UNIPAY DOCS

**🔐 1. Authentication Flow**

**Pages:**

* **Login Page**
* **Sign Up Page**
* **Forgot Password (optional)**

**Flow:**

* User signs in or registers.
* On successful login, redirect to **Dashboard/Home Page**.

**🏠 2. Homepage / Dashboard**

**Interface: /dashboard**

Shows 2 clear CTAs:

* ➕ Create New Payment Stream
* 🔗 Join Existing Payment Stream

**🔁 3. Create New Payment Stream**

**Interface: /payment-stream/create**

**Input fields:**

* Description (optional)
* Amount per payment (Float)

**Flow:**

* User fills and submits form.
* Unique code is generated.
* User is redirected to that stream’s homepage: /payment-stream/:code

**🔗 4. Join Existing Payment Stream**

**Interface: Dialog or /payment-stream/join**

**Input:**

* Stream Code

**Flow:**

* Validate stream code from paymentStream table.
  + ✅ Valid: Create userPaymentStreams & streamMembers entry → Redirect to /payment-stream/:code
  + ❌ Invalid: Show error and stay on modal/dialog

**📄 5. Payment Stream Homepage**

**Interface: /payment-stream/:code**

This is the central hub for a particular stream.

**Options:**

* 👁 View Payment Stream Details
* 💸 Make a Payment

**📊 6. View Payment Stream Details**

**Interface: /payment-stream/:code/details**

**Display:**

* Description
* Amount per payment
* Creator info
* Members (streamMembers)
* Payment history (payments)
* Dues & their status (dues)

**💵 7. Make Payment**

**Interface: /payment-stream/:code/pay**

**Input:**

* Payment method
* Note (optional)

**Flow:**

* Submit payment
* Create entry in payments with:
  + paid\_at, reference, amount, user\_id, etc.
* Mark related dues as paid (paid = true)
* Redirect to payment success screen or back to stream homepage

**📬 8. Optional Flows**

**a. Refund Request/Management**

* Display refund status in /payment-stream/:code/payments
* Option to request refund (if allowed)
* Admin/creator processes it and fills refunds table

**b. Leave a Stream**

* Button to remove user from userPaymentStreams and streamMembers

**c. Admin Panel (if applicable)**

* Stream creator can view:
  + All members
  + Total paid
  + Dues outstanding
  + Trigger refunds manually

**📱 Interfaces Summary (UI Pages/Modals)**

| **Route** | **Purpose** |
| --- | --- |
| /login | User login |
| /register | New user signup |
| /dashboard | Create/Join stream |
| /payment-stream/create | New stream creation |
| /payment-stream/join | Join with code |
| /payment-stream/:code | Stream home |
| /payment-stream/:code/details | Stream details |
| /payment-stream/:code/pay | Make payment |
| /payment-stream/:code/payments | View all payments (optional) |
| /payment-success | Payment confirmation (optional) |

**🧠 Backend Actions (API Endpoints you'll need)**

* POST /api/auth/login
* POST /api/auth/register
* POST /api/payment-stream/create
* POST /api/payment-stream/join
* GET /api/payment-stream/:code
* POST /api/payment-stream/:code/pay
* GET /api/payment-stream/:code/details
* GET /api/payment-stream/:code/payments
* POST /api/payment-stream/:code/refund

Resources I used:  
- [Setting Up Clerk in Nest.Js](https://dev.to/thedammyking/authentication-with-clerk-in-nestjs-server-application-gpm)