**Requirements Analysis**

Functional Requirements:

-Scan QR code of their inventory: to be able to easily add items to the inventory

-Add item to their virtual shopping basket: to keep track of what the customer would like to order when they are ready to checkout

-Order other sizes of same item from other stores: to allow the customer to change their mind regarding size and store for the items

-Check availability of item: so an item that isn’t available can’t be ordered

-Can checkout and pay for purchase through connection to online payment service: so a customer can purchase items in their shopping basket

-Use the Payment Service API to handle payments for the customer: easy handling of electronic payments using a third-party API

-Reasonably secure the app: keep all user info encrypted and on a secure server

-Staff can attach QR code and update store inventory with items: so that a new item can be associated to a QR code within store, and that the change in inventory is reflected on the system

Non-functional Requirements:

-Product: Assume users can be up to 1 million, but use a scalable backend

-To allow a large enough amount of users to concurrently use the service

-Performance: 1-2 seconds response time

-So that data is provided to the user within a reasonable amount of time

-Space 100MB

-To ensure that the application isn’t too large to download and store on the device.

-Usability: Clear and simple UI

-To make using the app as easy and accessible as possible.

-Security: Users should be able to log in to their account with a password and have their data stored securely in a database

-In order for customers to have peace of mind that their data is safe.

-Legislative: Ensure we adhere to GDPR requirements

-So that the EU doesn’t shut down our app due to poor data management standards

-Regulatory: Check with regulators of the fashion industry that the app meets any of their requirements

-To ensure that information provided to the customer about the products meets the standards of completeness and comprehensiveness

-Organisational: 6 employees, 6 months to design, develop and test

-A reasonably sized team to build and maintain the app

-Dependability: Ensure we test the app so that it’s reasonably bug-free, and maintain and update the app after release

-To make sure that the customer has the best experience possible using the app.

Ambiguous:

-What is the process of checking out? Add items to basket – view basket – guest checkout / signup – delivery address – payment handler

-How do users make an account? Email, password, address, name, phone number

-What can/can’t the staff do? Linking QR code to an item on the system, can’t add to bag and checkout for customer

-Number of expected users? Up to 1 million, scalable backend in case there are more

Customer

Description: Person who buys items from store

Actors: Customer

Basic flow of events:

1. Scans QR code
2. View availability in the current store
3. Check availability in other stores
4. Add items to basket
5. View/edit basket
6. Checkout
7. Login/create account/guest
8. Pay

Extensions:

Faulty code —> System informs the actor, not allowed to proceed

Can’t find item in any store —> Inform the user

Can’t connect/error with payment system handler —> Inform the user

Already have an existing account —> Inform the user, ask for another email

Password doesn’t meet requirements —> Ask to come up with a new password

Password doesn’t authenticate —> Show user error message from payment handler

Post conditions:

Deduct items from stock, email user with order confirmation

Special conditions:

Connected to Internet, uses compatible device

Staff

Description: An employee of the store

Actors: Staff

Preconditions: Work there, training in use the system, account with the system

Basic flow contents:

1. Create an account
2. System authenticates user as an employee
3. Scan QR code
4. Find item in the database to match with
5. Link the code and the item
6. Update inventory
7. System acknowledges the data has been added
8. Employee signs out of the system

Extensions:

System fails to authenticate the actor —> “Try again”, can’t proceed.

Item to be added has been already added —> Add a different item.

Not a clear QR code —> Scan again.

Can’t connect to stock management system —> Inform user, try again.

Post condition: Item exists in the system’s database.

Special requirements: Connected to Internet, employee of the company, compatible device.

Payment Handler

Description: Handler of the customer’s payment.

Actor: Payment Handler

Preconditions: Order with outstanding payment

Basic flow of events:

1. Takes card details and the amount owed.
2. Verifies and carries out the payment.
3. Inform the system if it has been successful or not.

Extensions:

Payment wasn’t successful —> Inform the user of why.

Can’t connect to payment system —> Inform the user, try again.

Post conditions: Payment successful, order completes.

Special requirements: None

Stock Management System:

Description: Database containing stock information.

Actor: Stock Management System

Preconditions: None

Basic flow of events:

1. Told whether to add or reduce stock.
2. Add or reduce stock by the specified amount.
3. Inform the system that the changes have been made.

Extensions:

Reduce stock and the quantity is already 0 —> Inform the user, item not available.

Item doesn’t exist —> Inform the system that the product doesn’t exist and needs to be added and linked to.