

Problem Domain and Requirements

David Chun (dc37875), Catie Cowden (cec3447), Ben Fu (byf69), Zi Zhou
Wang (zw3948)

EE 382C Software Architectures
Dr. Barber

February 14, 2018

Table of Contents

Table of Contents	2
1. The Domain Problem and Stakeholders	3
1.1. Domain Description	3
1.2. Envisioned System	3
1.3. Stakeholders for the Architecture	4
1.3.1. Consumers	4
1.3.2. Producers	5
1.4. Functional Requirements	6
1.4.1. Function Specifications	6
1.4.2. Scenario Specifications	22
1.4.3. Essential Scenario	25
1.5. Qualities and Constraints	25
1.6. Deployment Environment	28

1. The Domain Problem and Stakeholders

1.1. Domain Description

Online buying and selling has become an important part of many people's lives. Students and parents rely on the internet to acquire and sell textbooks at affordable prices, virtual stores allow people to shop from the comfort of their homes without the pressure of a salesperson, and online marketplaces provide a new and more convenient venue for the exchange of virtually all types of goods and services. The internet can be a great resource for shoppers looking to expand their choices of products to buy and an invaluable way for saving money. Online stores are highly competitive not only with other online stores, but also with brick-and-mortar competitors.

Both businesses and customers have embraced online sales as a cheaper and more convenient way to shop. The online retail market has been saturated with successful companies such as Amazon, who deal primarily with non perishable goods that can be stored in, and shipped from warehouses. However, the online market for perishable goods such as groceries has yet to be explored. According to a report from Food Marketing Institute and Nielsen American, consumers are expected to spend up to \$100 billion on online grocery items by 2025. This is where the design of our architecture comes in. Imagine picking out your groceries from the inventory of a supermarket while sitting at home comfortably in your pajamas. Our architecture is designed to transform the trite task of picking out your weekly groceries into a simpler, more enjoyable experience.

1.2. Envisioned System

The goal of an online grocer is to allow private individuals to purchase grocery products online, and arrange for either pickup from the grocery store or a delivery service to the private individual's residence. Our online grocer will be a variation of E-commerce, which is based on a brick and mortar supermarket, warehouse, or distribution center. Our users are given a web based interface in which they can browse and select their desired product based off proximity, price range, and customer ratings. They can then compare between products of different grocery stores, and select their mode of delivery or pick up.

The most basic method of ordering involves customers placing orders online and picking up their ready orders in the store on their way home. In store pickups will be offered for same day shopping as it does not require shipping or delivery, as such, it will also have a lower service fee. Our online grocer architecture will also support local delivery services, with drivers capable of delivering goods on demand. With this method, consumers can pick their delivery times and location, with advanced ordering being encouraged through discounted delivery fees as it allows for more efficient delivery routes.

Our online grocer can be managed by the supermarket's internal team. Additionally, our platform will be able to be re-used and customized to reflect the company brand and provide unique features.

1.3. Stakeholders for the Architecture

1.3.1. Consumers

1. Pre-family/No Family Customers

Expectations: The system should be based on modern, common, and convenient technology. It should integrate easily into a busy lifestyle. Pre-family/no family customers may be open to the system suggesting new products. The system should be stable and secure.

Role with respect to the development project: End User

Role with respect to the domain: Customer

Part of organization: Customer

Impact: Dissatisfied customers. Loss of business.

2. Family Customers

Expectations: The system should be simple to use. Family customers are looking to save time. They dislike time-consuming data entry with every use. They appreciate when the system makes it easy to buy their weekly essentials. The system should be stable and secure.

Role with respect to the development project: End User

Role with respect to the domain: Customer

Part of organization: Customer

Impact: Dissatisfied customers. Loss of business.

3. Third Age & Retired Customers

Expectations: The system should be simple to use. Products should be easy to locate. Third age/retired customers appreciate interacting with the system by following a clear workflow, from start to finish. The system should be stable and secure.

Role with respect to the development project: End User

Role with respect to the domain: Customer

Part of organization: Customer

Impact: Dissatisfied customers. Loss of business.

4. Brick and Mortar Store Employees

Expectations: The employee's goal is to meet the needs of the customer. The system should provide clear information on what products are purchased, when they will be picked up, and whether any payment is owed. The employee wants to avoid situations where purchased items are out of stock or order information is unclear.

Role with respect to the development project: End User, Order assembler

Role with respect to the domain: Supplier, Order assembler

Part of organization: Store employee

Impact: Employees feel ill-equipped to meet customer needs. Frustrated workforce.

5. Brick and Mortar Store Management

Expectations: The manager expects the system to meet the expectations of the customers and the employees. The system should provide data to aid in evaluating the value-add of the system. The system should provide a competitive advantage in the local market.

Role with respect to the development project: End User, Business expert

Role with respect to the domain: Supplier, Supervisor

Part of organization: Management

Impact: Dissatisfied managers may refuse to adopt the system. Loss of business.

