

Networking Application

Project Name: Couplet - Coupon Wallet

Project Purpose:

The goal of this project is to develop a mobile application that manages all user coupons and discounts from multiple sources in one convenient and user-friendly platform.

Group:

Aviv Duzy 207704040

Roni Kenigsberg 207106931

Doron Shen-Tzur 209366905

INTRODUCTION

Project overview

This project proposes a network-based mobile application that allows users to manage their coupons, discounts, and digital benefits in a secure and user-friendly experience.

While the system is built around a client-server architecture, sensitive coupon data is stored locally on the user's device for security reasons.

The backend server is responsible for user authentication, metadata management, notifications, and coordination between users, while coupon codes themselves never reside on the server.

In addition, the system supports peer-to-peer (P2P) communication for direct coupon sharing between users.

This project directly aligns with the goals of the Network Applications workshop, as it:

- Implements a clear Client-Server model.
- Relies on network communication for all core functionalities.
- Transfers structured data between client and server using APIs.
- Uses network-based services such as authentication and push notifications.
- Combines Client-Server architecture with Peer-to-Peer communication, creating a more advanced network model.

Functional Specification

System Overview

The system consists of:

- Mobile client that stores coupon data locally on the device.
- Backend server responsible for authentication, metadata, notifications, and coordination.
- Local storage layer on the client for secure coupon handling.
- P2P communication layer for coupon sharing between users.

Core Features

1. User Authentication (Network-Based)

- Users register and log in via the server.
- Authentication handled through network APIs.

2. Coupon Storage and Retrieval

- Coupon codes, barcodes, and QR codes are stored locally on the user's device.
- The server stores non-sensitive metadata only (category, expiration date, status).
- The client synchronizes metadata with the server over the network.
- This design reduces security risks related to centralized coupon storage.

3. Coupon Management

- Add, view, and update coupons.
- Filter and sort coupons by category or expiration date.

4. Expiration Notifications

- The server monitors coupon metadata expiration dates.
- Push notifications are sent to the client without accessing the actual coupon code.

5. Coupon Redemption View

- Coupon redemption is handled locally on the device.
- Only the coupon status (used / active / expired) is updated on the server.

6. User Connection and Coupon sharing

- A user can create a sharing group with one or other multiple users
- Connection requests are sent and approved via the server
- Each coupon has a clear owner (the user who added the coupon)
- Coupon owners can revoke the coupon permissions at any time (deletes the coupon from the group)
- When a connection is removed from a group, their access to shared coupons is revoked. Any coupons they own are also removed from the group.
- Search option by user, by coupon or by group

Wireframes - Main Screens

The wireframes illustrate the main screens of the Couplet app, organized into two rows of three screens each.

Top Row:

- Couplet:** The main login and registration screen. It features a header with the app name, tabs for 'Login' and 'Register', input fields for 'Email' and 'Password', a 'Login Button', an 'Or' separator, a 'Register Button', and a 'Forgot Password?' link.
- Home / Coupon List Screen:** The main screen for viewing coupons. It includes a 'Search Bar', a grid of coupon items (each with a thumbnail, title, description, and a 'Save' button), and an 'Add Coupon +' button at the bottom.
- Coupon Details Screen:** A screen for viewing a specific coupon. It displays the 'Coupon Title', 'Save' percentage, 'Expires' date, a 'QR / BARCODE', 'Balance', and an 'Info Details & Terms' button.

Bottom Row:

- Add Coupon Screen:** A screen for adding a new coupon. It has a 'Manual Coupon Input' section with input fields for 'Coupon Code', 'Description', 'Expiration Date', and 'Balance', and a 'Save Coupon' button.
- Connections Screen:** A screen for managing sharing groups. It shows 'My Sharing Groups' with a list of groups (Friends Group, Family Group, Work Group) and an 'Add Group +' button.
- Group Coupon Screen:** A screen for viewing shared coupons within a group. It lists 'Shared Coupon 1', 'Shared Coupon 2', and 'Shared Coupon 3', each with its 'Expires' date.

User Flow (Network Perspective)

1. Client authenticates with the server.
2. Server returns authentication token.
3. Client loads coupon data from local storage.
4. Client syncs coupon metadata with the server.
5. For coupon sharing, clients establish a P2P connection.
6. Coupon is transferred directly between devices.
7. Server updates metadata only.

