Bondr - User Stories

Part A: User Stories & On-Chain Requirements Document

- ### Core User Personas

1. **Migrant Workers (Senders)**
   Individuals working abroad who regularly send money back home to family/friends.

2. **Freelancers / Remote Workers**
   Self-employed professionals receiving payments from global clients; likely to send/receive frequently and in SPL tokens.

3. **Family Members / Recipients (Claimers)**
   Individuals receiving funds via QR. May or may not be familiar with crypto.

4. **Power Users / Loyalty Tier Users** *(Subset of senders)*
   High-frequency users who unlock NFT rewards for continued platform usage.

- ### Function Maps

1. **Migrant Workers / Freelancers (Senders)**

   Connect crypto wallet (to start the remittance process)

   Input remittance amount (and optionally receiver's address)

   Generate a remittance QR code (Solana Pay format)

   Fund the remittance (via SOL or SPL token transfer to PDA)

   Receive error messages (invalid input, reused reference seed, insufficient funds)

   Share the QR externally (email, WhatsApp, etc.)

Earn loyalty NFTs upon hitting milestones (3rd, 10th, 25th send)

2. **Family Members / Recipients (Claimers)**

   Scan QR to initiate claim process

   Connect wallet to verify identity

   Receive SOL/SPL in their wallet

   See confirmation message after successful claim

   Receive error feedback if QR is expired, claimed, or invalid

3. **Power Users / Loyalty Tier Users**

   Track milestone progress (e.g., 2/3 remittances sent)

   View and manage earned NFTs (gamified UX)

   Reuse saved receiver wallets from address book (future enhancement)

- ## Potential On Chain Requirements

1. Generate a QR code for the payment
   Must deterministically derive a PDA (remittance account) using:

   [b"remittance", sender, receiver, reference_seed]

   Validate that PDA doesn’t already exist.

   Must emit an event with QR metadata (sender, receiver, amount, ref_seed) for off-chain QR generation.

   Reference seed validation (!= 0, <= 100, prevent collisions)

2. Send the payment to Bondr escrow
   Create and initialize a Remittance account.

   Validate transfer amount > 0.

Accept payment (SOL or SPL token) into PDA-owned escrow:
If SOL: use system_program::transfer
If SPL: CPI call to token::transfer_checked

Support ATA creation for SPL transfers.

Store:

sender, receiver, amount, bump, token_mint (optional)

3. Get alerts for invalid entries or issues

   Validate:

   - amount > 0
   - reference_seed is within safe range
   - sender != receiver
   - Sender has enough balance (implicit via transfer failure)

   Return custom error codes

4. Scan the QR code to start receiving

   Decode QR off-chain into:
   - reference_seed
   - sender (for PDA derivation)
   - Possibly amount or mint

   Use it to derive correct remittance PDA to claim from.

5. See confirmation after claiming

   Emit success event on claim:

   - Include sender, receiver, amount, timestamp

   Or enable frontend to decode transaction metadata and log.

Part B: Process Appendix

- Part A: Initial User & Function Mapping
    - **AI Prompt:** My project's value proposition is -
      Frictionless Remittance via QR Codes
      Bondr simplifies cross-border payments by enabling users to send crypto through Solana Pay QR codes—eliminating the need to know or enter the recipient's wallet address.

      Beginner-Friendly UX for Crypto Newcomers
      The app is designed to be accessible to users unfamiliar with crypto, with guided onboarding and a first-time user flow that explains the QR-based payment process.

      Fast, Low-Cost, and Secure Transfers
      By leveraging Solana Pay, Bondr ensures transactions are processed quickly and affordably, making it ideal for high-frequency and low-value remittance needs.

      Loyalty System for Power Users
      Frequent senders are rewarded with NFTs, which can represent discount tiers or perks—introducing gamified retention mechanics missing in traditional wallets.

      Future-Proof Platform Extensibility
      With planned support for gift card purchases and a full browser extension rollout, Bondr is positioned to grow beyond basic remittance into broader peer-to-peer value transfer use cases.

      Here is a brainstormed list of all potential user types:

      Direct Users : The direct users of **Bondr** will be migrant workers wanting to send money back home, freelancers who want a fast, secure and reliable payment option to receive money from client/employer.
      Indirect Users/Beneficiaries : The indirect users can be family members of these workers who will be using the platform to receive the money, many crypto new users, and anyone interested in trying out new technologies.
      Administrators/Moderators : The moderator will be me as a developer for now, till the app is in active development and when launched on devnet , when the project

deploys to the mainnet then the community mods can manage and administer the protocol.
Stakeholders : Currently there are no stakeholders of **Bondr** , but will actively look for them.

