

ALDI

Order Management System



Group 10

Diana Almeida
Rohan Ganu
Vidhi Mistry
Rocky Dayaramani
Garima Bajaj

Table of Contents

1.Executive Summary	3
2. System Proposal and Problem Statement	4
3.Business process model using BPMN for the key business processes	7
4. Context Diagram	8
5. Use Case Diagram.....	9
6. Use Case Descriptions	10
7. Data Dictionary	11
8. Class Diagram.....	12
9.Sequence Diagram	12
Rocky/Garima.....	12
10. Functional Specification	13
11. Interface design	14
12. Database design.....	16
13. Complete Class Diagram	18
14. Software Design	19
15. Project Timeline	30
16. Minutes Of Meeting	31

1.Executive Summary

For over 40 years, ALDI US has stuck to the same guiding principle: Great quality shouldn't come at a high price; rather, great quality should come with everyday low prices. And with more demand comes more stores. Over the past decade, we've nearly doubled in size and by 2022, ALDI will bring its total number of US stores to nearly 2,500. The no-frills grocery shopping experience focuses on customers first - delivering high quality food they're proud to serve their family, responsive customer service, everyday low prices and a quick-and-easy shopping experience with only four to five aisles and all the essentials. Additionally, we partner with a wide variety of growers – including some local farmers – to offer a wide variety of fresh, in-season produce, including organic fruits and vegetables.

Because of its large customer base, the company decided to create an online grocery store in order to reach an even larger demographic and to increase shopping convenience. Unfortunately, alongside this opportunity, ALDI is facing the tremendous task of managing more orders across multiple channels and departments. Specifically, some of the challenges that the company is currently facing includes handling missing order statuses, improving order flaws across all sales channels, aggregating historic purchase order data for reporting and analytics, fulfilling complex order and shipping schedules, and eradicating inconsistent order information among all channels and departments within the organization.

In order to address these functional deficiencies, our team will like to propose an order management system (OMS) that is designed for the ALDI online grocery store. The purpose of an order management system is to secure and execute customer orders in an efficient and cost-effective manner. The order management plan will empower ALDI with features that will successfully transform and add an incredible amount of business value to the ALDI online supermarket. The features that the OMS will provide include a unified database and an integrated platform to enhance order aggregation across the organization. The unified database feature will allow for access to order information across all sales and distribution channels and order tracking for both customer and customer service team. In addition, the storage of historic customer, order, and inventory information in the database will allow ALDI to conduct extensive business reporting, evaluation and, analytics in order to increase business processes and better target potential customers. The OMS platform will also contain a cloud-based feature that will allow ALDI to intelligently scale their multi-channel operations and seamlessly integrate with third-party companies in order to reduce overhead cost and maximize revenue. This means that ALDI will be able to integrate different parts of their order systems with shipping providers such as USPS, DHL, and FedEx as well as various enterprise resource planning and accounting software. With the implementation of the order management system, ALDI can expect to see a growth of more than 40% in monthly sales at the end their third business quarter.

2. System Proposal and Problem Statement

Project Name ALDI Order Management System

Business Need

ALDI has been undergoing rapid change and growth in its online grocery business. During this time, ALDI has been experiencing difficulty managing its online order system as its customer base and order volume grows. Currently, ALDI is struggling with order tracking, inventory fulfillment, and meeting order delivery times and consequently is in an urgent need for a more efficient system. To this end, we are implementing an order management system that will integrate the company's database across all sales channels and will enhance the order processes under which they operate in. Based on our analysis, an order management system(OMS) is the most suitable system for ALDI's current needs for an E-Commerce model due to the rapid growth in online shopping. By implementing our OMS plan, ALDI will have an integrated database for order tracking information, enhanced inventory visibility and monitoring for inventory management, and greater business process performance.

Project Objectives

1. Database Integration

- a. Create a unified database to enhance order aggregation and access to order information across organization and all sales distribution channels.
- b. Simplify data-entry of products and pricing information across different departments
- c. Create unified database to verify and store customer information, product pricing information, and y information across all sales channels and department within ALDI in order to maintain data consistency integrity as well as to avoid data redundancy

2. Inventory Management

- a. Provide order details to warehouses for fulfillment and inventory forecasting in order to avoid stock-outs and in order to route orders from warehouses based on proximity to destination
- b. Share order information among all the distribution centers

3. Order Interface System

- a. Create in-app notification and direct-message tool to provide order status to customers and internal staff.
- b. Grant customers access to order customization, adjustment, and cancellation with in-app tools

- a. Improve payment process and efficiency by provide a platform that provides advanced payments methods that accepts customer orders regardless of money currency.

4. Reporting and Analysis

- a. Utilize data from database to create business reports and trends for targeting potential customers through various social media avenues

Project Scope

In-Scope:

