Date: [Date]
Time: [Time]
Group: 06

Group members:

MaoYe (Team Leader), YuHao Li (Team secretary), ZeJin Deng,

YuZe Yang, XinRan Tao, Sebastian Currie.

Catalog	Page

Requirement analysis

Functional requirements

Store detection • Store provided Wi-Fi connection can be used to detect which store is the customer in. • Phones' build in GPRS can be used to detect which store is the customer in.
QR code scanning • QR codes can be scanned by phones' build in QR code detector. • Failure in scanning shows a "try again" information. • Successfully scanning loads the product main page.
Product main page • Product main page displays *Name, Picture, Material, Description, Price, Size, Availability* of the wanted product. • *Availability* of the product is with respect to the shop that the customer currently in. • Customers can select *Quantity* of the wanted product. • "Add to shopping bag" button to add the product to the virtual shopping bag, with respect to the selected quantity. • Create an *Order* if the shopping bag is empty before adding the product above. • Failure in adding product shows a "fail to add product" error message. • Success in adding product shows a green tick (complete sign). • Regardless the *Availability*, choice-box is provided to switch between stores (include online store). • Every time the *Store* is changed, reload the product main page with respect to the selected *Store*.
Virtual shopping bag • Displays all the added products in a list from top to bottom. • Every product should be labelled with their *Name, Picture, Price, Availability, Quantity, Store*. • "Remove" button can be used to remove unwanted products. • Check-box is provided to select the products for check-out. • *Total Price* should be displayed at the bottom of the product list. • When checking out, products with *Availability* of "not available" should be filtered out automatically and leave in the shopping bag. • Check-out will ask for login or the customer *Address, Mobile-phone, Email*. • Check-box provided for any *Terms and conditions* agreements. • Failure in check-out shows information "check-out error", and display the cause of the failure. • Success in check-out invoke the *Payment API*. • Success in payment sends a *Confirmation* of the *Order* to the *Email* provided by the customer.
Store employee inventory management • Employees are able to log into the system by providing their *Username, Password, StoreID*. • Employees can update quantity available of products.

Non-functional requirements

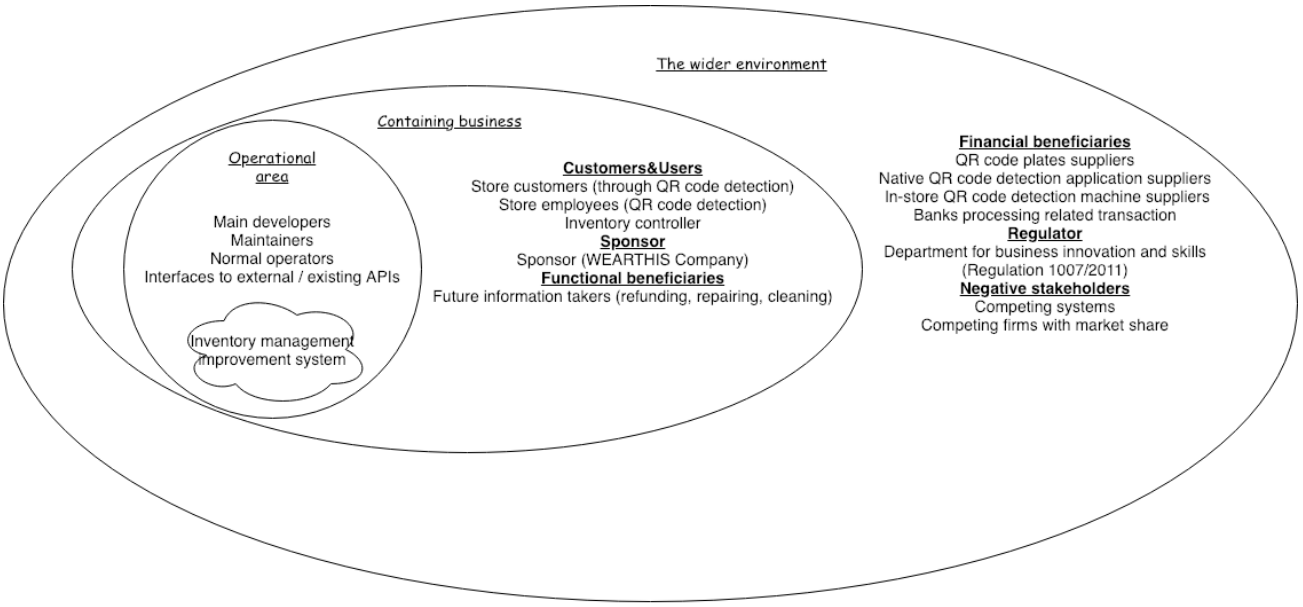
Performance • Store detection should be done within 2 minutes. • Scanning QR code should be done in 5 seconds. • Loading main page, loading virtual shopping bag should be done in 1 second. • Order creation and order confirmation should be done in 5 seconds. • Requirements above should be hold for request rate under 250 request per second.
Security • Customer information, *Email* and *Order* should be encrypted and stored. • Employees login information *Username, Password, StoreID* should be encrypted and stored. • All customers and employees' information should be controlled with respect to the *General Data Protection Regulation 2018 (GDPR)*.

