



Sir Syed CASE
Institute of Technology

5th May, 2024

Design & Architecture Plan

Submitted to: Dr. Shafaat A. Bazaz

Team: CASE Dev

**Project Title: Karyana Mart Sales and Inventory
Management System**

Members	Roll numbers
1. Jawad Hassan (AI)	2230-0035
2. Rizwan ul Hassan (CYS)	2330-0135
3. Muhammad Talha Ramzan (AI)	2330-0141
4. Usman Aslam (AI)	2330-0150
5. Adil Khan (AI)	2330-0030
6. Taha Malik (CYS)	2330-0156

Table of Contents

Preface	2
Introduction.....	3
Glossary.....	4
Team Details.....	5
Meetings	6
Specification Process.....	7
Prototype of UI	8
Contract	9
Elicitation	10
Analysis	14
Problems and Issues.....	15
Interviews	21
User Stories	24
Natural Language.....	26
Structured Language	31
User and System Requirement	32
Functional	33
Non- Functional	34
Test Cases.....	35
Graphical Notation.....	37
Requirements Validation.....	40
Index	41

Preface

This Design and Architecture Report for Hameed Mart has been prepared by the students of CASE Islamabad as part of the undergraduate Software Engineering course. The document is intended to provide a comprehensive design and architecture report for the Hameed Mart project, focusing on the use of Model-driven engineering (MDE) principles.

The development of this document involved the application of MDE principles, which includes the creation of four key models: Context models, Interaction models, Structural models, and Behavioral models. These models were developed based on a thorough understanding of Hameed Mart's requirements and needs, as well as the application of software engineering principles and methodologies learned during the course.

The purpose of this document is to serve as a guideline for the development team at Hameed Mart, providing a clear and structured approach to designing and architecting the system. It is our hope that this document will contribute to the successful development and deployment of the Hameed Mart project, meeting the needs and expectations of the client.

We would like to express our gratitude to our instructors and mentors for their guidance and support throughout the development of this document. We also extend our thanks to Hameed Mart for providing us with the opportunity to work on this project and gain valuable real-world experience in software engineering.

Students of CASE Islamabad

[Students]

1. Jawad Hassan
2. Rizwan ul Hassan
3. Muhammad Talha Ramzan
4. Usman Aslam
5. Adil Khan
6. Taha Malik

[Software Engineering]

[5th May, 2024]

Introduction

Karyana Store, a prominent retail establishment, is in need of a modern Inventory and Sales Point of Sale (POS) system to streamline its operations and enhance customer service. The current manual system is inefficient and prone to errors, leading to inventory discrepancies and customer dissatisfaction. To address these challenges, the students of CASE Islamabad have undertaken the task of developing comprehensive software solution that meets the specific needs of Karyana Store.

This document serves as a detailed design plan for the proposed POS system at Hameed Mart, utilizing Model-driven engineering (MDE) principles. It includes four key models: Context models, Interaction models, Structural models, and Behavioral models. These models provide a comprehensive overview of the system's design, outlining the functional and non-functional requirements, use cases, system architecture, and user interface design. The goal is to provide a clear understanding of the system's design and capabilities, ensuring that it aligns with the expectations and requirements of Hameed Mart.

The development of this document follows the principles and methodologies of software engineering taught in our undergraduate course. It represents our commitment to applying theoretical knowledge to practical, real-world scenarios, and showcases our ability to deliver high-quality software solutions.

We believe that the proposed POS system will significantly improve the efficiency of Karyana Store's operations, leading to better inventory management, accurate sales tracking, and ultimately, improved customer satisfaction. We are excited about the opportunity to work on this project and look forward to seeing the positive impact it will have on Karyana Store's business

Glossary

Context Models: These models describe the external environment of the system, including entities such as users, other systems, and physical devices. They help in understanding the interactions and dependencies between the system and its environment.

Interaction Models: Interaction models describe how users and external systems interact with the POS system. They include use cases, which are descriptions of specific interactions or scenarios between a user and the system to achieve a goal. Use cases help in identifying and defining the system's functionality from a user's perspective.

Structural Models: Structural models depict the internal structure of the POS system, including its components, their relationships, and how they are organized. These models help in understanding the system's architecture and how different parts of the system interact with each other.

Behavioral Models: Behavioral models describe the dynamic behavior of the POS system, including how it responds to stimuli, events, or inputs. These models help in understanding the system's functionality over time and how it behaves in different scenarios.

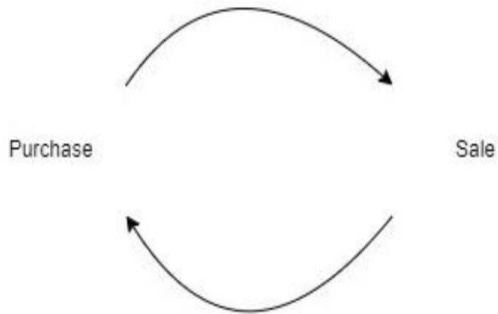
UML (Unified Modeling Language): A standardized modeling language used to visualize, specify, construct, and document the artifacts of a software-intensive system. UML offers a standard way to visualize a system's architectural blueprints, including elements such as activities, actors, business processes, database schemas, components, and interfaces.

Background:

This is a simple flow chart we made in the previous specification phase when we did not learn about model driven engineering for designing phase:

Flowchart

Purchase and sale in a karyana store are in a continuous loop:



Design plan:

While making specification report we elicited some data from the clients and made task cards, we will use those task cards to design and define our system boundaries, object classes and all kinds of models. Following are the tasks that will be used as references in designing models:

1. Inventory Management Tasks:

- Enter data for each item into the inventory system.
- Maintain records of expiration dates and highlight expiring items.
- Set inventory level alerts when stock is low.

2. Supplier Management Tasks:

- Create a database of supplier information.
- Maintain records of transactions with each supplier.
- Set up automatic notifications for new shipments.

3. Customer Management Tasks:

- Create a customer database with contact details and previous purchase records.
- Implement a loyalty program to reward regular customers.

4. Sales and Transaction Tasks:

- Maintain records of each transaction including item details, quantity, price, date, and payment method.
- Keep track of discounts and promotions and analyze their impact.

5. Reporting and Analysis Tasks:

- Generate daily, weekly, and monthly sales reports.
- Analyze sales trends and identify best-selling items.

6. Security and Access Control Tasks:

- Create user accounts and define their access levels (Admin, Staff, etc.).
- Implement data encryption and conduct regular security audits.

