# CS 303 Final Project Design Doc

# **PAYPAY Bill Split Web Application**

# Project Members:

Haozhe Dong, Xinyao Feng, Shutao He, Ruiling Qian, Yusen Wang

# **1. Use Case & Interaction Design**

1.1 User Roles

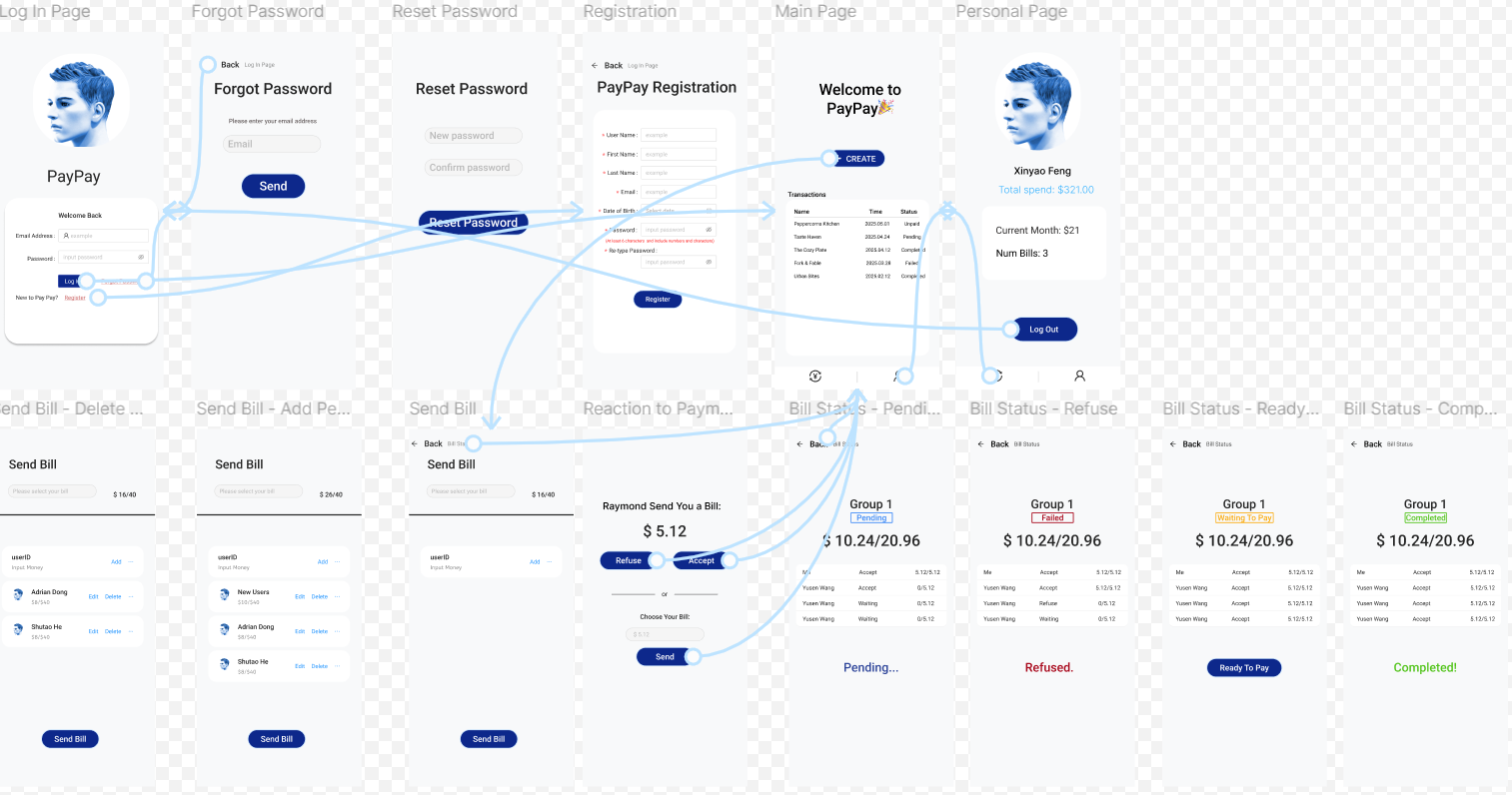
* Group Leader: Create a group payment session and initiate the split bill.
* Group Member: Join group and accept payment and group.