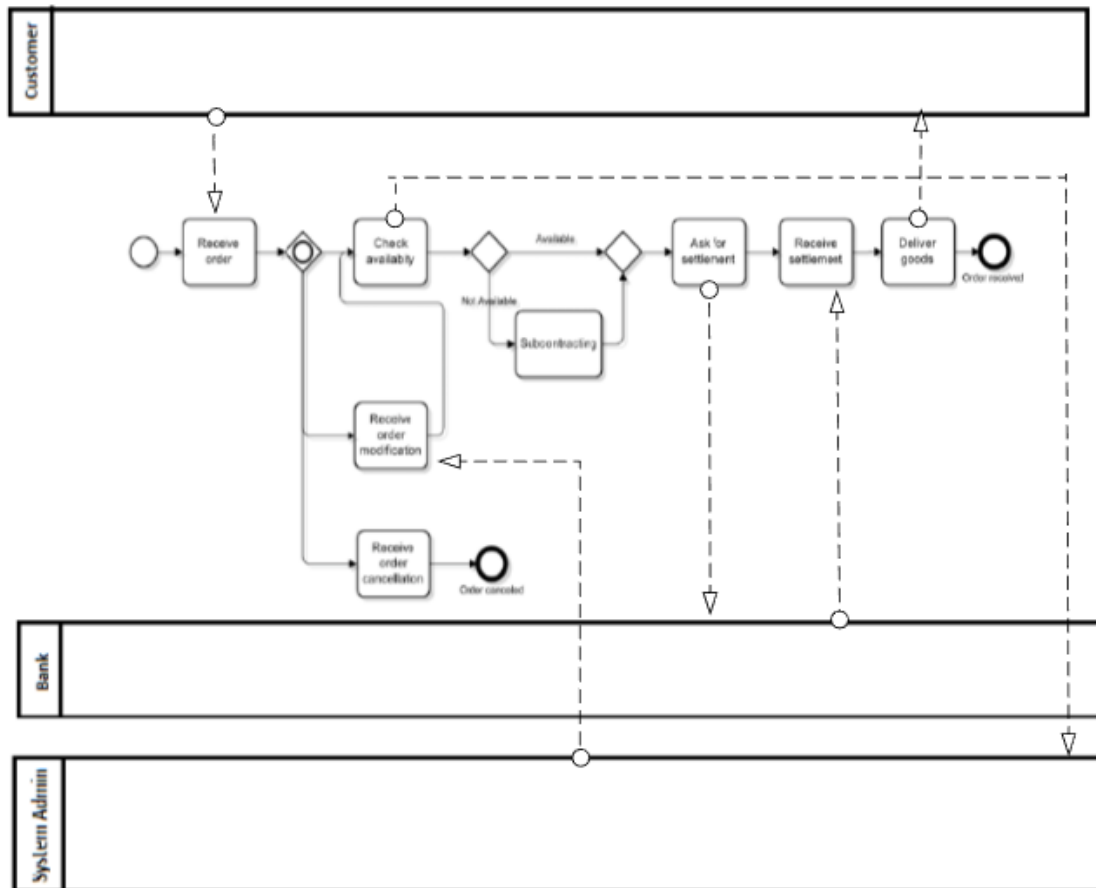
The in-scope of the project includes that the customer and products information should be properly stored in database based on the supply and demand. This system should track the products in the warehouse as well on the system and should update the products based on the market trends. It should also handle missing order statuses and keeping updates to improve order flaws across all sales channels and aggregating purchases of order made previously and fulfilling shipping order schedules. The stakeholders must be ready to supply additional amount if needed and should be ready for any risks in future. The budget should be well calculated beforehand and further processes should run accordingly, keeping in mind the budget. We also need to provide a good customer service and customer interaction and keep the track of complaints and the feedbacks from the customer if any and responding them. Tracking of payments made by customer and receipts received and providing encrypted of customer information for example, the card information and its pin/password should only be accessible to customer and no other person. The estimated time of the project is 90 days, and the budget is around \$200,000. Order flaws across all sales channels will be improved by 95% and this will increase the number of ALDI online shoppers.

PROJECT SUCCESS CRITERIA

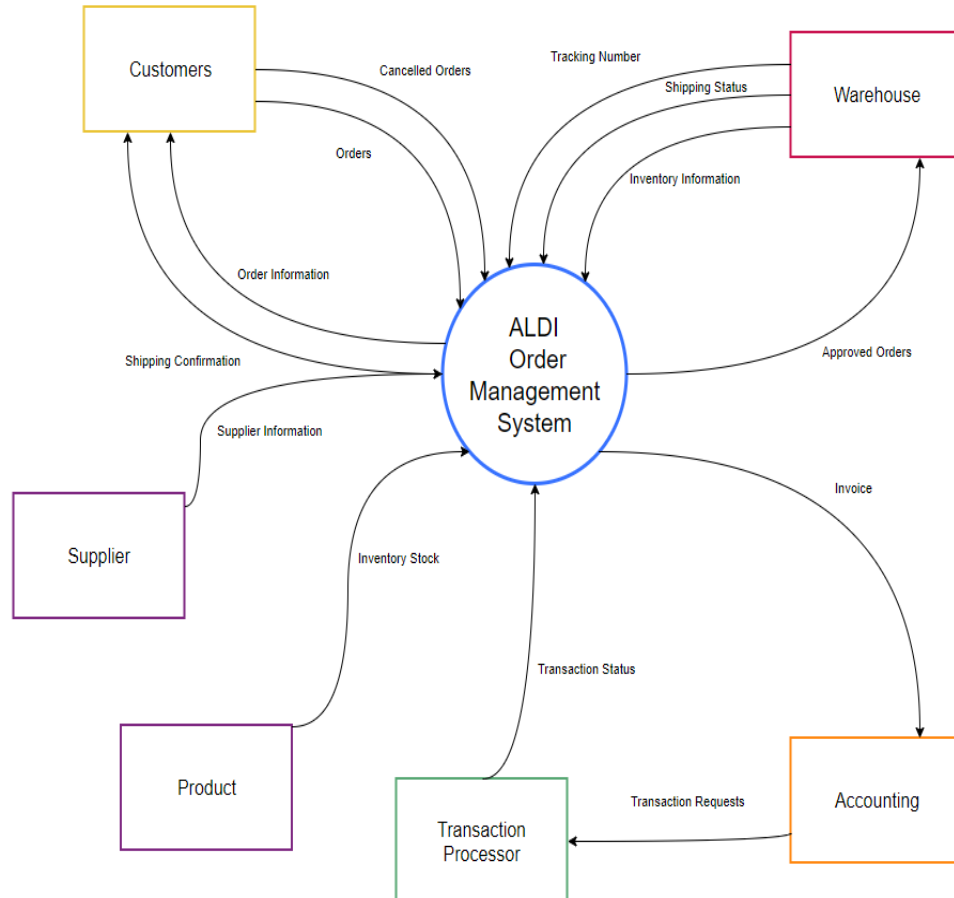
1. The project is considered successful when all or most of the following criteria are met:

- a.** The integrated OMS is compatible with the existing vendors and accounting system. Previous orders and customers' information can be accessed for reporting and analytics.
- b.** All the information is accessible by all the locations so that inventories can be shipped from other warehouses if inventories in the nearest store are not available.
- c.** Warehouse staff can check stock level in sync with order updates, remove spoilage with authorization, and control inventory capacities.
- d.** In-store staff can check limited customers' personal and order information stored in the OMS. Because of a reduction of human error, staff can prepare orders more accurately. This will reduce error rate of orders by 50%.
- e.** New inventories can be added into OMS through code-bar scanning. It will greatly reduce time spent on inventories management by 50%.
- f.** OMS can save annual system maintenance fees by 10% - 30%.
- g.** OMS can safely store customers' orders and reduce loss of order amounts by 90%.
- h.** Items added in customers' shopping carts through their phones can be synced at the same time with the website.
- i.** Customers can track packages using tracking numbers or pick-up at the curbside using their unique order numbers.
- j.** The new OMS will provide ALDI group a comprehensive overview of all customer and order data across the entire order lifecycle and achieve the objective of better customer service.
- k.** Efficient payment processing through invoice-to-cash management.
- l.** The project should be completed by 90 days and in \$200,000 budget.

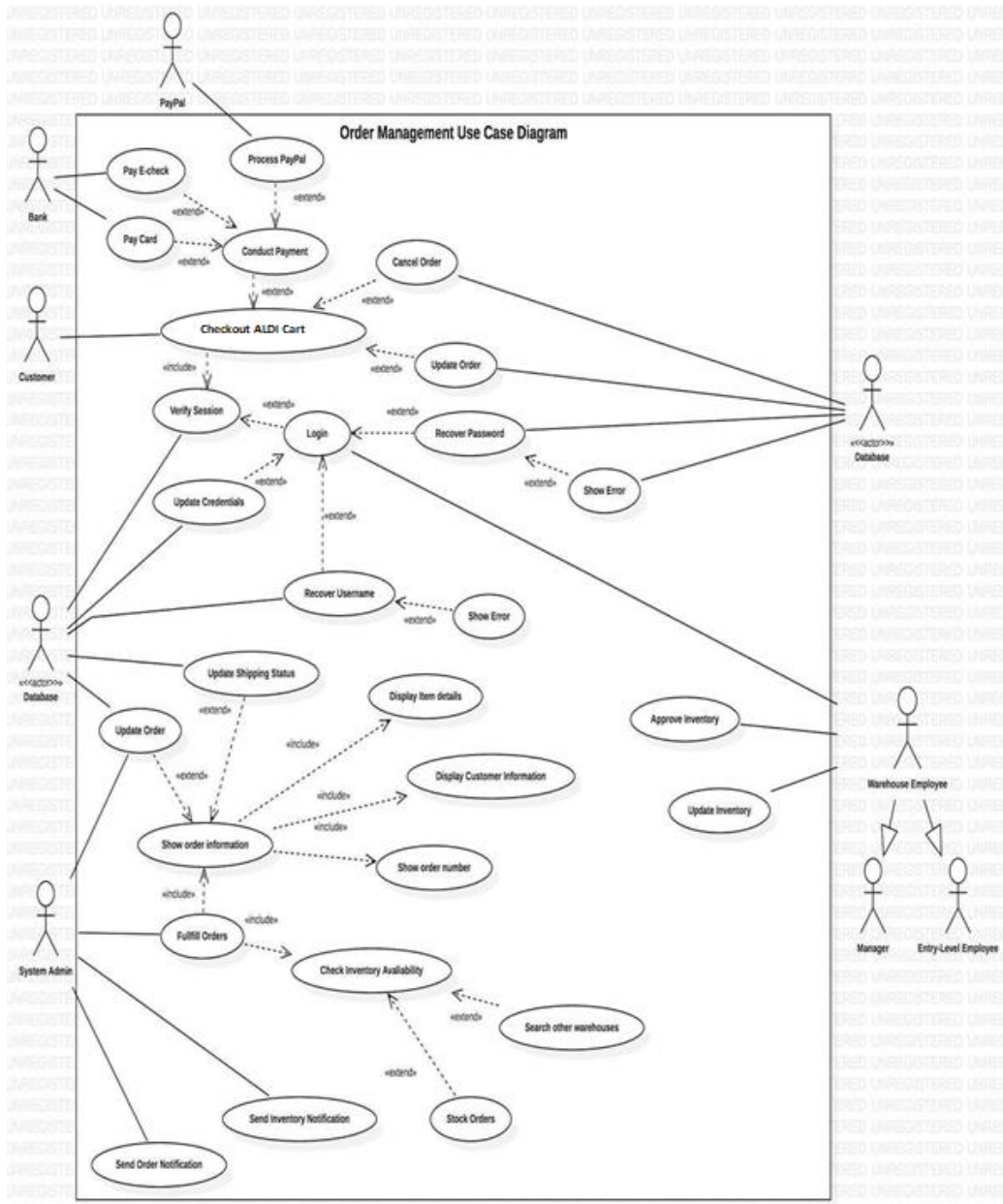
3. Business process model using BPMN for the key business processes



