Expense Share

Mason Bott, Tyler Haskins, Connor Julson, Mariana Vadas-Arendt, and Avery Wagner

What is our application?

This application will help you organize your finances with other people. This application allows users to create groups and then send expenses to each other. You can also calculate potential expenses and see how much you would owe. You can also withdraw and deposit money into it. This application will send notifications regarding your account. Expense share also allows users to upload pictures of receipts to store data from them. This application is aimed at people living with each other because they normally share expense of some sort.

Tools we used

Project Tracker: Github Project Board

Description:

This board helped us keep track of what the other group members were working on and what we still have to work on.

Ranking:

We give this 5 stars because it was easy to use and easy to track others projects.

VCS Repository: Github

Description:

This repository tracked different branches for different features to allow multiple different changes to happen at the same time.

Ranking:

We give this 5 stars because it was easy to use and easy to track others commits and progress.

Tools we used

Database: PostgreSQL

Description:

This is a relational database management system that supports SQL we used to create our database.

Ranking:

We give this 5 stars because it was easy to create a database and manage the database.

IDE: Visual Studio Code

Description:

We used this to type code and debug our project as we worked.

Ranking:

We give this 5 stars because it was easy to use and easy to track our progress and files.

Tools we used

UI: HTML, handlebars, tailwindcss

Description:

We used these tools to make the website look nice and allow the user to interact with the website easily.

Ranking:

We give this 4 stars because it was easy to include these features and they made our website look more professional.

Application server: NodeJS

Description:

We used this to implement the routes of our website and the other tools and API we used.

Ranking:

We give this 4 stars because it was moderately easy to use and we were able to get everything implemented.

Tools we used

Deployment environment: Docker

Description:

We used this to see how our website is coming along before deploying it to the world. We were able to see changes we were making and test features we were implementing.

Ranking:

We give this 5 stars because it was easy to use and access our website on.

External API: Mindee

Description:

We used this API to read files and get data off of a picture that was uploaded by a user.

Ranking:

We give this 5 stars because it was moderately easy to use and we were able to get everything implemented.

Tools we used

Testing tool: Mocha and chai

Description:

We used these tools to make sure our website was working correctly. These applications helped us implement our test cases that was testing our application.

Ranking:

We give this 4 stars because it was moderately easy to write test cases.

Methodology:

This falls under the iterative category.

Additional tools: Nodemailer, SMTP, and multer

Description:

We used nodemailer to send users emails about registering for the application. We used multer to allow users to upload files to the website securely. We used SMTP to send emails over the internet.

Ranking:

We give this 5 stars because it was easy to use.

Methodologies

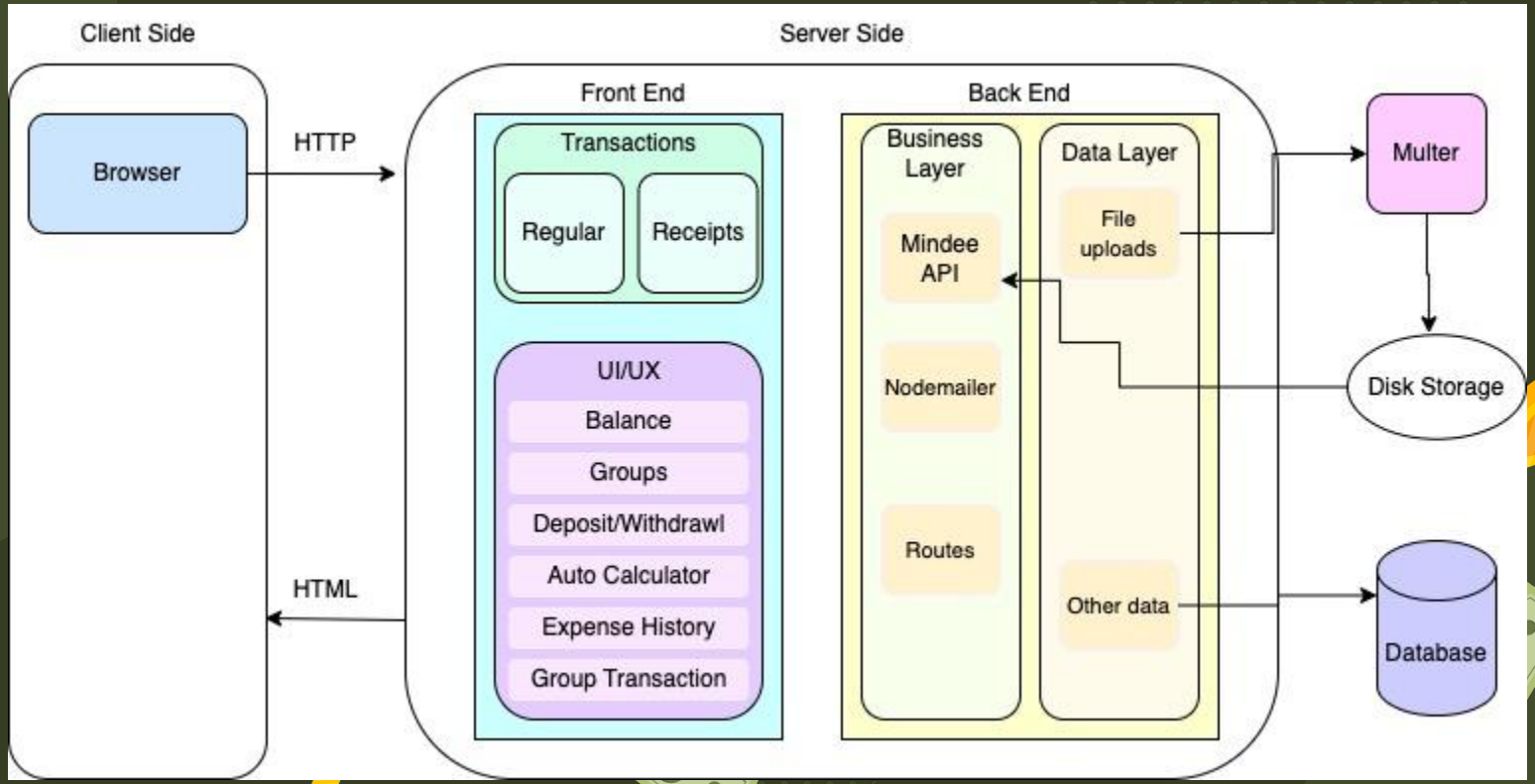
The main methodologies we used were agile, pair programming, and peer code review.

We met weekly for about 1 - 2 hours and discussed what each other was working on and problems we have encountered. During this time, people who were working on the same or similar things would work together on the feature.

We made sure to test our code before deploying it or merging it to the main branch.

We discussed with our group members a lot to make sure the thing we are implementing meets the requirements.

Architecture diagram



Challenges

We all faced some challenges with how to navigate Github, branching, and resolving merge conflicts.

There were challenges implementing the group feature as well. We wanted to implement a secure way for users to join groups using a unique token, but were unable to get this fully functional in time. Our MVP allows users to join by just using a group name.

We also faced a few challenges with the API implementation and the notification system. An API was used to implement the OCR feature. The notification system involved a SMTP to send emails to users over the internet.

Enhancements

01

Add Paypal or Venmo

Adding this API would allow users to seamlessly actually pay the other group members

02

Receipt Transactions to Group Transactions

This would help expedite transaction processes. If a user uploaded a receipt, the total amount would automatically be turned into a group expense.

03

Send daily/weekly/ monthly Expenses

This would ease the user more since they can set up for an expense to be billed every certain time period.

Demo