6. Corporate Management

Expectations: The system should reach a new segment of the market, drive bottom line revenues, and increase competitive advantage. The system should be self-sustaining and supportable. The cost of implementing the system should be reasonable. The system should be easily expanded to support future brick and mortar locations.

Role with respect to the development project: Business expert

Role with respect to the domain: Product Owner

Part of organization: Senior Management

Impact: Low ROI.

1.3.2. Producers

1. Product Owner

Expectations: The system should meet all external requirements as defined by the stakeholders.

Contributions: The product owner meets with stakeholders and defines requirements for the engineering team. The product owner also defines the priorities for which requirements need to be met first.

Impact of Failure: If the visions of the product owner are not met, the system will most likely not meet the requirements, which means the whole project will be unsuccessful.

2. Product Designer

Expectations: The system should look and feel friendly to the consumers.

Contributions: The product designer works with the design team to create mockups of the application. The designer also works with the engineering team in evaluating the implementation of the design.

Impact of Failure: If the needs of the product designer are not met, consumers are likely not going to be pleased with the look of the product. Look and feel is very important in this project, so if the design is bad then customers likely will not use the product at all.

3. Software Engineering Team

Expectations: The system should implement the design and requirements defined by the product owner and designer. The system should also be scalable and easy to update.

Contributions: The engineering team does the bulk of the work in developing the actual application and infrastructure. Ideally, the team should have daily meetings

to collaborate and also meet with the product owner and designer frequently to make sure they are implementing the requirements correctly.

Impact of Failure: If the software engineering team's vision is not met, then the final product is nonexistent.

4. QA Engineering Team

Expectations: The system should be well-tested inside development and production environments.

Contributions: The QA team is in charge of testing the product thoroughly in dev, QA, and production environments. They impose that the software will operate correctly within different operating environments and devices.

Impact of Failure: If the QA visions are not met, then the software will likely face many issues once deployed in production. In addition, many hard-to-track bugs related to specific devices and operating systems will likely surface.

1.4. Functional Requirements

1.4.1. Function Specifications

1. Create a Customer Account: This function allows a customer to create a new account within the system. After receiving the user ID and password, this function verifies that the username is unique across all other users. If so, the account is created and available for future use.

Input:

User ID [External]
Password [External]
Event (Request to create account) [External]

Output:

Account [Required for *all* functions]

Performers:

Customers, regardless of market segment.

Resources:

Computer/laptop/phone, Internet access, Data store with user information

Performance Locations:

Not restricted, but typically in the home of the customer.

Precondition:

Event (User requested Create a Customer Account function) and Input (Username) and Input (Password)

Postcondition:

Event (New account created) and Event (User logged into system)

2. Authenticate User: This function allows customers with accounts to login to the system.

Input:

User ID [External]
Password [External]

Output: Event (Request to login) [External]
 Login Session [Required for *all* functions]
Performers: Customers, regardless of market segment.
Resources: Computer/laptop/phone, Internet access, Data store with user information
Performance Locations: Not restricted, but typically in the home of the customer.
Precondition: Event (User requested Authenticate User function) and Event (User requested Create a Customer Account function) and Input (User ID) and Input (Password)
Postcondition: Event (User logged into system)

3. Choose a Preferred Brick & Mortar Store: This function allows customers to select the brick & mortar store where they plan to pick up their purchases. The side effects of choosing a store include price, inventory, and available pickup times.

Input: Store [External]
 Event (Request to choose a Preferred Store) [External]
 Brick and Mortar Stores database [Get All Brick and Mortar Stores]
Output: Store [Schedule a Pickup Time, Select Item in Store]
Performers: Customers, regardless of market segment.
Resources: Computer/laptop/phone, Internet access, Data store with all available brick & mortar locations
Performance Locations: Not restricted, but typically in the home of the customer.
Precondition: Event (User requested Choose a Preferred Brick & Mortar Store function) and Event (User requested Authenticate User function) and Input (Store)
Postcondition: Data Updated (User's preferred store)

4. Select Item in Store: This function allows customers to select an item from a product catalog.

Input: Event (Request to select an item) [External]
 Store [Choose a Preferred Brick & Mortar Store]
 Item Inventory [Internal]
Output: Selected Item [Add Item to Cart, Check Expiration Date, Add Item to Wishlist, Verify Item Available in Inventory]
Performers:

Customers, regardless of market segment.

Resources:

Computer/laptop/phone, Internet access, Data store with product catalog for customer's preferred store

Performance Locations:

Not restricted, but typically in the home of the customer.

Precondition:

Event (User requested Select Item in Store function) and Event (User requested Choose a Preferred Brick and Mortar Store function)

Postcondition:

Item is selected

5. Add Item to Cart: This function allows customers to add an item to their shopping cart to purchase later. This function requires that user is logged in, has chosen a preferred store, and has selected an item. The item must be available in their preferred store for it to be added to the cart.

Input:

Item [Select Item in Store]
Event (Request to Add Item to Cart) [External]
Item Can be Added to Cart [Verify Item Available in Inventory]

Output:

Shopping Cart (updated) [Make a Purchase]

Performers:

Customers, regardless of market segment.

