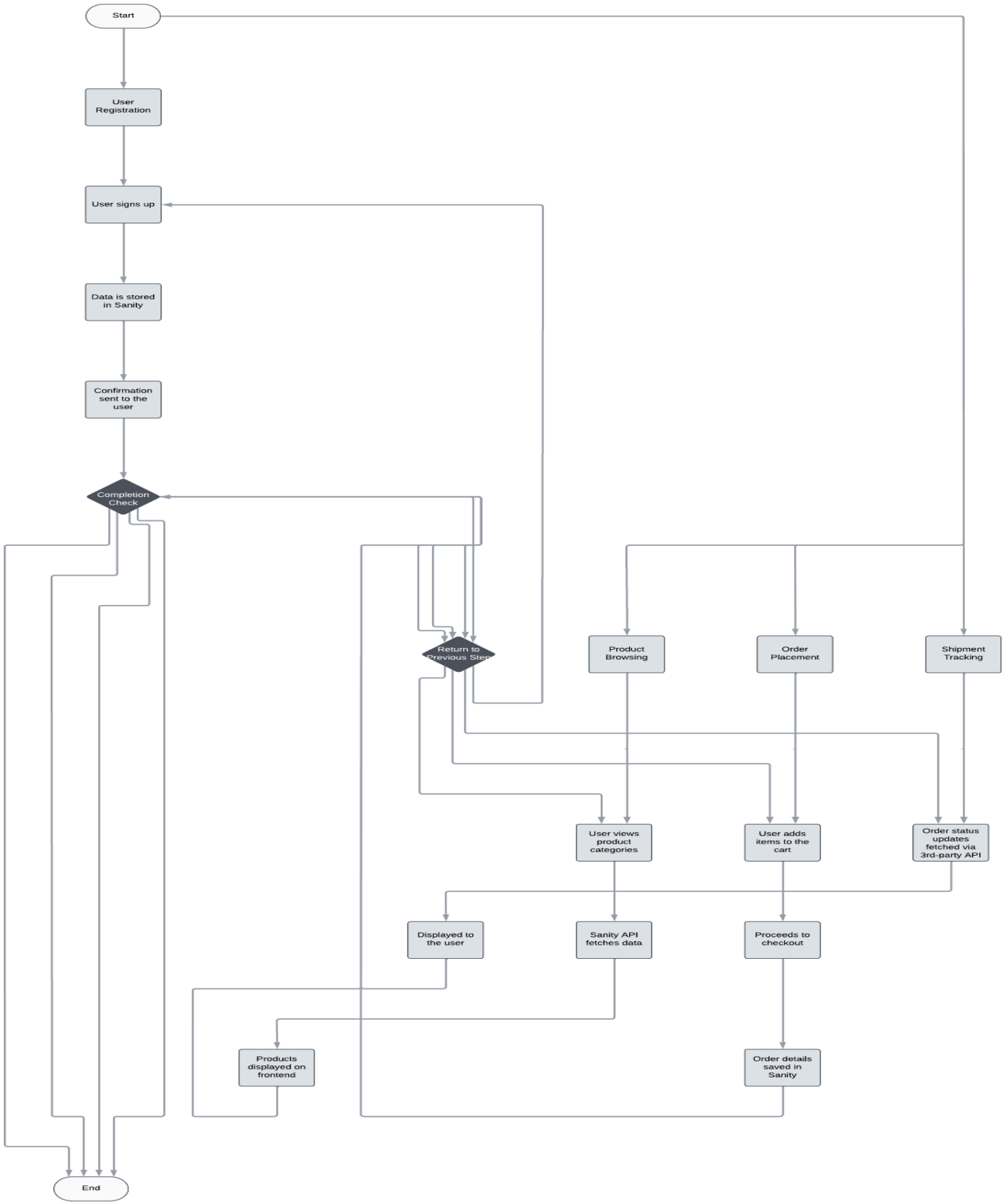
DAY 2 PLANNING THE TECHNICAL FOUNDATION

Key Workflows:



Step-by-step details of interactions between user and components:

Starting with User Registration and Authentication:

The process begins when a new user starts the registration process. They enter their information into the registration form on the frontend. This data is then stored in Sanity CMS, creating a new user record. The system sends a confirmation to the user, typically via email, to verify their account. At this point, the system performs a completion check - if successful, the user can proceed to use the marketplace features.

Moving to the Main Marketplace Functions:

After successful registration, the user encounters three main pathways:

1. ### Product Browsing Workflow:

   When a user enters the marketplace, they first see product categories. The frontend makes a request to Sanity's API to fetch the product catalog data. The data flows back to the frontend, where it's transformed into a user-friendly display. Users can view individual products, with each product view triggering another specific data fetch from Sanity to get detailed product information.

2. Order Placement Process:

As users browse, they can add items to their cart. Each "add to cart" action is first handled by the frontend, which maintains the cart state. When the user is ready to purchase, they proceed to checkout.

At this stage:

- The Sanity API receives the cart data
- The system validates inventory and pricing
- Order details are saved in Sanity's database
- A confirmation is generated for the user

3. Shipment Tracking Integration:

Once an order is placed, the system creates a tracking workflow:

- Order status updates are fetched via the third-party API
- This information is displayed to the user through the frontend
- Users can check their order status at any time, triggering new status checks with the tracking API

Decision Points and User Flow Control: The diagram shows two key decision diamonds:

- The first checks if registration is complete
- The second verifies the previous step's success before allowing progression

Each of these paths can loop back if needed - for instance, if a user needs to modify their cart or check different product categories. The system maintains session state throughout these interactions, ensuring a coherent user experience regardless of which path they take.

Error Handling and Validation:

Though not explicitly shown in the diagram, each step includes validation:

- Registration data must meet system requirements
- Product availability is checked before adding to cart
- Payment information must be valid
- Order details must be complete before processing

This creates a robust system where users can move smoothly between different functions while maintaining data integrity and user session consistency.