# **2. Technology Stack**

| Layers | Frameworks | Reasons |
| --- | --- | --- |
| Frontend | Next.js + Ant Design + Vercel | Nextjs learned in the class, and it is a good time to practice. Ant Design is a UI library constructed by Ant Group Inc., which is widely used in many big companies. Learning Ant Design is a good practice to prepare for internships. Vercel is a tool to make our frontend become public, allowing the web services be accessed from any devices. |
| Backend | Django + AWS Lambda / API | Django provides built-in features like auth and ORM. AWS Lambda offers scalable, serverless deployment with low maintenance. |
| Database | AWS RDS + mySQL + S3 | AWS RDS provides fully managed database services, supporting high availability, automatic backups, and elastic scaling. It reduces the complexity of database management, making it suitable for rapid development and deployment.  AWS S3 provides an economical and efficient file storage service that can be used to store billing attachments, user avatars, or other uploaded files, helping to reduce database load and improve performance and scalability. |
| Reatime Library | Web Sockets | Enables low-latency, real-time communication needed for agreement tracking without polling. |

### **Realtime Design:** WebSockets let the frontend and backend maintain a persistent connection, allowing two-way message exchange without the need to send an HTTP request each time. Use Websockets to realize realtime payment, a WebSocket connection is established when a user enters a session, when user agree to pay, the client sends a message like: { "type": "agree", "user\_id": "u123" }, and the server would updates agreement status in the database and broadcasts to all clients, frontend then updates UI with current agreement status.If all members agree, server will sends like: { "type": "payment\_ready" }, and all user will change to payment\_ready status.

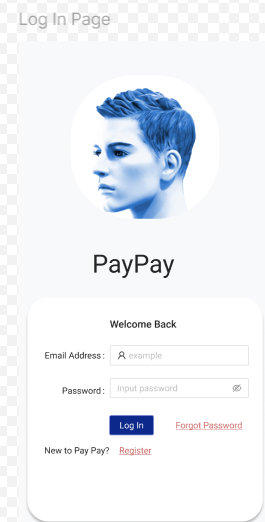
# **3. Figma Design of All Pages**



Figma View Only Link: <https://www.figma.com/design/nIQCqkuaVIuUGmdLOz8lWO/Bill-Split?node-id=0-1&t=avVeBTixIv7pkLLS-1>

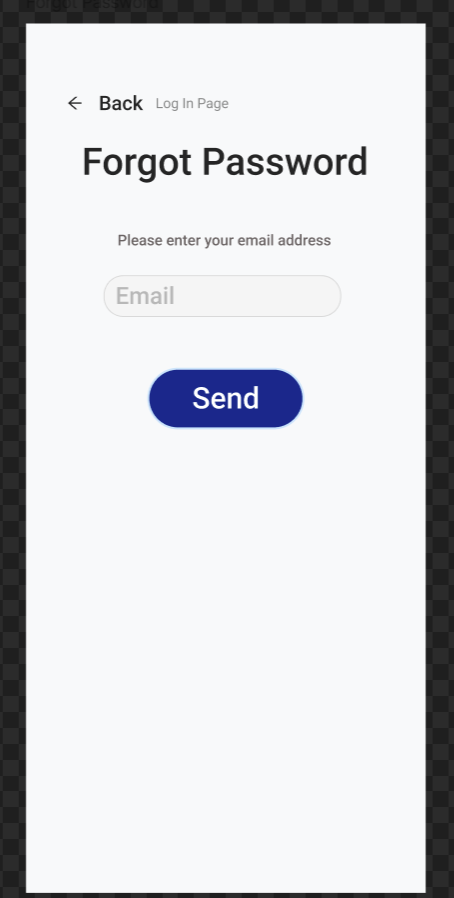
# **4. Detailed Description of Functionalities in Each Page**

