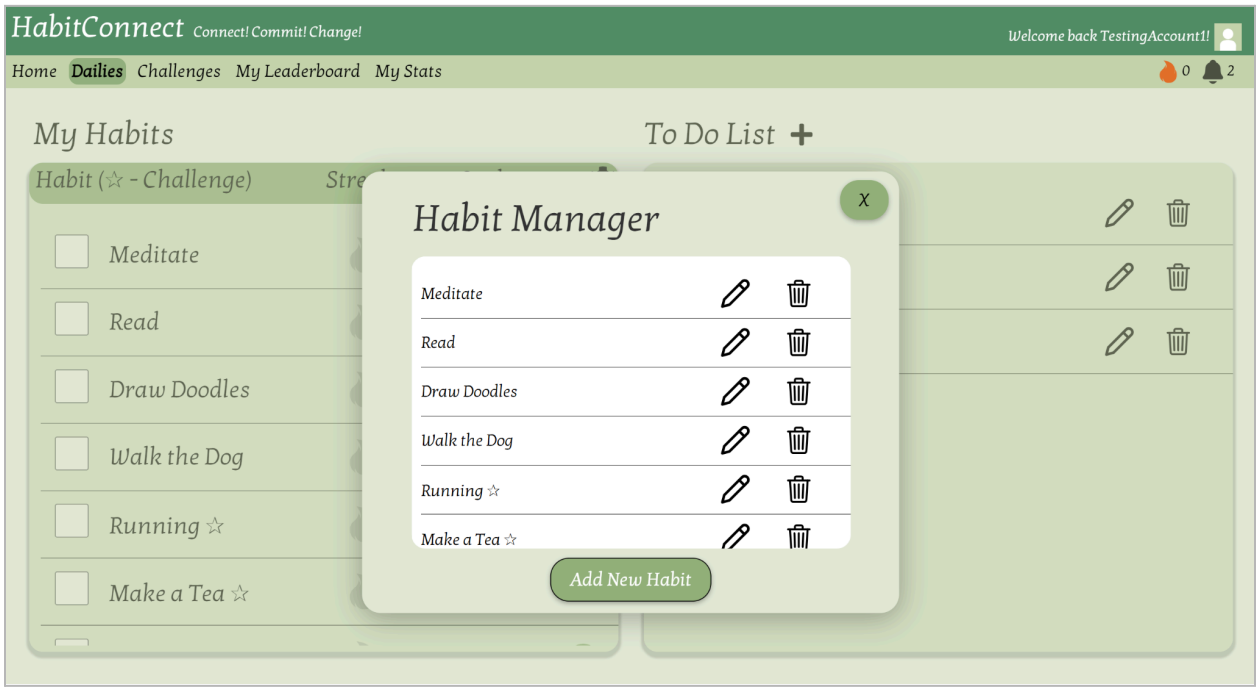
User homepage.



Notifications pop up.



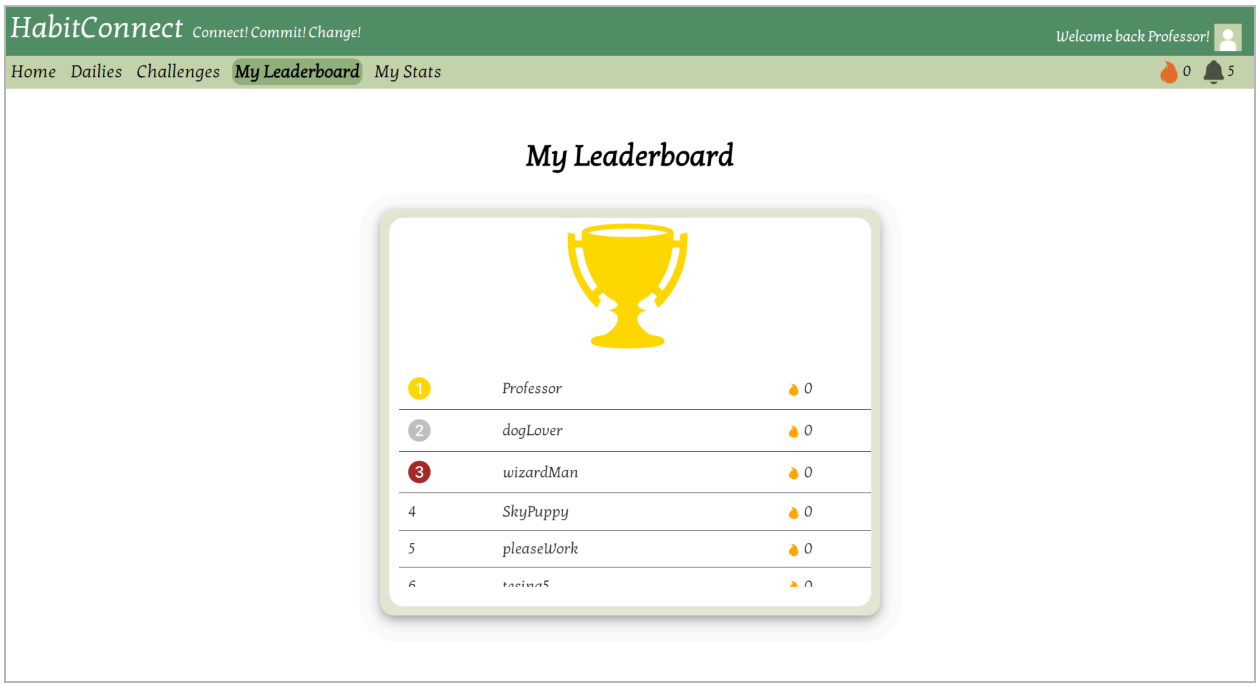
Dailies page.



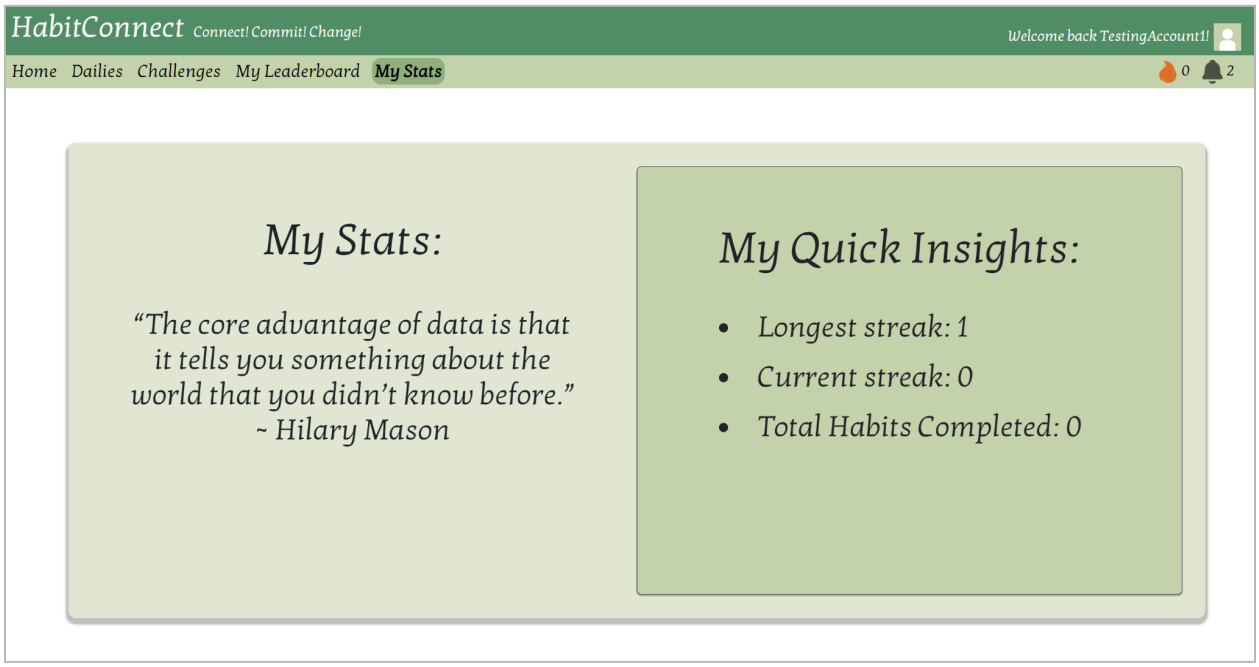
Habit Manager pop up.