4. Context Diagram



5. Use Case Diagram



6. Use Case Descriptions

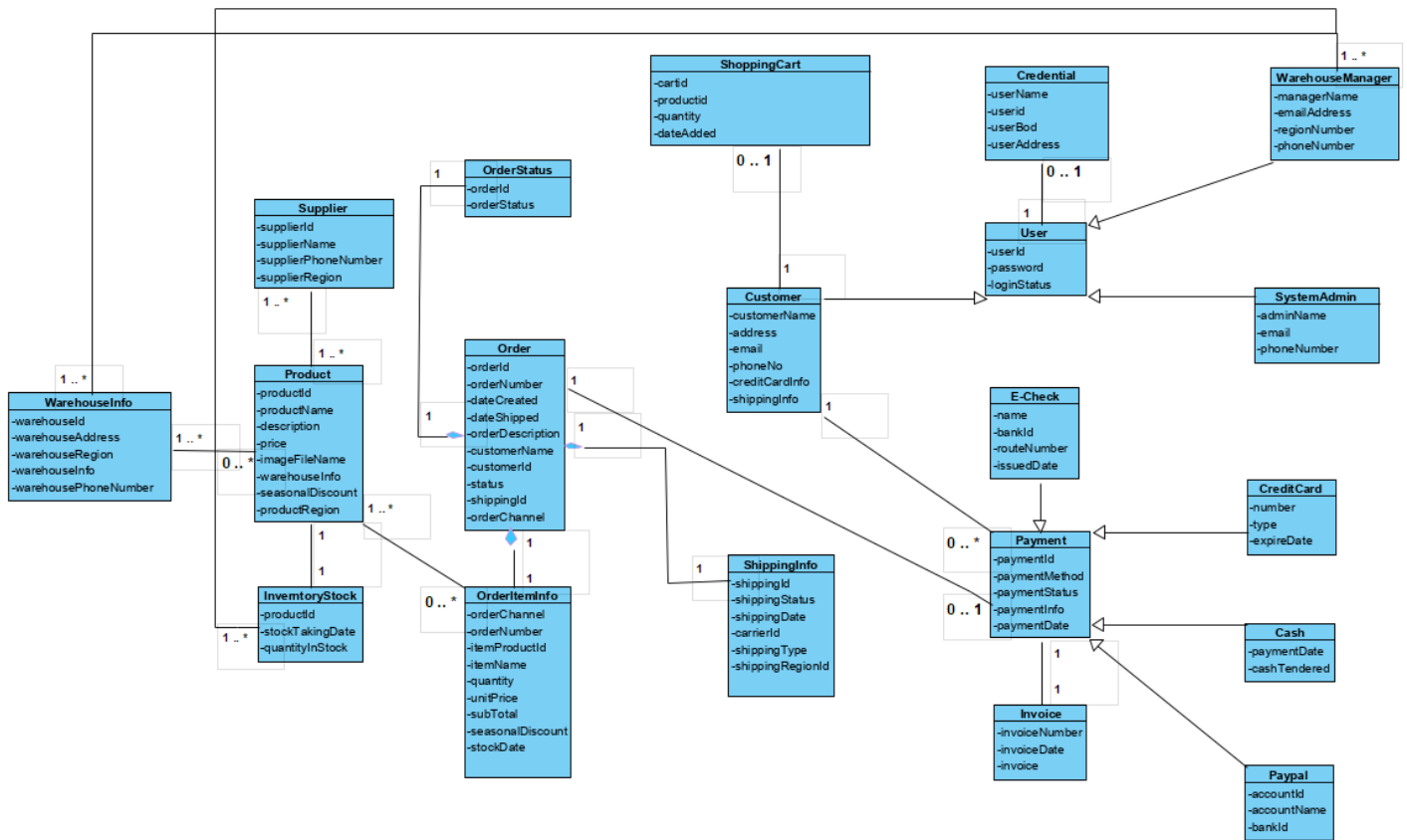
Use Case Name:	Order management use case diagram
Primary Actor:	Customer, System Admin, Database.
Brief Description:	The use case diagram shows an order management system.
Stakeholders:	Store Manager: Jerry Lim Warehouse Manager: Tom Fisher
Normal flow of events:	<p>(1) The Customer checkout ALDI online shopping cart.</p> <ol style="list-style-type: none"> The payment is conducted by Conduct Payment using pay card and e-check to bank or by processing PayPal. The Database verifies session and show order status to Customer. The Customer can cancel order at checkout. The Customer can update order at checkout. <p>(2) The System Admin updates order to the Database and Show the order information.</p> <ol style="list-style-type: none"> The Show order information displays Item Details. The Show order information displays the Customer Information such as name, address, contact. The Show order information displays the Order Number. <p>(3) The Show order information displays the Updated Shipping Status stored in Database.</p> <p>(4) The Show order information displays Fulfilled orders by System admin.</p> <p>(5) The System Admin Checks Inventory Availability into Database.</p> <ol style="list-style-type: none"> The System Admin Search Other Warehouses when order not fulfilled. The System Admin updates the Stock Order to Database. <p>(6) The System Admin Send Order Notifications to Customers during shipment and delivery.</p> <p>(7) The System Admin Send Inventory Notifications to Warehouse Employee.</p> <p>(8) The Warehouse Manager checks the order and approves Inventory into Database.</p> <p>(9) The Warehouse Employee updates the Inventory into Database.</p>
Sub flows:	<p>(1) The Customer Login using his username/password.</p> <p>(2) The Database Show Error if incorrect credentials.</p> <p>(3) The Customer can recover the username</p> <p>(4) The Customer can recover password.</p> <p>(5) The Customer updated credentials stored in Database.</p>

