

ĐỀ KIỂM TRA

HỌC PHẦN: LẬP TRÌNH MOBILE ĐỀ SỐ: 05 - FLUTTER FIREBASE

Thời gian làm bài: 120 phút

Tổng điểm: 100 điểm

Được phép: Sử dụng ChatGPT, tài liệu, Internet

MÔ TẢ DỰ ÁN

⚠ YÊU CẦU BẮT BUỘC:

- Tên project Flutter phải đặt là: `flutter_app_<MaSinhVien>` (Ví dụ: `flutter_app_2151061234`).
- Package name (Android Application ID) phải là: `com.example.<MaSinhVien>` (Ví dụ: `com.example.2151061234`).
- Nếu sử dụng Firebase Project mới, tên Project ID nên chứa Mã Sinh Viên.

Bạn cần xây dựng một ứng dụng Flutter quản lý nhà hàng với Firebase Firestore. Ứng dụng cho phép khách hàng xem menu, đặt bàn, đặt món, thanh toán, và quản lý đặt bàn của nhà hàng.

📋 YÊU CẦU CƠ SỞ DỮ LIỆU

Cấu trúc Firestore Collections:

Collection 1: `customers`

- `customerId` (String, Document ID): ID duy nhất của khách hàng
- `email` (String): Email đăng nhập
- `fullName` (String): Họ và tên đầy đủ
- `phoneNumber` (String): Số điện thoại
- `address` (String): Địa chỉ
- `preferences` (Array of Strings): Sở thích ăn uống (["vegetarian", "spicy", "seafood"])
- `loyaltyPoints` (Integer): Điểm tích lũy (mặc định 0)
- `createdAt` (Timestamp): Thời gian tạo tài khoản
- `isActive` (Boolean): Trạng thái tài khoản

Collection 2: `menu_items`

- `itemId` (String, Document ID): ID duy nhất của món ăn
- `name` (String): Tên món ăn
- `description` (String): Mô tả món ăn
- `category` (String): Danh mục ("Appetizer", "Main Course", "Dessert", "Beverage", "Soup")
- `price` (Double): Giá món ăn
- `imageUrl` (String): URL hình ảnh món ăn
- `ingredients` (Array of Strings): Danh sách nguyên liệu
- `isVegetarian` (Boolean): Món chay không

- `isSpicy` (Boolean): Món cay không
- `preparationTime` (Integer): Thời gian chế biến (phút)
- `isAvailable` (Boolean): Còn phục vụ không
- `rating` (Double): Đánh giá trung bình (0.0 - 5.0)
- `createdAt` (Timestamp): Thời gian thêm món

Collection 3: `reservations`

- `reservationId` (String, Document ID): ID duy nhất của đặt bàn
- `customerId` (String): ID của khách hàng (reference đến `customers`)
- `reservationDate` (Timestamp): Ngày giờ đặt bàn
- `numberOfGuests` (Integer): Số lượng khách
- `tableNumber` (String, nullable): Số bàn được phân (chỉ điền khi confirmed)
- `status` (String): Trạng thái ("pending", "confirmed", "seated", "completed", "cancelled", "no_show")
- `specialRequests` (String, nullable): Yêu cầu đặc biệt
- `orderItems` (Array of Maps): Danh sách món đã đặt
 - `itemId` (String): ID món ăn
 - `itemName` (String): Tên món
 - `quantity` (Integer): Số lượng
 - `price` (Double): Giá tại thời điểm đặt
- `subtotal` (Double): Tổng tiền món ăn
- `serviceCharge` (Double): Phí phục vụ (10% subtotal)
- `discount` (Double): Giảm giá (từ `loyaltyPoints`)
- `total` (Double): Tổng cộng
- `paymentMethod` (String, nullable): Phương thức thanh toán ("cash", "card", "online")
- `paymentStatus` (String): Trạng thái thanh toán ("pending", "paid", "refunded")
- `createdAt` (Timestamp): Thời gian đặt bàn
- `updatedAt` (Timestamp): Thời gian cập nhật

Lưu ý về quan hệ:

- Một `customer` có thể có nhiều `reservation` (quan hệ 1-nhiều)
- Một `reservation` chứa nhiều `menu_item` (quan hệ nhiều-nhiều qua array `orderItems`)
- Khi đặt bàn: tạo reservation với status = "pending"
- Khi xác nhận: set status = "confirmed", phân `tableNumber`
- Khi thanh toán: tính `loyaltyPoints` (1% total), cộng vào `customer`
- `preferences` là array chứa sở thích ăn uống của `customer`

YÊU CẦU CHỨC NĂNG

Phần 1: Thiết lập Firebase (10 điểm)

1. **Cài đặt Firebase trong Flutter** (3 điểm)
2. **Tạo Firestore Database** (4 điểm)
3. **Tạo Service Class** (3 điểm)

Phần 2: Tạo Model Classes (15 điểm)

1. **Model Customer** (5 điểm) - Xử lý array preferences
 2. **Model MenuItem** (5 điểm) - Xử lý array ingredients
 3. **Model Reservation** (5 điểm) - Xử lý array orderItems, nullable fields
-

Phần 3: Repository Pattern - CRUD Operations (40 điểm)

3.1. Customer Repository (12 điểm)

1. **Thêm Customer** (3 điểm)
2. **Lấy Customer theo ID** (2 điểm)
3. **Lấy tất cả Customers** (2 điểm)
4. **Cập nhật Customer** (3 điểm)
5. **Cập nhật Loyalty Points** (2 điểm)
 - Method `Future<void> updateLoyaltyPoints(String customerId, int points)`

