System Requirements for Peer-to-Peer Gifting

1. Problem Description

Traditional gifting methods often result in duplicate gifts, unnecessary spending, and logistical challenges for both guests and event organizers. A **Peer-to-Peer Gifting Platform** is needed to streamline the gifting process by allowing event hosts to create wishlists, enabling guests to contribute towards gifts, and preventing duplicate purchases. The system should support **gift pooling**, **cash gifts**, and **seamless online transactions** while integrating with retailers to facilitate direct purchases.

The **goal of the system** is to simplify the gifting experience, encourage collaborative contributions, and ensure a smooth, transparent process for all users.

2. Actors

1) Event Organizer – Creates the wishlist, manages funds, and receives funds or gifts.

2) Guest – Contributes towards gifts, makes cash donations, comments on media and uploads event media.

3) Affiliate Partner – Provides product information and links for wishlist items.

4) Payment Processor (Stripe Connect) – Handles transactions, gift pooling, and fund disbursement.

5) Administrator – Manages user accounts, platform settings, handling disputes, regulations/bans and monitors transactions.

3. Use Cases

Wishlist Management

1) Create a Wishlist – The event organizer adds gift items to their wishlist from online retailers.

2) Edit or Remove Wishlist Items – The event organizer updates or removes items from the wishlist.

3) Share Wishlist with Guests – The system generates a shareable link or QR code for guests.

Gift Contributions & Transactions

4) Contribute to a Wishlist Item – A guest selects an item and contributes funds toward its purchase.

5) Pool Money for a Gift – Multiple guests can contribute towards a high-value gift.

6) Send a Cash Gift – Guests can send direct cash gifts to the celebrant.

7) Prevent Duplicate Purchases – The system updates the wishlist in real-time to avoid duplicate gifts.

8) Process Payments Securely – Stripe Connect handles transactions for gift contributions and cash gifts.

9) Disburse Funds to Organizer – The system releases funds to the event organizer or retailer.

Affiliate & Retailer Integration

10) Fetch Product Details from Retailers – The system retrieves product information using API integration.

11) Redirect Users to Affiliate Links – The platform earns commission when gifts are purchased through affiliate links.

Event Gallery & Social Features

12) Upload Event Photos/Videos – Guests can upload media from the event to a shared gallery.

13) View & Comment on Event Media – Guests can engage with uploaded photos and videos.

User & System Management

14) User Registration & Authentication – Users can sign up and log in securely.

15) Manage User Roles & Permissions – The administrator manages user access.

16) Secure and Encrypt Transactions – The system ensures data and transaction security.

17) Backup and Store User Data – The system periodically backs up wishlist and transaction data.

18) Monitor Transactions for Fraud – The administrator reviews flagged transactions for security compliance.

Reporting & Notifications

19) Notify Organizer of Contributions – The system alerts organizers when guests contribute gifts or cash.

20) Send Payment Confirmations – The system emails guests receipts after successful transactions.

21) Generate Financial Reports – The system provides event organizers with an overview of received funds.

4. Scenarios

1. Create Wishlist

Actors: Event Organizer
Preconditions:

- The Event Organizer must be logged into the system.
- The Event Organizer must have an event created.

Postconditions:

- A new wishlist is successfully created and associated with the event.

Main Flow:

1. The Event Organizer navigates to the “My Events” section.
2. The Event Organizer selects an event and clicks “Create Wishlist”.
3. The system prompts the organizer to enter a wishlist name.
4. The Event Organizer confirms and clicks “Save”.
5. The system creates an empty wishlist and associates it with the event.

Alternative Flows:

- (A1) The organizer tries to create a wishlist without an event
- The system displays an error: “You must create an event before adding a wishlist.”
- (A2) Organizer cancels wishlist creation
- The system discards any input and returns to the event page.

2. Send Cash Gift

Actors: Guest, Payment Processor (Stripe), Event Organizer
Preconditions:

- The guest has access to the event.
- The system is connected to the payment processor.

Postconditions:

- The cash gift is successfully transferred to the Event Organizer.

Main Flow:

1. The guest selects “Send Cash Gift”.
2. The system prompts the guest to enter the amount.
3. The guest selects a payment method.
4. The system processes the payment.
5. The system transfers the amount to the Event Organizer.
6. The system notifies the Event Organizer of the received cash gift.

Alternative Flows:

- (A1) The payment fails
- The system displays an error message and prompts the guest to try again.

5. Complexity & Justification

- The system involves multiple **actors** with distinct roles, making it a **multi-faceted platform**.
- Covers **wishlist management, payment processing, social interaction, security, and reporting**.
- Requires **real-time updates** to prevent duplicate gifts and ensure transparency.
- Integrates **third-party services (Stripe, Retailer APIs, Affiliate Networks)**, adding technical depth.
- **Security considerations** (fraud detection, data encryption, privacy compliance) enhance system robustness.

Used Tools

- **AI Tools:**

- Chat GPT
    - Prompts:
        - How to resize a box in draw.io without resizing it’s inner contents;
        - Generating classes for Class Diagram (members + methods);
        - Important bullet points for System Requirements
        - Refining the Use Cases for atomic propositions
- PlantUML
    - For creating the Class Diagram
- DiagramGPT
    - Not very useful, in our personal opinion, we used it but the Use Cases were too chaotic, there weren’t generated aspects regarding relations between Use Cases like include or extend, a little bit hard to understand
- draw.io
    - For creating UML Diagram
    - This was done in parallel with the System Requirements bullet points
    - Problems with resizing outer box (where the Use Cases are located)