1. System Modelling:

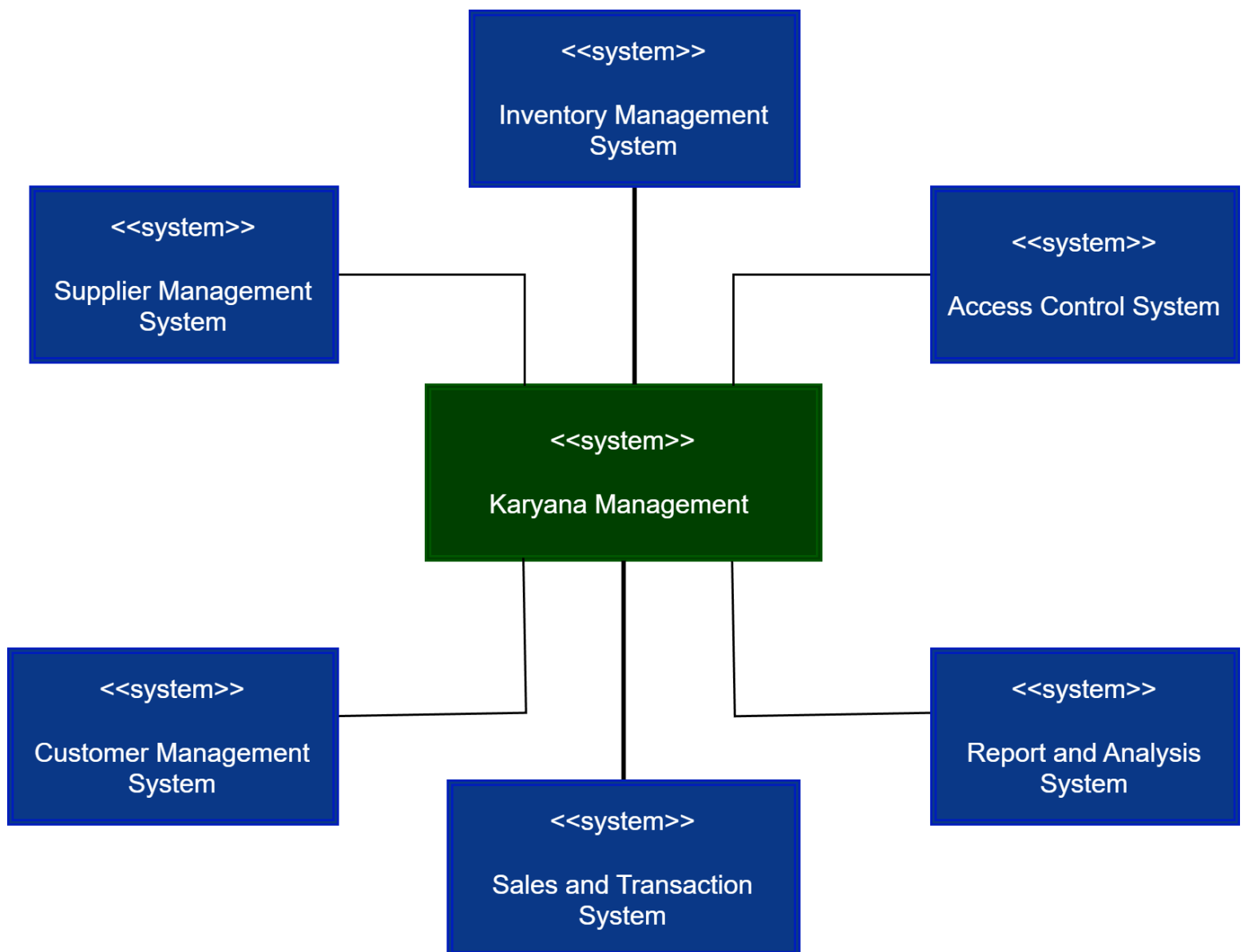
We are using Unified Modeling Language to design and create graphical notations to show interactions between systems, subsystems, external agents (users and external systems), object classes and their dynamic behavior. Each model shows a different perspective, models are outlined in the following sequence:

1.1 Context Models:

This model shows operational context of the system in which we define system boundaries as in what is considered as part of the system or external.

Architectural Model:

The most basic model which shows the relationship between different sub systems of main: Hameed Karyana Management System:

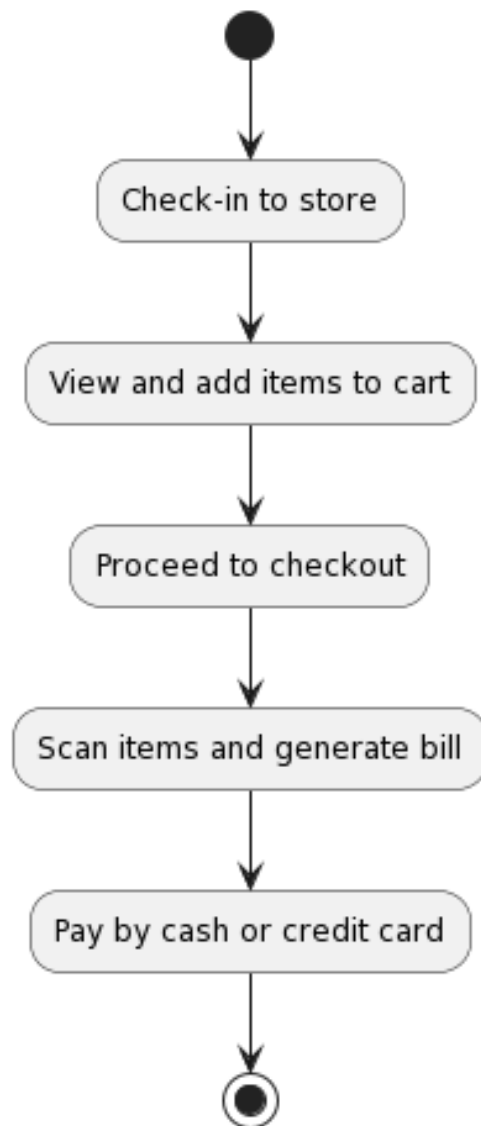


Business Process Model:

In order to show Hameed Karyana Broader Business Process we use following UML activity diagrams:

Simple Activity Diagram:

This activity diagram was made before studying the design phase:

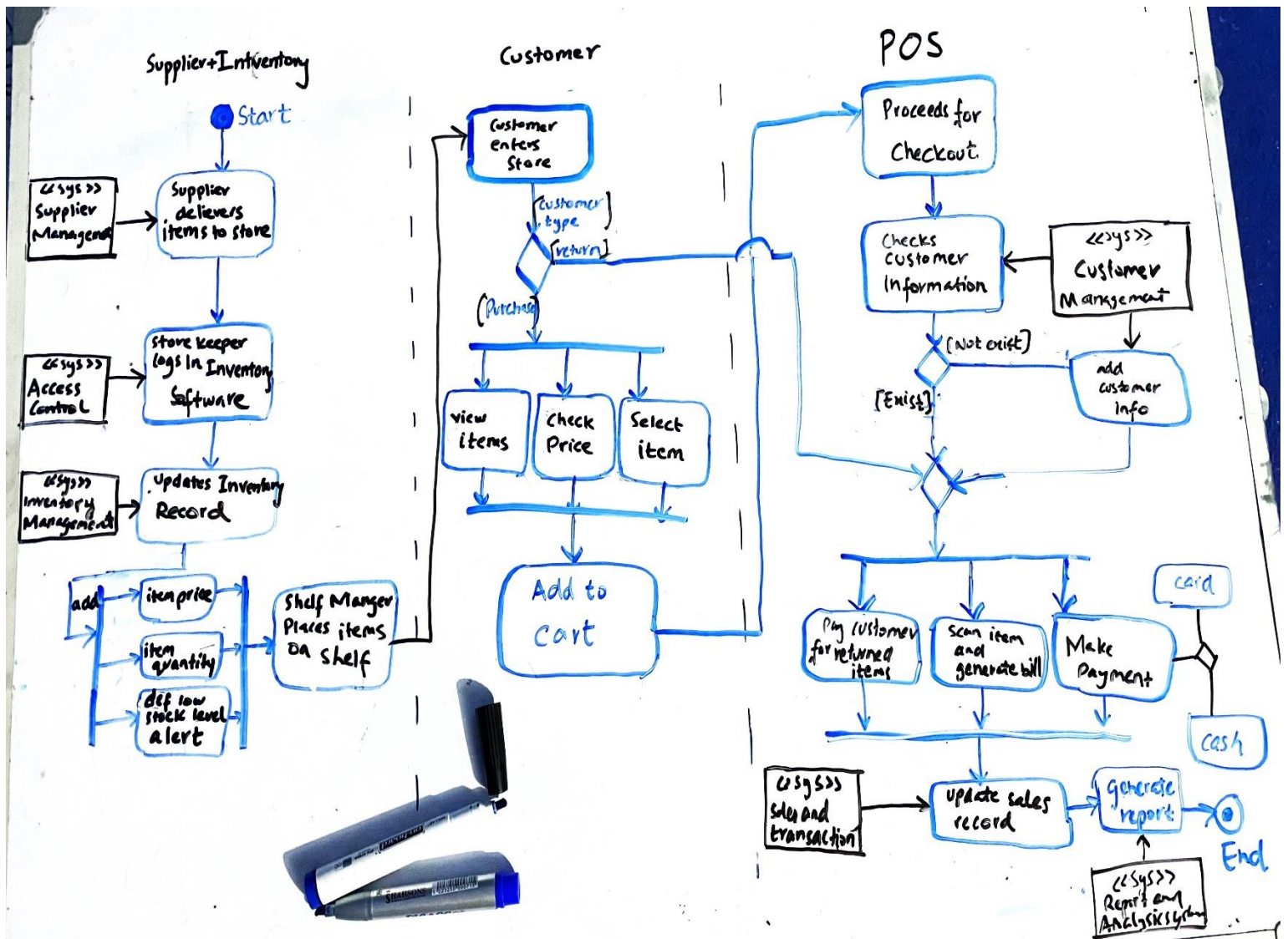


Explanation:

This simple activity diagram uses only start, end and arrow functions only to show the business process of POS system where customer enters the karyana store, select item, proceeds for checkout and pays the bill.

