PYCO App Technical Specifications

Akemi Sai, Edward Marecos

November 12, 2024

PYCO App Overview

This section outlines how the PYCO app meets the requirements specified in the class assignment for CS501, covering essential features, architectural best practices, use of APIs, and offline functionality.

1 Network Requests and API Integration

- **Retrofit**: Retrofit is used as the primary library for making HTTP requests. It integrates with external services like fashion-related APIs to retrieve data such as brand information and item suggestions. Retrofit supports Kotlin coroutines, ensuring network operations are executed asynchronously, which is essential for maintaining app responsiveness.
- **Firebase Firestore**: Firebase Firestore is used for authentication, real-time data storage, and syncing across devices.
- **ML Kit**: ML Kit is leveraged for image background removal, improving user-generated content like clothing images.
- **Coroutine Integration**: Coroutines are used to manage asynchronous tasks efficiently, ensuring UI responsiveness during API calls and database operations.

2 Jetpack Tools and Modern Architecture

- **ViewModel**: The app uses ViewModel components to manage UI-related data in a lifecycle-conscious manner, ensuring data persists across configuration changes like screen rotations.
- **Room Database**: The Room database stores clothing inventory data locally for offline functionality. When the network connection is restored, WorkManager handles background synchronization to update remote data sources.
- **LiveData/StateFlow**: These components are employed to observe data changes and update the UI accordingly, ensuring real-time data consistency between the ViewModel and UI.

3 Separation of Concerns and UI Controllers

- **Single Activity Architecture**: The app follows the single activity architecture with composables for different screens, maintaining clean navigation flows and reducing overhead.
- **Repository Pattern**: The ViewModel interacts with repository modules, handling data operations like fetching data from Room or making network requests via Retrofit.

4 Offline Functionality

- **Room Database and WorkManager**: The Room database ensures offline data persistence, while WorkManager handles data synchronization with Firebase Firestore once the connection is restored.
- **Cached Data Access**: Room ensures users can access previously fetched data, providing a continuous experience without visible interruptions during connectivity loss.

Synopsis and Goals

PYCO (Put Your Clothes On) is a social media application that connects individuals seeking fashion inspiration with users who have a natural flair for styling. The app allows users to catalog their wardrobe, request help creating outfits, and interact with a community passionate about fashion.

Goals

- Provide a platform for fashion advice and inspiration.
- Enable users to catalog their clothing inventory.
- Empower aspiring designers by giving them a platform to showcase their skills.
- Evolve into a marketplace for style inspiration and potential clothing transactions.

Target Audience

- **Fashion Seekers**: Users looking for fashion advice or inspiration.
- **Aspiring Designers**: Individuals with an interest in fashion and styling.
- **General Public**: Fashion enthusiasts and trend followers.
- **Young Adults and Students**: Those needing affordable styling solutions or help for specific occasions.

Core Features and Functionality

Account Types

- **Standard Account**: Default account type for users to catalog wardrobe, create outfit requests, and interact with others.
- **Designer Account**: Users can opt into designer mode, accessing analytics and participating in promotional content.

Social Features

- **Followers and Friends**: Users can follow others and send friend requests.
- **Block Feature**: Users can block others to ensure a safe environment.
- **Friends-Only Feed**: Users can filter their feed to view only friends' content.

User Inventory Management

- **UI Elements**: LazyVerticalGrid and ScrollableTabRow are used for displaying and navigating clothing categories.
- **CameraX API**: Allows users to take pictures of clothing items and add them to their wardrobe.
- **ML Kit/Adobe Creative SDK**: Used for background removal for clothing images.

Outfit Requests

- **UI Elements**: Card composable and LazyColumn to display outfit requests.
- **Functionality**: Users can select items to highlight and set privacy for requests.

Outfit Creation and Interaction

- **UI Elements**: DragGestureDetector for dragging and dropping clothing items.
- **Functionality**: Designers can create outfits using a mannequin-based interface.

Feed Structure

- **UI Elements**: LazyRow for trending outfits and Pager for full-screen scrollable posts.
- **Integration**: Firebase Firestore ensures real-time updates, and WorkManager handles offline syncing.

Recognition and Leaderboards

- **Badges and Honor Points**: Users earn badges for achievements.
- **Leaderboard**: Displays top designers based on various metrics.
- **Rotating Spotlight**: Features rising stars to ensure exposure for new designers.

Notifications and User Engagement

- **Notifications**: Firebase Cloud Messaging sends notifications for comments, followers, and leaderboard placements.
- **Customization**: Users can manage notifications through settings.

Privacy and Security

- **Post Visibility Options**: Users can control post visibility (friends-only or public).
- **Data Protection**: Firebase Authentication ensures secure login, and EncryptedSharedPreferences are used for storing sensitive data.