**4.1. Login Page**



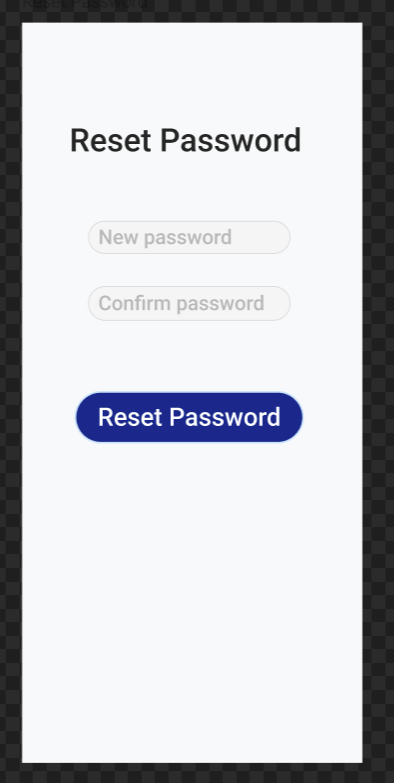
This page includes fields for the user’s email address and password to log in. When the user enters their credentials, the frontend sends an HTTP POST request to the backend to verify whether the email and password match the records in the database. If they match, the user is redirected to the Main Page, as shown in the image above. If the credentials do not match, an alert will notify the user that the information entered is incorrect. There are also two links, “Forgot Password” and “Register,” which navigate to separate pages for password recovery and account registration.

**4.2. Forgot Password**



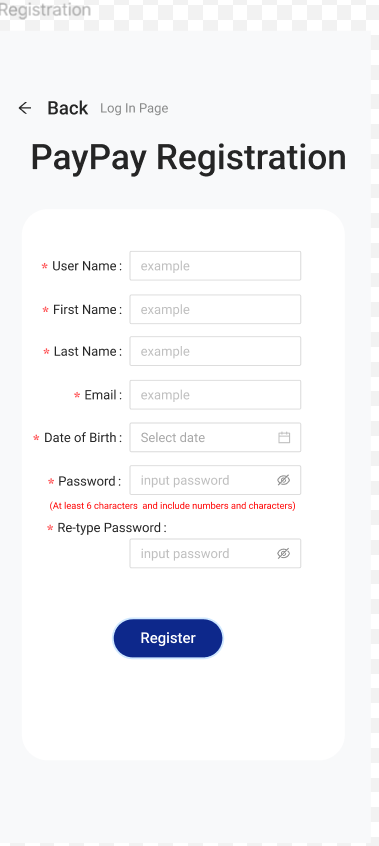
When a user forgets their password, they can enter their email address, and a password reset link will be sent to that email. By clicking the link, the user will be redirected to the Reset Password page

