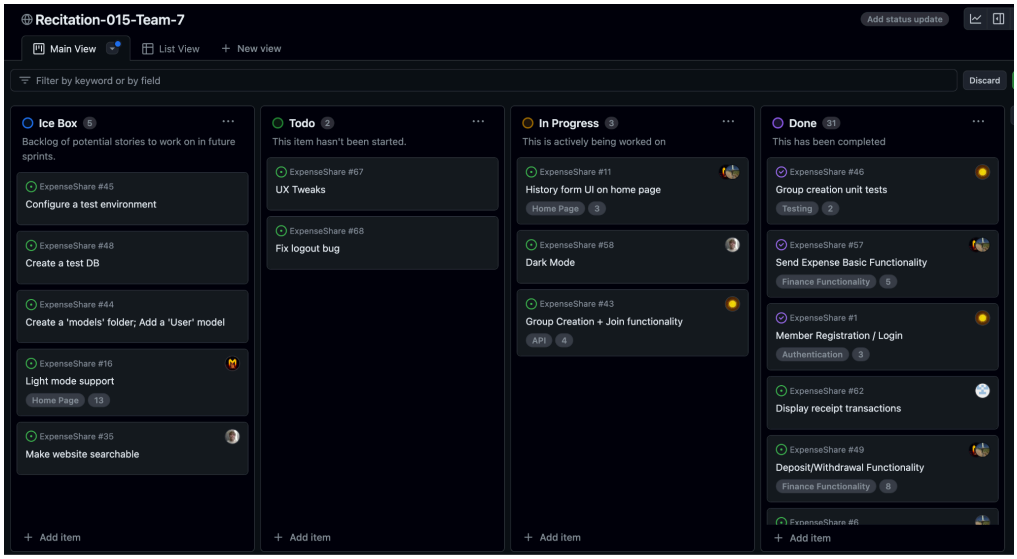
# ExpenseShare®

**Members:**
Connor Julson
Tyler Haskins
Avery Wagner
Mason Bott
Mariana Vadas-Arendt

**Project Description:**
Expense Share is a user based finance platform that allows users to join groups to calculate shared expenses. Each user has their own account and ability to deposit and withdraw money from their Expense Share account to and from their bank. Users can create groups and invite their friends. Once other users join their group they can send expenses to the group that are automatically divided and sent to each group member's individual account where they have the option to pay. We roped all our features besides register, log in, and log out onto the home page as it acts as the central hub for the user to see their account information. On the home page each user can see their transaction history as well as their current balance and group information. This is where group expense requests will appear and users will have the option to pay expenses, as well as deposit and withdraw from their Expense Share account. The application will also send users emails regarding their account. Our application features a receipt scanning API where users are able to upload photos of receipts to be automatically scanned and placed in the transactions section of the home page. We used Postgres running SQL and NodeJS to run the site.

**Project Tracker:**
https://github.com/users/averywagnerr/projects/1/views/2?filterQuery=




**Video:**
**https://drive.google.com/file/d/1XBPjqVOxjZeMor-wwyqeF6HkdDQuR5rl/view?usp=sharing**

**VCS:**
**https://github.com/averywagnerr/ExpenseShare.git**

**Contributions:**

Connor Julson:
I initially helped get the docker container setup and running, including configuration of the .env and docker-compose files. Additionally, I designed the login and register pages which were implemented using html, css and nodejs. Finally, I designed and implemented the API route for the deposit feature using NodeJS.

Tyler Haskins:
The first thing I did was make the original wireframes for our site. Also, I designed the UI for our home page and landing page including making custom css. I implemented the css framework Tailwind which could be used across our

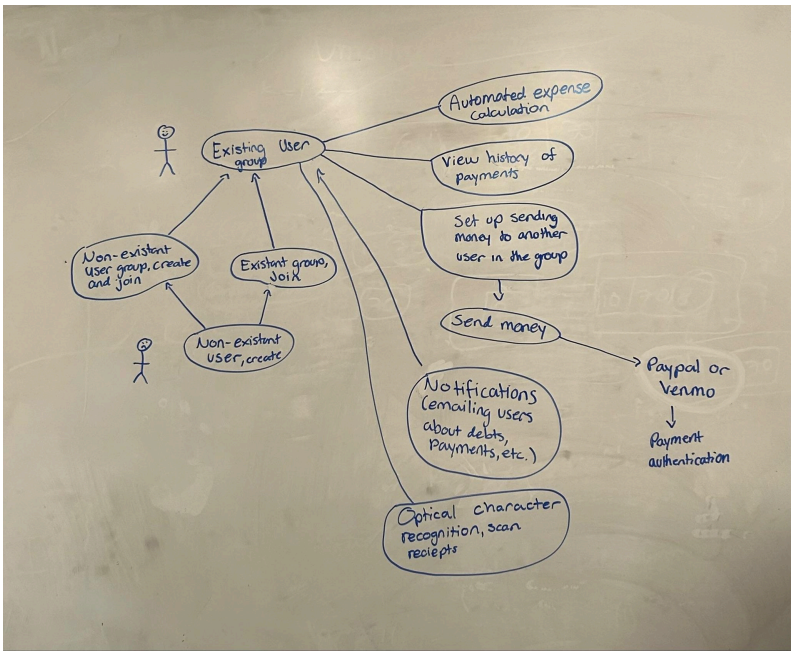
whole site. Using Tailwind I implemented a dark mode to our site. For dark mode it will automatically detect your system preference but you can also change it manually. Finally, I did all of the deployment of our website using Microsoft Azure.

