Dear Mark and Matt,

In this document I have considered the assumptions, acceptance criteria and potential issues that could arise when testing the below user stories.

To ensure that the user stories are tested and defect free, I would follow the below acceptance criteria to derive positive and negative tests for both the “happy path”, as well as edge cases.

To ensure that the development and testing progresses in as efficient a manner as possible, I would:

* Make sure genuine reproduceable issues are reported and any defect reports contain the exact steps to recreate an issue, any error logs, release versions, so as to save the developers time and effort,
* Advocate for high priority issues to be resolved first, especially if they are business-critical (e.g. ability to charge for subsciptions),
* Highlight any blockers in a timely manner, so they can be taken care of in good time,
* Be as diplomatic as possible when presenting my findings.

I look forward to discussing these with you in more detail soon.

Regards,

Emilia

============================================================================

**Test Engineer Interview Scenario**

**Background:**

Customers can use their app to manage their subscription. Subscriptions are paid for weekly and are paid up front.

Drivers use an app to tell them what needs to be delivered.

==========================================================================

1. **Story #1:**

As a Customer

I need to be able to increase and decrease the quantity of an item to be delivered on a given day

So that I can manage my deliveries

e.g. change from 1 pint of milk to be delivered on Mondays to 2 pints milk.

* 1. **Assumptions (A):**

A1. The functionality will be implemented for both one-time and repeat orders.

A2. The customer has already placed an order and wants to edit the quantity of items of an existing order yet to be delivered from the customer app.

A3. The order can be for a single item or multiple items.

**1.2 Acceptance criteria (AC):**

These acceptance criteria concern only **editing existing orders** and not new orders or the basket functionality. It is assumed that the customers have the ability to edit existing orders within a certain time frame.

**1.2.1 Acceptance criteria for both one-time orders and repeat orders:**

AC1: The customer can increase the quantity of items to be delivered for before the cut-off time.

AC2. The customer cannot increase the quantity of items to be delivered after the cut-off time.

AC3. The customer can decrease the quantity of items to be delivered before the cut-off time.

AC4. The customer cannot decrease the quantity of items to be delivered after the cut-off time.

AC5. Following from AC1, AC3: Display message asking customer to confirm change of order.

AC6. Following from AC2, AC4: Display message to user saying that the quantity of items to be delivered cannot be changed after the cut-off time.

AC7. Display message to customer if the change of quantity of items has been successful.

AC8. The quantity is increased correctly, by the correct increment.

AC9. The quantity is decreased correctly, by the correct increment.

AC10. The quantity can be changed repeatedly (e.g. it can be increased, decreased and then increased again).

AC10. The customer can change the quantity of the items to be delivered multiple times, as long as they do so before the cut-off time.

AC11. The changes of quantity of items to be deliveredare reflected correctly in the driver app.

AC12. The customer can abandon any changes and the order will remain as is.

AC13. For orders of multiple items, the quantity of 1 item only can be changed correctly and the quantity of other ordered items stays the same.

AC14. For orders of multiple items, the quantity of all ordered items can be changed correctly.

**1.2. 2 Additional acceptance criteria for repeat orders:**

**Assumptions**: The customer has 3 deliveries per week, on 3 different days of the week.

AC15. The customer can increase the quantity of items to be delivered on a particular day with the remaining days being unchanged.

AC16. The customer can decrease the quantity of items to be delivered on a particular day with the remaining days being unchanged.

AC17. The customer can increase the quantity of items to be delivered on any two days with the remaining day being unchanged.

AC18. The customer can decrease the quantity of items to be delivered on any two days with the remaining day being unchanged.

AC19. The customer can increase the quantity of items to be delivered on all days.

AC20. The customer can decrease the quantity of items to be delivered on all days.

AC21. The customer can increase the quantity of items to be delivered on some days, decrease it on other days, and leave yet other days unchanged.

**1.3. Potential issues:**

-The quantity of items to be delivered is not updated correctly.

-The change of quantity is not updated in the driver app.

- For orders of multiple items, the change of quantity for one item only incorrectly triggers a change in the quantity of another item.

-For repeat orders, any change of quantity for a given day triggers a change for another or all delivery days.

- The customer cannot reverse any changes made if they change their mind.

- The customer requests a change of quantity by mistake when they don’t mean to (“Are you sure?” messages can be used to prevent this – please see AC5.)

=================================================================================

1. **Story #2**

As a Customer

I should be refunded or charged for any changes I make to my subscription

So that I am paying the correct amount

* 1. **Assumptions:**

The changes that can be made to a subscription are:

* Pausing a subscription
* Restarting a subscription
* Cancelling a subscription

The following acceptance criteria **do not** take into consideration new subscriptions, which would have been covered in a separate user story.