Challenges page.

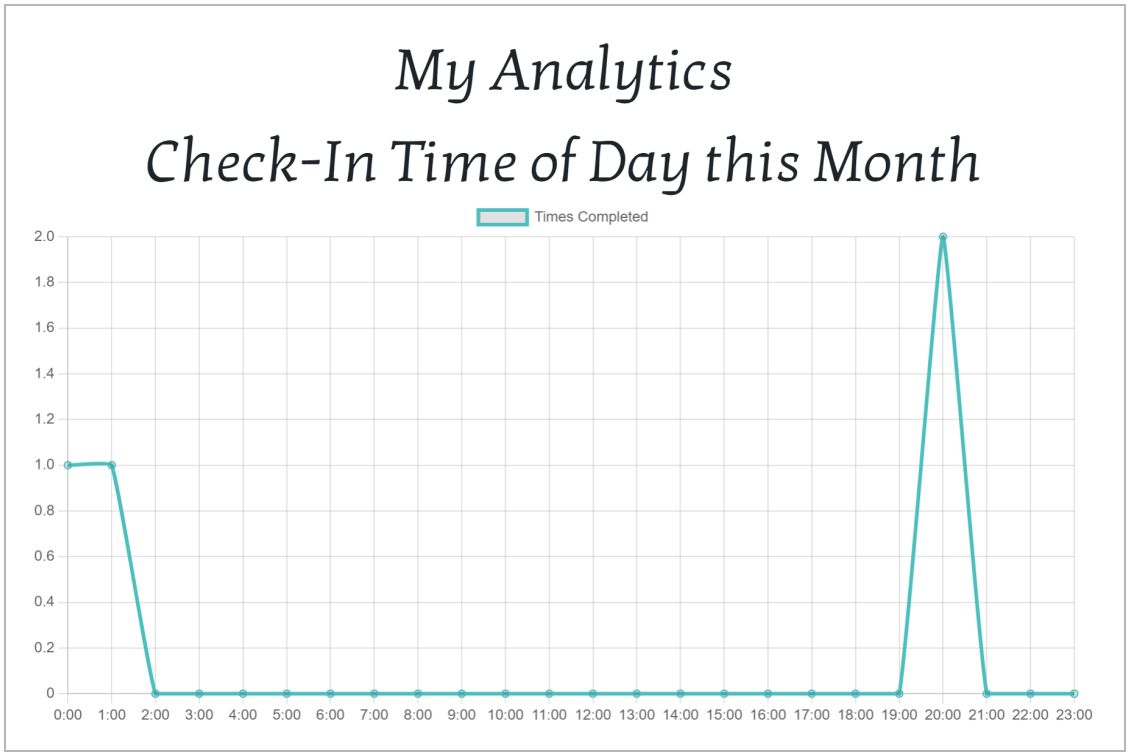
Challenge manager pop up.



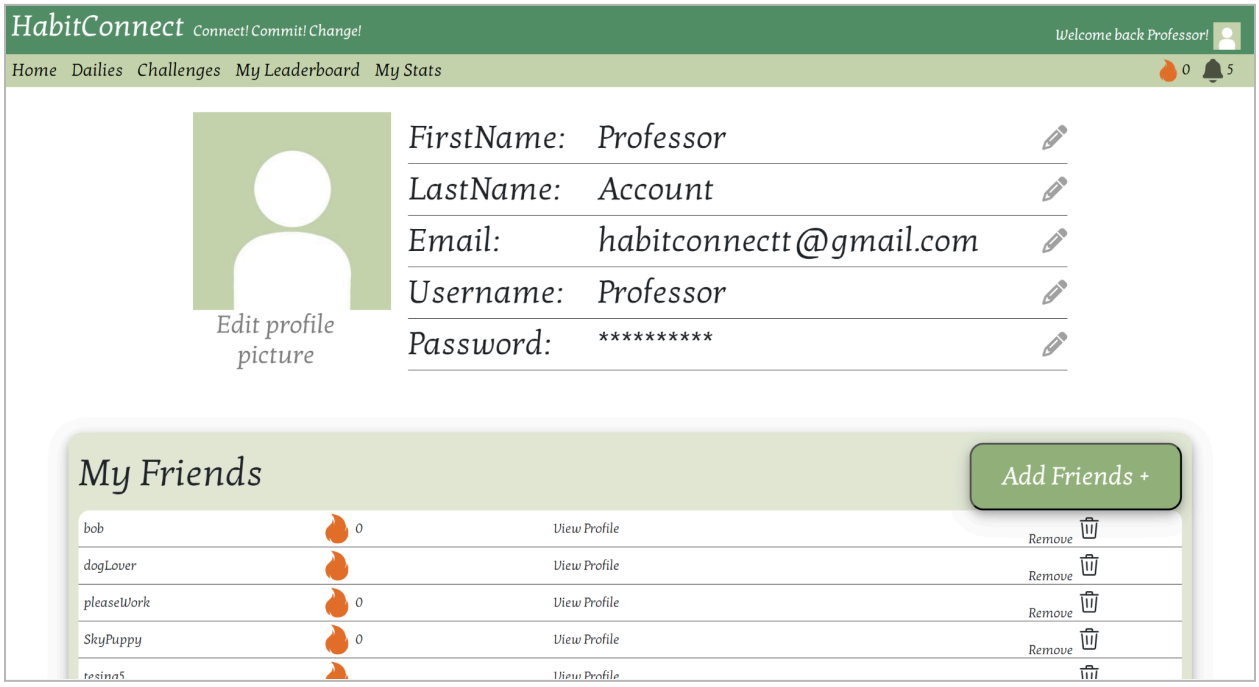
My Leaderboard page.



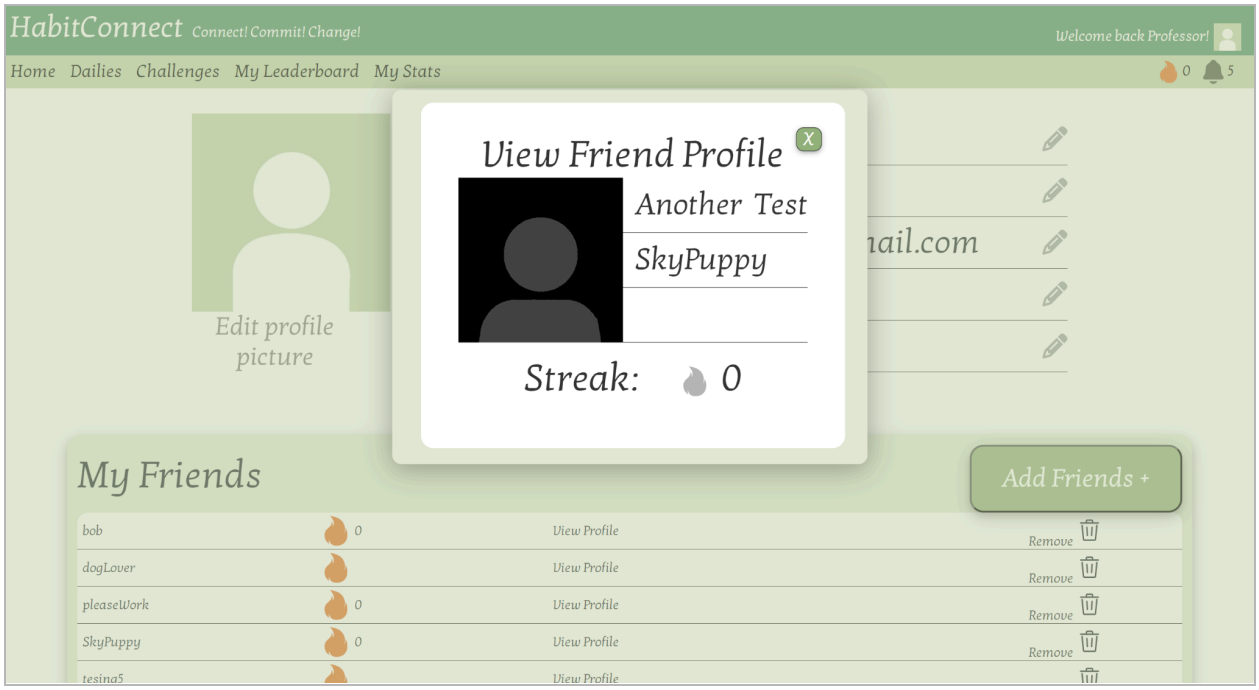
My Stats page (top).