Resources:

Computer/laptop/phone, Internet access, Data store with all available brick & mortar locations

Performance Locations:

Not restricted, but typically in the home of the customer.

Precondition:

Event (User requested Add Item to Cart function) and Event (User requested Select Item in Store function) and Event (Verify Item Available in Inventory function) and Input (Item)

Postcondition:

Data Updated (Item in User's cart)

6. Verify Item Available in Inventory: This function verifies that an item from the product catalog is available in the preferred store's inventory.

Input:

Item [Add Item to Cart]
Event [Request to Add Item to Cart]
Store [Choose a Preferred Brick & Mortar Store]

Output:

Item Can be Added to Cart [Add Item to Cart]

Performers:

Employees, and other authorized parties.

Resources:

Data Store with product inventory for all items in product catalog for all stores

Performance Locations:

System

Precondition:

Event (Add Item to Cart function requested inventory availability) and Event (User requested Add Item to Cart)

Postcondition:

Event (Item added or failed to add to cart) and Notification (Inform user of failure)

7. Update Store Inventory: This function will update the store inventory when a transaction is processed.

Input:

Event (Made Successful Purchase)
Items [Add Item to Cart]
Store [Choose a Preferred Brick & Mortar Store]

List of Item Successfully Purchased [Make a Purchase]
Inventory [Internal]

Output:

Inventory Updated [Internal]

Performers:

Employees, and other authorized parties.

Resources:

Data Store with product inventory for all items in product catalog for all stores

Performance Locations:

System

Precondition:

Event (User requested to Make a Purchase)

Postcondition:

Data Updated (Available inventory for Store)

8. Schedule a Pickup Time: This function allows customers to select what time they will pick up their purchases.

Input:

Time [External]
Store [Choose a Preferred Brick & Mortar Store]
Brick and Mortar Store database [Get All Brick and Mortar Stores]

Output:

Update Pending Orders [Marks Order as Picked Up]
Event (New Pending Order is Available)

Performers:

Authenticated customers, regardless of market segment.

Resources:

Computer/Laptop/Phone, Internet Access, Data store with available pick up times for Store

Performance Locations:

Not restricted, but typically in the home of the customer.

Precondition:

Event (User requested to Make a Purchase)

Postcondition:

Data Updated (Expected Pickups updated for Employee use)

9. Create an Employee Account: This function allows an employee to create a new account within the system. After receiving the employee ID and password, this function verifies that the request to create an employee account is from an authorized party. If so, the account is created and available for future use.

Input:

Password [External]
Employee ID [External]
Event (Request to create account) [External]

Output:

Account [Required for *all* functions]

Performers:

Employees, and other authorized parties.

Resources:

Computer/laptop/phone, Internet access, Data store with employee information

Performance Locations:

System

Precondition:

Event (Employee requested Create an Employee Account function) and Input (Employee ID) and Input (Password)

Postcondition:

Event (New account created) and Event (User logged into system)

10. Authenticate Employee: This function allows employees with accounts to login to the system.

Input:

Employee ID [External]
Password [External]
Event (Request to login) [External]

Output:

Login Session [Required for *all* functions]

Performers:

Employee, and other authorized parties

Resources:

Computer/laptop/phone, Internet access, Data store with employee information

Performance Locations:

System

Precondition:

Event (Employee requested Authenticate Employee function) and Event (Employee requested Create a Employee Account function) and Input (Employee ID) and Input (Password)

Postcondition:

Event (Employee logged into system)

11. Mark Order as Picked Up: This function allows employees to mark items as picked up. This function requires that the employee is logged in, and is from the designated store. The

item must have been marked as available for pick up in the designated store to be marked as picked up.

Input:

Order [External]
Event (Request to Mark Order as Picked up) [External]
Pending Orders [Update Pending Orders]

Output:

Pending Orders (updated)

Performers:

Employee, with authorized store location.

Resources:

Data Store with list of all currently available orders for pick up

Performance Locations:

System

Precondition:

Event (User picks up order from location), Event (Employee marks order as picked up) and Input (Order)

Postcondition:

Data Updated (Orders available for pickup updated)

12. Update System with new inventory: This function allows employees to update the system with new inventory. This function requires that the employee is logged in, and is from the designated store. The item must be one that is recognized by the database.

Input:

Items [External]
Event (Adds Items to inventory) [Internal]

Output:

Item inventory (updated) [Internal]

Performers:

Employee, with authorized store location.

Resources:

Data Store with current item database

Performance Locations:

System

Precondition:

Event (Employee updates database) and Input (Items)

Postcondition:

Data Updated (Items added to database)

13. Track Previous Purchases: This function allows a user to view their previously purchased items. This function requires a customer to be logged in.