Example Usage Scenarios

Scenario 1:

A user shares a coupon with another user. The coupon is transferred directly between devices using a peer-to-peer connection, while the server updates only the ownership metadata.

Scenario 2:

The server sends a push notification when a coupon is about to expire.

Scenario 3:

The user searches for a store name, and the system displays all matching coupons (local data + metadata).

Scenario 4:

A user shares a coupon with a group using P2P communication, subject to permissions.

Scenario 5:

When the user enters a shopping mall or commercial area, the system detects the location and suggests relevant coupons. (optional)

Scenario 6:

The system integrates coupons into a digital wallet (optional)

Scenario 7:

The user can add, view or remove users from a group as the group owner

Scenario 8:

A user creates a named collective sharing group and sends connection requests to other users to join the group.

Design and Technical Architecture

Architecture Overview

The application follows a hybrid network architecture

Client (Mobile App):

- User interface
- Sends HTTP requests to server
- Receives responses and notifications

Server (Backend):

- REST API
- Authentication and authorization
- Coupon logic and validation
- Notification scheduling
- Client-Server for authentication, metadata, and notifications.
- Peer-to-Peer for secure coupon sharing.

Client Side Design

Technology: React Native / Flutter

Responsibilities:

- UI rendering
- User input handling
- Network communication
- Local state management

Server Side Design

Responsibilities:

- User authentication and authorization
- Coupon metadata management
- Notification scheduling
- Peer discovery and session coordination

* No storage of sensitive coupon data

Database Design

Users

- user_id
- email
- user_name
- password_hash
- created_at
- coupon_id_list

Coupon

- coupon_id
- owner_id
- category
- redeemable_stores
- expiration_date
- balance
- status: active/expired/used

Group

- group_id
- group_id
- user_id_list
- coupon_id_list
- admin_user_id

Client Side Stored Data

- Coupon codes / QR / barcodes.
- Coupon owner identifier.
- Local sharing permissions.

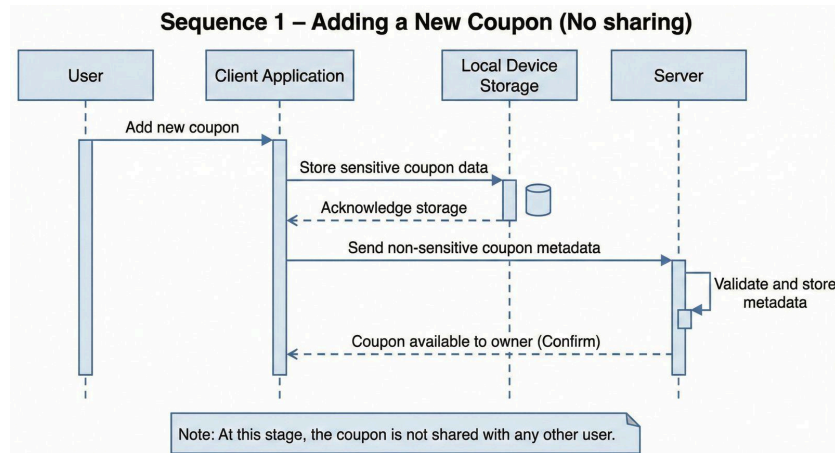
External Tools and Technologies

- Mobile framework: React Native / Flutter.
- Backend: Node.js with Express.
- Authentication: JWT / Firebase Auth.
- Notifications: Firebase Cloud Messaging.
- P2P communication: To be determined (e.g., WebRTC).

