Payback: The App That Gets Your Money Back

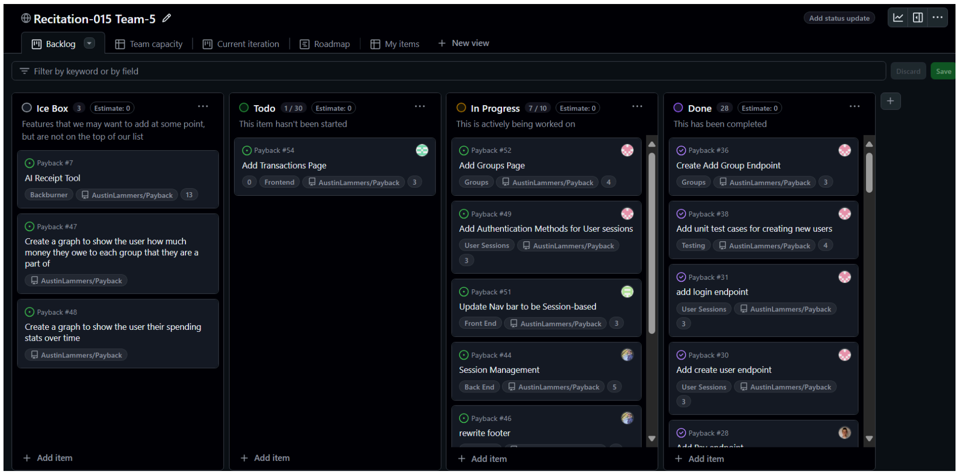
Group Members: Austin Lammers, Gilberto Corral, Alexander Drozdzewicz, Xavier Pena, Shaan Chauhan
Github Usernames: AustinLammers, GCorra19, Alex-Drozdzewicz, xpena10, shaanCh

Description:

Payback is a web application that allows bills to be split amongst groups easily. Our app provides the ability to request money from everyone who is responsible for a bill at once, streamlining the process of manually splitting bills and ensuring everyone sends the money they owe toward that bill. Rent, Dinner, or anything else, our app aims to ensure the person who foots the bill is paid back. In the app, you can create a “group” which contains one or more people who need to contribute to a bill. This group holds a total amount the members must work toward fulfilling. Each member of the group can then log into the app and send payments to this group, chunking away at the total group cost and forwarding the payment to the requesting user. Our app allows you to add friends so you can easily send small one-off payments that may not necessarily need an entire group to facilitate. You are also able to track your incoming and outgoing payments using the profile page for easy accounting of your funds. Get paid back, with Payback.

GitHub Project Tracker:

[Backlog · Recitation-015 Team-5]



Ice Box 3 Estimate: 0
Features that we may want to add at some point, but are not on the top of our list

Payback #7
AI Receipt Tool
Backburner · AustinLammers/Payback · 13

Payback #47
Create a graph to show the user how much money they owe to each group that they are a part of
AustinLammers/Payback

Payback #48
Create a graph to show the user their spending stats over time
AustinLammers/Payback

Add item

Todo 1 / 30 Estimate: 0
This item hasn't been started

Payback #54
Add Transactions Page
0 · Frontend · AustinLammers/Payback · 3

Add item

In Progress 7 / 10 Estimate: 0
This is actively being worked on

Payback #52
Add Groups Page
Groups · AustinLammers/Payback · 4

Payback #49
Add Authentication Methods for User sessions
User Sessions · AustinLammers/Payback · 3

Payback #51
Update Nav bar to be Session-based
Front End · AustinLammers/Payback · 3

Payback #44
Session Management
Back End · AustinLammers/Payback · 5

Payback #46
rewrite footer

Add item

Done 28 Estimate: 0
This has been completed

Payback #36
Create Add Group Endpoint
Groups · AustinLammers/Payback · 3

Payback #38
Add unit test cases for creating new users
Testing · AustinLammers/Payback · 4

Payback #31
add login endpoint
User Sessions · AustinLammers/Payback · 3

Payback #30
Add create user endpoint
User Sessions · AustinLammers/Payback · 3

Payback #28
Add Pay endpoint

Add item

Video:

(Github)
https://github.com/AustinLammers/Payback/blob/main/MilestoneSubmissions/Group%20015-5%20Project%20Demo.mp4

(Google Drive)
https://drive.google.com/file/d/1OWCxtjscwopDDnfLYoQLdc_gwGyHjrhy/view?usp=sharing

VCS Link:

AustinLammers/Payback

Contributions:

Shaan: I played a key role in designing the front-end pages for both the profile and payment pages, leveraging Handlebars, HTML, and CSS to create visually appealing and functional interfaces. Additionally, I contributed significantly to the overall design of various other front-end pages, ensuring a consistent and user-friendly experience across the platform.