Input:

User [Customer account to use][Internal]

Output:

Previously purchased items [External]

Performers:

Authenticated customers, regardless of market segment

Resources:

Database of customer purchased items

Performance Locations:

Not restricted

Precondition:

User (User is properly authenticated and identified)

Postcondition:

Previous purchases retrieved (Items returned from the database)

14. Create an Administrator Account: This function allows corporate management to create an administrator account for the system which adds privileges to a normal Employee account. After verifying the creation credentials supplied in the admin ID field, the account is created.

Input:

Password [External]

Administrator ID [External]

Event (Request to create account) [External]

Output:Admin Account [Required for *all* functions]**Performers:**

Administrator, management, and other authorized parties.

Resources:

Computer/laptop/phone, Internet access, Data store with administrator information

Performance Locations:

System

Precondition:

Event (Administrator requested Create an Admin Account function) and

Input (Admin ID) and Input (Password)

Postcondition:

Event (New admin account created) and Event (User logged into system)

15. Authenticate Administrator: This function allows an administrator to login to the system.

Input:

Admin ID [External]

Password [External]

Event (Request to login) [External]

Output:Login Session [Required for *all* functions]**Performers:**

Administrator or an authorized party

Resources:

Computer/laptop/phone, Internet access, Data store with administrator information

Performance Locations:

System

Precondition:

Event (Employee requested Authenticate Administrator function) and Event

(Administrator requested Create a Administrator Account function) and

Input (Administrator ID) and Input (Password)

Postcondition:

Event (Administrator logged into system)

16. Send Notification: This function sends a notification to a user, notifying them of some event occurrence.

Input:

Event (What to Send) [Internal/External]
User [Internal]

Output:

Notification (Sent to User specified) [External]

Performers:

Administrator or Employee

Resources:

Computer/laptop/phone, Internet access, database of all users (Admin, Employee, and Customers)

Performance Locations:

System (processes notification and some events), not restricted (for requests or other updates)

Precondition:

Event (Status has changed, inventory is updated, customer changed order, etc) and User (An end user must be specified so the notification can be sent to them)

Postcondition:

Notification (Sent to User specified)

17. Schedule Refund: This function allows a customer to request a refund for a purchase. The items requested must come from their recent purchases and the user requested a refund must be authenticated properly.

Input:

Item from Past Order [View Orders]
Request (Reason for Return) [External]

Output:

Return Ticket [Process Return Ticket, Issue Refund]

Performers:

Authenticated customers, regardless of market segment,
Employee

Resources:

Computer/laptop/phone, Internet access, Data store with customer purchase information

Performance Locations:

System

Precondition:

User (properly authenticated and signed in) and Item (must be a recent purchase by the customer and eligible for return)

Postcondition:

Return Ticket (A return ticket is created in the database and employees can view return request)

18. Process Return Ticket: This function allows an employee to view a customer's return ticket and evaluate it for return. The customer should be present with the item to return and their payment information must be valid for a return to be successfully completed.

Input:

Return Ticket [Schedule Refund]

Payment Information [~~Enter Payment Information~~Internal]**Output:**

Item returned for post-return processing [External]

Performers:

Employee

Resources:

Computer/laptop/phone, Internet access, database for returned items, return tickets, and customer information

Performance Locations:

System (location is flexible as long as both customer and employee are presents)

Precondition:

Return Ticket (Customer must have requested a valid return) and Payment Information (Customer must have a valid payment method on file).

Postcondition:

Item is returned (Item add to returned items, return ticket is marked completed)

Commented [1]: Would it be better to have the return issued to the payment initially used for the transaction?**Commented [2]:** Either way. Somehow you need to connect it back to the order then, and the payment info they entered**Commented [3]:** Would it be better to have the return issued to the payment initially used for the transaction?**Commented [4]:** Either way. Somehow you need to connect it back to the order then, and the payment info they entered**19. Add Item to Wishlist:** This function allows a customer to mark an item for their wishlist, allowing them to schedule a notification when the item is in inventory.**Input:**

User Account [Internal]

Selected Item [Select Item in Store]

Notification Request [External]

Output:

Item added to wishlist [Internal]

Performers:

Authenticated customers, regardless of market segment.

Resources:

Computer/laptop/phone, Internet access, database of all possible inventory

Performance Locations:

Not restricted (wherever the user wishes)

Precondition:

User (User is properly authenticated), Item (Item exists in possible inventory), and Notification Request (Should the system schedule a notification if previously unavailable item is restocked)

Postcondition:

Wishlist updated (database of user wishlists) and Notification scheduled (if requested)

20. Check Expiration Date: This function allows a customer or employee to check the expiration date of any item in the inventory.**Input:**

Selected Item [Select Item in Store]

Output:

Expiration Date [External]

Performers:

Customer, employee, administrator, or other parties

Resources:

Computer/laptop/phone, Internet access, Data store with inventory and expiration date information

Performance Locations:

Not restricted (anywhere a properly identified user can access the internet)

Precondition:

Event (Employee requested Authenticate Administrator function) and Event (Administrator requested Create a Administrator Account function) and Input (Administrator ID) and Input (Password)