Sequence Diagrams

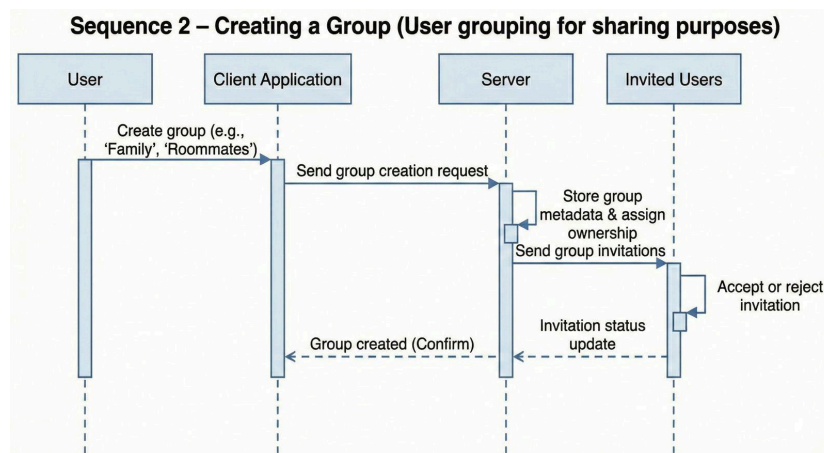
Sequence 1 – Adding a New Coupon

- User adds a new coupon through the client application.
- The client stores the sensitive coupon data locally on the device.
- The client sends non-sensitive coupon metadata to the server.
- The server validates and stores the metadata.
- The coupon becomes available only to the owner user.



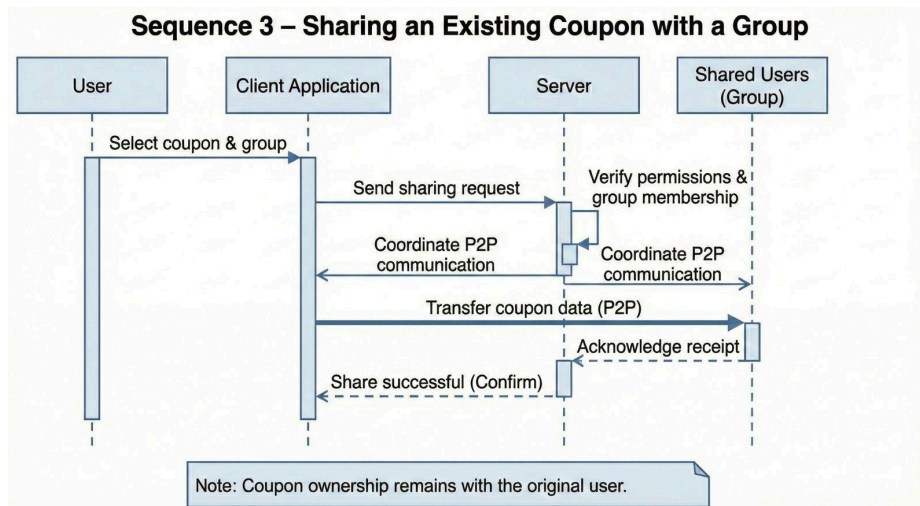
Sequence 2 – Creating a Group

- User creates a new group (e.g., “Family”, “Roommates”).
- Client sends group creation request to the server.
- Server stores group metadata and assigns ownership.
- Other users receive group invitations.
- Invited users accept or reject the invitation.



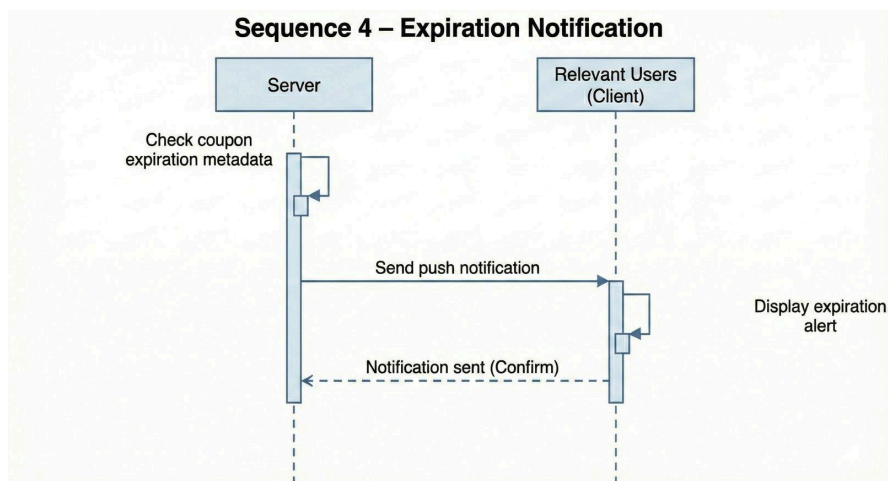
Sequence 3 – Sharing an Existing Coupon with a Group

- User selects an existing coupon from their coupon list.
- User selects a group to share the coupon with.
- Client sends a sharing request to the server.
- Server verifies permissions and group membership.
- Server coordinates P2P communication between users.
- Coupon data is transferred directly between devices.
- Shared users receive access to the coupon.
- Coupon ownership remains with the original user.