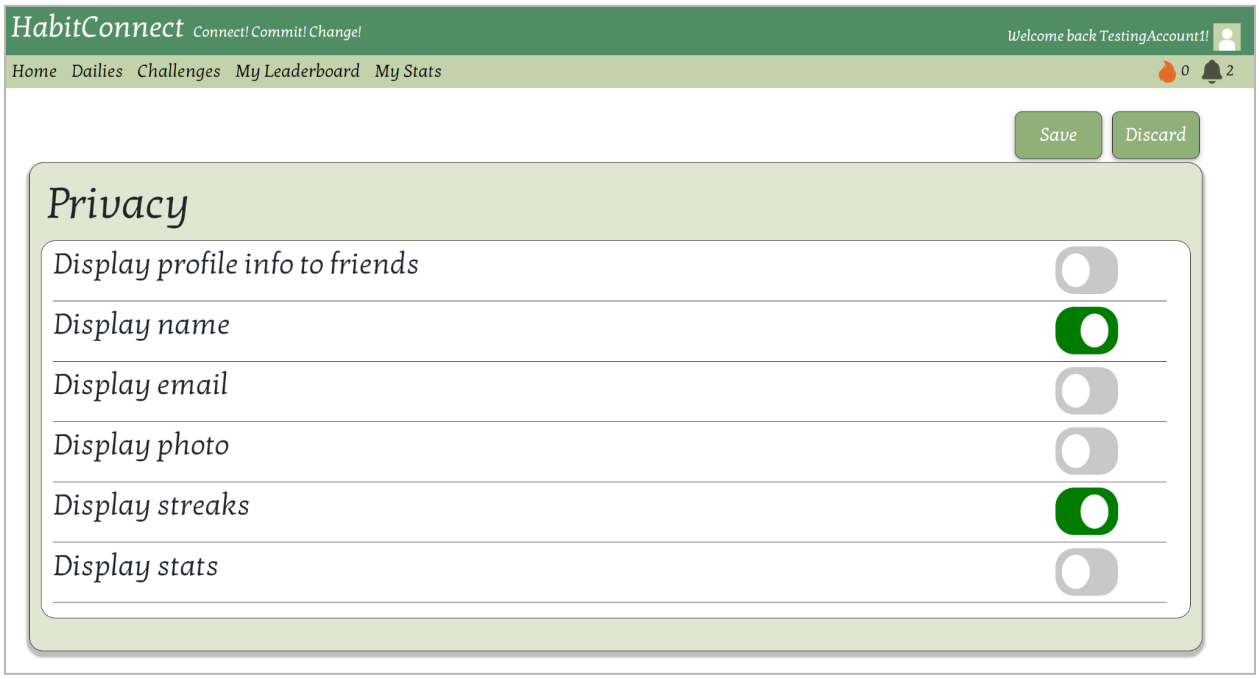
Graph on the My Stats page (bottom).



User profile page.



View Friend Profile pop up.



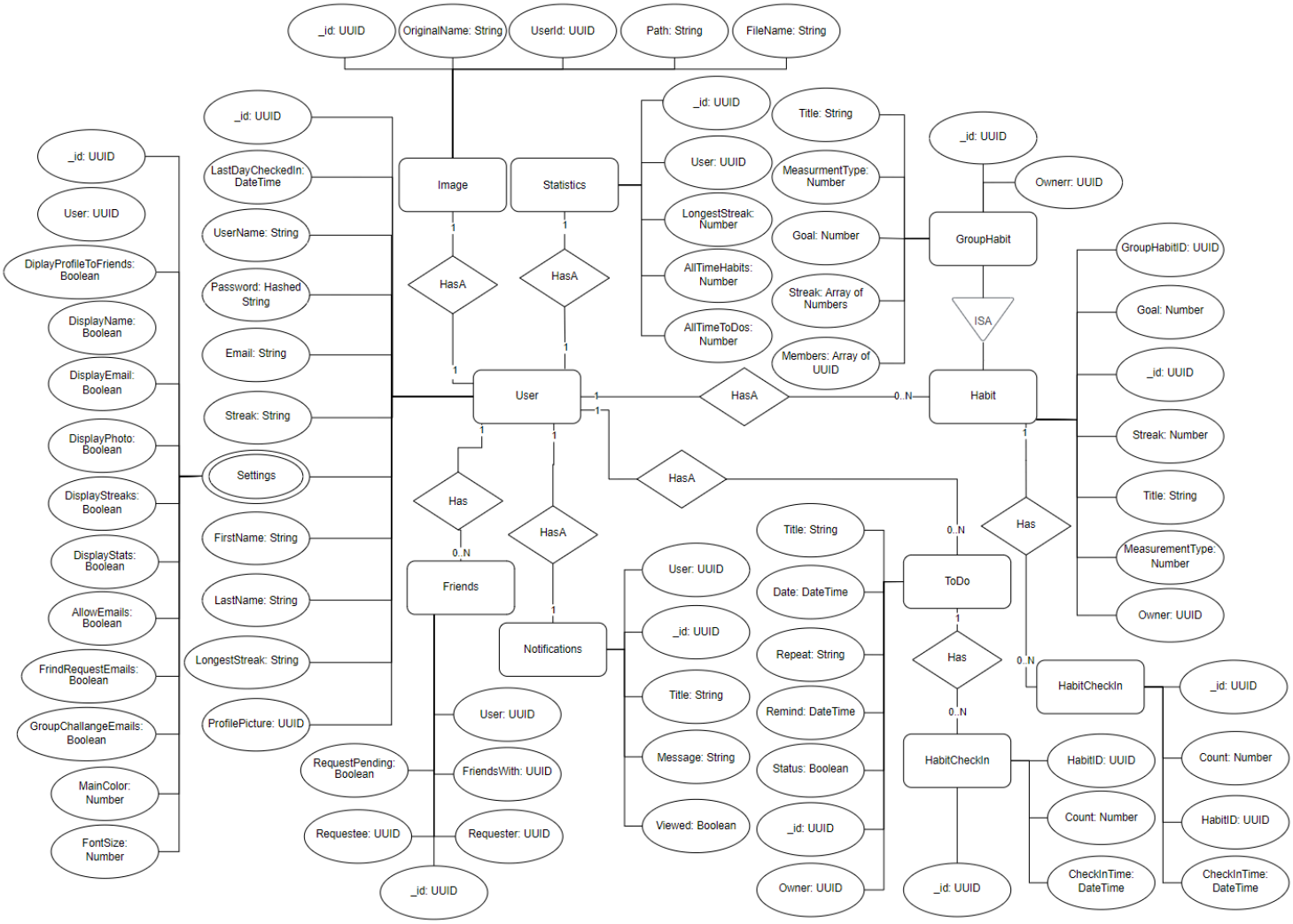
Settings page.

Back-End

Starting in Sprint 1, the back-end team focused their efforts on designing and implementing the beginning stages of our back-end work. We focused on designing the database in a way that would allow for easy implementation or modifications of other components later on. We ended up adding fields and tables after the initial design as the scope of the project became more clear, but the design in the first sprint was a good base template for the backend to start with.

Our MongoDB database was created based on these designs, and a manual for the rest of the team to be able to access the database was created. Development proceeded quickly with the actual implementation of the database. All the tables and schemas were created with test data to ease the front-end's implementation. One of the first functionalities developed in the backend was the signup and login functions. With a secure password that had a salt added to it, then was hashed and stored in the database.

The back-end team decided to divide the work in such a way where Lysa worked on everything related to the User Profile, Friends, Settings, and Notifications, and Robbie worked on everything related to Todo, Habits, Group Challenges, and Statistics. Later the work was divided up with Lysa working on Login, Sign Up, User Profile, Friends, Leaderboard, Forgot Password, Settings, Notifications, Group Habits, Images, User Streaks, and Sending Emails. Robbie worked on the Habits, Habit Check Ins, Todos, Todo Check Ins, and Statistics.



Testing and Results

Testing was a large focus for our team throughout the development of this project. Kacie focused efforts on developing manual test cases for front-end aspects of the site, while the back-end team implemented automatic testing to have high coverage of implemented functions.