Postcondition:

Expiration date provided

21. Make a Purchase: This function allows a customer to finalize their purchase.

Input:

Confirmation [External]
Shopping Cart [Add Item to Cart]
Payment Information [Enter Payment Information]

Output:

Event [Send Notification of order confirmation to customer]
Event (Made Successful Purchase) [Update Store Inventory]
Event (Request Delivery) (optional)
Order Number [Enter Delivery Address]
List of Item Successfully Purchased [Update Store Inventory]

Performers:

Authenticated customers, regardless of market segment

Resources:

Computer/laptop/phone, Internet access, Data store with inventory and expiration date information

Performance Locations:

Not restricted (anywhere a properly identified user can access the internet)

Precondition:

Event (Customer confirms purchase) and Event (Customer payment information entered) and Input (Customer ID) and Input (Password)

Postcondition:

Event (Purchase record added to database) Event (Customer is charged)

22. Enter Payment Information: This function allows a customer to enter their payment information

Input:

Payment Information [External]

Output:

Payment information entered into database [Save Payment Information, Pay for Delivery]

Performers:

Authenticated customers, regardless of market segment

Resources:

Computer/laptop/phone, Internet access, Data store with inventory and expiration date information

Performance Locations:

Not restricted (anywhere a properly identified user can access the internet)

Precondition:

Event (Customer enters payment information) and Input (Customer ID) and Input (Password)

Postcondition:

Database updated with customer's payment information

23. Confirm Delivery: This function allows a customer to confirm delivery of the their order

Input:

Event (Request Delivery) (optional) [Make a Purchase]

Output:

Event [Notification of delivery sent to customer]

Performers:

Authenticated customers, regardless of market segment

Resources:

Computer/laptop/phone, Internet access, Data store with inventory and expiration date information

Performance Locations:

Not restricted (anywhere a properly identified user can access the internet)

Precondition:

Event (Requests delivery or order), Event (Customer has paid for delivery), Event (Customer has entered delivery address) and Input (Customer ID) and Input (Password)

Postcondition:

Database updated with customer's delivery request

24. Pay for Delivery: This function allows a customer to pay for an order delivery

Input:

Payment Information [External or Enter Payment Information]

Output:

Payment information entered into database
Event (Completed Payment)

Performers:

Authenticated customers, regardless of market segment

Resources:

Computer/laptop/phone, Internet access, Data store with inventory and expiration date information

Performance Locations:

Not restricted (anywhere a properly identified user can access the internet)

Precondition:

Event (Customer enters payment information) and Input (Customer ID) and Input (Password)

Commented [5]: Need a way to lookup the price of the delivery

Commented [6]: Need a way to lookup the price of the delivery

Postcondition:

Database updated with customer's payment information

25. Enter Delivery Address: This function allows a customer to enter their delivery address

Input:

Delivery address [External]
Order Number [~~Make a Purchase~~External]

Output:

Delivery Address Information [Save Delivery Address Information]

Performers:

Authenticated customers, regardless of market segment

Resources:

Computer/laptop/phone, Internet access, Data store with inventory and expiration date information

Performance Locations:

Not restricted (anywhere a properly identified user can access the internet)

Precondition:

Event (Customer enters delivery address) and Input (Customer ID) and Input (Password)

Postcondition:

Database updated with customer's delivery address

26. Enable Two-Factor Authentication: This function allows a customer to enable two-factor authentication for more secure logins.

Input:

User Toggle [External]

Output:

Two-Factor Authentication enabled

Performers:

Authenticated customers, regardless of market segment, Employees, Administrators

Resources:

Computer/laptop/phone, Internet access

Performance Locations:

Not restricted (anywhere a properly identified user can access the internet)

Precondition:

Event (Customer already logged in)

Postcondition:

Database updated with two-factor authentication setting and second device

27. Save Payment Information: This function allows a customer to save their payment information for more convenient payments in the future

Input:

Payment Information [Enter Payment Information]

Output:

Payment information saved

Performers:

Authenticated customers, regardless of market segment

Resources:

Computer/laptop/phone, Internet access

Performance Locations:

Not restricted (anywhere a properly identified user can access the internet)

Precondition:

Event (Customer already logged in)

Postcondition:

Database updated with saved payment information

28. Save Delivery Address Information: This function allows a customer to save their delivery address for more convenient deliveries in the future

Input:

Delivery Address Information [Enter Delivery Address]

Output:

Delivery Address Information

Performers:

Authenticated customers, regardless of market segment

Resources:

Computer/laptop/phone, Internet access

Performance Locations:

Not restricted (anywhere a properly identified user can access the internet)

Precondition:

Event (Customer already logged in)

Postcondition:

Database updated with saved address(es)

29. View Popular Products: This function allows a customer to view items that are popular (large amount of views and/or purchases)

Input:

Item Inventory [Internal]

Item View Counts [Internal]

Item Purchase Counts [Internal]

Output:

Popular Products [External]

Performers:

System (automatic/manual trigger)

Customers, regardless of market segment.