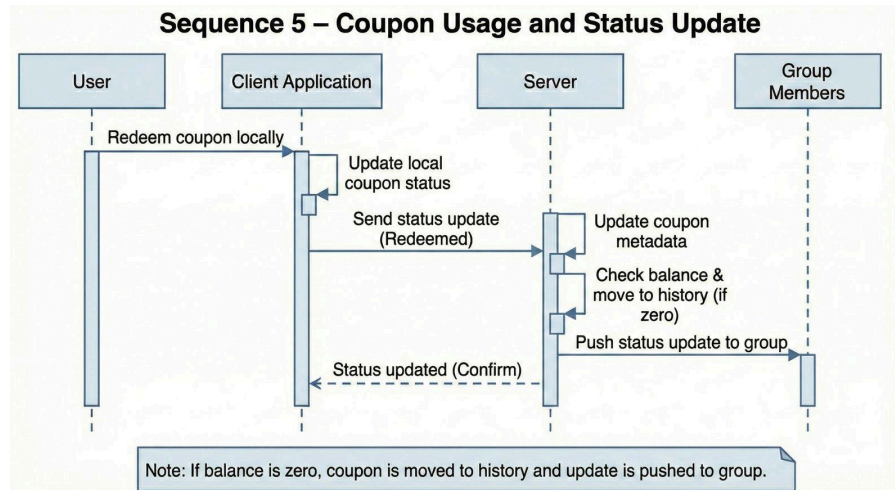
Sequence 4 – Expiration Notification

- Server checks coupon expiration metadata.
- Server sends push notifications to all relevant users.
- Client displays expiration alert.



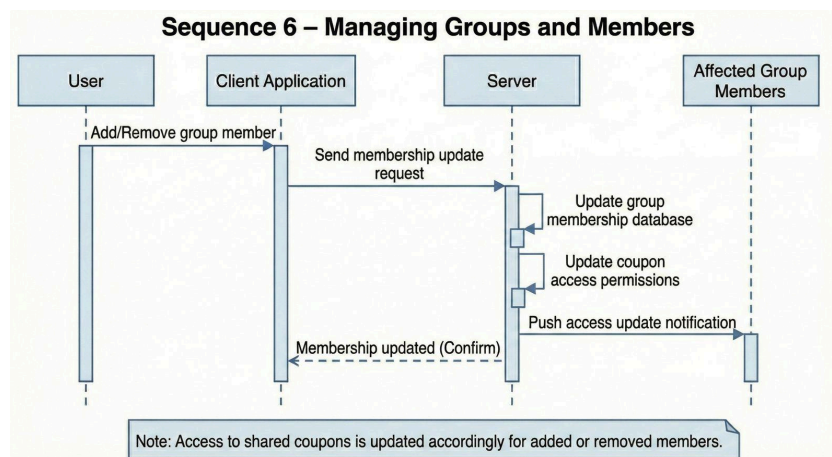
Sequence 5 – Coupon Usage and Status Update

- A user redeems a coupon locally on the device.
- Client updates coupon status.
- Client sends status update to the server.
- Server updates metadata.
- If the coupon balance reaches zero:
 - The coupon is moved to history.
 - The update is reflected for all users in the group.



Sequence 6 – Managing Groups and Members

- User adds or removes members from a group.
- Client sends update to the server.
- Server updates group membership.
- Access to shared coupons is updated accordingly.



Sequence 7 - Searching for Coupons by Store

- User enters a store name in the search bar within the mobile application.
- The client sends a search request to the server with the store identifier.
- The server queries coupon metadata to find coupons associated with the requested store.
- The server returns a list of matching coupon identifiers to the client.
- The client filters the locally stored coupons using the received identifiers.
- The client displays all matching coupons available for the selected store.
- If no matching coupons are found, the client displays a relevant message to the user.