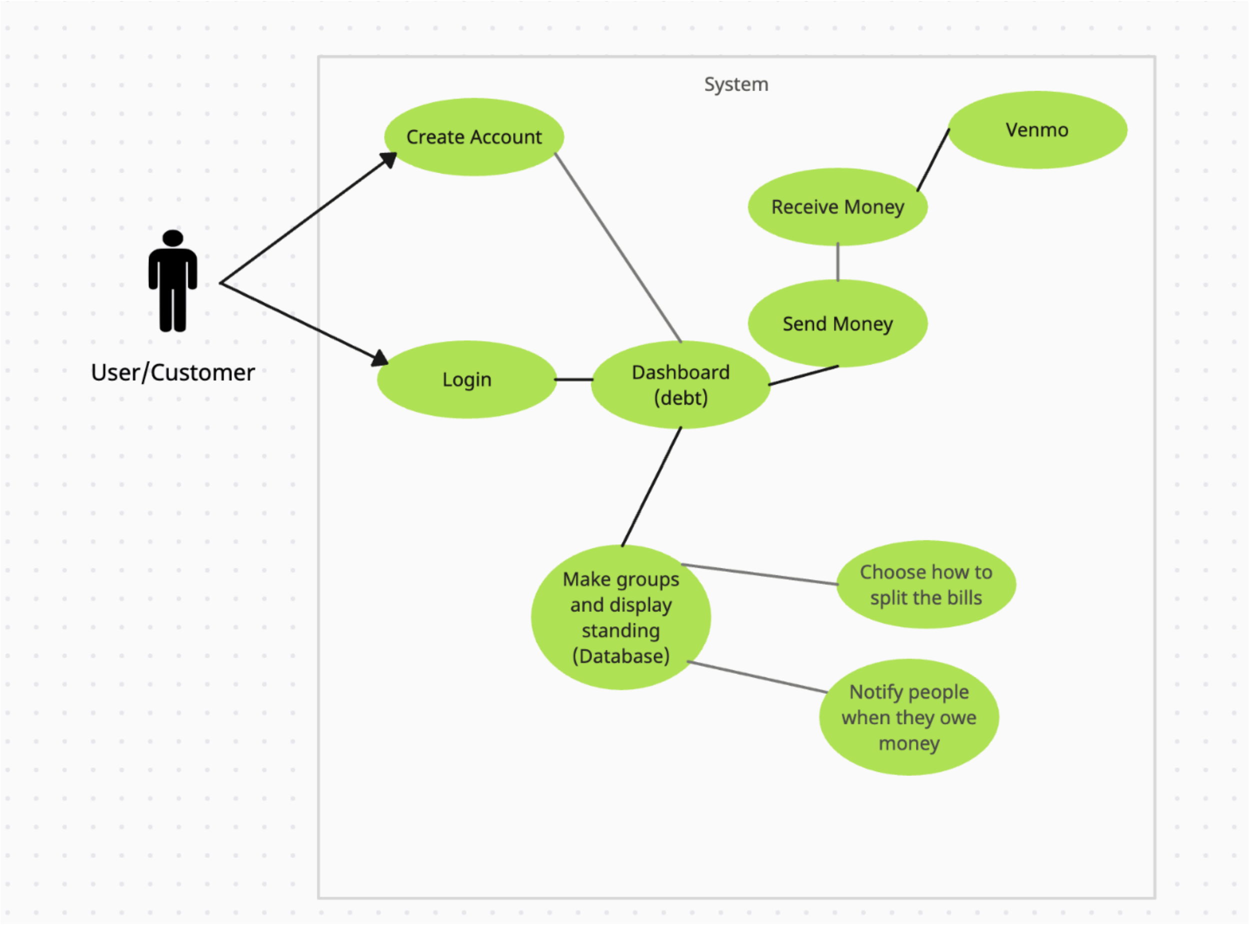
Xavier: The majority of my contributions were contained within front-end handlebars for the home page, footer, and navigation bar. I also made a few function calls within these pages so I could display relevant group information for the user, such as the total amount owed by the group and those who were assigned the same group as the user. I also helped to troubleshoot session management issues especially with the nav bar.