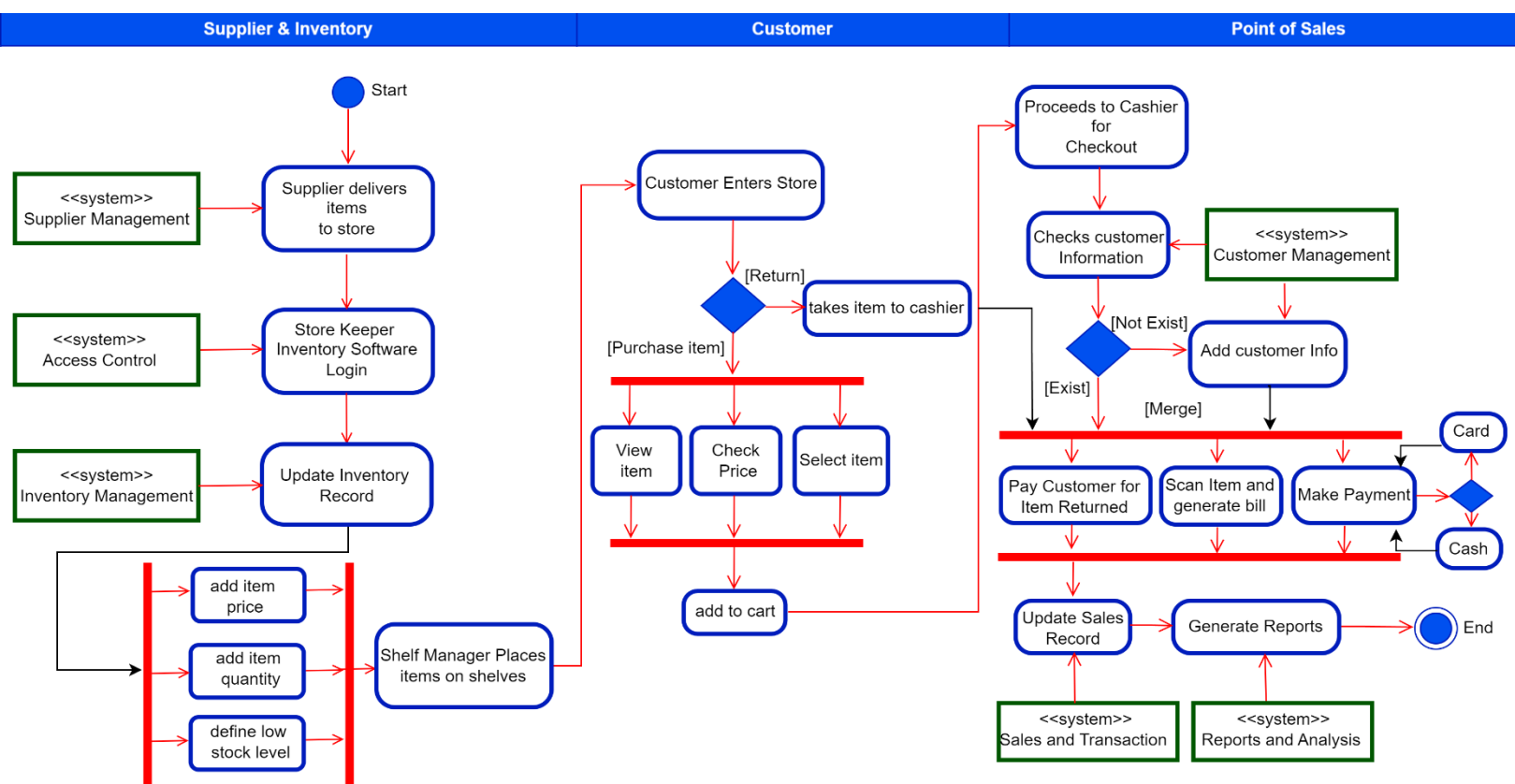
Rough White Board design: by Jawad Hassan

For mind mapping first we designed a rough activity diagram on a white board:



Detailed Activity Diagram for Karyana Management System:

After we learned designing on UML, we designed a neat diagram using 'draw.io'. In this diagrams we will see the detail of business flow of a karyana store from a supplier's, shopkeeper, customer, and cashier's perspective, as written in the blue boxes at the top, and the system components being used in each activity, the green rectangle shape shows system component whereas blue rounded rectangle shows the activity being performed. Orange bars are synchronization bars and blue diamond shapes are decision making points.



1.2 Interaction Models:

This model perspective shows interactions of two kinds:

- Interaction b/w external agents (actors/ users or external systems) and system components.
- Interaction b/w system components.

Entire System Use Case Diagram:

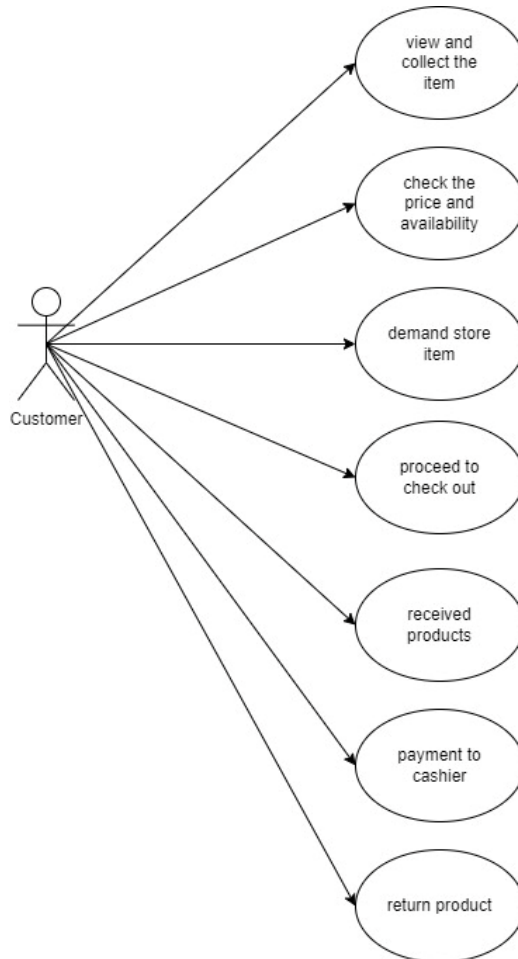
For interaction between user and system components we model with help of Use Case Diagram. Following is an overall Use Case Diagram of Karyana Management System, which models interaction b/w 6 actors:



Break Down of Use Cases:

For a detailed and elaborated system and component understanding, we broke down the diagram into smaller parts where each part represents an interaction of a separate actor.

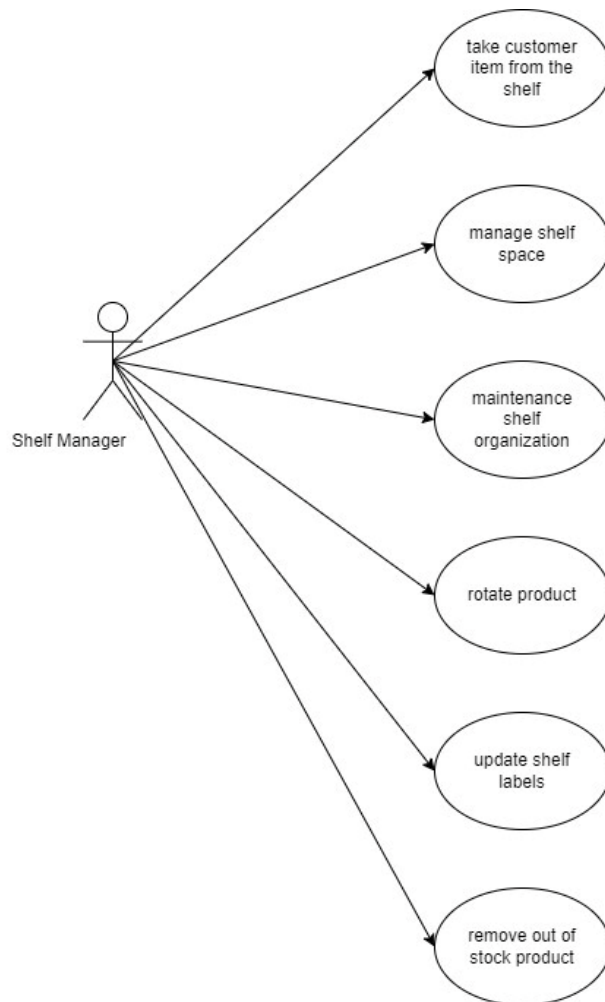
Use cases in the Karyana system involving the role 'Customer': by Talha Ramzan



Tabular description:

Use case	Description
View and collect item	Customer view and collect the item they want to purchase.
Check price and Availability	Customer check price and availability of the select item.
Demand store item	Customer selects the item they want to purchase.
Proceed to check out	Customer proceeds to the checkout counter to make payment.
Receive Product	Customer receives the product after making payment.
Payment To Cashier	Customer makes payment to the cashier.
Return Product	Customer initiates the return process for a product.