Manual Testing

By the end of the project, we had 59 manual test cases, with 53 tests passing by the end of development. All test cases, with steps to run them, sample data and detailed results can be found in our Testing Master Spreadsheet. The manual test cases are mainly focused on pages where users could input invalid values into text fields. They cover the sign up, sign in, dailies and user profile pages, as well as contrast and visibility requirements. We also had plans to focus on other kinds of testing for the front-end, including accessibility, cross-browser, and performance testing. Due to time constraints at the end of the project, this testing was done in a more informal capacity, though they were always considerations while we were developing the website, since everyone used different browsers anyways. Testing was handled in order of priority.

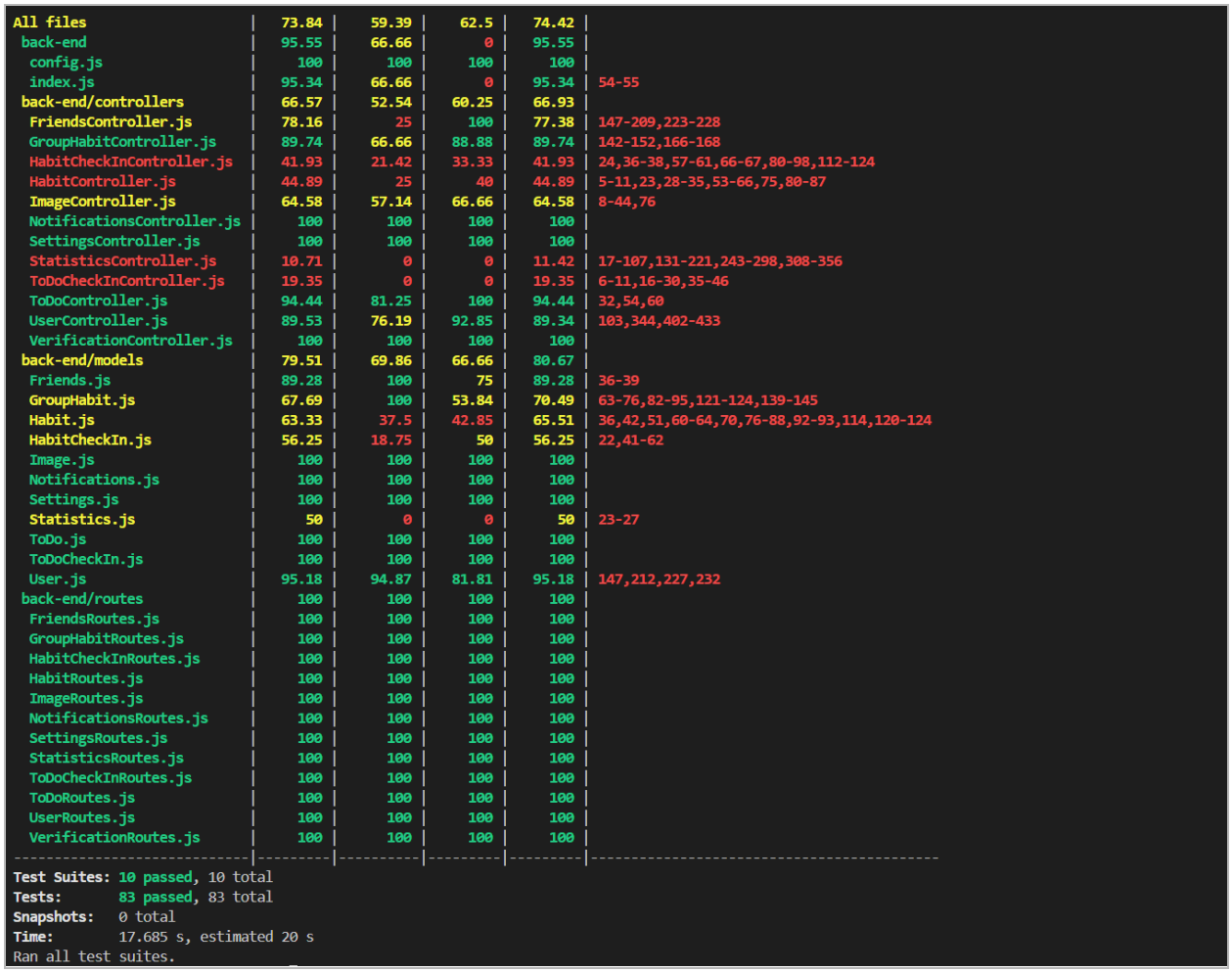
Automatic Testing

Automatic testing for back end API was important for us to begin implementing during our second and third sprints so we could quickly find and resolve any potential new bugs introduced to the system in new pull requests, and to reduce the frequency in which we would need to complete manual testing. We have 83 tests, which at the end of the project were all passed. We had a lot higher coverage at the end of sprint 2, but there were a lot of changes made in the last week that changed the coverage a lot unfortunately.

We used the dependency called Jest to help us implement our automatic testing. Additionally, we have ensured that around each of our controller functions there is error handling, incase of any unexpected errors that may occur.

There are a couple API's that we were unable to test automatically. These include all the API's that deal with sending emails, because sometimes they would time out, and this caused inconsistencies in our results. Also, the upload image API was unable to be tested automatically since we couldn't find a way to send images through automatic testing. But both of these exceptions were tested rigorously using manual testing so these cases were still covered.

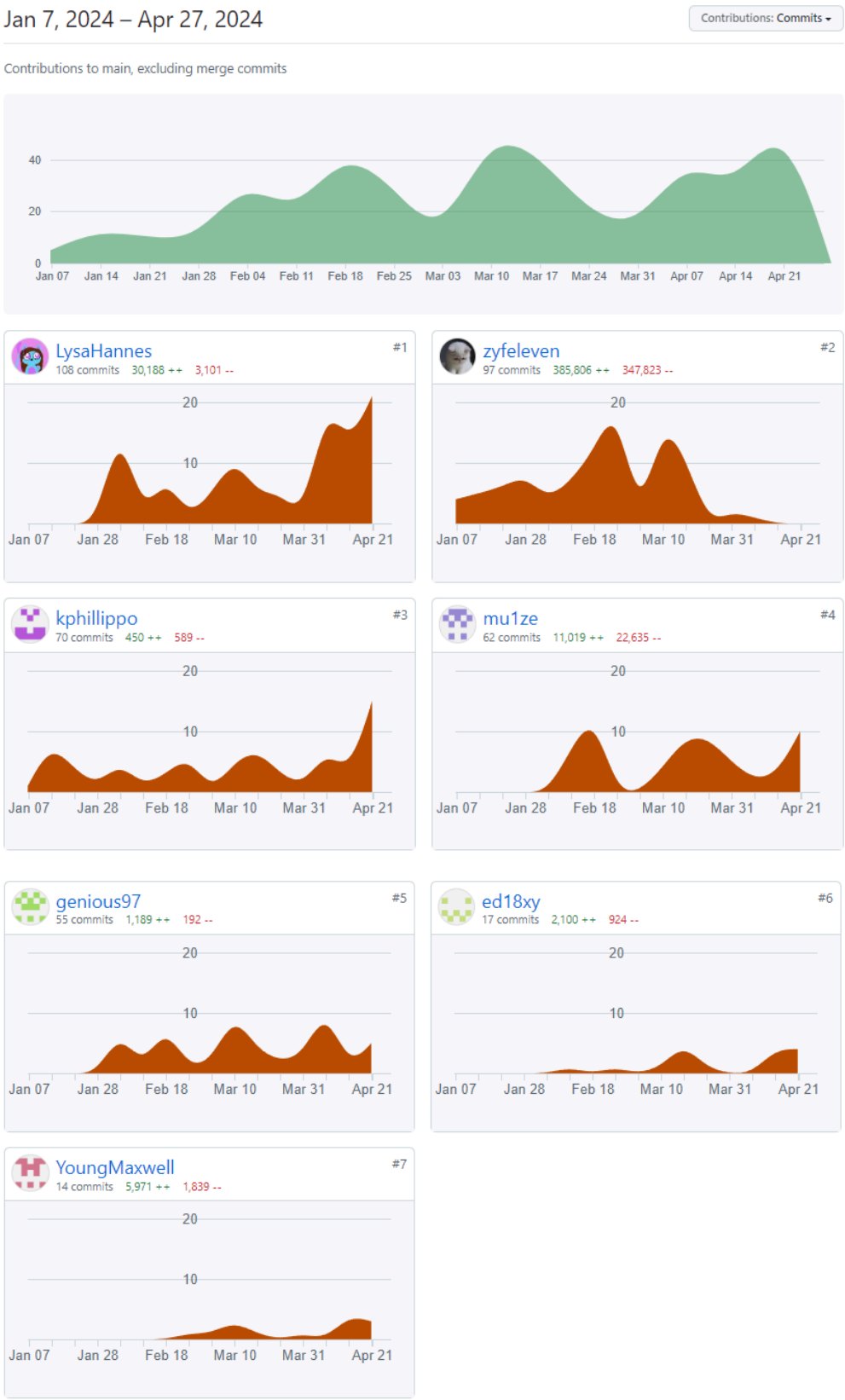
In order to run the backend manual test you must first "cd back-end", and then "npm test". Below is our automatic testing coverage report:



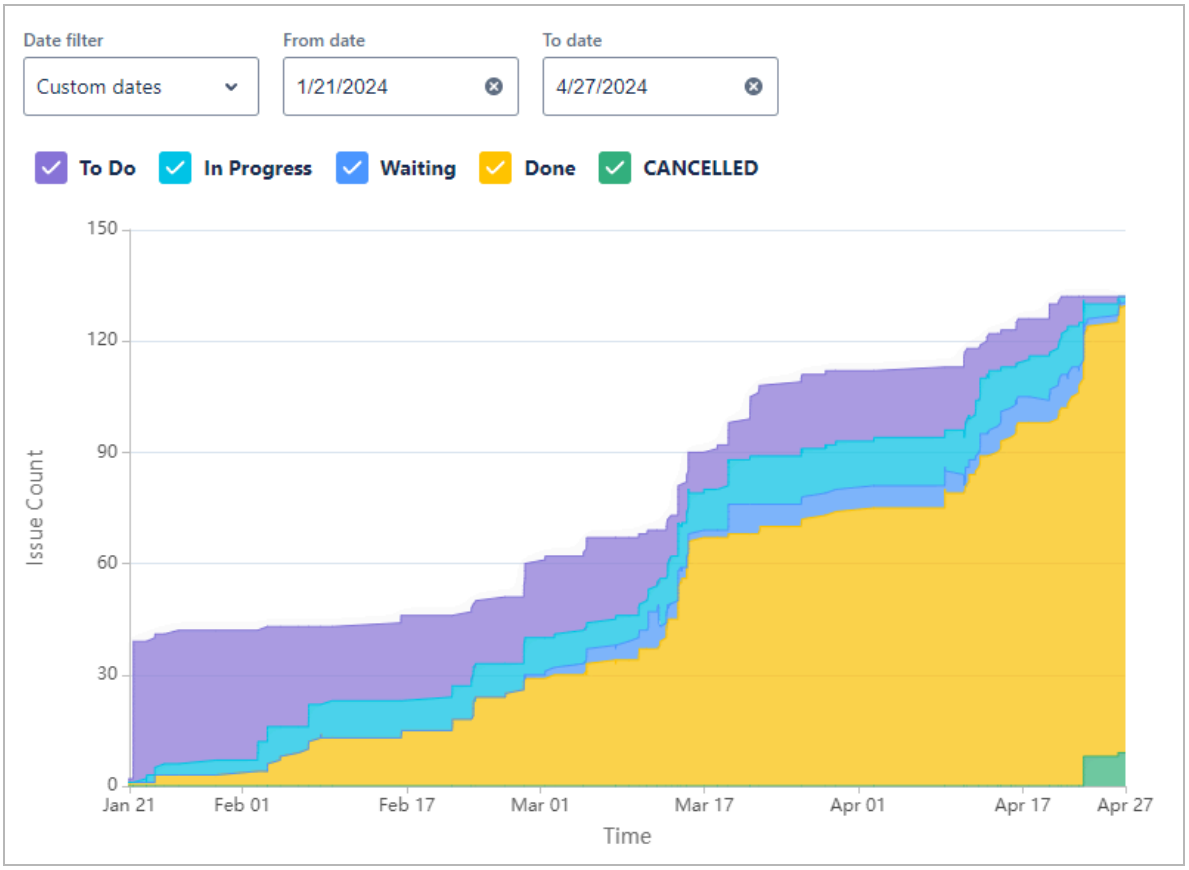
```
All files                    |   73.84 |    59.39 |    62.5 |   74.42 |
 back-end                    |   95.55 |    66.66 |       0 |   95.55 |
  config.js                  |     100 |      100 |     100 |     100 |
  index.js                   |   95.34 |    66.66 |       0 |   95.34 | 54-55
 back-end/controllers        |   66.57 |    52.54 |   60.25 |   66.93 |
  FriendsController.js       |   78.16 |       25 |     100 |   77.38 | 147-209,223-228
  GroupHabitController.js    |   89.74 |    66.66 |   88.88 |   89.74 | 142-152,166-168
  HabitCheckInController.js  |   41.93 |    21.42 |   33.33 |   41.93 | 24,36-38,57-61,66-67,80-98,112-124
  HabitController.js         |   44.89 |       25 |      40 |   44.89 | 5-11,23,28-35,53-66,75,80-87
  ImageController.js         |   64.58 |    57.14 |   66.66 |   64.58 | 8-44,76
  NotificationsController.js |     100 |      100 |     100 |     100 |
  SettingsController.js      |     100 |      100 |     100 |     100 |
  StatisticsController.js    |   10.71 |        0 |       0 |   11.42 | 17-107,131-221,243-298,308-356
  ToDoCheckInController.js   |   19.35 |        0 |       0 |   19.35 | 6-11,16-30,35-46
  ToDoController.js          |   94.44 |    81.25 |     100 |   94.44 | 32,54,60
  UserController.js          |   89.53 |    76.19 |   92.85 |   89.34 | 103,344,402-433
  VerificationController.js  |     100 |      100 |     100 |     100 |
 back-end/models             |   79.51 |    69.86 |   66.66 |   80.67 |
  Friends.js                 |   89.28 |      100 |      75 |   89.28 | 36-39
  GroupHabit.js              |   67.69 |      100 |   53.84 |   70.49 | 63-76,82-95,121-124,139-145
  Habit.js                   |   63.33 |     37.5 |   42.85 |   65.51 | 36,42,51,60-64,70,76-88,92-93,114,120-124
  HabitCheckIn.js            |   56.25 |    18.75 |      50 |   56.25 | 22,41-62
  Image.js                   |     100 |      100 |     100 |     100 |
  Notifications.js           |     100 |      100 |     100 |     100 |
  Settings.js                |     100 |      100 |     100 |     100 |
  Statistics.js              |      50 |        0 |       0 |      50 | 23-27
  ToDo.js                    |     100 |      100 |     100 |     100 |
  ToDoCheckIn.js             |     100 |      100 |     100 |     100 |
  User.js                    |   95.18 |    94.87 |   81.81 |   95.18 | 147,212,227,232
 back-end/routes             |     100 |      100 |     100 |     100 |
  FriendsRoutes.js           |     100 |      100 |     100 |     100 |
  GroupHabitRoutes.js        |     100 |      100 |     100 |     100 |
  HabitCheckInRoutes.js      |     100 |      100 |     100 |     100 |
  HabitRoutes.js             |     100 |      100 |     100 |     100 |
  ImageRoutes.js             |     100 |      100 |     100 |     100 |
  NotificationsRoutes.js     |     100 |      100 |     100 |     100 |
  SettingsRoutes.js          |     100 |      100 |     100 |     100 |
  StatisticsRoutes.js        |     100 |      100 |     100 |     100 |
  ToDoCheckInRoutes.js       |     100 |      100 |     100 |     100 |
  ToDoRoutes.js              |     100 |      100 |     100 |     100 |
  UserRoutes.js              |     100 |      100 |     100 |     100 |
  VerificationRoutes.js      |     100 |      100 |     100 |     100 |
-----------------------------|---------|----------|---------|---------|-------------------------------------------
Test Suites: 10 passed, 10 total
Tests:       83 passed, 83 total
Snapshots:   0 total
Time:        17.685 s, estimated 20 s
Ran all test suites.
```