Gilberto: I helped construct part of the database by creating the tables for users, groups, users_to_groups, and expenses. I also designed the groups page and connected it with the database and the API routes for the /createGroup POST API and /groups GET API, which were API routes that I also created and refined. I then created two unit test cases for the createGroup API route to ensure its proper functionality. This successfully allowed for our applications ability to successfully map users to groups that they had been added to and retain that information in memory through the database for each user's profile. This also allowed for users to create new groups, update current groups, and display each group they were members of as well as how much of the total balance for the group was remaining. I also helped build our project's initial wireframes.

Alexander: I assisted with optimizing the front end by converting HTML into handlebars for the home page, footer, nav bar, and other pages. I also used CSS, HTML, and Bootstrap to design the page and assisted with session management on the backend as well as some fixes.

Austin: My contributions for this project consisted mostly of backend work with PostreSQL, Express, pgpromise, NodeJS, and Handlebars. I developed many of the endpoints of our app that related to the friends and payments page. This includes the endpoints which pull the information about friends/groups from the database and render the page with it as well as the endpoints which handle the addition of friends and payments to the database. I also assisted with the group listing functionality of our groups page as well as the linkage of the profile page. I also made contributions to the database layout

Use Case Diagram:


System
Create Account
Venmo
Receive Money
User/Customer
Send Money
Login
Dashboard (debt)
Make groups and display standing (Database)
Choose how to split the bills
Notify people when they owe money

Wireframes:

https://docs.google.com/presentation/d/1RIjoe3C9u9Nr1-9aNQN5ll0lS0Qt2BV7HtFM301egvk/edit?usp=sharing

Test Results:

For all tests, the server was hosted on Render (https://payback-p4gt.onrender.com)

1.) User should be able to request payment/create a group

- **Acceptance Criteria**
    - The user was able to navigate to the groups page and create a group asking for a payment
    - The database properly stored this group and the users added to the group are also able to see this group
- **Test Results**
    - The users had initially had problems finding how to request a payment
    - After a quick pointer, they navigated to the groups page and they were able to easily add a group with a list of various given users
- **User Acceptance Testers Information**
    - "Jonathan Eldar" <Jonathan.Eldar@colorado.edu>
    - "Aidan Rowlett" <Aidan.Rowlett@colorado.edu>
    - "Manogya Thapa" <Manogya.Thapa@colorado.edu>

2.) The user should be able to add a friend and see a list of friends on the friends list

- **Acceptance Criteria**
    - The user was able to navigate to the friends page
    - The user was able to add a friend and the database added it
    - The user should be able to see all previously added friends
- **Test Results**
    - All of the users immediately navigated to the friends page through the navigation bar
    - Each user was able to add at least three friends
- **User Acceptance Testers Information**
    - "Jonathan Eldar" <Jonathan.Eldar@colorado.edu>
    - "Aidan Rowlett" <Aidan.Rowlett@colorado.edu>
    - "Manogya Thapa" <Manogya.Thapa@colorado.edu>

3.) User should be able to find where their previous transactions are and from where

- **Acceptance Criteria**
    - The user should be able to go to their profile page where they can find previous transactions
    - The page successfully pulls the user's previous transactions
- **Test Results**
    - A few of the testers went to the payment page, where they unsuccessfully didn't find previous transactions
    - After a little bit of exploration, they found the information listed on the profile page
- **User Acceptance Testers Information**
    - "Jonathan Eldar" <Jonathan.Eldar@colorado.edu>
    - "Aidan Rowlett" <Aidan.Rowlett@colorado.edu>

        - "Manogya Thapa" <Manogya.Thapa@colorado.edu>

4.) The user should be able to find other users who are also added to the same group as them

- **Acceptance Criteria**
    - The user was able to navigate the home page and find the users of a specified group
    - The data is pulled from the database and displayed to the user
- **Test Results**
    - After adding the users to various groups from alternative user accounts with various other users, all of the testers were able to navigate on the home page and select a group that displayed all of the users within their groups
- **User Acceptance Testers Information**
    - "Jonathan Eldar" <Jonathan.Eldar@colorado.edu>
    - "Aidan Rowlett" <Aidan.Rowlett@colorado.edu>
    - "Manogya Thapa" <Manogya.Thapa@colorado.edu>

Deployment:

Payback is hosted using Render: **Link to Payback**