Based on the value proposition, which 2-5 of these user types are the most critical to focus on for an initial Proof-of-Concept? For each user you recommend, provide a brief rationale explaining why they are essential for proving the project's core value.

- **Edited Output:** Priority user types for PoC are
  a. Migrant Workers (Sender) -
  Role: Direct User
  Why critical**:**
  They are the **primary target persona** for remittance platforms. Validating a flow where they can easily send money via QR without typing wallet addresses will prove Bondr's core UX value.
  Agree with AI , because this was the first market type which I personally thought.
  b. Freelancers / Remote Workers -
  Role: Direct User
  Why critical:
  They represent **higher-value, frequent senders**. Testing the loyalty NFT mechanics (after 3+ sends) depends on this user type to validate the retention loop Bondr promises.
  Agreed with AI here also, as thought this also
  c. Family Members / Recipients (Receiver) -
  Role: Indirect User
  Why critical:
  They experience the **claiming flow**. Ensuring that scanning a QR and receiving funds is seamless validates the Solana Pay + QR model. Their ease of use is vital for real adoption.
  Kinda disagree with AI here as I don't know how much the families will be able to adopt it.

- Part B: Adversarial Analysis & Granularity Check
  - **AI Prompt:** Review my core user functions/stories -
    Migrant Workers (Senders)
    Primary persona for validating Bondr's core UX + remittance flow

    Freelancers / Remote Workers

Power users to validate high-frequency use and NFT reward loop

Family Members / Recipients (Claimers)

Supportive persona — involved only during claim step

- **Edited AI Output :** updated user stories aligning to these granularities -

  Migrant Workers / Freelancers (Senders)

  "As a sender, I want to connect my wallet so I can authorize transactions securely."

  "As a sender, I want to enter an amount and an optional recipient wallet so I can generate a remittance."

  "As a sender, I want to generate a Solana Pay-compatible QR so the receiver can claim the funds without sharing their address upfront."

  "As a sender, I want to fund the remittance using SOL or SPL tokens so the payment is escrowed until claimed."

  "As a sender, I want to see feedback if I enter an invalid amount, reuse a reference seed, or my balance is insufficient."

  "As a sender, I want to be rewarded with a loyalty NFT when I hit usage milestones (e.g., 3rd transfer)."

  Family Members / Receivers (Claimers)

  "As a receiver, I want to scan a QR code and connect my wallet to claim the remittance sent to me."

  "As a receiver, I want to receive SOL or SPL tokens into my wallet if the remittance is valid."

  "As a receiver, I want to see a clear success message after claiming, including transaction details."

  "As a receiver, I want to be notified if the QR is invalid, expired, or already claimed."

  Granular Technical Requirements:

  Wallet Integration

  - Detect and manage connection status

- Access publicKey and signer via provider

Remittance Initialization

- Anchor program instruction initialize(amount: u64, reference_seed: u8)
- PDA derived via [sender, receiver, reference_seed]

Token Handling

- Support both:
  system_program::transfer for SOL
  token::transfer_checked for SPL tokens
- Pass is_token_transfer boolean, token_mint, and relevant token accounts

Claiming Flow

- Anchor claim() instruction with:
  reference_seed, is_token_transfer, amount, decimals
- Transfers from PDA to receiver
- Closes PDA

Loyalty NFT System

- Stats PDA tracks tx_count per user
- After 3rd, 10th, 25th tx → emit LoyaltyMilestoneEvent
- NFT minting triggered off-chain via webhook/indexer or separate instruction

- Part C: Granularity & Clarity Refinement
  - **Action:** De-jargonify all the above points so that user can understand them easily
  - **Refined Stories** :

Migrant Workers / Freelancers (Senders)

Connect wallet to begin sending money

As a sender, I want to connect my crypto wallet so I can securely begin the remittance process.

Enter payment details

As a sender, I want to enter the amount I wish to send and optionally choose or enter the recipient's wallet address.

Generate a QR code for the payment

As a sender, I want to create a QR code that contains the payment information so I can send it to the recipient.

Send the payment to Bondr escrow

As a sender, I want to send SOL or tokens to a temporary holding account so the receiver can later claim it.

Get alerts for invalid entries or issues

As a sender, I want to be alerted if the amount is too low, I reused a QR code, or I don't have enough funds.

Earn rewards for regular use

As a sender, I want to earn a digital reward when I reach certain sending milestones.

Family Members / Receivers (Claimers)

Scan the QR code to start receiving

As a receiver, I want to scan a QR code sent to me so I can begin the process to claim the money.