GitHub and Jira Logs

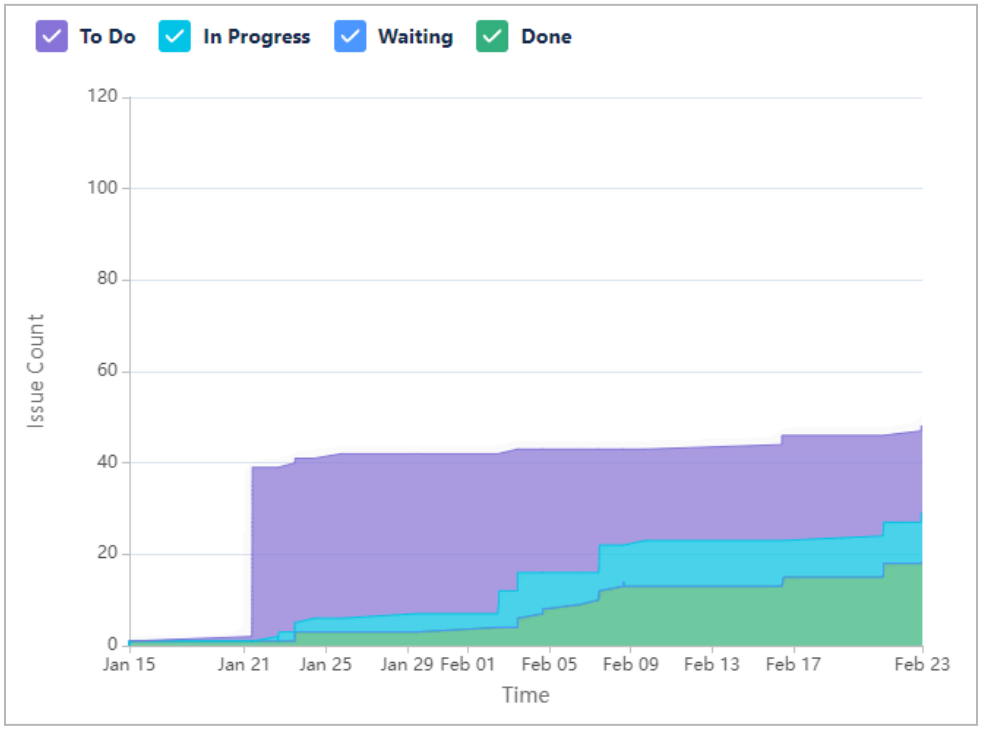
Below are some screenshots of reports generated by GitHub and Jira to track our progress during the software engineering process. We encourage you to focus more heavily on the “Contributions” section of this report, since this provides a more accurate representation of the full breadth of responsibilities and work put in by each group member.



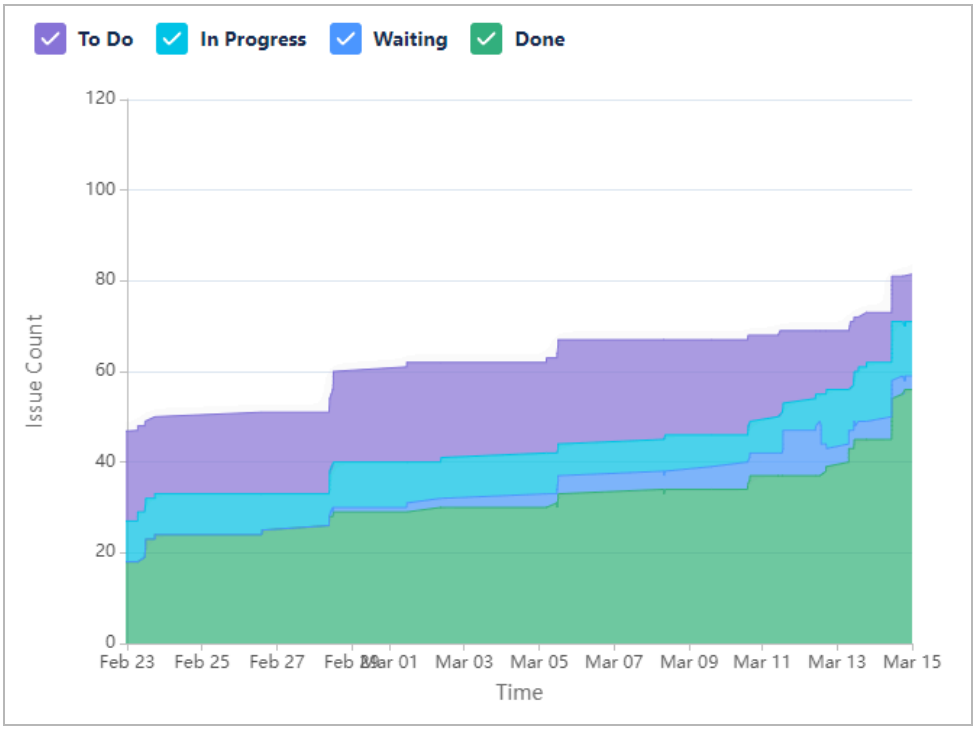
Report of contributions to the main branch (GitHub).



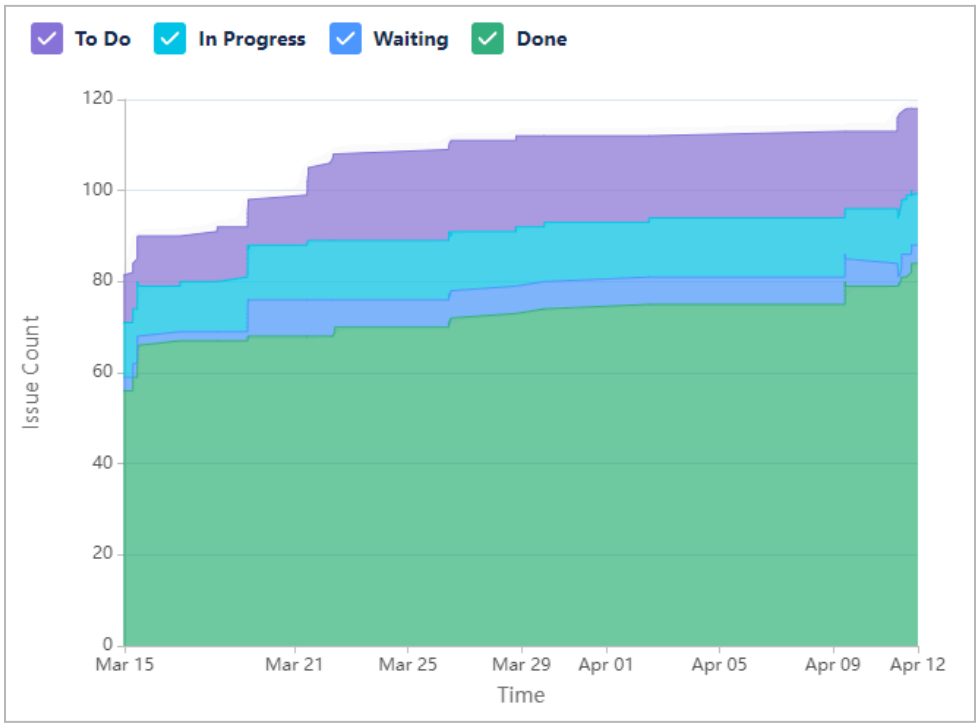
Cumulative flow diagram (full project) (Jira).