**4.3. Reset Password**



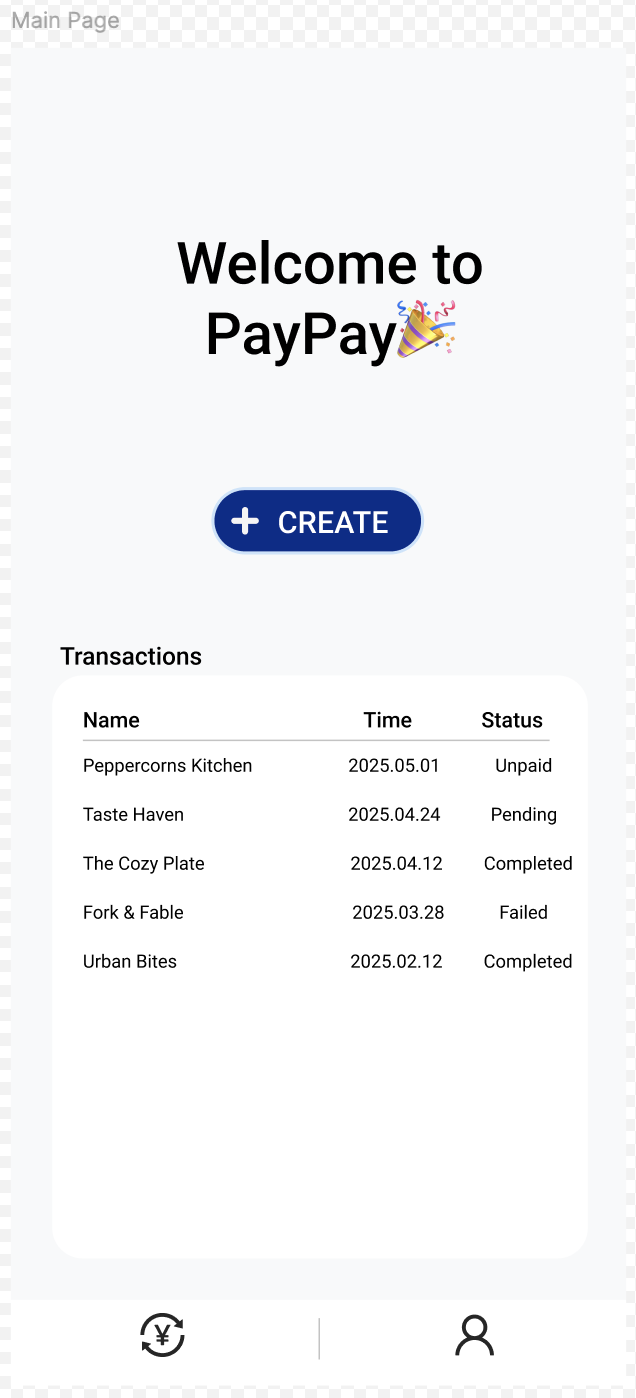
When the user clicks the reset password link from their email, they will be redirected to the Reset Password page. On this page, the user will enter a new password and confirm it. Once submitted, the system will validate and update the password, then redirect the user to the login page with a success message.

**4.4. Registration**



This is a registration page, users will input their information and that information will be entered into our database. We will do a username verification before entering into the database because we need to make sure that usernames are unique. Otherwise, there will be an alert message for the user to input again. After user successfully registered, they will be automatically entered into log in age to log in using its email address and password (Password should be at least 6 characters long and include numbers and English characters).

**4.5. Main Page**

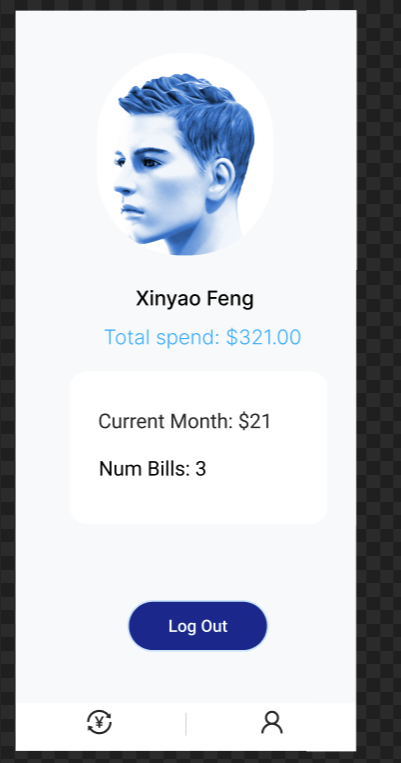


This is the main page, where users can navigate to key functions. The top of the page creates a friendly atmosphere with a welcome message and the brand name "PayPay," enhancing user familiarity. The central "+ CREATE" button is prominently centered, allowing users to quickly create new splits with the shortest operation path. The "Transactions" area below displays the user's historical transaction records, categorized by bill name, time, and status, making it easy to browse quickly and check statuses. Users can click on a bill to enter the detailed bill information interface.