Avery Wagner:
Initially, I got the team's repository up and running and created the GitHub projects board that we would use for SCRUM planning, as well as helping to set up our docker file and readme file. I designed the ExpenseShare logo and helped style the navbar, landing page, and the home page. I also helped to make sure password encryption was implemented for user login, and created the group routes. For the login, register, and group routes, I implemented unit tests.

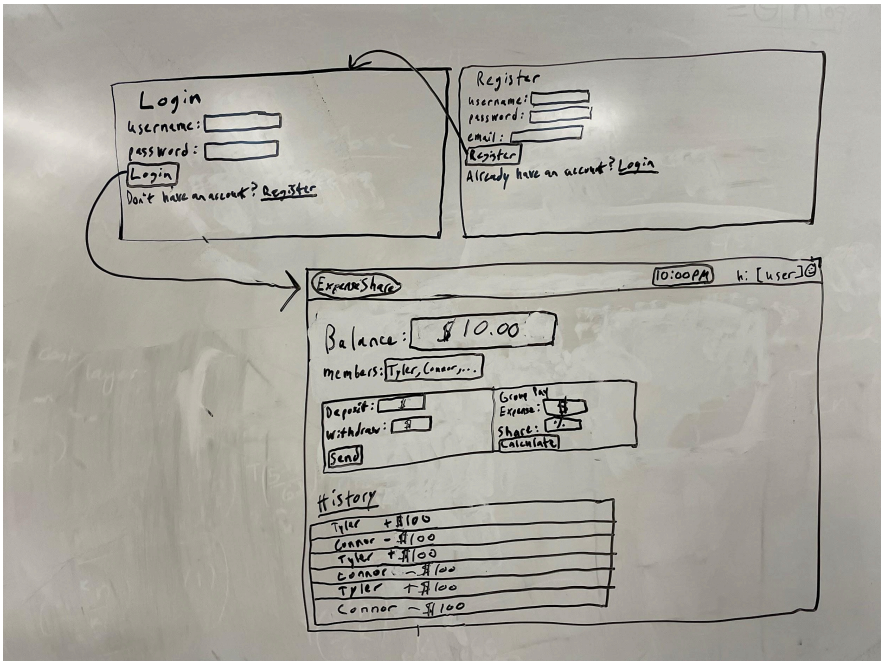
Mason Bott:
I was the database engineer behind ExpenseShare with main contributions regarding the flow of data around the application. With the database in place, I made further contributions regarding data transfer when going to different routes in the database. I made the route for transactions which involved two many-to-many database tables being processed and many new rows in the database being added depending on the size of the group, among helping with other routes. Finally, I was also involved in UI touch ups and final design changes.

Mariana Vadas-Arendt:
I helped get the index.js started with basic routes. I worked on a notification system which will send the user emails regarding their account. I implemented a calculator that would tell the user how much of an expense they owe. I implemented a feature that allows users to upload files to the website and then have the mindee API scan the file and pull information off of it. I then parsed the output of the API for the relevant information we want, that information is then displayed on the website.

## Use Case Diagram:




## Wireframes:



**Test Results:**

Use Case 1: A user with roommates is able to send their weekly grocery bill to all their roommates collectively instead of individually sending them expense requests.

Use Case 2: A user is seeking an automatic way to scan receipts and upload them as transactions to a finance platform.

Use Case 3: A user who has not registered an ExpenseShare account is able to create an account and join their friends group.

Use Case 4: A user is able to see their constant running balance and history of past group transactions.

**Deployment:**

In order to deploy expense share you must have docker installed. You need to clone the github repository and navigate to the /ExpenseShare/ProjectSourceCode/src directory and create a .env file with the contents:

#database credentials
POSTGRES_USER="postgres" POSTGRES_PASSWORD="password"
POSTGRES_DB="users_db"

#Node vars
SESSION_SECRET="super duper secret!"
API_KEY="45c59448df5500383d09c23125d4f5f7" PASS="yqgz czqm jpcm evaa"

Then run docker-compose up -d and navigate to http://localhost:3000/ in your browser and you will be able to access the program