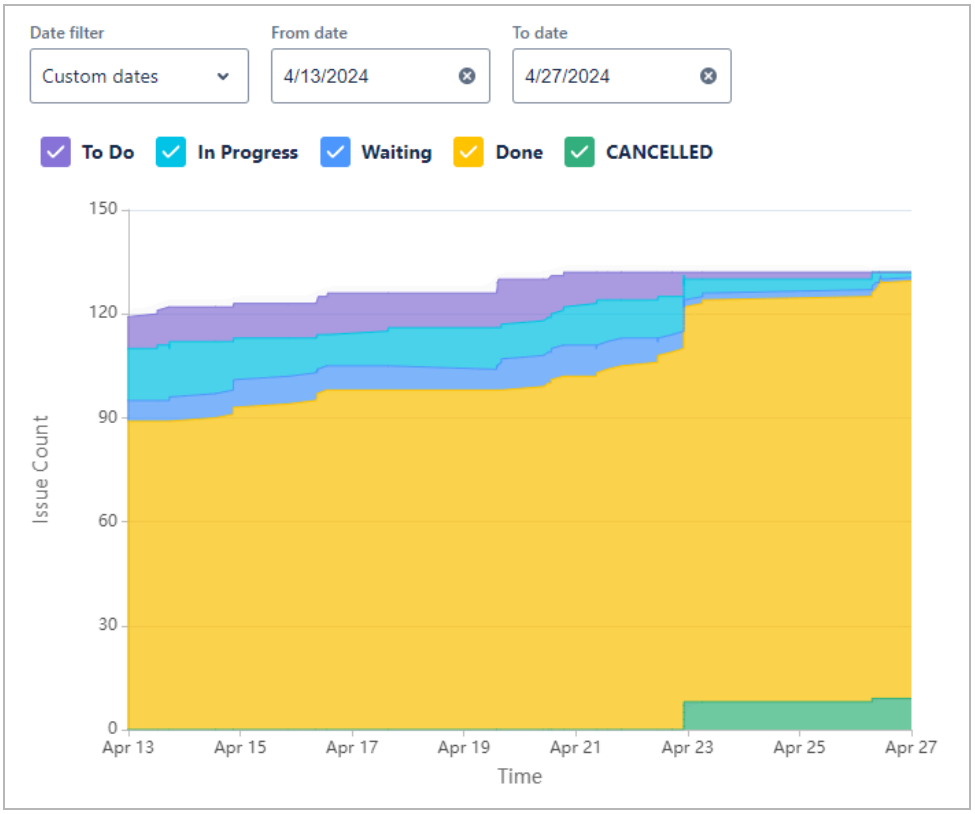
Cumulative flow diagram (Sprint 1) (Jira).



Cumulative flow diagram (Sprint 2) (Jira).



Cumulative flow diagram (Sprint 3) (Jira).



Cumulative flow diagram (Sprint 4) (Jira).

Software Engineering Process

Throughout our software engineering process, we went through 4 scheduled Sprints:

- Sprint 1, January 17th - February 22nd, Sprint1Tasks.xlsx
- Sprint 2, February 23rd, March 15th, Sprint2Tasks.xlsx
- Sprint 3, March 16th - April 12th, Sprint3Tasks.xlsx
- Sprint 4, April 13th - April 26th, Sprint4Tasks.xlsx

For each of these scheduled sprints, we completed Sprint Planning, Sprint Review, and Sprint Retrospective meetings, to make sure that we were staying on top of our progress and we could reprioritize tasks as required to make sure all requirements and deadlines were being met. These detailed meeting notes, as well as meeting notes from our 'daily' scrum meetings twice a week, can be found in the meeting-notes folder on our GitHub page. Between scheduled meeting times, we continued to communicate through a constantly active Discord channel, where we kept each other updated about our current progress, and reached out when we needed help. This was an excellent way for us to compensate for being unable to meet daily as with regular scrum processes due to responsibilities in our other courses.

We consistently utilized our Jira page for task/user story assignment and prioritization. We did our best to complete all tasks within their scheduled sprints. However, we did not always accomplish this goal. With our lack of experience, we found it difficult to estimate the amount of time it would take to do specific tasks, leading us to have to push tasks back in Sprints

frequently. However, by the end of the project, we completed almost everything we were hoping to accomplish, so our overall project scope was correct. Detailed updates on all tasks per sprint can be viewed in the Sprint#Tasks.xlsx files linked above.

We had meetings twice a week. It was during these meetings where we updated each other on our progress since the last meeting, asked each other for help, and updated the Jira to make sure tasks were being updated appropriately. It was here where we also made sure that our workload was being distributed relatively evenly. As was stated earlier, for communication between these meetings, we had an active Discord channel to make sure communication was consistent throughout the week and all important information was remembered.

Challenges

We encountered many challenges throughout the development of HabitConnect as we were learning how to actively participate in the software engineering process.

At the beginning, it was quite difficult for us to get to know each other. For many of us, it was our first time interacting and working with each other, so it took time to learn each other's skill sets and learning/working styles. Once we got over this hurdle, it became much easier to communicate with each other and we got into a much more natural flow where task assignment became obvious. Additionally, we learned who to go to for help in certain areas. For many of us it was our first time coding as part of a group, so we were very happy and comfortable with our group dynamic by the end of the project.

For many of us, it was our first time using software engineering tools like Jira, GitHub and Figma. Our first sprint for many of our members was spent learning the tools and how to properly maintain version control through GitHub.

Another challenge we have faced during our time working on this project is managing the weight of this project with the responsibilities of our other classes. Many of us are in our final semester at Brock, and taking many upper year courses which are very demanding. As such, some group members were unable to contribute as much as others between our scrum meetings at different times during our sprints, in order to prioritize other assignments and tests/exams. However, our 2 meetings a week did a good job at keeping everyone accountable for the work they contributed to the project, while not overwhelming the members.

As the project went on we learned how to better estimate the time it would take to complete certain tasks, which also allowed for better task assignment throughout the semester. However, at the beginning, it was more difficult to predict how long specific tasks would take, and what we would be able to accomplish within one scheduled sprint. This led to many tasks carrying over to later sprints. While it may appear that we were not meeting deadlines, we accomplished almost everything we were planning on adding to the website. While the scheduling of our sprints may have been out of scope at times, the scope of our project as a whole was accurate and reasonable within the assigned project deadlines.

While we faced quite a few challenges throughout this project, we view them as learning opportunities, and have developed plenty of new skills, both technical and otherwise, by the end of this project course.

Successes

There are many successes we experienced throughout this project we feel are important to highlight, as we are proud of the work we have done throughout this course.

Firstly, we were very proud of the amount of time that we invested in the design of the website, both in the front and back-end. This made implementation of features and pages on our site much simpler, since there were designs to work directly off of. We knew the design process was important and feel as if we completed this work to an excellent level.

We are also proud of the amount of time we invested in manual and automatic testing. This is also something we felt was extremely important in the software engineering process. We did not want to leave it all until the last minute, as this is not where testing is necessarily its most useful. Our manual and automatic tests, as described above in this report, were extensive, and we found focusing on this process to be extremely useful throughout this project in ensuring we were meeting all user requirements.

Finally, we were extremely proud of our close adherence to the software engineering process. We used a modified version of the scrum development process, where we met twice a week, with constant communication through our group's Discord server. All members made it to nearly all meetings, with only a few absences (but never from our more important Sprint meetings). We have detailed meeting notes for every meeting we conducted, and our documentation (within and outside of these reports) is extensive. We are confident that anyone looking through our GitHub page in its entirety would easily be able to see the commitment we made to ensuring we were fully investing ourselves in the software engineering process.

We are most excited by the amount we learned throughout this semester, both about programming and the software engineering process as a whole. We believe that skills we learned in the development of HabitConnect will take us far beyond the scope of this course and into our future endeavors.

Contributions from the Development Team

Name	Role	Student ID	Brock Email	GitHub Username
Kacie Phillippo	Scrum Master	6877591	kp19to@brocku.ca	kphillippo
Muiz Odebiyi	Product Owner	6849509	mo19qw@brocku.ca	mu1ze
Elya Denysova	Project Manager	6667596	ed18xy@brocku.ca	ed18xy
Yifan Zhu	Front-End Lead	7345671	yz21mm@brocku.ca	zyfeleven
Maxwell Young	Front-End Lead	6769608	my19if@brocku.ca	YoungMaxwell
Lysa Hannes	Back-End Lead	6695100	lh18wt@brocku.ca	LysaHannes
Robert Pierik	Back-End Lead	6773832	rp19dq@brocku.ca	genious97

Kacie Phillippo (6877591) Scrum Master

- Contributed to the creation and editing of all the user stories and tasks to be added to the product backlog.
- Wrote and maintained the GitHub README section with relevant information.
- Led the scrum meetings which occurred twice a week, directing conversation, ensuring that all members are contributing and all important topics are discussed before the end of the meeting time. Led all Sprint Retrospective, Sprint Review meetings, and Progress Report meetings. Was one of the leads for the final presentation.
- Wrote and uploaded meeting minutes for all scrum meetings, all progress report meetings, all three 4P02 online tutorials, the Sprint Review meetings, the Sprint Retrospective meetings and the Sprint Planning meetings. Created and uploaded the Jira task rundown spreadsheets from the Sprint Review meetings.
- Design ideas were the basis for the site identity currently behind HabitConnect, including colour scheme, navigation bar, pop ups, and reusable component designs. Led the designs of the Dailies, Sign In, User Profile, Guest Homepage, Notification dropdown and My Stats pages in figma.
- Managed branch merging and version control on GitHub, reviewing all pull requests to the development branch (later named testing-branch) and making edits to code as required. Handled all changes to the main branch.
- Was the primary contributor to all content in Progress Report 1, 2, and the Final Report, compiling the entire group's contributions to display and doing the written work.
- Created sign in/sign up activity diagram for the first Progress Report.
- Designed the Statistics and Notifications tables for the database, and updated the database structure diagram appropriately.
- Debugging of many pages, smaller fixes throughout the site as the project came to an end. Did the beginning of the My Stats page implementation (basic .css only). Did an overhaul of the guest user homepage to have a more responsive design.

