# WishWhiz - Shopping Application Flow

## 1. Overview of User Journey and Microservices

This document outlines the user journey and technical microservices interaction for the WishWhiz shopping application. The application allows users to browse products, add items to a cart, proceed to checkout, and place orders. The architecture is built on microservices to ensure scalability and maintainability.

Key Microservices:  
1. Product-Service – Manages product catalog and inventory.  
2. Cart-Service – Handles cart operations and stores user-selected products.  
3. Order-Service – Manages order creation, validation, and checkout.  
4. User-Service – Authenticates users and manages user profiles.  
5. Notification-Service – Sends order confirmations and notifications to users.

## 2. User Journey Flow

1. Browsing Products:  
 - The user visits the app to browse products. Product information is fetched from the Product-Service.  
 - No other services are involved at this stage.  
  
2. Adding to Cart:  
 - The user adds products to the cart. Cart-Service stores product IDs and quantities.  
 - Cart-Service may optionally validate stock by querying Product-Service.  
  
3. Direct Buy (Skipping Cart):  
 - Users can directly purchase items using the 'Buy Now' feature. This triggers Order-Service directly.  
 - Order-Service calls Product-Service to check stock and reserve the item.  
  
4. Checkout Process:  
 - Users proceed to checkout from the cart or directly from the product page.  
 - Order-Service manages the checkout process and communicates with User-Service for authentication and user details.  
 - Product stock is deducted after successful payment.  
  
5. Order Confirmation:  
 - After successful checkout, Notification-Service sends an order confirmation to the user via email or SMS.

## 3. Technical Flow (Microservices Interaction)

1. \*\*Browse Products:\*\*  
 - Frontend → API Gateway → Product-Service  
  
2. \*\*Add to Cart:\*\*  
 - Frontend → API Gateway → Cart-Service  
 - Cart-Service ↔ Product-Service (Optional: Check Stock)  
  
3. \*\*Direct Buy (Buy Now):\*\*  
 - Frontend → API Gateway → Order-Service  
 - Order-Service → Product-Service (Reserve Stock)  
  
4. \*\*Checkout and Place Order:\*\*  
 - Frontend → API Gateway → Order-Service → User-Service (User Authentication)  
 - Order-Service → Product-Service (Deduct Stock)  
 - Order-Service → Cart-Service (Clear Cart after Order)  
  
5. \*\*Order Confirmation:\*\*  
 - Order-Service → Notification-Service (Send Email/SMS)

## 4. Diagram Explanation

The diagram represents the user journey from product browsing to order confirmation. It shows the flow of data between microservices, with API Gateway acting as the entry point for frontend requests. Each microservice handles a specific responsibility, ensuring that the application is modular and scalable.