Food Delivery App - QA Notes & Test Scenarios

Meeting notes from QA planning session - needs to be formalized

Session Notes - Jan 15, 2025

Okay so we discussed the main test scenarios today. For the **user registration flow (TEST-001)**, we need to verify that when someone downloads the app and tries to create an account, they can sign up with email OR phone number OR social login (Google/Apple). The password requirements are the usual - 8 characters minimum, needs a number and special character. After signup, they should get a verification code via email or SMS depending on what they used. This is **high priority** obviously since nobody can use the app without an account. Oh and we need to test that the referral code field works - if they enter a valid code, both users should get $5 credit.

For **restaurant search (TEST-002)**, the user should be able to search by cuisine type, restaurant name, or specific dishes. The results should show distance, rating, delivery time estimate, and whether there's a promo. **Medium priority**. Filters should work - price range ($, $$, $$$), dietary restrictions (vegan, halal, gluten-free), and minimum rating. If GPS is off, it should ask to enable or let them enter address manually. Need to test the "near me" feature with different radiuses. Type is functional testing.

Order placement (TEST-003) is critical - this is where the money is! User adds items to cart, can customize (extra cheese, no onions, etc.), sees real-time price updates. When they checkout they should see subtotal, taxes, delivery fee, any discounts applied, and tip options (15%, 20%, 25%, custom). Payment methods: saved cards, Apple Pay, Google Pay, PayPal, and cash on delivery for some areas. After successful payment, order confirmation with estimated delivery time (should be realistic based on restaurant prep time + driver distance). Order should appear in restaurant's tablet app within 30 seconds. This is integration testing with payment gateway and restaurant system. Priority: CRITICAL!

Sarah mentioned we need to test **what happens when a restaurant is closed (TEST-004)** - if someone tries to order from a closed restaurant, they should see "Opens at 11 AM" or whatever, and option to schedule order for later. Pre-ordering should work up to 7 days in advance. Priority is medium, type is functional and edge case.

Driver tracking (TEST-005) - once order is picked up, customer should see live map with driver location updating every 10 seconds. ETA should adjust based on traffic. They can message driver through in-app chat (messages should be masked, no personal phone numbers exposed). If driver is delayed more than 10 min past original ETA, automatic notification to customer with new estimate and option to cancel for full refund. Priority high, integration type.

We almost forgot about **refunds and complaints (TEST-006)**. If food arrives cold, wrong, or missing items, customer can report within 2 hours of delivery. They upload photo, describe issue, and submit. Should get automatic partial refund for missing items, or full refund option if order completely wrong. Support ticket created and assigned to restaurant for response. Credits should appear in account within 24 hours. Test the escalation flow if restaurant disputes - goes to customer service manager. This is definitely high priority for customer satisfaction. Error handling and business rules testing.

Promo codes (TEST-007) - various scenarios: percentage off (20% up to $10), flat discount ($5 off orders over $25), free delivery, BOGO on specific items. Need to test that codes can't be stacked (unless marked stackable), expired codes show clear error, usage limits work (one per customer, first 100 uses, etc.). Someone tried entering SQL injection in promo field last month so add security test for that - sanitize input! Medium priority mostly functional but that SQL test is critical security.

Almost forgot **TEST-008 for ratings and reviews**. After delivery marked complete, customer should get prompt to rate (1-5 stars) for food, delivery, and overall. Can add text review and photos. Reviews should appear on restaurant page within 1 hour (moderation for profanity). Restaurant can respond to reviews. Test that you can only review orders you actually placed. Low priority, UI/UX type testing.

TEST-009: App performance - we talked about load times. Home screen should load in under 2 seconds on 4G. Search

results under 1.5 seconds. Restaurant menu with images under 3 seconds. Cart operations (add/remove/update) should feel instant, under 500ms. Need to test on older phones too - iPhone 8, Samsung Galaxy S9. This is performance testing, medium priority but CEO really cares about this so maybe bump to high.

Last one: **TEST-010 offline behavior**. If connection drops while browsing, show cached content with "offline" banner. If drops during checkout, queue the order and submit when back online (within 5 min window, otherwise cancel and notify). Saved payment methods should work offline for small orders under $50 with reconciliation later. Edge case testing, low to medium priority.