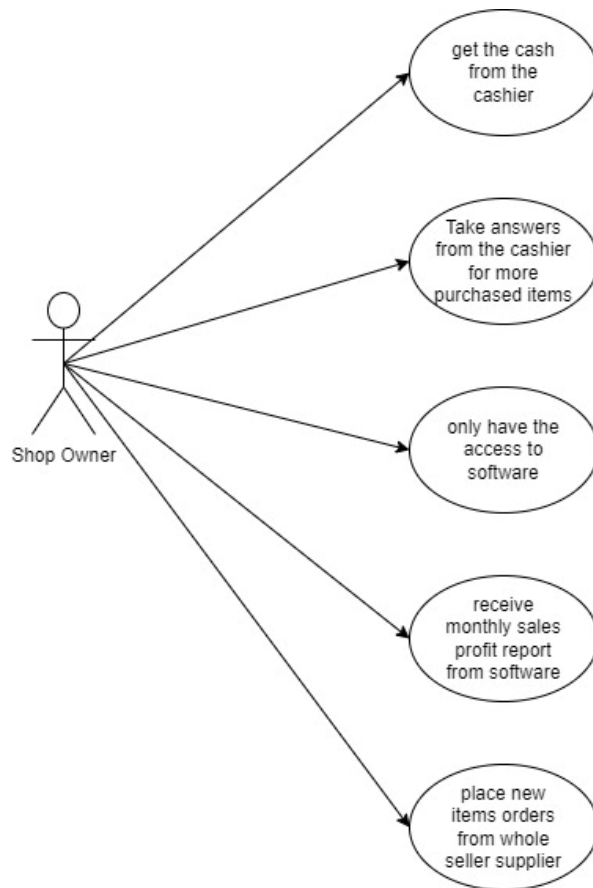
Use cases in the Karyana system involving the role 'Shelf Manager': by Usman Aslam



Tabular description:

Actors	Shelf Manager
Description	<ol style="list-style-type: none">1. Placing Customer Items on Shelves:<ul style="list-style-type: none">- The Shelf Manager takes products delivered to the store and places them on the appropriate shelves.2. Managing Shelf Space:<ul style="list-style-type: none">- This involves organizing and planning the layout of the shelves to maximize space and product visibility.3. Maintaining Shelf Organization:<ul style="list-style-type: none">- Regular checks and maintenance of shelf organization to ensure that everything is orderly and products are easy to locate.4. Rotating Products:<ul style="list-style-type: none">- Periodically rotating products to ensure that older stock is sold before the expiry date and to keep the display fresh.5. Updating Shelf Labels:<ul style="list-style-type: none">- Updating the shelf labels to reflect current prices, promotions, and product information.6. Removing Out of Stock Products:<ul style="list-style-type: none">- Regularly checking for and removing any products that are out of stock or have been discontinued.

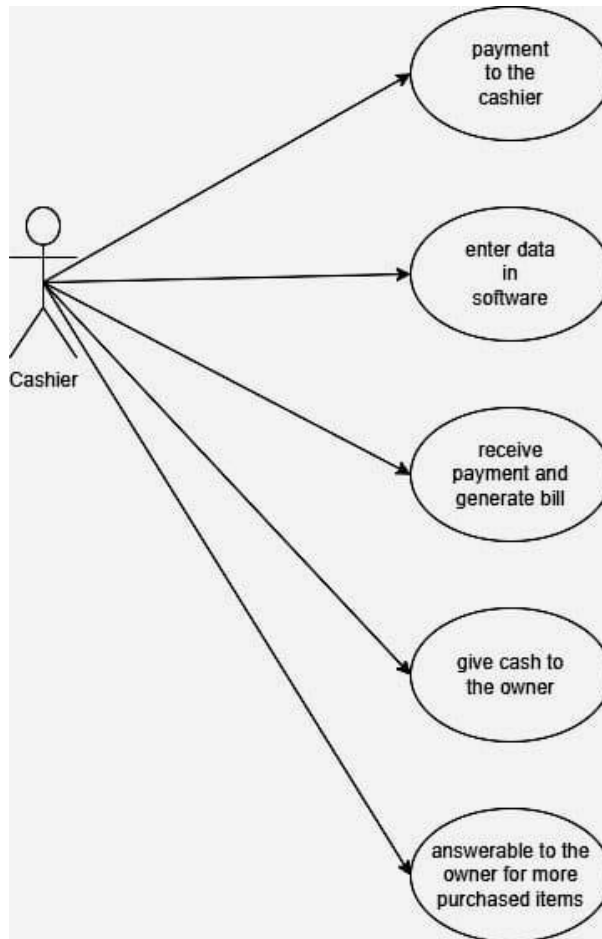
Use cases in the Karyana system involving the role 'Shop Owner': by Adil Khan



Tabular description:

Actors	Shop Owner
Description	<ol style="list-style-type: none">1. Get Cash from Cashier:<ul style="list-style-type: none">- The shop owner collects the day's earnings from the cashier, ensuring that the cash amount matches the recorded sales.2. Inquire About Transactions:<ul style="list-style-type: none">- The shop owner asks the cashier for clarifications regarding any discrepancies or additional purchased items not accounted for in the software.3. Software Access:<ul style="list-style-type: none">- The shop owner uses the management software to review transaction records, update inventory, and verify sales data.4. Receive Sales Report:<ul style="list-style-type: none">- At the end of each month, the shop owner receives a sales profile report generated by the software, which includes detailed analytics on monthly sales, popular items, and trends.5. Place Orders with Supplier:<ul style="list-style-type: none">- Based on the sales report and current inventory levels, the shop owner places orders for new items from the wholesaler supplier to restock the shop.

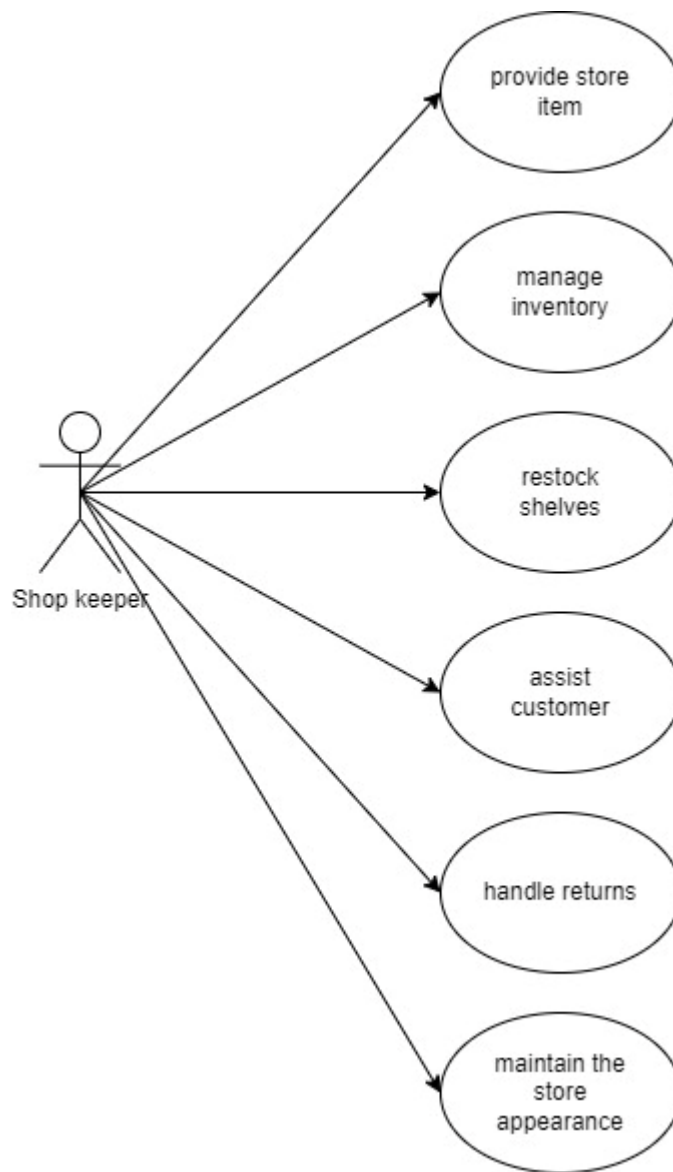
Use cases in the Karyana system involving the role 'Cashier': by Rizwan ul Hassan



Tabular description:

Use case	Description
Collect Payment to the cashier	Payment is collected from the customer.
Enter data in software	Sales data is entered into the software.
Receive payment and generate bill	Payment is received and a bill is generated.
Give cash to the owner	Cash is deposited with the owner.
Answerable to the owner for more purchased item	The cashier provide the owner with details of the product see the bill and purchased item.