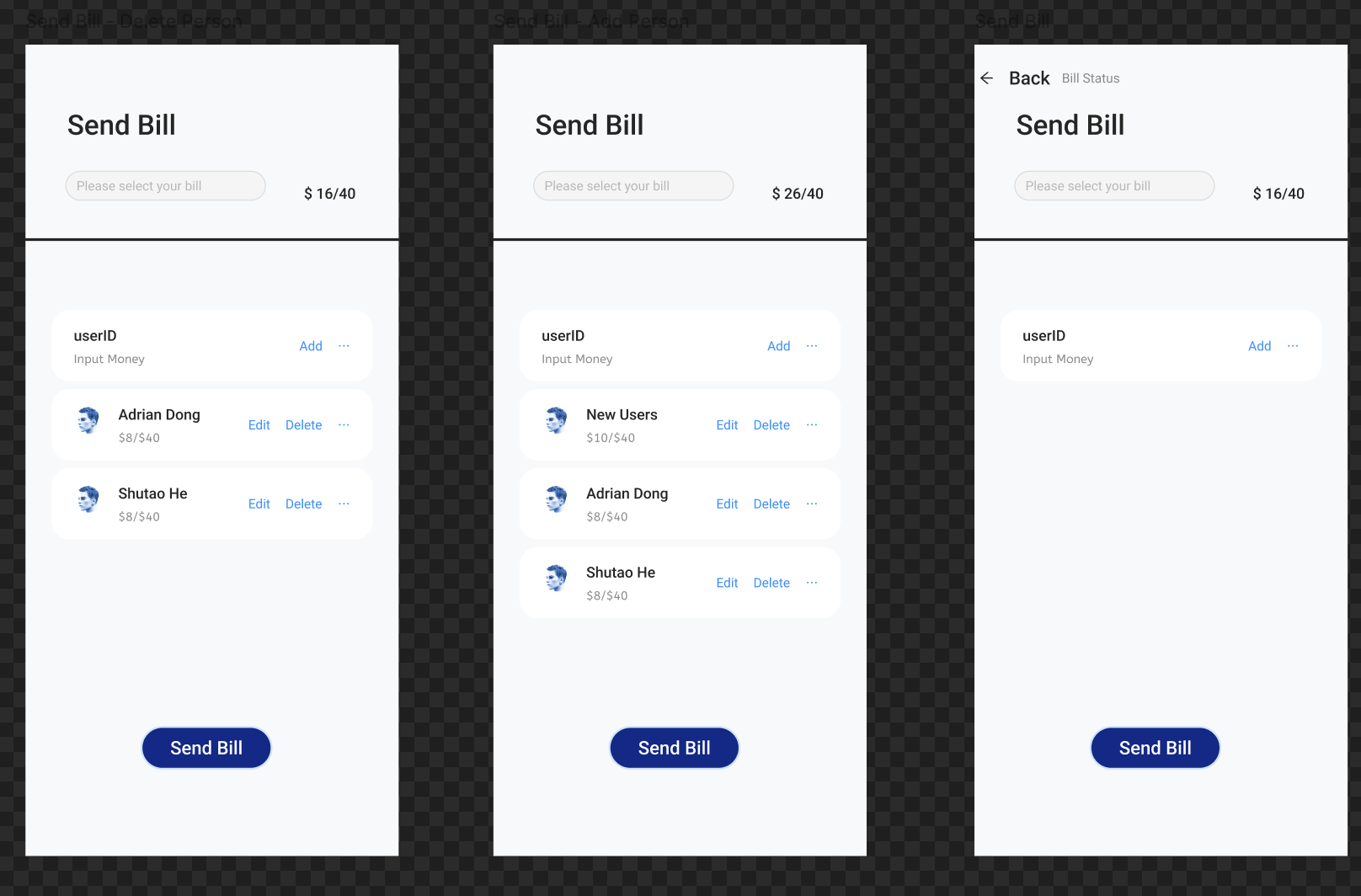
The status of the bill includes four types: "Unpaid," "Pending," "Completed," and "Failed." "Unpaid" means the user needs to make a payment for this bill, and clicking on the bill will redirect to the "Reaction To Payment" interface. "Pending" means the user has completed their part of the payment and is waiting for other users in this bill to complete their payments. "Completed" means this bill has been completed, and all users have made their payments. "Failed" means that a user has rejected the splitting terms in this bill. In Transactions, for bills with the statuses "Pending," "Completed," and "Failed," clicking will enter the "Bill Status" interface.