Resources:

Internet Access

Performance Locations:

Not restricted

Precondition:

Event (Customer already logged in)

Event (At least one item has been viewed and/or purchased)

Postcondition:

Customer home page updated with popular products section

30. View Similar Products: This function allows a customer to view items that are similar to ones they have purchased.

Input:

Item Inventory [Internal]

Output:

Similar Products [External]

Performers:

System (automatic/manual trigger)
Customers, regardless of market segment.

Resources:

Internet Access

Performance Locations:

Not restricted

Precondition:

Event (Customer already logged in)
Event (Customer has viewed or purchased an item)

Postcondition:

Customer home page updated with similar products section

31. Create Subscription: This function allows a customer to easily reorder an item by selecting a subscription option at checkout (similar to Amazon's subscription items)

Input:

Item Inventory [Internal]

Output:

User Subscriptions (updated) [Edit or Delete Subscriptions, Reorder subscriptions, View Subscriptions]

Performers:

Authenticated customers, regardless of market segment

Resources:

Phone/computer, Internet Access

Performance Locations:

Not restricted (anywhere with internet access)

Precondition:

Event (Customer already logged in)
Event (Customer has added item to cart)
Event (Customer has reached checkout page)

Postcondition:

Subscription registered to database

32. Edit or Delete Subscriptions: This function allows a customer manage his or her current subscriptions including editing and/or deleting them.

Input:

User Subscriptions [Internal]

Output:

User Subscriptions (updated) [Reorder Subscription, View Subscriptions]

Performers:

Authenticated customers, regardless of market segment

Resources:

Phone/computer, Internet Access

Performance Locations:

Not restricted

Precondition:

Event (Customer already logged in)

Event (Customer has registered one or more subscriptions)

Postcondition:

Subscription changes saved to database

33. Reorder Subscription: This function describes the replacement of a subscription order for the user, or when the user reorders a subscription manually.

Input:

User Subscriptions [Internal]

Item Inventory [Internal]

Output:

Subscription order placed

Performers:

Store (automatic trigger)

Authenticated customers, regardless of market segment

(manual trigger)

Resources:

Phone/computer, Internet Access

Performance Locations:

Not restricted

Precondition:

Event (Customer already logged in)

Event (Customer has registered one or more subscriptions)

Postcondition:

Subscription and order saved to database

34. View Orders: This function allows a customer to view orders that he or she has placed.

Input:

Order History [Internal]

Output:

List of orders the user has placed [External]

Performers:

Authenticated customers, regardless of market segment

Resources:

Computer/Phone, Internet Access

Performance Locations:

Not restricted

Precondition:

Event (Customer already logged in)

Event (Customer has placed at least one order)

Postcondition:

None

35. View Subscriptions: This function allows a customer to view subscription orders that he or she has placed.

Input:

User Subscriptions [Internal]

Output:

List of subscriptions the user has placed [External]

Performers:

Authenticated customers, regardless of market segment

Resources:

Computer/Phone, Internet Access

Performance Locations:

Not restricted

Precondition:

Event (Customer already logged in)

Event (Customer has at least one subscription)

Postcondition:

None

36. Compare Items: This function allows a customer to compare similar items among the inventories of stores

Input:

Items [Internal]

Item Inventory [Internal]

Output:

List of items and their attributes specified by the user [External]

Performers:

Customers, regardless of market segment.

Resources:

Computer/Phone, Internet Access

Performance Locations:

Not restricted

Precondition:

Event (Customer already logged in)

Event (Customer has at least one item to compare)

Postcondition:

None

37. Issue Refund: On the successful return of an item, this function will repay the amount of the item bought by the customer.

Input:

Return Ticket [Schedule Refund]

User Payment Info [Internal]

Output:

Repayment issued to customer

Performers:

Store

Resources:

Internet Access, Payment database

Performance Locations:

Not restricted

Precondition:

Event (Customer already logged in)

Event (Customer has successful return ticket processed)

Postcondition:

Event (Payment issued to customer)

38. Update Pending Orders: Once a customer schedules a delivery, the new pending order must be combined with the existing pending orders.

Input:

New Pending Order [Schedule a Pickup time]

Output:

Pending Orders [Select Order in Store, Mark Orders as Picked Up]

Performers:

Employees, Store

Resources:

Computer/laptop/phone, Internet access

Performance Locations:

Not restricted

Precondition:

Event (Employee already logged in)

Event (New Pending Order is Available)

Postcondition:

Event (Pending Orders Updated)

39. Get All Brick and Mortar Stores: This returns a list of all the Brick and Mortar Stores currently signed up for the service under a specific provider, allowing customers to pick a preferred store.