Use cases in the Karyana system involving the role 'Shop Keeper': by Taha Malik

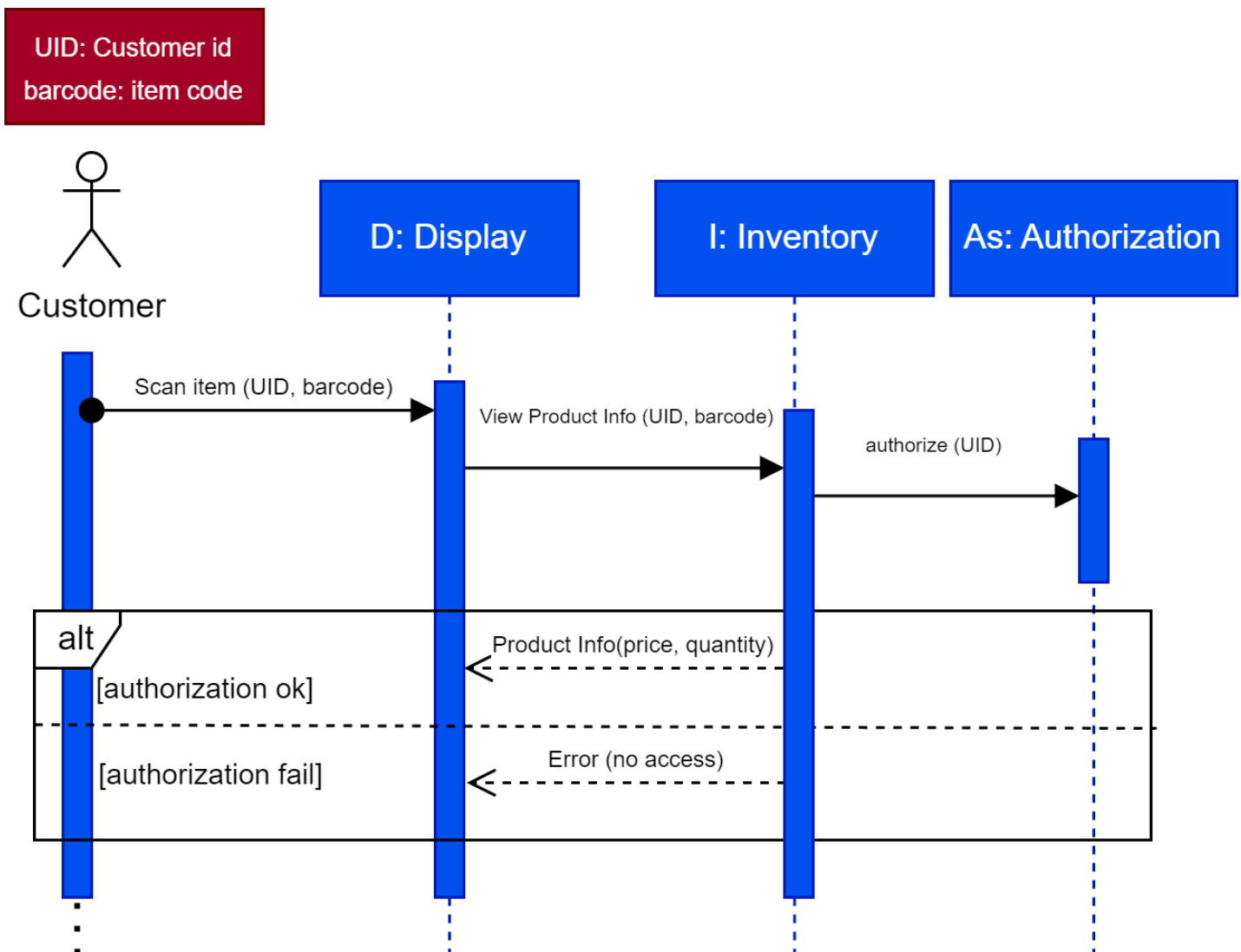
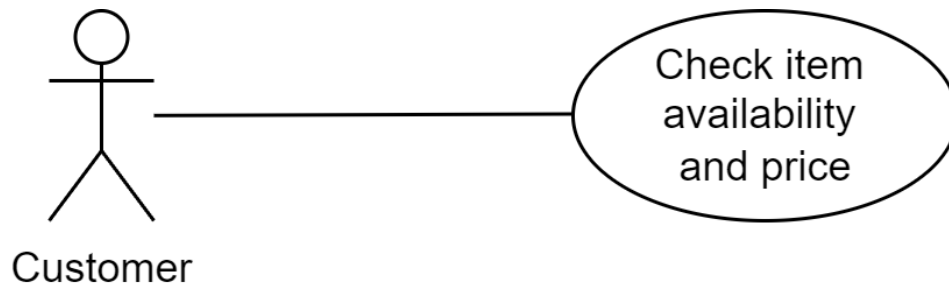


Sequence Diagrams:

It is used for showing interactions between the actors and the objects within a system. Each Use Case of an actor will have its own sequence diagram. We picked each those use cases that involve some instance of an object with in a system and its interaction with the user/ actor. Which will make it easy for developers to implement software and for customers to understand and communicate.

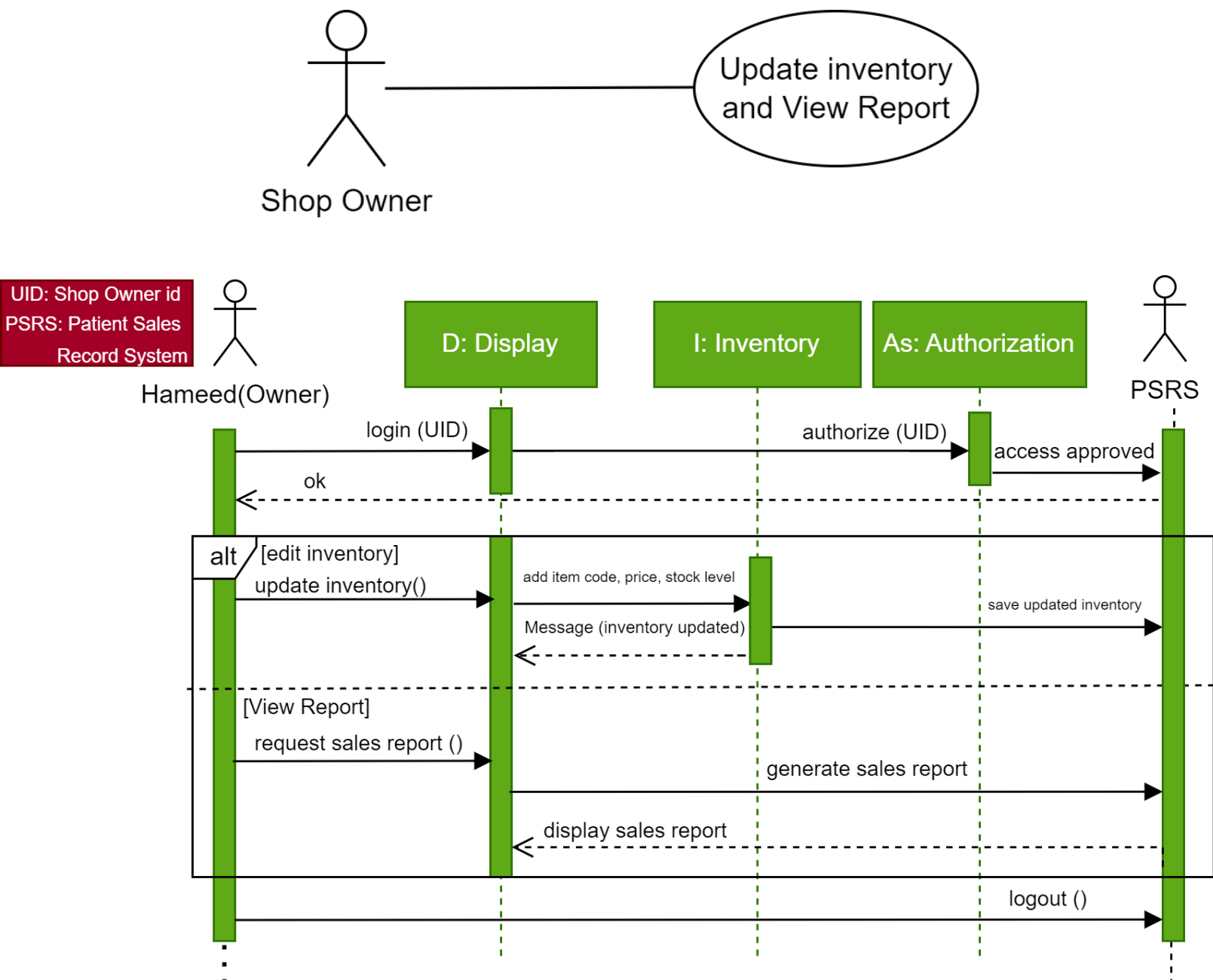
Sequence diagram for View Product Information (Price &Quantity) by 'Customer':