Alternate/Exception flow:	(1) The Customer login is incorrect, it will Show Error. (2) Only the Warehouse Manager approves the Inventory.
---------------------------	--

7. Data Dictionary

Rocky/Garima

8. Class Diagram



9. Sequence Diagram

Rocky/Garima

10. Functional Specification

1. Manage Orders

- 1.1** The system shall store orders' details from all the sales channels in one database, including order's number, customer's home address, customer's billing address, phone number, and order items' details.
- 1.2** The system shall check inventory availability in warehouses within 30 miles of the customer's location.
 - 1.2.1** If quantities in a warehouse are sufficient, a staff of the warehouse shall process the order;
 - 1.2.2** If quantities are insufficient, the system shall search nearby warehouses to meet customers' needs;
 - 1.2.3** The system shall show staff orders' numbers, customer name, home address, and item details.
- 1.3** The system shall accept payment from the bank (credit cards or e-check) or PayPal online payment portal.
- 1.4** The system shall update tracking numbers to customers' accounts once orders have been shipped.
- 1.5** The system shall notify customers when orders have arrived in designated locations.
- 1.6** If a customer cancels an order, the system shall notify the Warehouse Manager.

2. Manage Inventories

- 2.1** When new inventories arrive, quantities are input by a warehouse staff, approved by Warehouse Manager, and automatically updated by the system to the unified warehouse database.
- 2.2** The system automatically calculates the average unit price of a specific inventory.
- 2.3** The system shall notify the Warehouse Manager if certain inventory is low in stock.
- 2.4** If a specific inventory is already out of stock, the system shall show "out-of-stock" on all channels.

3. Manage Accounts

- 3.1** Customers can create accounts using their emails.
- 3.2** When customers log-in to their accounts, the system shall verify credentials against the database.

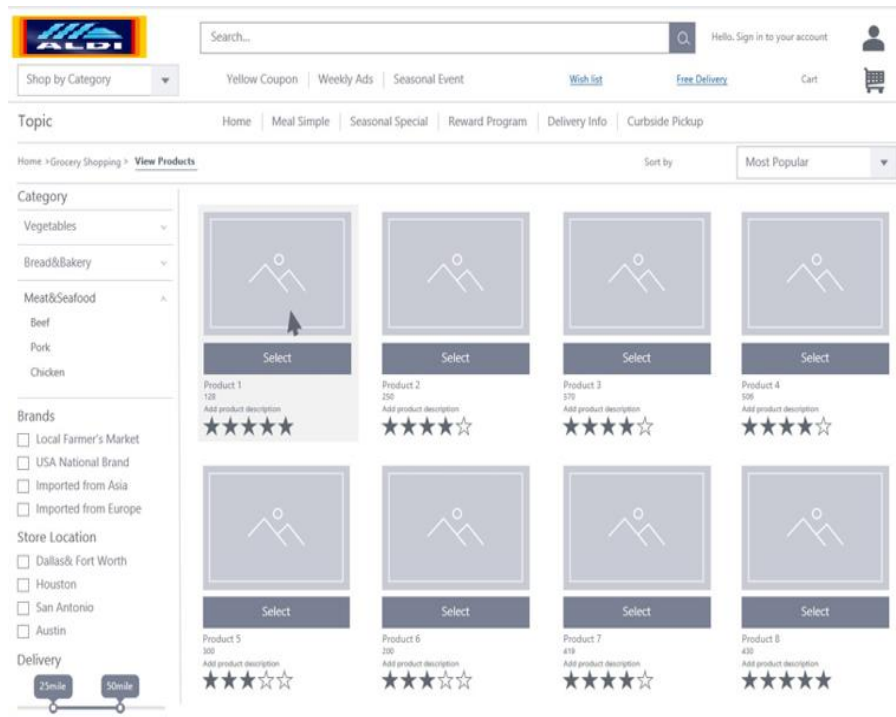
3.2.1 If username and password do not match, the system shall show the error message.

3.2.2 The system can send reset email to customers

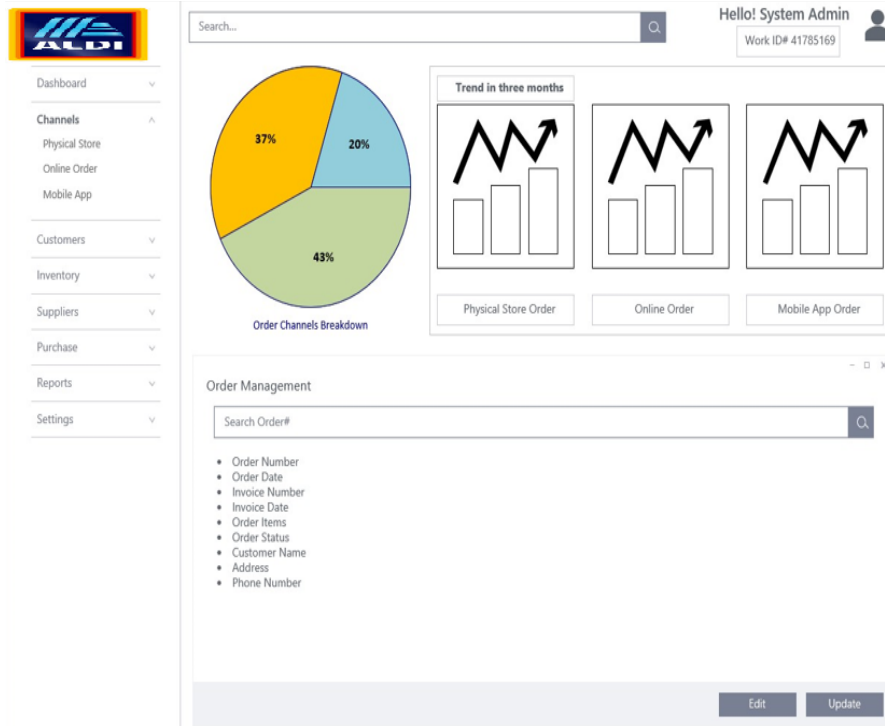
3.3 Customers can change their names, home addresses, contacts, and payment methods.

