**Technical Preview**

**I. Schedule**

**II. Guiline**

**1. Entity Files (Đại Diện Cho Các Table Trong Database)**

Những file này định nghĩa cấu trúc dữ liệu, giống như "cột" trong bảng Excel. Mỗi entity là một table, với @PrimaryKey là khóa chính (ID duy nhất).

|  |  |  |  |
| --- | --- | --- | --- |
| **File Name** | **Thành Phần Chính** | **Công Dụng** | **Kiểu Dữ Liệu Chính (Giải Thích Cho Người Mới)** |
| **Item.kt** | - @Entity(tableName = "items") - Data class với fields: id, name, price, quantity, date, store, category, receiptId. - @ForeignKey liên kết với Receipt (xóa cascade). | Lưu thông tin chi tiết sản phẩm từ hóa đơn (receipts - input từ user). Một receipt có nhiều item. | - id: Int (số nguyên, tự động tăng). - name: String (chuỗi văn bản, tên sản phẩm). - price: Double (số thực, giá tiền). - quantity: Float (số thực, số lượng như 100.5g). - date: Date (ngày tháng, dùng converter). - store: String (tên cửa hàng). - category: String (phân loại như "thịt"). - receiptId: Int? (số nguyên nullable, liên kết với receipt). |
| **Recipe.kt** | - @Entity(tableName = "recipes") - Data class với fields: id, title, description, ingredients, instructions, nutrition, totalTime, dateSaved. | Lưu công thức nấu ăn được AI tạo ra (recipes - output sau phân tích). Không liên kết trực tiếp với item/receipt. | - id: Int (tự động tăng). - title: String (tiêu đề món ăn). - description: String (mô tả ngắn). - ingredients: String (JSON chuỗi, danh sách nguyên liệu). - instructions: String (hướng dẫn từng bước). - nutrition: String (JSON, thông tin dinh dưỡng). - totalTime: Int (thời gian phút). - dateSaved: Date (ngày lưu). |
| **User.kt** | - @Entity(tableName = "users") - Data class với fields: id, username, password, phone, savedRecipes. | Lưu thông tin tài khoản người dùng, bao gồm công thức đã lưu (profile). | - id: Int (tự động tăng). - username: String (tên đăng nhập). - password: String (mật khẩu đã hash). - phone: String (số điện thoại). - savedRecipes: String (JSON, danh sách ID công thức đã lưu). |
| **Receipt.kt** | - @Entity(tableName = "receipts") - Data class với fields: id, date, source, status. | Group nhiều item thành một receipt (hóa đơn input từ user). Status theo dõi quá trình xử lý. | - id: Int (tự động tăng). - date: Date (ngày hóa đơn). - source: String (nguồn cửa hàng). - status: String (trạng thái như "Pending"). |

**2. DAO Files (Truy Vấn Dữ Liệu)**

|  |  |  |  |
| --- | --- | --- | --- |
| **File Name** | **Thành Phần Chính** | **Công Dụng** | **Kiểu Dữ Liệu Chính (Trong Query/Hàm)** |
| **ItemDao.kt** | - @Dao - Query: getAllItems (ORDER BY name), getItem, searchItems. - Insert, Update, Delete. | Truy vấn/xử lý dữ liệu sản phẩm từ receipts. Tìm kiếm case-insensitive. | - Trả về: Flow<List<item>> (danh sách reactive). - Query params: String (query tìm kiếm), Int (id). - Hàm suspend: Item (object).</item> |
| **RecipeDao.kt** | - @Dao - Query: getAllRecipes (ORDER BY dateSaved), getRecipe, searchRecipes. - Insert, Update, Delete. | Truy vấn/xử lý công thức nấu ăn (recipes). Tìm kiếm theo title/description. | - Trả về: Flow<List<recipe>>. - Query params: String, Int. - Hàm suspend: Recipe.</recipe> |
| **UserDao.kt** | - @Dao - Query: getAllUsers (ORDER BY username), getUser, searchUsers. - Insert, Update, Delete. | Truy vấn/xử lý tài khoản người dùng. Tìm kiếm theo username/phone. | - Trả về: Flow<List<user>>. - Query params: String, Int. - Hàm suspend: User.</user> |
| **ReceiptDao.kt** | - @Dao - Query: getAllReceipts (ORDER BY date), getReceipt, searchReceipts. - Insert, Update, Delete. | Truy vấn/xử lý nhóm hóa đơn (receipts). Tìm kiếm theo source/status. | - Trả về: Flow<List<receipt>>. - Query params: String, Int. - Hàm suspend: Receipt.</receipt> |

**3. Repository Files (Trung Gian Giữa UI Và DAO)**

|  |  |  |  |
| --- | --- | --- | --- |
| **File Name** | **Thành Phần Chính** | **Công Dụng** | **Kiểu Dữ Liệu Chính** |
| **ItemsRepository.kt** | - Interface với hàm: getAllItemsStream, getItemStream, searchItems, insertItem, deleteItem, updateItem. | Trừu tượng hóa truy cập dữ liệu item (receipts). | - Flow<List<item>>, Flow&#x3C;Item?>, suspend Item.</item> |
| **OfflineItemsRepository.kt** | - Class implement ItemsRepository, gọi DAO. | Implementation cục bộ cho item. | Giống trên (gọi DAO). |
| **RecipesRepository.kt** | - Interface tương tự cho recipe. | Trừu tượng hóa truy cập recipe. | Flow<List<recipe>>, Flow&#x3C;Recipe?>, suspend Recipe.</recipe> |
| **OfflineRecipesRepository.kt** | - Implement RecipesRepository. | Implementation cục bộ cho recipe. | Giống trên. |
| **UsersRepository.kt** | - Interface tương tự cho user. | Trừu tượng hóa truy cập user. | Flow<List<user>>, Flow&#x3C;User?>, suspend User.</user> |
| **OfflineUsersRepository.kt** | - Implement UsersRepository. | Implementation cục bộ cho user. | Giống trên. |
| **ReceiptsRepository.kt** | - Interface tương tự cho receipt. | Trừu tượng hóa truy cập receipt. | Flow<List<receipt>>, Flow&#x3C;Receipt?>, suspend Receipt.</receipt> |
| **OfflineReceiptsRepository.kt** | - Implement ReceiptsRepository. | Implementation cục bộ cho receipt. | Giống trên. |

|  |  |  |  |
| --- | --- | --- | --- |
| **File Name** | **Thành Phần Chính** | **Công Dụng** | **Kiểu Dữ Liệu Chính** |
| **InventoryDatabase.kt** | - @Database với entities: Item, Receipt, Recipe, User. - Abstract fun cho các DAO. - Singleton instance. - @TypeConverters(Converters::class). | Khởi tạo và quản lý toàn bộ database (singleton để tránh nhiều instance). fallbackToDestructiveMigration xóa data khi migrate (dev only). | Không trực tiếp (chỉ config). |
| **AppContainer.kt** | - Interface AppContainer với val cho các repository. - Class AppDataContainer lazy init offline repositories. | Dependency Injection: Cung cấp repository cho app (dễ test/mở rộng). | Không trực tiếp (inject objects). |
| **Converters.kt** | - Class với @TypeConverter cho Date <-> Long. | Chuyển đổi Date thành Long (timestamp) để lưu trong SQLite. | Long (timestamp), Date. |