Customer picks up an item from a shelf and is curious to know the product details, there is a bar code scanner with a display system at karyana store where customers can verify product price and quantity themselves. Without calling anyone for help. This system saves time and if customer wants to buy item in bulk, he can get to know about product quantity in the back store without visiting it with the help of inventory software system. There is also an authorization system called Access Control system in software that checks the level of access for a specific role, when Customer mentions his specific id, the software returns relevant product info and not sensitive information that only shop owner can access (amount of sales, report etc.).



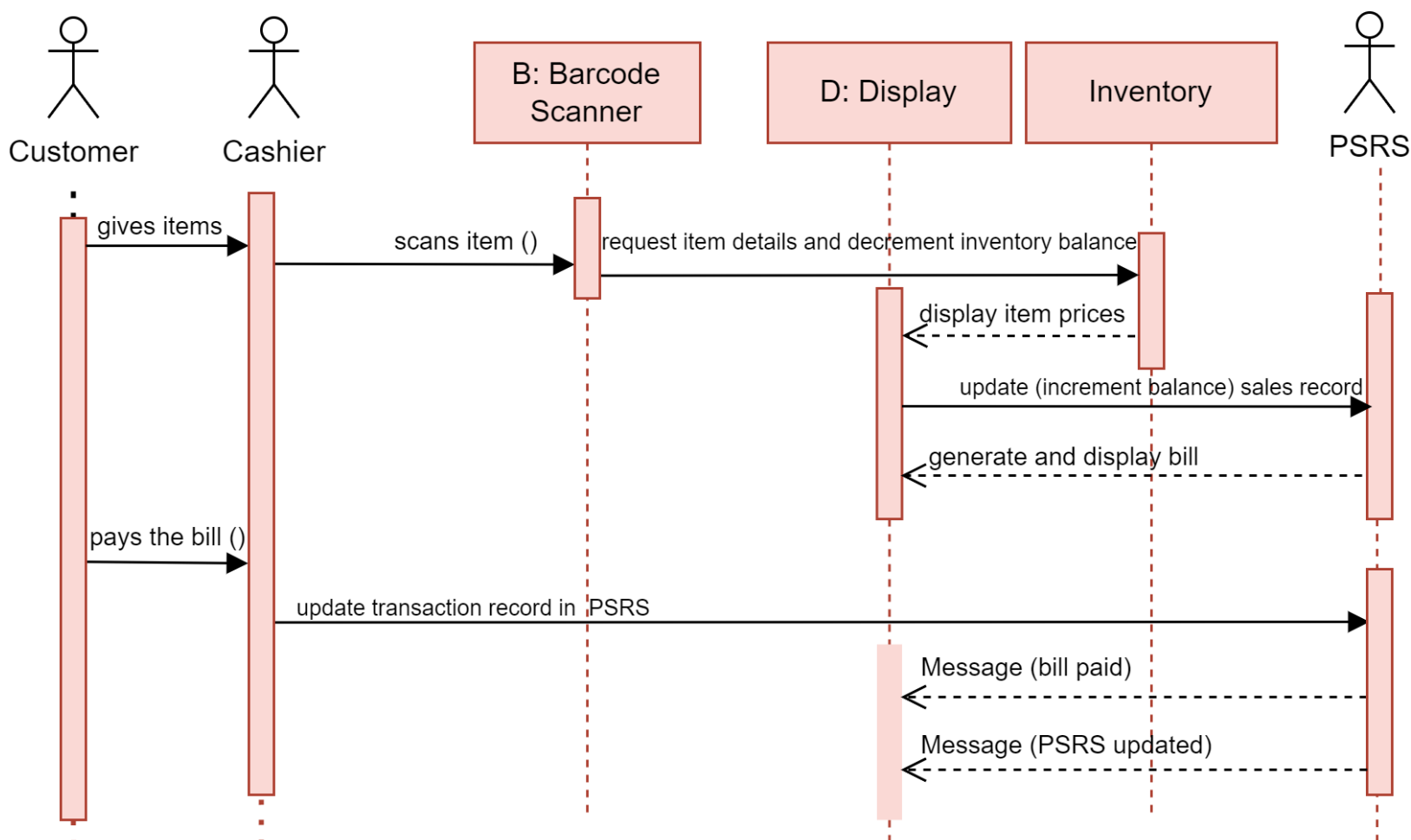
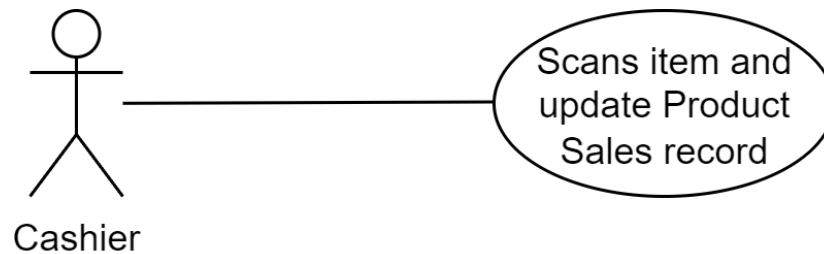
Sequence diagram for Inventory update after item delivery from supplier and View Sales Summary Report and Analysis by 'Shop Owner-Hameed':

After Supplier delivers the ordered items on the shop. The Shop owner 'Hameed', puts a barcode sticker on the items and one by one manually enters the item record (if not already there) such as (item code, price, quantity, flavor and defines item low stock level for alert). This process is called inventory update. Shop owner also views his shop's sales report and analysis stored in Product Sales Record System (PSRS) an external sales record system. The first step is owner logging in PSRS system by passing through an authorization system called Access Control system in software that checks the level of access for a specific role, when shop owner 'Hameed' mentions his id he gets access to sensitive information such as total amount of sales, daily/weekly/monthly reports, supplier information, trendy items, dead stock and access to update inventory. Each role's user id is specific and kept as private information to avoid misuse.



Sequence diagram for Customer's item scanned via barcode scanner and bill payment received and sales transaction record updated in PSRS by 'Cashier':

After customer selects products and adds them to cart, he proceeds for checkout and stands in front counter where cashier scans the customer products via barcode scanner and item prices are displayed on monitor. At the end these prices adds up and final bill is generated. Customer pays the bill and cashier updates the sales and transaction record in Product Sales record system (PSRS).