11. Interface design

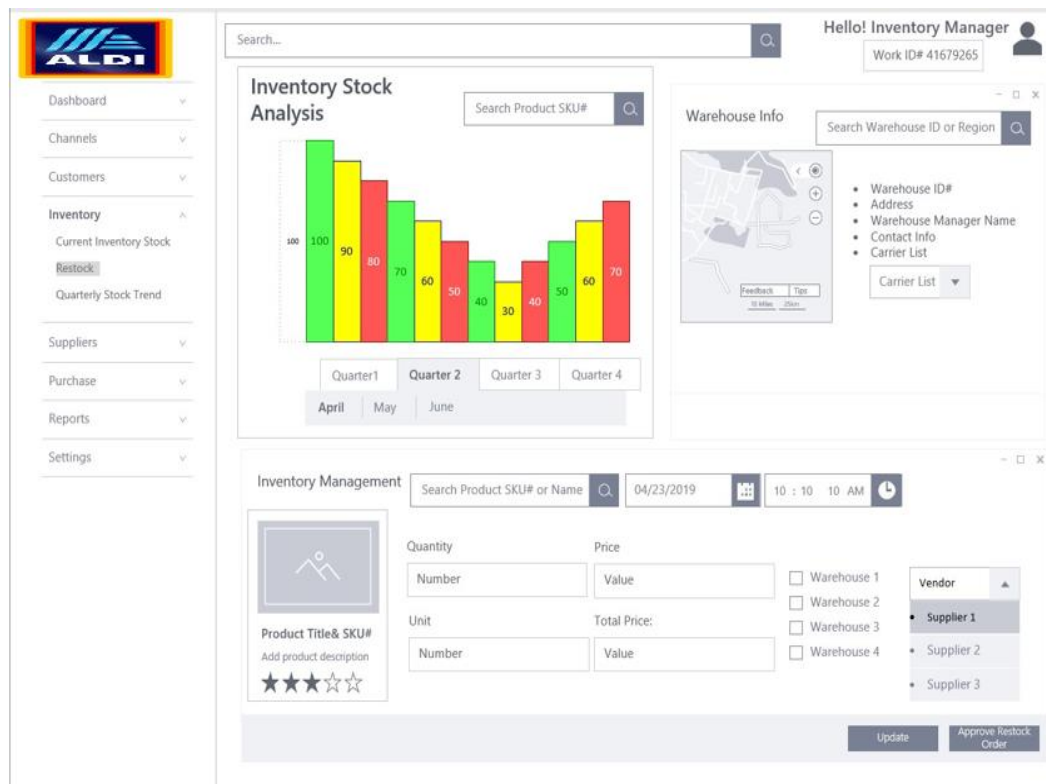
Customer/User Interface



Administrator Interface



Inventory Manager Interface



12. Database design

Shopping Cart (**cartId #**, productId #, quantity# , dateAdded)
cartId# is the Primary Key and should be unique and not null
productId# is the Foreign Key and should be not null

OrderStatus (**orderStatusId #**, orderStatus)
orderId # is the Primary Key and should be unique and not null

Order (**orderId #**, orderNumber, dateCreated, dateShipped, orderDescription, customerName, customerId, status, shippingId, orderChannel)
orderId# is the Primary Key and should be unique and not null
customerId is the Foreign Key and should be not null

Product (**productId#**, productName, description, price, imageFileName, warehouseInfo, seasonalDiscount, productRegion)
productId# is the Primary Key and should be unique and not null
warehouseInfo is the Foreign Key and should be not null

Supplier (**supplierId#**, supplierName, supplierPhoneNumber, supplierRegion)
supplierId # is the Primary Key and should be unique and not null

Customer (**customerName#**, address, email, phoneNo, creditCardNumber, shippingId)
customerName # is the Primary Key and should be unique and not null
creditCardNumber is the Foreign Key and should be not null
shippingId is the Foreign Key and should be not null

Credential (**userId#**, userName , userBod ,userAddress)
userId # is the Primary Key and should be unique and not null

WarehouseManager(**managerName** , managerAddress, regionNumber, phoneNumber)
managerName is the Primary Key and should be unique and not null

WarehouseInfo(**warehouseId#** warehouseAddress, warehouseRegion, warehouseInfo, warehousePhoneNumber)
warehouseId# is the Primary Key and should be unique and not null

User (**userId#** , password, loginStatus)
userId# is the Primary Key and should be unique and not null
password should not be null

SystemAdmin (**adminName** , email, phoneNumber)
adminName is the Primary Key and should be unique and not null