Domain requirements

Domain definition:

The domain is "fashion inventory management", the motivation of this domain analysis is to meliorate inventory management upon the existing systems and QR code detection technology, with increase in on-line sales as an outgrowth, for the fashion retailer *WEARTHIS*.

Customers, users and stakeholders:



Terminology, methodology, standards and analysis

- *Inventory management system*:

System that tracks inventory movement across all sales channels in real time.

- *EU Regulation 1007/2011:*

This regulation specified rules about labelling textile products information.

When developing search or ordering features there are noticeable requirements:

1. expressions “100%”, “pure” and “all” should be considered as the same, without other similar expression are allowed to be used.
2. “Wool” product should have further derivatives such as “fleece wool”, “virgin wool”, and same for other materials.
3. The expression of a material may also be referred to the indication of its composition.
4. Textile products containing two or more textile components of two or more different textile fibers must bear a label stating the fiber composition of each component.

Further details might refer to the regulation guidance:

https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/513963/BIS-16-193-textile-labelling-regulations-guidance.pdf

- *ISO/IEC 18004:2015 QR Code standard*

This standard specifies the QR Code symbology characteristics, symbol format, dimensional characteristics, error correction rules, reference decoding algorithm, production quality requirements, and user selectable application parameters.

Further details might refer to the ISO: https://www.iso.org/standard/62021.html

The QR code generator/reader class should be in such a standard.

- *ISO 8559-3:2018 Cloth size standard*

This standard describes the principles of the establishment of tables for body measurements, defines the categories of tables (related to intervals), and lists the population groups (infants, girls, boys, children, women, men) and sub-groups to be used for developing ready-to-wear garments.

Further details: https://www.iso.org/standard/67334.html

Maximum request rate analysis

How it is done: Taken the annual sales / revenue within the UK market from three large clothing retailer --- H&M, Next, Newlook --- then dividing by factors according to assumptions made:

1. Assuming every order has price from 0 to 100 pounds and taking the average 50.
2. Assuming the daily working hours are 10 hours.
3. Assuming that all the sales / revenue come from in-store purchases.
4. Assuming our system are going to cover all of the in-store purchases as maximum.
5. Assuming that each purchasing process contains 10 requests to the CPU.

Retailers	Annual sales / revenue (Millions)	Annual working Second	Price per order	Maximum visiting rate (per second)	Maximum request rate (per second)
Next (2016)	2374	365 * 10 *3600 = 13140000	50	4	40
Next (2017)	2305	13140000	50	4	40
H&M (2016)	15058	13140000	50	23	230
H&M (2017)	14580	13140000	50	22	230
Newlook (2016)	1491	13140000	50	2	20
Newlook (2017)	1455	13140000	50	2	20

(https://www.nextplc.co.uk/~/media/Files/N/Next-PLC-V2/documents/2017/Copy%20of%20WEBSITE%20FINAL%20PDF.pdf