Input:

Event (Requests for Stores) [External]

Output:

Brick and Mortar Store database [Choose a Preferred Brick & Mortar Store, Schedule a Pick Up Time]

Performers:

Authenticated Customers, Employees, Store

Resources:

Computer/laptop/phone, Internet access

Performance Locations:

Not restricted

Precondition:

Event (Customer is already logged in)

Event (Customer requested brick and mortar store selection)

Postcondition:

None

1.4.2. Scenario Specifications

1. Buy An Item

Sequence of Functions:

1. *Create a Customer Account*
2. *Authenticate User*
3. *Choose Preferred Brick & Mortar Store*
4. *Select Item in Store*
5. *Add Item to Cart*

6. *Verify Item Available in Inventory*
7. *Schedule a Pickup Time*
8. *Make Purchase*
9. *Update Store Inventory*

Environment: This scenario occurs when a customer, regardless of market segment, wishes to make a purchase using the online system.

Precondition: New customer wishes to make purchase.

Post-condition: New customer purchases chosen items online.

2. Complete a Customer Pick-up

Sequence of Functions:

1. *Authenticate User*
2. *Update Store Inventory*
3. *Mark order as picked up*

Environment: This scenario exists when an employee at the store must complete the transaction, handing off the purchases to the customer.

Precondition: Customer purchased item online.

Post-condition: Employee gives them their purchases, and the transaction is complete.

3. Modify Preferred Brick & Mortar Store

Sequence of Functions:

1. *Authenticate User*
2. *Choose Preferred Brick & Mortar Store*

Environment: This scenario occurs when a user relocates or wishes to select a different store (maybe closer to their work location, child's daycare, etc.).

Precondition: A pre-existing customer wishes to make an account modification.

Post-condition: The customer's preferred store is updated in the system.

4. Request a notification on Item Restock

Sequence of Functions:

1. *Authenticate User*
2. *Add Item to Wishlist*
3. *Send Notification*

Environment: This scenario occurs when a user wishes to purchase an item not currently in a store's inventory and wants to be notified when it becomes available.

Precondition: A pre-existing customer wishes to save a unstocked inventory item and wants to be notified when it is back in stock.

Post-condition: The customer is notified of item availability.

5. Return an item

Sequence of Functions:

1. *Authenticate User*
2. *Schedule Refund*
3. *Process Return Ticket*
4. *Issue Refund*

Environment: This scenario occurs when a user wants to return an item they purchased while it is able to be returned.

Precondition: A pre-existing customer wishes to return a valid purchased item.

Post-condition: The customer is issued a refund only if the return ticket process completes successfully.

6. Buy An Item for Delivery

Sequence of Functions:

1. *Authenticate User*
2. *Select Item in Store*
3. *Add Item to Cart*
4. *Verify Item Available in Inventory*
5. *Enter Delivery Address*
6. *Make Purchase*
7. *Pay for Delivery*
8. *Confirm Delivery*
9. *Updates Store inventory*

Environment: This scenario occurs when a customer, regardless of market segment, wishes to make a purchase and have it delivered using the online system.

Precondition: Existing customer wishes to make purchase for delivery.

Post-condition: Existing customer purchases chosen items online for delivery.

7. Request Delivery of Existing Order

Sequence of Functions:

1. *Authenticate User*
2. *Select Order to be Delivered*
3. *Enter Delivery Address*
4. *Pay for Delivery*
5. *Confirm Delivery*

Environment: This scenario occurs when a customer wished to have an existing order delivered.

Precondition: Existing customer wishes to have order delivered.

Post-condition: Existing customer's order is marked for delivery.

8. Create a Subscription

Sequence of Functions:

1. *Authenticate User*
2. *Select Item in Store*
3. *Add Item to Cart*
4. *Verify Item Available in Inventory*
5. *Enter Delivery Address*
6. *Create Subscription*
7. *Pay for Delivery*
8. *Confirm Delivery*
9. *Updates Store inventory*

Environment: This scenario occurs when a customer places a subscription order rather than a one-time normal order..

Precondition: An existing user wishes to place a subscription order.

Post-condition: The order is placed and is registered as a subscription.

9. Reorder a Subscription Manually

Sequence of Functions:

1. *Authenticate User*
2. *View Orders*
3. *Verify Item Available in Inventory*

4. *Place Subscription Order*

5. *Update Store inventory*

Environment: This scenario occurs when a customer wishes to reorder a subscription order earlier than the renew date.

Precondition: An existing user wishes to reorder an existing subscription.

Post-condition: The subscription order is placed.

10. Cancel a Subscription

Sequence of Functions:

1. *Authenticate User*

2. *View Subscriptions*

3. *Edit/Delete Subscriptions*

Environment: This scenario occurs when a customer wants to cancel an existing subscription.

Precondition: An existing user wishes to cancel a registered subscription.

Post-condition: The subscription is removed.

1.4.3. Essential Scenario

The essential scenario would be “Buy an Item” because the sale of items is the core of the grocery business model. The other scenarios are derivative of this scenario. At its core, the system must allow customers to complete transactions-- the support of subscriptions and various delivery processes simply builds upon this functionality,

1.5. Qualities and Constraints

1. Reliability of System

Description:

System should be robust and allow customers to browse product catalog, add items to cart, and successfully complete transactions.