Payment (**paymentId#**, paymentMethod, paymentStatus, paymentInfo, paymentDate)
paymentId #is the Primary Key and should be unique and not null

E-Check (**name** , bankId#, routeNumber, issuedDate)
name is the Primary Key and should be unique and not null

CreditCard (**number#** , type, expireDate)
Number# is the Primary Key and should be unique and not null

Cash (**id#**, paymentDate, cashTendered)
id is the Primary Key and should be unique and not null

PayPal (**accountId#**, accountName, bankId#)
accountId #is the Primary Key and should be unique and not null
bankId# is the Foreign Key and should be not null

Invoice (**invoiceNumber#**, invoiceDate, invoice)
invoiceNumber #is the Primary Key and should be unique and not null

ShippingInfo (**shippingId#**, shippingStatus, shippingDate, carrierId#, shippingType, shippingRegionId)
shippingId #is the Primary Key and should be unique and not null

OrderItemInfo (**ordernumber#**,orderChannel#, shippingStatus, shippingDate, carrierId#, shippingType, shippingRegionId#)

ordernumber #is the Primary Key and should be unique and not null

shippingStatus is the Foreign Key and should be not null

shippingType is the Foreign Key and should be not null

shippingRegionId# is the Foreign Key and should be not null

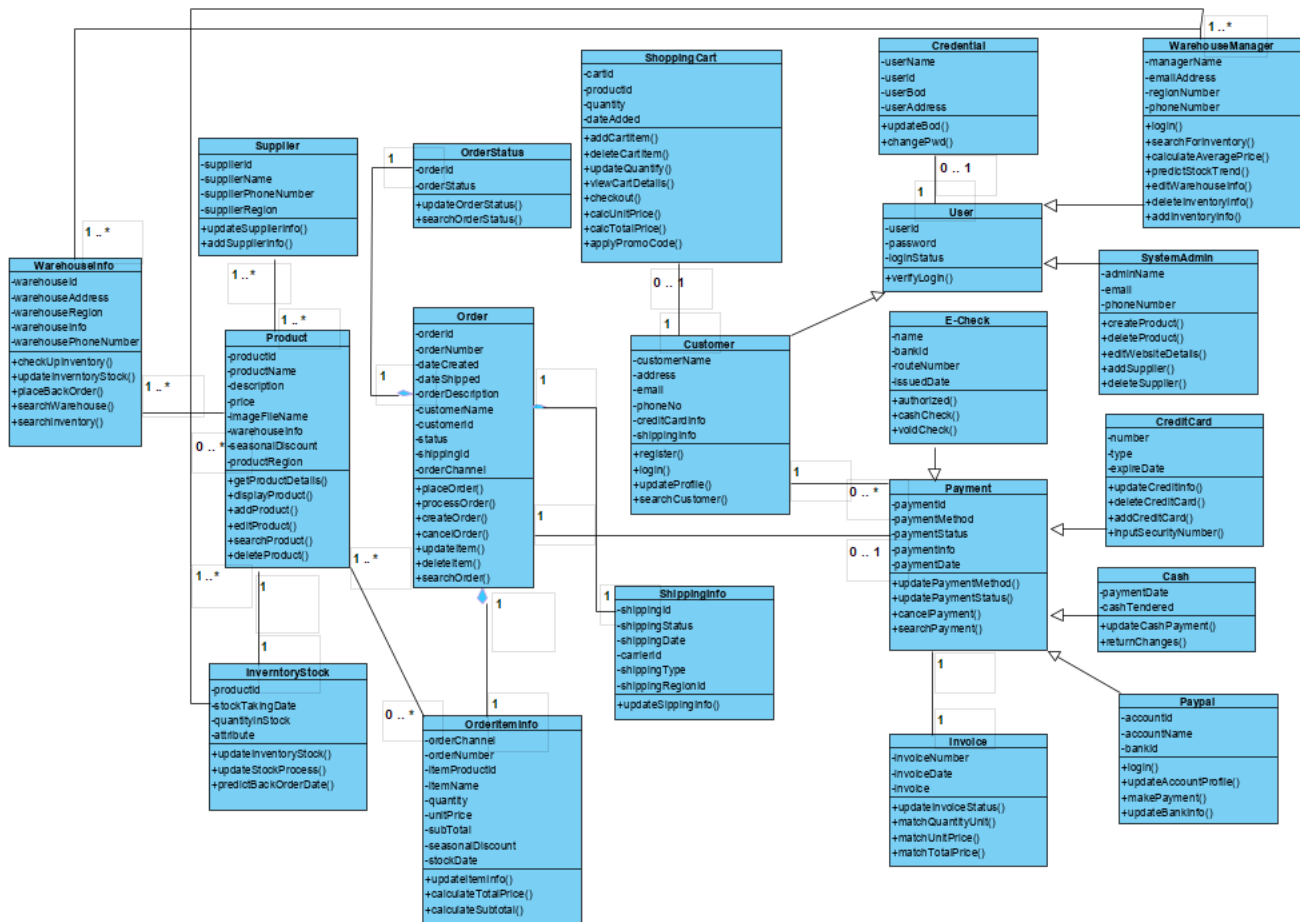
carrierId# is the Foreign Key and should be not null

InventoryStock (**inventoryId#**,productId#,stockDate, quantityInStock)

inventoryId #is the Primary Key and should be unique and not null

productId is the Foreign Key and should be not null

13. Complete Class Diagram



14. Software Design

1.