- Was a main contributor to the Testing Ideas document, brainstorming manual tests that need to be verified on the site. Created the Testing Master Spreadsheet, outlining the specifics of each manual test which will be run. Manually ran all tests, and communicated features which needed to be improved upon to the appropriate group members.
- Wrote and maintained the Installation Guide and User Manuals, and developed the updated list of user requirements, for the Final Report.
- Wrote and created the templates and generated the .html code for all emails to be sent from the website.
- Created the final presentation powerpoint, and wrote the general outline of what needed to be said throughout the final presentation and by whom. Was a presenter for the final presentation.
- Voiced the beginning and ending portions of the instructional YouTube video for HabitConnect. Posted the video with the written description.

Muiz Odebiyi (6849509) Product Owner

- Contributed to the creation and editing of all the user stories and tasks to be added to the product backlog.
- Transferred list of requirements from shared document into Jira, and managed initial sprint scheduling and task assignments.
- Acted as the voice of the front-end team for Progress Report meetings, updating the TA on the team's progress and next steps.
- Implemented the sign in page with form requirements according to specifications.
- Implement the sign up page with a working form to collect user information.
- Designed email-verification popup in signup page. Implemented emailing functionality across the site, and set up email verification for site sign up.
- Changed styling of the navigation bar to be consistent with the rest of the pages, as per the designs.
- Implemented Add/Edit/Delete and join GroupHabit functionality in the challenges page.
- Led implementation of the Leaderboard page.
- Bug fixing with sign-in and sign-up pages.
- Designed a forgot password function and implemented a way for users to change their password if forgotten.
- Made design changes to Dailies and Challenges pages.

Elya Denysova (6667596) Project Manager

- Contributed to the creation and editing of all the user stories and tasks to be added to the product backlog.
- Transferred list of requirements from shared document into Jira, and managed initial sprint scheduling and task assignments.
- Created Figma board and Jira Scrum project for design and project management tasks.
- Collaborated on creating and inputting initial product back-log for 3 sprints.
- Researched existing software (apps and websites) similar to our habit tracker, sketched out initial ideas for the tracker.

- Collected screenshots that are the most relevant and contain interesting design ideas Presented for review for the entire team in Figma.
- Designed Login Page, User Profile Page, Viewing Friend Popup, Update info popup, Challenges Page, Leaderboard Page, and Settings Page.
- Collaborated on the designs of Home and Dailies pages as well as pop-ups for Dailies page.
- Implemented Home and User Profile pages.
- Monitored Jira product back-log progress.
- Designed Mobile view of basic components and Challenges page (in progress).
- Enabled API connection on User Profile Page, Home Page, and Notifications.
- Implemented various popups for the user page to make it fully functional.
- Updated and organized tasks in Jira.

Yifan Zhu (7345671) Front-End Lead

- Contributed to the creation and editing of all the user stories and tasks to be added to the product backlog.
- Implemented navbar and routings, created reusable API request functions in front-end.
- Provided possible front-end and back-end technical stacks for the team.
- Managed branch merging and version control on GitHub.
- Utilized React props and states in the front-end to convert static web pages to dynamic pages.
- Implemented user authorization system (sign-in, sign-up, sign-out) in front-end and integrated the system with our back-end.
- Created and continuously updated API document and libraries version control document.
- Set up automated testing for functions in the front-end.
- Implemented dynamic Dailies page with real-time data from back-end.
- Implemented Add/Edit/Delete Habit function on Dailies page.
- Implemented Add/Edit/Delete Todo function on Dailies page.
- Manually tested all existing functions on Dailies page in the front-end.
- Was responsible for maintaining the Dailies page.

Max Young (6769608) Front-End Lead

- Contributed to the creation and editing of all the user stories and tasks to be added to the product backlog.
- Worked on implementing the Dailies page based on designs created.
- Added tables for the To Do list and My Habits.
- Came up with a scrapped Leaderboard design and made it in Figma.
- Fixed To Do and habit tables.
- Added a plus button to the column bar of the habits table.
- Worked on implementing and created the html and css for the “Add Habit” popup and the “Add ToDo” popup.
- Worked on implementing and created the html and css for the “Edit Habit” popup and the “Edit ToDo” popup.

- Worked on implementing and created the html and css for the “Habit Manager” popup and a “Are you sure you want to delete” popup.
- Worked on the html and css for the Settings Page.
- Finished the functional portion of the Settings Page. Implemented the display profile info to friends, display name, display email, display stats, and display streaks parts of the settings table.
- Made the display values connect to the back-end when you save on the settings page.
- Implemented the discard capability of the settings page.
- Implemented the friend email request for the settings page which is connected to the back-end.
- Made the visual and functional portions of the Stats Page.
    - Made a graph for the stats page.
    - Implemented a quick insights portion of the Stats Page which gets data from the back-end.
- Implemented the Guest User Home page, the home page that a user sees upon entering the site.

Lysa Hannes (6695100) Back-End Lead

- Contributed to the creation and editing of all the user stories and tasks to be added to the product backlog.
- Acted as the voice of the back-end team for Progress Report meetings, updating the TA on the team’s progress and next steps.
- Worked on the backend part of the reports.
- Designed the database with Robbie.
- Initialized the setup of the database.
- Created about half the tables in the database.
- Created User API’s (login and sign up) + testing (100% coverage)
- Made password storage safe with salt + hashing.
- Set up automated testing using jest for back end.
- Created a template for test cases.
- Created the base templates for routers and controllers for each schema.
- Created API’s for the user profile + testing (100% coverage).
- Created API’s for all friends functions + testing (100% coverage).
- Created API to return the leaderboard + testing (100% coverage).
- Created API for sending emails + testing (100% coverage).
- Created API’s for settings + testing (100% coverage).
- Created API’s for notifications + testing (100% coverage).
- Updated the ER Diagram whenever it needed to be updated.
- Created API’s for Group Habits + testing (about 70% coverage)
- Created API’s for forgot password
- Created API’s for profile pictures (images)
- Created API’s for user streaks and longest streak tracking
- Worked on lots of last of minute back end bug fixes and requests
- Edited and put together the video for the video tutorial.

Robbie Pierik (6773832)

- Contributed to the creation and editing of all the user stories and tasks to be added to the product backlog.
- Collaborated in the design and initialization of HabitConnect's database.
- Made our database info safe when pushing to GitHub.
- Created a couple tables in the database.
- Creating habit back-end API with routes and controllers.
- Created API's for the habits.
- Created Automated Tests for Todos.
- Created API's for the todos.
- Started on the check in system.
- Created Automated tests for HabitCheckIn.

Resources Used

Chat GPT (and other AI tools)
As instructed that this was allowed within the scope of this course, we utilized ChatGPT to help us with this project. We used it for both idea generation and as a coding assistant. We feel that because we were able to do so, we learned how to utilize AI to help us effectively and efficiently code. This skill will help us with our jobs after university, since being able to utilize AI to help code is becoming the new norm within the industry. It helped a lot with debugging, catching small mistakes that might have taken hours to figure out, and learning how to code using the MERN stack (which was new to most group members).

The MongoDB website was useful for helping with setting up the database.
https://www.mongodb.com/docs/atlas/

This YouTube playlist was used to help development of the login and sign up functionalities in the back-end. It also helped us learn how to use MVC in the back-end.
https://www.youtube.com/watch?v=WsRBmwNkv3Q&list=RDCMUCW5YeuERMmlnqo4oq8vwUpg&start_radio=1

This is the link to the website we used for the profile picture.
https://stock.adobe.com/ca/images/default-profile-picture/64676383?asset_id=64672736