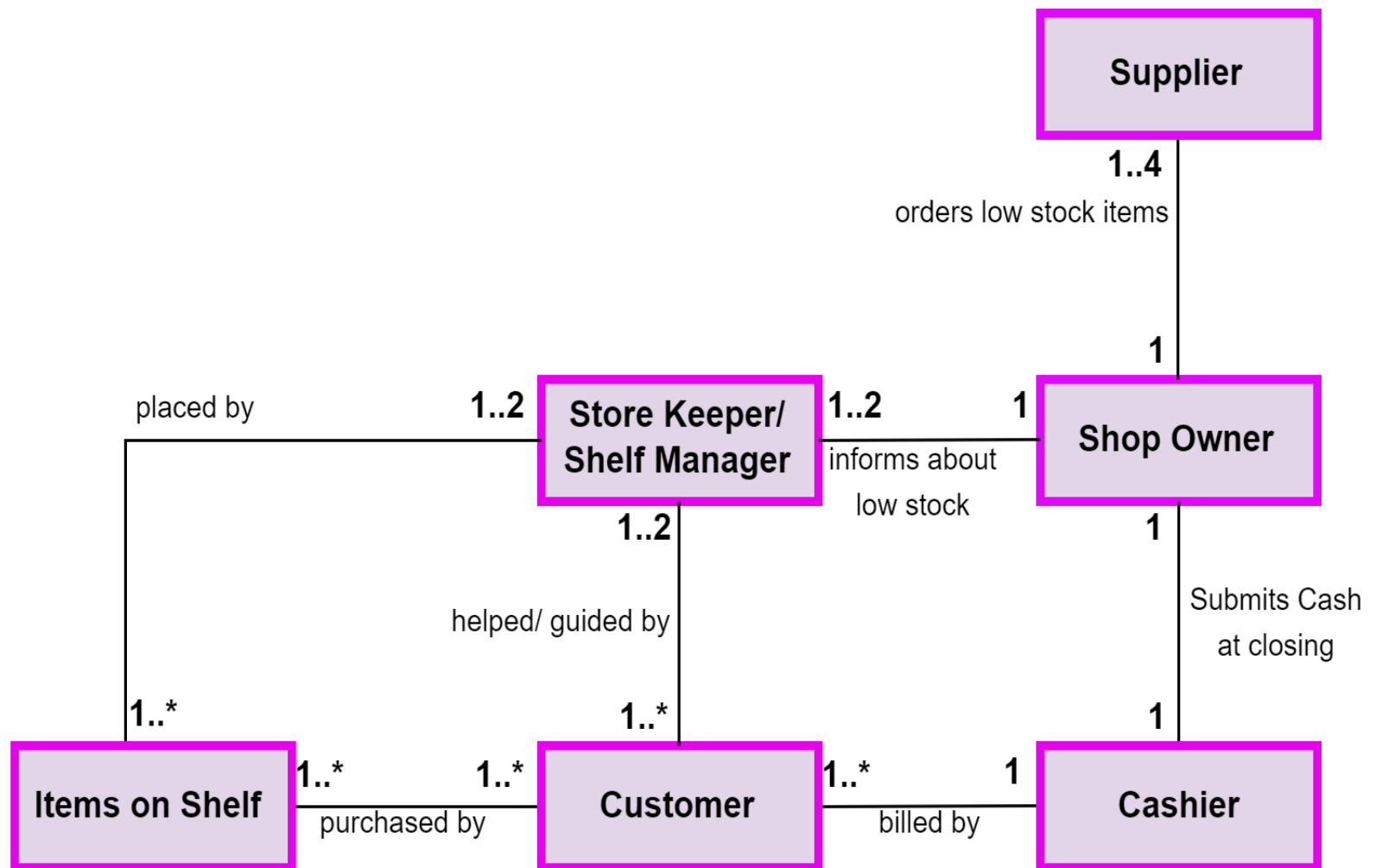
1.3 Structural Model:

It shows the structure and organization of system and its components. In this model we will be designing the system architecture.

Class Diagrams:

Over here we are developing an object oriented system so we need to show relationship between different object classes called association or link. (1) shows there is one object, (1..*) shows there are indefinite number of objects and (1..4) shows there are 4 definite objects.

Classes and Association in Karyana Management system:



The 'Point of Sales' class (Attributes and Operations):

Point of Sales
Customer id Cashier Cart Date Time Items Barcode Scanner Bill Cash Cash Drawer
Scan() Add to Cart() Calculate Bill() Print Bill() Update Sales Record() Update transaction Record()

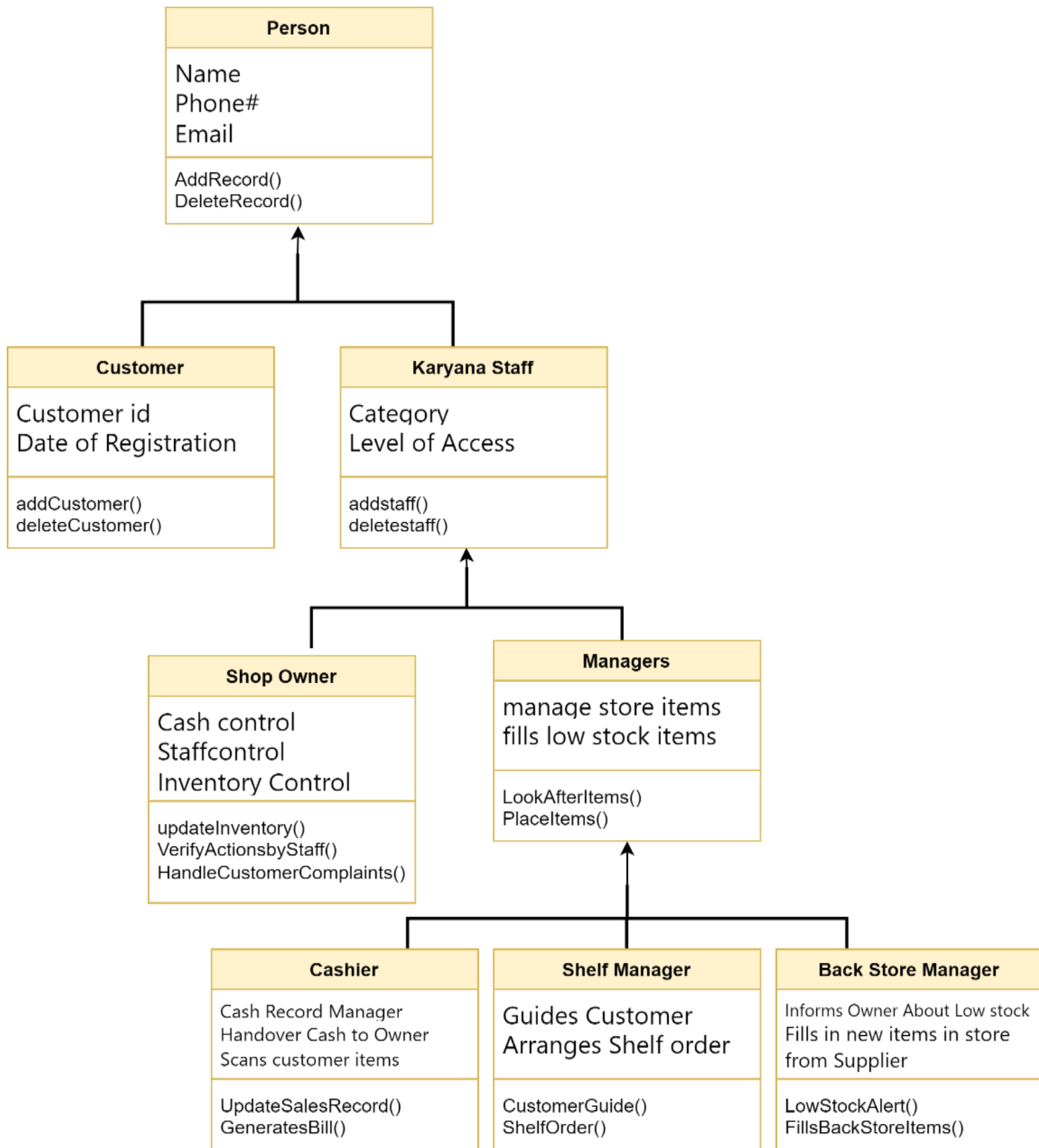
The 'Inventory Management' class (Attributes and Operations):

Inventory Management
Product code Barcode Price Manufacturer Supplier Stock level Shop owner Flavor Quantity Expiry Date
AddNewItem() DeleteItem() PriceUpdate() InventoryUpdate() Supplier Update() StockLevelAlert()