The two icons at the bottom of the page represent "Main page" and "Personal Page," and clicking them will navigate to the respective pages.

**4.6. Personal Page**

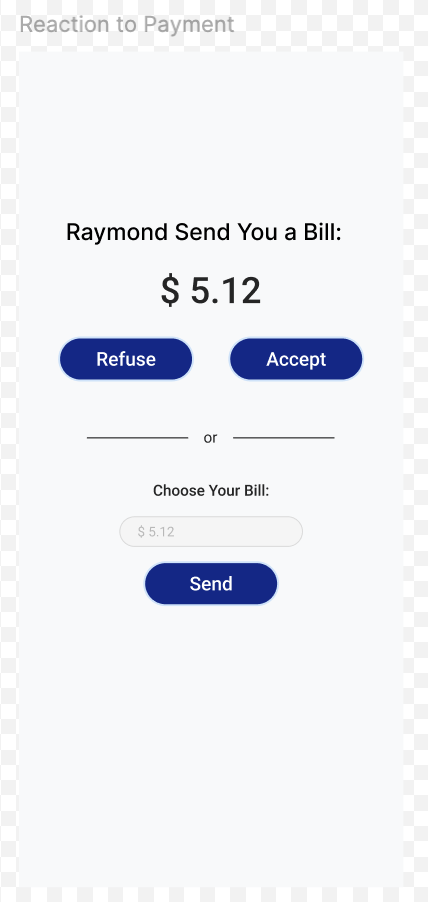
****

On this page, the user will see the total amount spent in the current month and the number of orders they have placed. The page also contains a Log Out button. When clicked, the user will be logged out and redirected to the login page.

**4.7. Send Bill (3 Different Pages)**

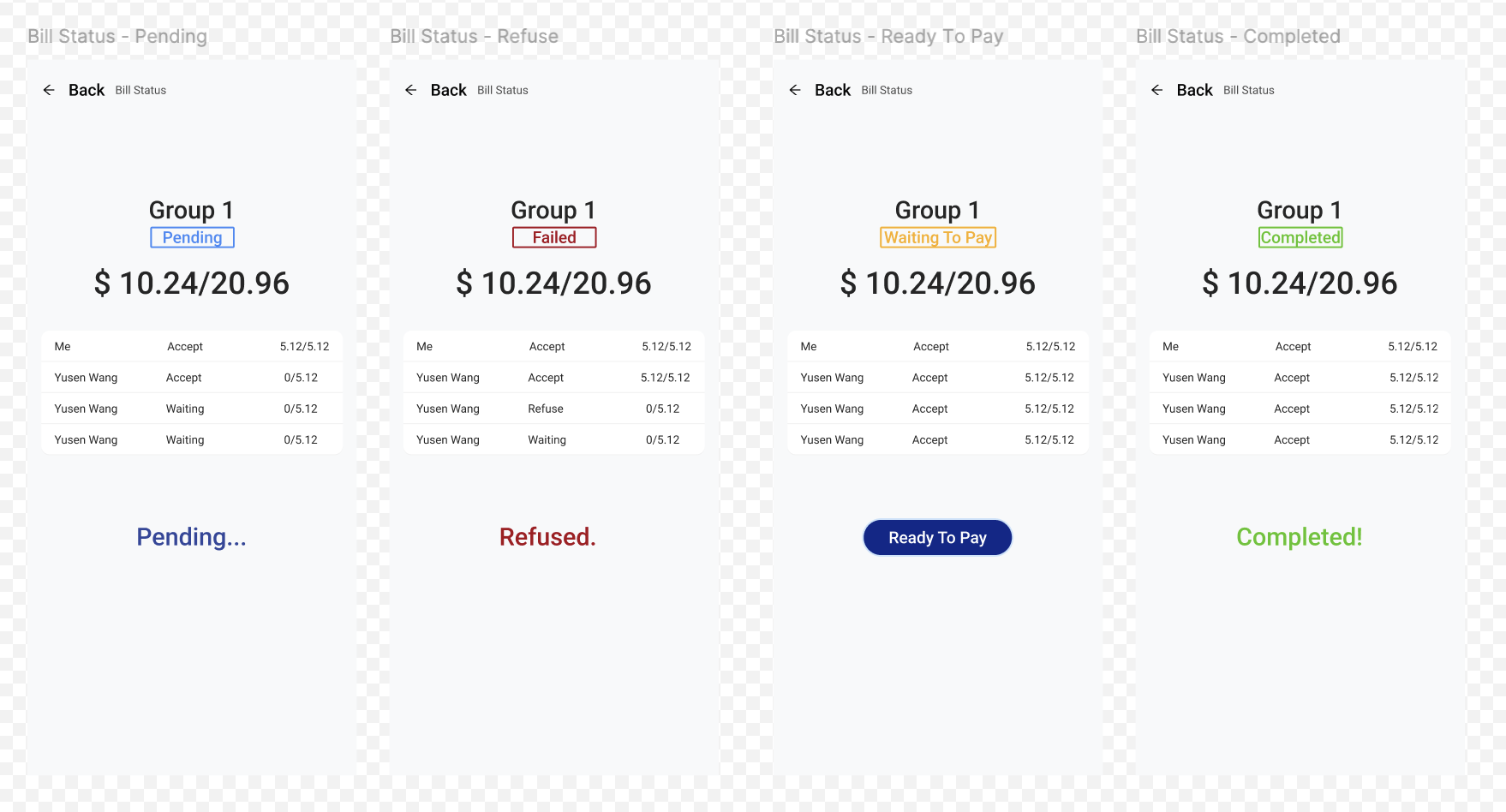
On this page, users can send bills to everyone in their group. They can add a new recipient by entering the user's ID and specifying the amount to be paid. After adding all recipients, users can send the bill to everyone in the group. Additionally, users can easily edit or remove existing group members by clicking the "Edit" or "Delete" button.