* 1. **Acceptance criteria (AC)**
     1. **Acceptance criteria for refunds:**

AC1: Issue a refund when the subscription has been cancelled and a subscription payment has been taken.

AC2. Don’t issue a refund when a subscription has been cancelled, but a subscription payment hasn’t been taken. (E.g. customer cancels before the next weekly subscription payment is due.)

AC3. Don’t issue a refund when the subscription has been paused and a subscription payment has been taken. (Assumption: Customers will come back and re-activate their subscription).

AC4: A refund is issued to the correct customer.

AC5: A refund is issued to the correct bank card.

AC6: Don’t issue a refund to a different customer.

AC7: Don’t issue a refund to a bank card different from the one used by the customer.

AC8. Refund the correct amount of money.

AC9. The refund cannot exceed the amount of money charged for the subscription.

AC10. The refund cannot be 0.

AC11. The refund amount is a positive number with a maximum of two decimal places.

AC12: Notify the customer of any problems with issuing the refund (e.g. expired bank card).

AC13. Notify the customer of successfully processed refunds.

**2.2.2 Acceptance criteria for subscriptions:**

AC14. Don’t charge a subscription payment when the customer has paused subscriptions.

AC 15. Charge a subscription payment when the customer restarts subscriptions.

AC 16. Charge the correct amount of money when reactivating subscriptions.

AC 17. Charge the correct bank card when reactivating subscriptions.

AC 18. The subscription amount is not 0.

AC 19. The subscription amount is a positive number with a maximum of 2 decimal places.

AC 20. Notify the customer of any payment processing problems when reactivating a subscription.

AC 21. Notify the customer of successful reactivation of subscription.

AC 22. Notify the customer of successful pausing of subscription.

**2. 3 Potential issues**

- A refund is issued when it shouldn’t be.

- The amount of the refund is incorrect.

- A refund is not issued when it should be.

- A subscription is charged when it shouldn’t be (e.g. when it is cancelled or paused).

- A subscription is not charged when it should be.

- The amount charged for a subscription is incorrect.

- Customer is not notified of refunds or subscription charges and is confused as to what is going on with their account.

**2. 4. Comments:**

From a business perspective, a situation where a subscription isn’t charged would be of the highest priority, because it results in loss of income. It is also important that refunds are processed in a timely manner to prevent complaints and reputation damage for the company.

=================================================================================

**3. Story 3**

As a Driver

I need to know how many of an item needs to be delivered to a customer

So that I can fulfil their orders

**3.1 Assumptions:**

A1. The total items to be delivered on any round cannot exceed the space available in the delivery vehicle.

A2. The quantity of items ordered cannot exceed the existing stock.

**3.2 Acceptance criteria (AC):**

AC1. The driver can see the correct quantity of items to be delivered for single-item orders.

AC2. The driver can see the correct quantity of items to be delivered for multiple-item orders.

AC3. The driver can see the correct quantity of items to be delivered for one-time orders.

AC4. The driver can see the correct quantity of items to be delivered for repeat orders.

AC5. The quantity of items to be delivered in the driver app matches the quantity chosen by the customers in the customer app.

AC6. The quantity of the items to be delivered is not 0.

AC7. The quantity of the items to be delivered is a positive integer.

AC8. The quantity of the items to be delivered is view-only.

AC9. The quantity of items to be delivered is not editable by the driver.

AC10. The quantity of items to be delivered does not exceed (X) (where X is derived from a previously discussed and agreed upon number that takes into consideration stock levels, the maximum number of items that can be delivered at a time and the space available in the delivery vehicle).

**3.3 Potential issues:**

- Mismatch of the quantity - incorrect values passed to the driver app from the customer app – e.g. 20 instead of 2 (2.0)

- Quantity of items to be delivered not displayed.

- Quantity of items to be delivered displayed incorrectly – e.g. a decimal number shown for boxes of eggs.

- Incorrect quantity shown on orders for multiple items.

- Quantity of items editable for drivers – it could cause situations where the driver accidentally edits the quantity.

- Edge case: Since the quantity of items to be delivered should be view-only in the driver app, the customer app could potentially be affected, and the customers might not be able to edit the quantity of an item in the customer app.

- Edge case: The total quantity of items to be delivered on one round is too large or exceeds the space in the delivery vehicle.

Final note: The above acceptance criteria could be written as Given / When / Then scenarios, e.g.

Given I am logged in as a driver

And orders for my round have been placed

When I check my deliveries

Then I should see how many of an item I should deliver.