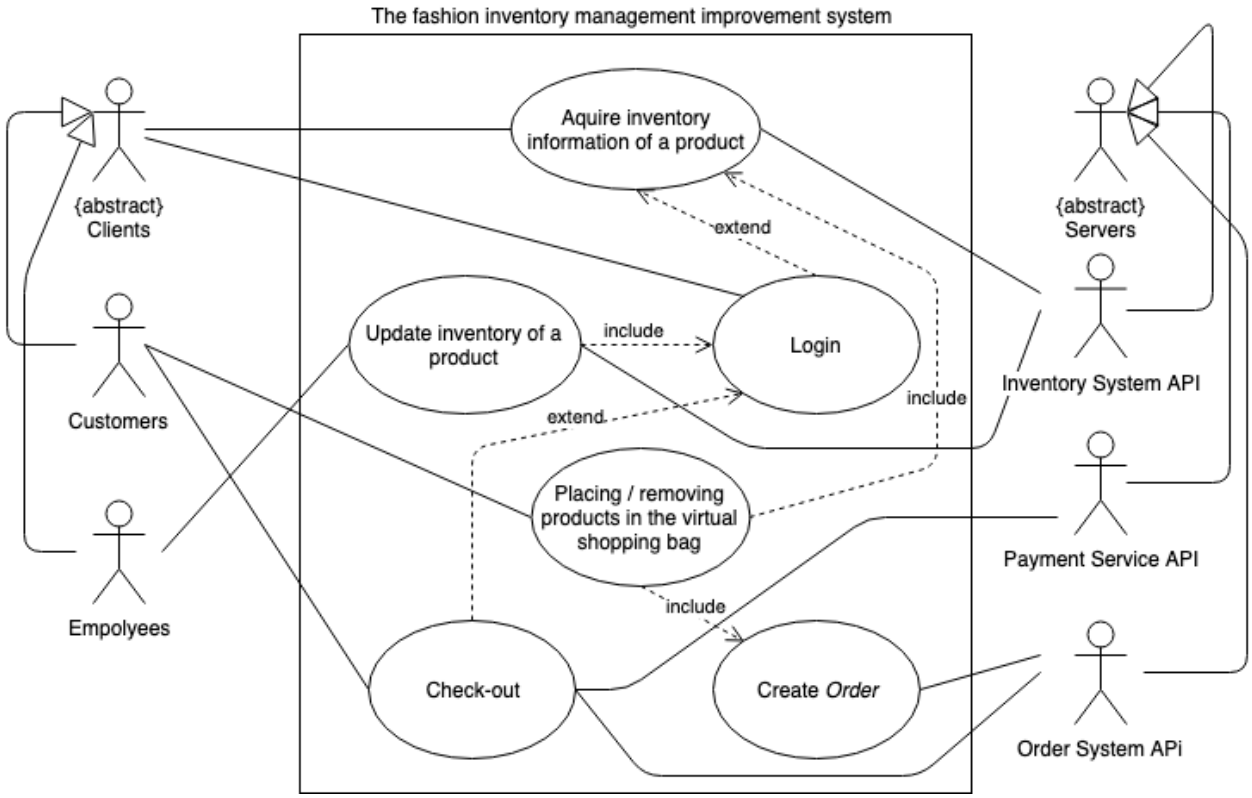
https://about.hm.com/content/dam/hmgroup/groupsite/documents/en/Annual%20Report/Annual%20Report%202017.pdf

https://www.newlookgroup.com/system/files/uploads/financialdocs/fy17-annual-report-06-06-2017.pdf)

As it shown, the maximum estimated request rate (per second) stays under 250 request per second, but the safe position for our system will be determined only after concrete tests such as load test or actual performance test. Necessary cascading failure prevention should also be applied to the system.

Use case

Use case diagram



Use case description

System	The fashion inventory management improvement system
Actors	• Customers • Employees • Inventory System API • Order System API • Payment Service API
Use cases	• Acquire inventory information of a product. Actors: Clients, Inventory System API 1) Trigger: Customers / employees scan the QR code attached on the product, or employees access through inventory manage page. 2) getAvailability method of Inventory System API is invoked. 3) The Inventory System API returns the available number of the product of a specific size. 4) For customers, only “available” or “not available” is shown, for employees accessing through the manage page, actual number can be shown.

- Login
  Actors: Clients
  1) Trigger: Employee supply *Username, Password, StoreID* and Customer supply *Username, Password* and click "log in" button.
  2) *Username / Password / StoreID* is invalid, retry getting the same login details up to three times, record error then wait for five minutes before allowing next login.
  3) *Username / Password / StoreID* is valid, enter the inventory manage page.
  4) The page provide sections for displaying the current available number of a product or update inventory by entering items added to a product.
  5) Stay in the inventory manage page until the employee press "quit" button.

- Update inventory of a product.
  Actors: Employees, Inventory System API
  1) Trigger: Employee at the inventory manage page enter number of items need to be added, press "update" button.
  2) Inventory System API invoke method to update availble number of the product.
  Failed update, System return an error message and report the cause of the failure.
  3) Successful update, System return a success message.

- Placing / removing products in the virtual shopping bag.
  Actors: Customers
  1) Trigger: Customer select size, quantity and store they want to order the product from at the product main page, then press "Add to shopping bag".
  2) For products that are "not available", customer will be asked whether still want to add them to the shopping bag and suggest order from stores which have inventories.
  3) Intends to add the product to the *Order* object.
  4) Failure in adding the product shows a "fail to add product" message, and the possible cause of the failure.
  5) Success in adding the product shows a green tick or a "product added" message.
  6) Customer can view the shopping bag by clicking "view shopping bag" button.
  7) Customer should be able to click "remove" button next to a product to remove products from the shopping bag.

- Check-out.
  Actors: Customers, Payment Service API, Order System API
  1) Trigger: Customer click "check-out" button at the shopping bag page.
  2) Filter out automatically all the "not available" products and inform the customer.
  3) Ask the customer to log in, or provide *Address, Mobile-phone, Email*.
  4) Validate the customer details, if no error occurred then invoke the *Payment API*.
  5) Validate whether the payment success or not, if success then submit order and send order *confirmation* to the *Email* provided by the customer.

- Create *Order*.
  Actor: Order System API.
  1) Trigger: First product intends to be added to the shopping bag.
  2) Add the first product to the *Order*.