Method Name: createOrder	Class Name: Customer
--------------------------	----------------------

Description of responsibilities: Click “Add to Cart” on the website to create a new order to an existing customer, keeping the selected inventories in a sorted order by time/date.	
Arguments Received: anOrder customerID	Data Type: Order Object Integer
Return Value: orderResult	Data Type: Boolean
Message and Example: addOrder(customerID, anOrder): Boolean orderResult = addOrder(“12345”, “Order#1”)	
Algorithm Specification: If order is not null then Add the order to the shoppingCart Insert the order into the database Return true Else Return false	

2.

Method Name: confirmOrder	Class Name: System Admin
Description of responsibilities: Implement the necessary step to fulfill an existing order, which is sorted by order number.	
Arguments Received: ItemID anOrder warehouseID	Data Type: Integer Order Object Char
Return Value: confirmOrder	Data Type: Boolean
Message and Example:	

```

checkQuantity(itemID, warehouseID): Boolean
confirmOrder = checkQuantity("001", "WH#1")

```

Algorithm Specification:

```

If itemID is not null then
    Retrieve itemID, warehouseNum and item quantity from the database
    --Check for valid itemID and warehouseNum
    If valid itemID and warehouseNum
        --Check for item quantity
        If item quantity >= ordered quantity
            Add the item to the shippingInventory
            Update the order in the database
            Return true
        Else
            Return false
    Else
        Return false
Else
    Return false
End if itemID is null

```

3.

Method Name: updateOrder	Class Name: Warehouse Manager
Description of responsibilities: Implement the necessary step to update an existing order, which is sorted by shipping document number.	
Arguments Received: shippingNum anOrder	Data Type: Integer Order Object
Return Value: updateOrder	Data Type: Boolean
Message and Example: confirmPickup(shippingNum, anOrder): Boolean updateOrder = confirmPickup("01", "Order#1")	
Algorithm Specification: If shippingNum is not null then	

```
Add the item to the shippingInventory
Update the order in the database
Return true
Else
Return false
```

15. Project Timeline

WEEKLY SCHEDULE		
FROM	TO	TASKS
19 th August 2019	25 th August 2019	Formation of project group
26 th August 2019	1 st September 2019	Brainstorming of project ideas
2 nd September 2019	8 th September 2019	Finalized on OMS system for ALDI
9 th September 2019	15 th September 2019	Identification of problem statement and feasibility of the new system
16 th September 2019	22 nd September 2019	Discussed on the business flow and BPMN diagrams
23 rd September 2019	29 th September 2019	Identification of Use Cases and Designed the context diagram
30 th September 2019	6 th October 2019	Designing of Class and sequence diagrams
7 th October 2019	13 th October 2019	Documentation of Data Dictionary
14 th October 2019	20 th October 2019	Use Case descriptions
21 st October 2019	3 rd November 2019	Discussion on Interface Design
4 th November 2019	10 th November 2019	Design database and complete Class Diagram
11 th November 2019	17 th November 2019	Discussion of Software Design
18 th November 2019	24 th November 2019	Finalized the Software Design
25 th November 2019	1 st December 2019	Report Layout preparation and start working on the report Report preparation
2 nd December 2019	7 th December 2019	Report preparation
8 th December 2019	13 th December 2019	Review and Proof reading of report and submission

16. Minutes Of Meeting

ALDI Order Management System Project meeting minutes

Location:	UTD JSOM
Date:	02/16/2019
Time:	Kick-off meeting, 11:00 a.m. - 1:00 p.m.
Attendees:	Gift Chima, Aarushi Kaushik, Karan Syal, Fanjing Xu (recorder), Chun Zhang

Agenda items

1. Overview of the requirements of the ALDI Order Management System (OMS).
2. Discussed the Software Development Life Cycle (SDLC) and conducted project initiation and planning.
3. Defined the business problem, discussed project charter, determined project scope and identifies risks.
4. Reviewed the project proposal and discussed the assumption and scope of the OMS project.
5. Assigned tasks to each team member and released the deadlines for them. Scheduled next group meeting on February 20, 6:00 p.m.

Action items	Owner(s)	Deadline	Status
Executive Summary Background and Justification	Gift Chima	Feb.20 6:00 p.m.	In Progress
Assumption and Deliverables	Karan Syal	Feb.20 6:00 p.m.	In Progress
Project Success Criteria	Chun Zhang Fanjing Xu	Feb.20 6:00 p.m.	In Progress

Charter Project Scope (in)	Aarushi Kaushik	Feb.20 6:00 p.m.	In Progress
WBS Project Scope (out)	Chun Zhang Fanjing Xu	Feb.20 6:00 p.m.	In Progress

ALDI Order Management System Project meeting minutes

Location:	JSOM
Date:	2-20-2019
Time:	Second meeting, 7:00 pm - 9:00 pm
Attendees:	Aarushi Kaushik, Chun Zhang (recorder), Fanjing Xu

Agenda items

1. Looked through the Project Plan and discussed changes.
2. Changed Executive Summary to be more suitable to the project scope.
3. Discussed the scope of the project, including questions about the definition of “scope”.
4. Discussed the structure of WBS.
5. Classified WBS into different categories and discussed on each one.

Action items	Owner(s)	Deadline	Status
Executive Summary Background and Justification	Gift Chima	Feb.20	Completed
Assumption and Deliverables	Karan Syal	Feb.24	In Progress (Edit bullet points into paragraph)
Project Success Criteria	Chun Zhang Fanjing Xu	Feb.24	In Progress (Refine the criteria towards stakeholders)

Charter Project Scope (in)	Aarushi Kaushik	Feb.24	In Progress (integration of all parts in a template)
WBS Project Scope (out)	Fanjing Xu Chun Zhang	Feb.24 Feb.24	In Progress Completed