Connect wallet to claim funds

As a receiver, I want to connect my wallet to prove that I am the intended recipient.

Receive the money into my wallet

As a receiver, I want to get the funds (SOL or tokens) directly into my wallet once I claim them.

See confirmation after claiming

As a receiver, I want to see a success message showing that the transfer was completed.

Be notified if something is wrong

As a receiver, I want to know if the QR has expired, is already used, or is invalid.

Part D: Defining Potential On-Chain Requirements

Migrant Workers / Freelancers (Senders)

User Story: Enter payment details

On-chain Requirements:

- None directly, but:

    - Amount must be passed to the contract method at runtime.
    - Receiver wallet address must be passed (or blank).
    - A reference seed must be provided for PDA derivation.

User Story: Generate a QR code for the payment

On-chain Requirements:

- Must deterministically derive a PDA (remittance account) using:
    - [b"remittance", sender, receiver, reference_seed]
- Validate that PDA doesn’t already exist.
- Must emit an event with QR metadata (sender, receiver, amount, ref_seed) for off-chain QR generation.
- Reference seed validation (!= 0, <= 100, prevent collisions).

User Story: Send the payment to Bondr escrow

On-chain Requirements:

- Create and initialize a Remittance account.
- Validate transfer amount > 0.
- Accept payment (SOL or SPL token) into PDA-owned escrow:
    - If SOL: use system_program::transfer
    - If SPL: CPI call to token::transfer_checked
- Support ATA creation for SPL transfers.
- Store:
    - sender, receiver, amount, bump, token_mint (optional)

Family Members / Receivers (Claimers)

User Story: Scan the QR code to start receiving

On-chain Requirements:

- Decode QR off-chain into:
    - reference_seed
    - sender (for PDA derivation)
    - Possibly amount or mint
- Use it to derive correct remittance PDA to claim from.

User Story: Receive the money into my wallet

On-chain Requirements:

- From remittance PDA:
    - Transfer SOL (via system_program) OR
    - Transfer SPL (via token::transfer_checked)
- Create receiver ATA if needed (for SPL).
- Close remittance PDA (refund rent to receiver).

User Story: Be notified if something is wrong

On-chain Requirements:

- Handle and surface errors:
    - ConstraintSeeds if remittance PDA is incorrect
    - AccountAlreadyInitialized if claimed already
    - ConstraintHasOne if receiver mismatch
    - Missing remittance account → "QR expired/invalid

Part C: Refinement Log

List of all the before and after of the de-jargonify:

1. Before - “Connect wallet via browser extension or dApp”
   After - “Connect my crypto wallet so I can securely begin the remittance process”
   Rationale - Removed technical jargon (e.g., “dApp”), clarified intent.
2. Before - “Enter amount + receiver’s address (or select from contacts)”
   After - “Enter the amount I wish to send and optionally choose or enter the recipient’s wallet address”
   Rationale - Made atomic and de-jargonized.
3. Before - “Generate remittance QR (Solana Pay format)”
   After - “Create a QR code that contains the payment information”
   Rationale - Removed reference to Solana Pay for clarity to non-technical readers.
4. Before - “Fund remittance (SOL/SPL transfer to PDA)”
   After - “Send SOL or tokens to a temporary holding account”
   Rationale - Rephrased “PDA” as “temporary holding account” for accessibility.
5. Before - “See feedback if I enter an invalid amount, reuse a reference seed, or my balance is insufficient”
   After - “Be alerted if the amount is too low, I reused a QR code, or I don’t have enough funds”
   Rationale - De-jargonized and clarified the scenarios.
6. Before - “Be rewarded with a loyalty NFT when I hit usage milestones”
   After - “Earn a digital reward when I reach certain sending milestones”
   Rationale - Removed “NFT” (can be reintroduced in UI); phrased in simple terms.
7. Before - “Scan QR via Bondr app or website”
   After - “Scan a QR code sent to me to begin the process”
   Rationale - Broadened access channel and clarified the flow.
8. Before - “Connect wallet to verify ownership”
   After - “Connect my wallet to prove I am the intended recipient”
   Rationale - Made outcome clearer and reworded passively stated action.
9. Before - “Claim remittance (receive SOL/SPL)”
   After - “Get the funds directly into my wallet once I claim them”
   Rationale - Clearer phrasing; focused on outcome.
10. Before - “See confirmation of receipt”
    After - “See a success message showing that the transfer was completed”
    Rationale - Clarified result of action.
11. Before - “Know if QR is invalid, expired, or already claimed”
    After - “Be notified if the QR has expired, is already used, or is invalid”
    Rationale - Better clarity and user-focused language.