**4.8. Reaction To Payment**



When the user is added to the group, he/she receives the payment page. The user can know the person who sent the amount and the amount to be paid. The user can choose to accept the transaction or refuse it, or choose a payment amount that is acceptable to the user and send it back to show to the rest of the group.

**4.9. Bill Status (4 Different Pages)**



When the group is created, it will enter each of the four states and two different processes in the diagram:

1. pending->Failed: When someone in the group declines the payment, or chooses to pay another amount, the transaction changes from pending to Failed status and ends the transaction.

2. Once everyone in the group has accepted the amount, the status will change from Pending to Waiting To Pay, then by clicking on the Ready To Pay button, the platform will collect the amount from everyone and pay it to the merchant. Once the payment is made, the transaction will change from Waiting To Pay to Completed and the transaction will be closed.

The screen in the figure shows the name of the group, the status of the transaction, the amount that has been accepted and the total amount, and a list of the members of the group as well as their status.

# **5. Work Split by Team Members.**

**Haozhe Dong:** Designed the Sent Bill Page, front-end design and code implementations of all pages. Draw the use case diagram

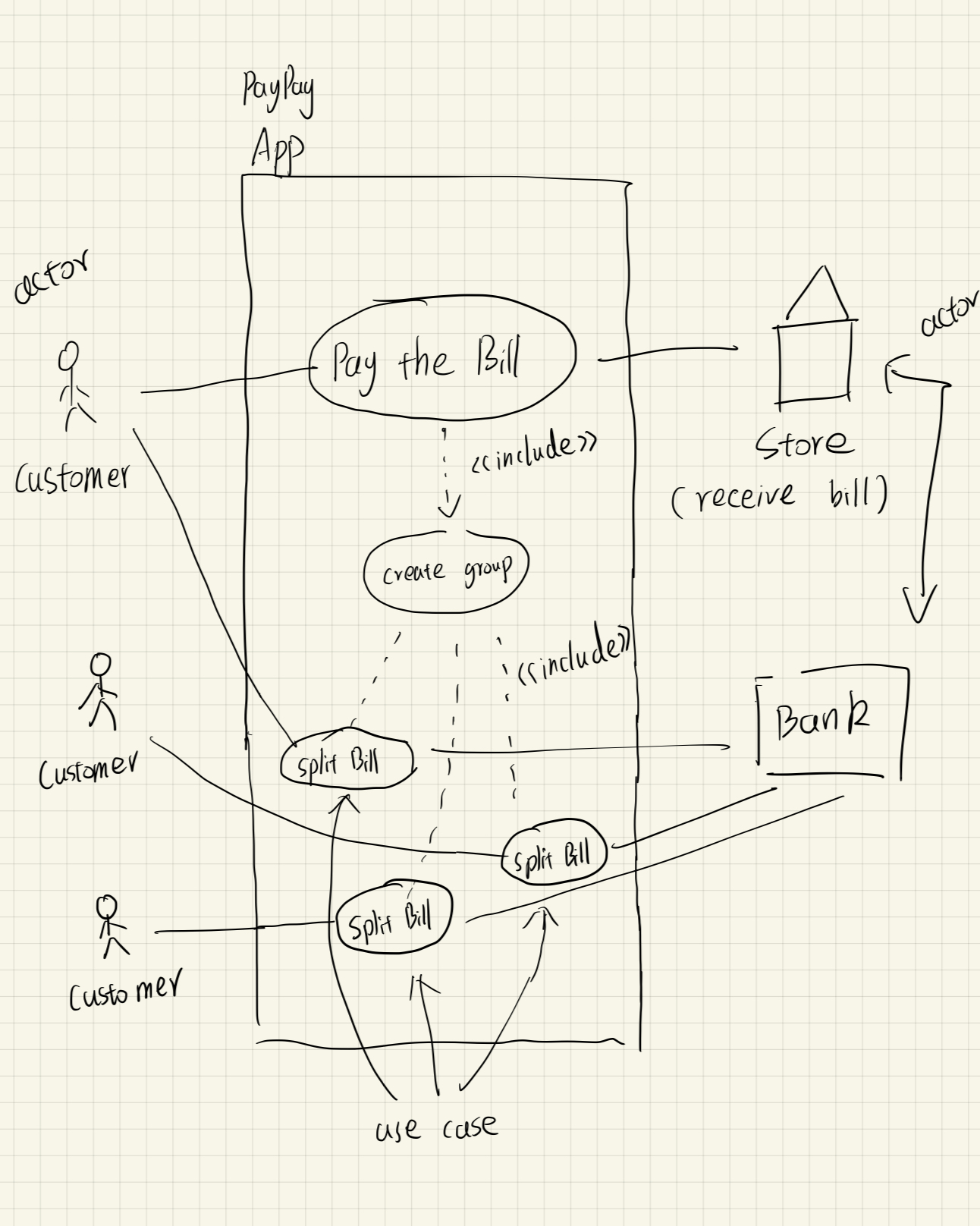
**Xinyao Feng:** Designed Personal Center, forgot password and reset password page, the back-end logic design and code implementation.

**Shutao He:** Designed the payment flow, bill status and payment pages, the back-end logic design and code implementation.

**Ruiling Qian:** Design the Figma for the login page and registration page. Also, the frontend code implementations of all pages in this project

**Yusen Wang:** Designed the Main page, the back-end logic design and code implementation.

# **6. Use Case Diagram**



# **7. Probably Future Implementations**

1. **Get users' bank information**

In our bill-splitting app, we might use **Plaid** to securely connect users' bank accounts without collecting sensitive login information ourselves. When a user wants to link their bank, they go through Plaid’s interface to log in, and we receive a secure token that allows us to access their basic account details, such as account and routing numbers, and verify their balance. This data is then used to facilitate payments through a third-party processor like Stripe or Dwolla, enabling seamless and secure money transfers between users.

1. **Scannable QR code for group session**

Instead of adding new users by entering their usernames, we might design a scannable QR code that others can scan to join the group.