Category:

Performance

Stakeholder Source:

Customer, all market segments

Scope:

Entire System

Quality Evaluation:

How often the system crashes unintentionally

2. Usability of System

Description:

System should be hosted on platforms familiar to users. The desired workflow (performing a transaction to completion) should be intuitive.

Category:

Usability

Stakeholder Source:

Customer, all market segments. Employees.

Scope:

Entire System

Quality Evaluation:

Feedback from users, usability testing.

3. Scalability

Description:

System should be easily extended to support future brick & mortar stores, a larger product catalog, and a larger user base in the case of widespread adoption.

Category:

Performance

Stakeholder Source:

Corporate Management

Scope:

Entire System

Quality Evaluation:

How many steps are required to on-board a new brick & mortar store, add items to the product catalog, or support an increased user base.

4. Security

Description:

System should protect sensitive stored customer data from hacking while also maintaining a secure payment system for the brick & mortar stores.

Category:

Performance

Stakeholder Source:

Customers, corporate management

Scope:

Entire System

Quality Evaluation:

How many system entry points are exposed to potential hacking attempts and how much of the system can a malicious actor access after defeating a gateway.

5. Recoverability

Description:

System should be able to recover from database corruption or deletion with minimal loss of productivity.

Category:

Performance

Stakeholder Source:

Customers, Employees, Corporate Management

Scope:

Entire System

Quality Evaluation:

How long does it take to recover a working system state from catastrophic failure given typical guidance and operating habits.

6. Response Time

Description:

System should offer low latency for all user requests over common network speeds.

Category:

Performance

Stakeholder Source:

Customers, Employees, Corporate Management

Scope:

Entire System

Quality Evaluation:

How long does each function take to execute and how long does a consumer have to wait for visual feedback of their actions.

7. Adaptability

Description:

System should be able to be able to adapt to specific themes and cultures of specific companies or holidays/seasons

Category:

Usability

Stakeholder Source:

Customers, Employees, Corporate Management

Scope:

Entire System

Quality Evaluation:

How long does it take or how difficult it is to adapt the system to a new company with an unique identity.

8. Evaluability

Description:

System should be able to be evaluated and critiqued, by both customers and employees

Category:

Usability

Stakeholder Source:

Customers, Employees, Corporate Management

Scope:

Entire System

Quality Evaluation:

How long does it take or how difficult it is to collect the opinions of customers and employees on the performance of the system.

9. Maintainability

Description:

System should be able to be maintained easily and quickly

Category:

Usability

Stakeholder Source:

Employees, Corporate Management

Scope:

Entire System

Quality Evaluation:

How long does it take or how difficult it is to perform maintenance on the system when needed

10. Availability

Description:

System should always be up and available, except for appropriate circumstances

Category:

Usability

Stakeholder Source:

Customers, Employees, Corporate Management

Scope:

Entire System

Quality Evaluation:

How often, long, or if ever, the system has to be offline and unable to function normally

1.6. Deployment Environment

1. Computing Platforms

Description:

Client and server side system endpoints should be accessible to 99% of prospective users. Thus client software must run on any low end device (1gb ram, dual core processor minimum) with browser capabilities with manageable latency or performance problems. Server side platforms should be stable and provide scalability to host the website for large traffic amounts, making the choice of website hosting choice key to customer satisfaction.

Stakeholder Source:

Consumers require a quick and stable client side implementation and Brick & Mortar stores want easy to configure, fast, and stable website hosts.

2. Operating System/Runtime Environment

Description:

As above, the system should be available to any customer who meets the minimum hardware specifications with internet browser access. However, the system will be optimized for viewing on the all modern web browsers for viewing design.

Stakeholder Source:

Consumers should be able to access the system from a wide variety of devices with a similar experience.

3. Connectivity

Description:

The system should be available to all areas with a minimum of 3G data speeds. This restricts the amount of client and server interactions while still making the system available to almost everyone in the target markets.

Stakeholder Source:

Consumers and Corporate Management both want widespread access to the proposed system.

4. Database

Description:

The system should offer a reliable and scalable database for large numbers of read and write transactions simultaneously, making choices like Oracle SQL and MySQL preferable due to their reliability, familiarity, and customer support.

Stakeholder Source:

Brick & Mortar stores need a stable database to avoid dealing with system instability and the accompanying customer complaints.

5. Physical Location

Description:

While the servers and client side applications themselves are off-site, the system must be able to be configured from any device and location with browser support. This allows for flexibility and transportability. However, some system functions will require on site presence to complete their transaction at a brick & mortar store.

Stakeholder Source:

Customers must be able to access the system from their devices while also knowing where to complete their system transactions at a brick & mortar store.

6. Libraries

Description:

To reduce costs, open source libraries should be used when necessary rather than their commercial counterparts except in case where customer support is necessary and a commercial version has no supported alternative.

Stakeholder Source:

System administrators/developers need easy to maintain and fixable implementations while Brick & Mortar stores/chains want lower cost for system purchase or subscription.