3.2. MenuItem Repository (14 điểm)

1. **Thêm MenuItem** (3 điểm)
2. **Lấy MenuItem theo ID** (2 điểm)
3. **Lấy tất cả MenuItems** (2 điểm)
4. **Tìm kiếm MenuItems** (4 điểm) - Tìm trong name, description, ingredients
5. **Lọc MenuItems** (3 điểm)
 - Theo category
 - Theo isVegetarian
 - Theo isSpicy

3.3. Reservation Repository (14 điểm)

1. **Đặt Bàn** (5 điểm)
 - Method `Future<void> createReservation(String customerId, Timestamp reservationDate, int numberOfGuests, String specialRequests)`
 - Kiểm tra không đặt trùng thời gian (nếu cần)
 - Tạo reservation với status = "pending"
2. **Thêm Món vào Đơn** (3 điểm)
 - Method `Future<void> addItemToReservation(String reservationId, String itemId, int quantity)`
 - Kiểm tra item isAvailable
 - Thêm vào orderItems array
 - Tính lại subtotal, serviceCharge, total
3. **Xác nhận Đặt Bàn** (2 điểm)
 - Method `Future<void> confirmReservation(String reservationId, String tableNumber)`
 - Set status = "confirmed", tableNumber

4. Thanh toán (3 điểm)

- Method `Future<void> payReservation(String reservationId, String paymentMethod)`
- Tính discount từ loyaltyPoints (1 point = 1000đ, tối đa 50% total)
- Tính lại total sau discount
- Set paymentStatus = "paid", status = "completed"
- Cộng loyaltyPoints cho customer (1% total)
- Trừ loyaltyPoints đã dùng

5. Lấy Đặt Bàn (1 điểm)

- Method `Future<List<ReservationModel>> getReservationsByCustomer(String customerId)`
- Method `Future<List<ReservationModel>> getReservationsByDate(String date)`

Phần 4: UI Implementation (25 điểm)

Yêu cầu chung cho UI: Trên thanh AppBar của màn hình chính (Home Screen), bắt buộc phải hiển thị tiêu đề có chứa Mã Sinh Viên của bạn. Ví dụ: "Restaurant App - 2151061234".

4.1. Màn hình đăng ký/đăng nhập (5 điểm)

1. Màn hình đăng ký (3 điểm)

- Form: email, fullName, phoneNumber, address
- Multi-select cho preferences

2. Màn hình đăng nhập (2 điểm)

4.2. Màn hình Menu (8 điểm)

1. Hiển thị danh sách món ăn (4 điểm)

- StreamBuilder real-time
- GridView/ListView với image, name, price, rating
- Badge "Hết món" nếu !isAvailable
- Icon cho vegetarian, spicy

2. Tìm kiếm và lọc (2 điểm)

- TextField tìm kiếm
- Filter: category, vegetarian, spicy

3. Chi tiết món ăn (2 điểm)

- Hiển thị đầy đủ thông tin
- Danh sách ingredients
- Nút "Thêm vào đơn" (chỉ hiện nếu isAvailable)

4.3. Màn hình đặt bàn (7 điểm)

1. Form đặt bàn (4 điểm)

- DatePicker cho reservationDate
- TimePicker cho giờ
- Number picker cho numberOfGuests
- TextField cho specialRequests
- Gọi `createReservation()`

2. Thêm món vào đơn (3 điểm)

- Hiển thị danh sách món đã thêm
- Cho phép tăng/giảm số lượng
- Tính subtotal, serviceCharge, total
- Nút "Xác nhận đặt bàn"

4.4. Màn hình đặt bàn của tôi (5 điểm)

1. Danh sách đặt bàn (3 điểm)

- Hiển thị đặt bàn của customer hiện tại
- Mỗi item: reservationDate, numberOfGuests, status, total
- Màu sắc theo status

2. Chi tiết đặt bàn (2 điểm)

- Hiển thị đầy đủ thông tin
- Danh sách món đã đặt
- Nút "Thanh toán" (chỉ cho status = "seated")
- Hiển thị loyaltyPoints hiện có

Phần 5: Xử lý Real-time và Error Handling (10 điểm)

1. Real-time Updates (5 điểm)

2. Error Handling (5 điểm) - Món hết, đặt bàn trùng, v.v.

📝 YÊU CẦU KỸ THUẬT

- Tổ chức code theo cấu trúc chuẩn
- State Management
- UI/UX đẹp

📦 PACKAGES ĐƯỢC PHÉP SỬ DỤNG

```
dependencies:  
  flutter:  
    sdk: flutter  
  firebase_core: ^latest
```

```
cloud_firestore: ^latest
shared_preferences: ^latest
```

CHECKLIST NỘP BÀI

- Project hoàn chỉnh, có thể chạy được
- Firebase project đã tạo và kết nối
- Có ít nhất 5 customers mẫu
- Có ít nhất 20 menu_items mẫu (phân bổ các category)
- Có ít nhất 10 reservations mẫu (nhiều trạng thái khác nhau)
- Tất cả chức năng CRUD hoạt động
- UI hiển thị dữ liệu từ Firestore
- Real-time updates hoạt động
- Error handling đầy đủ
- Code tổ chức rõ ràng
- File README.md

Chúc các bạn làm bài tốt! 🎉