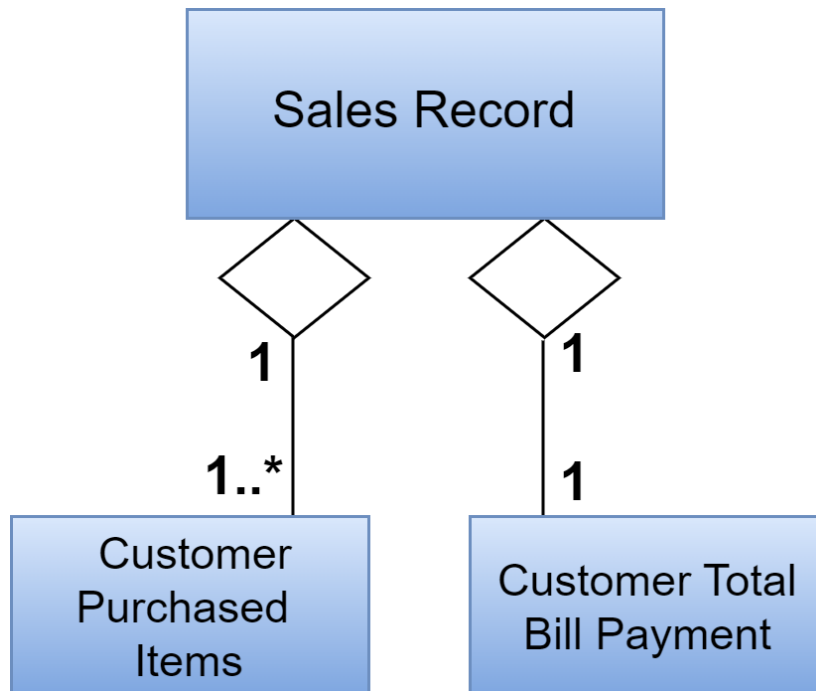
The 'Sales and Report' class (Attributes and Operations):

Sales and Report Analysis
Product ID Quantity Total price Timestamp Sales Data Mean Average Trendy Item
getSalesbyProduct() getSalesbyCategory() getSalesDaily() getSalesWeekly() getSalesMonthly() MostSoldStock() DeadStock()

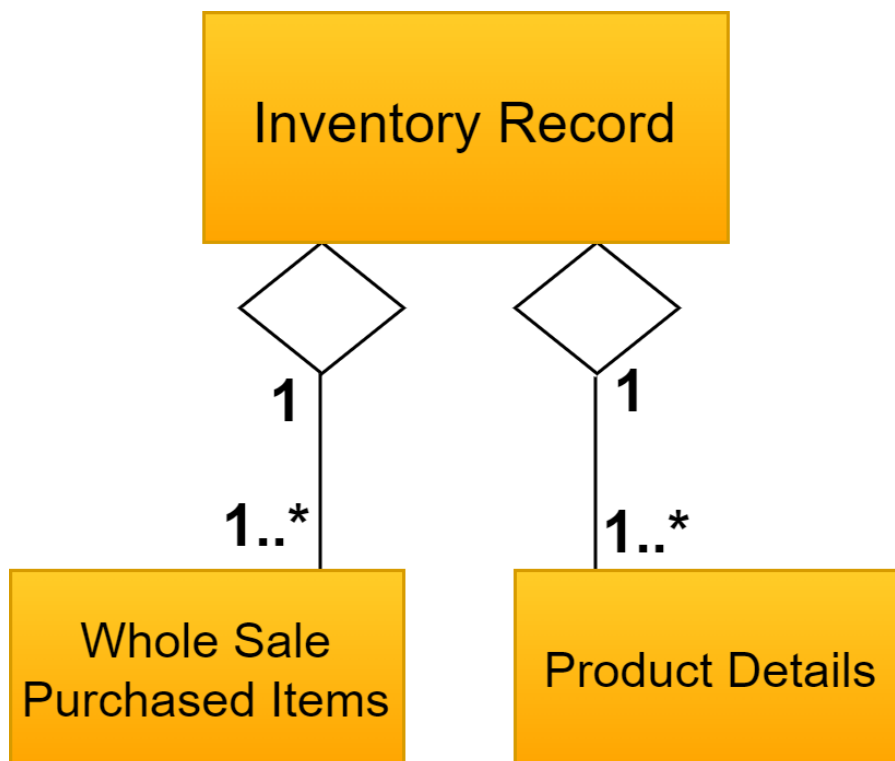
A generalization hierarchy with added detail for 'Persons Related to Shop':



The aggregation association Of Sales Record:



The aggregation association Of Inventory Record:



1.4 Behavioral models:

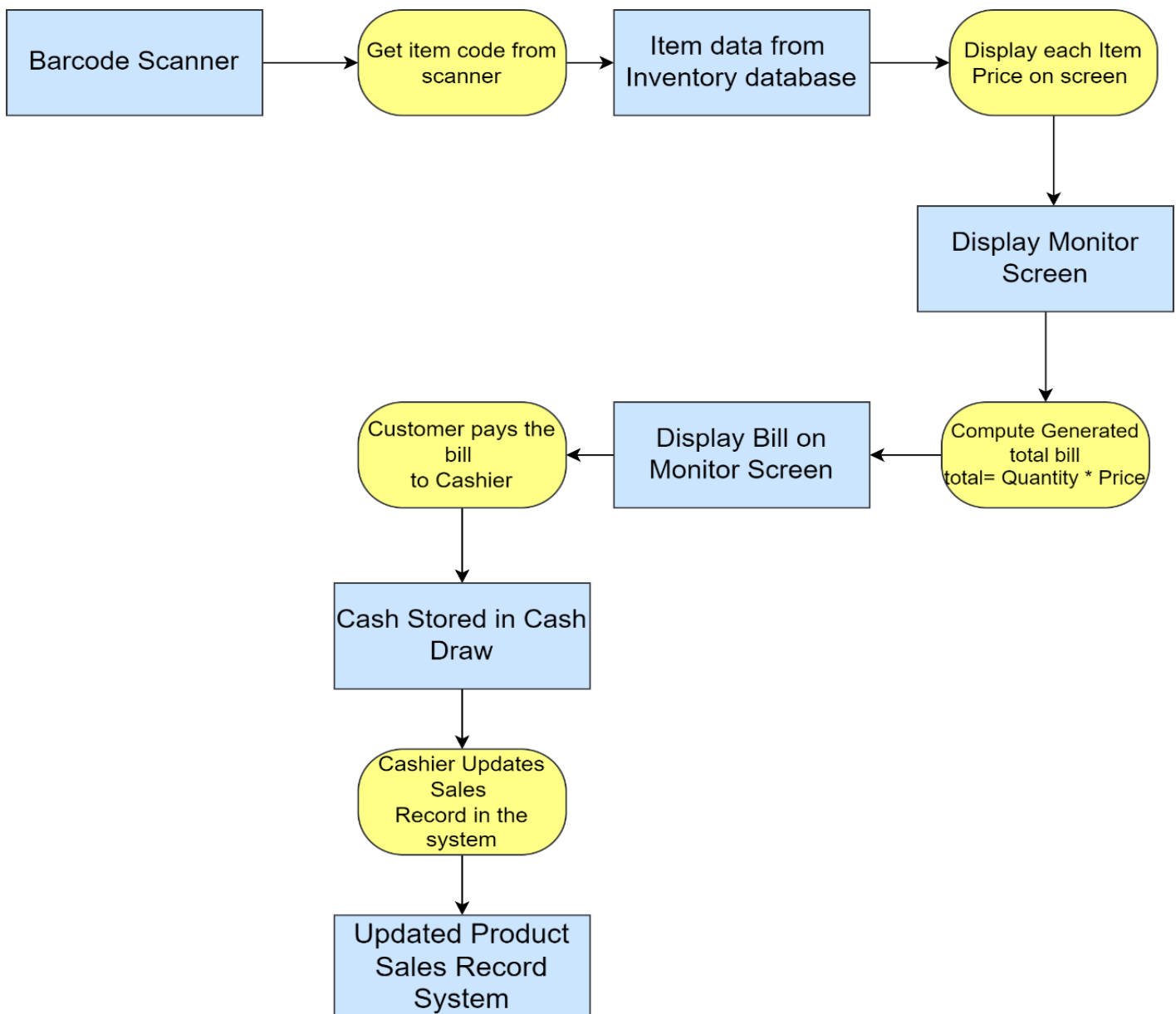
It shows the dynamic behavior of the system as it is executing in response to a stimulus, which is either An event or data.

Data-driven modeling:

In Karyana system most work is physical from the start when customer enter the stores, selects item, add to cart until he reaches the cashier for checkout, At the point of sales, an event occurs when cashier scans the customer's selected barcode. Then that barcode extracts product data from inventory, then the whole process takes place, the product data (item price, quantity, flavor etc.) are shown on display and total bill is generated. Customer pays the bill via cash or card and check-out, the sale record is updated in database. And inventory stock data is updated as well. This whole flow of process is demonstrated in following diagram:

An activity model of Sales Transaction Record at Karyana Store:

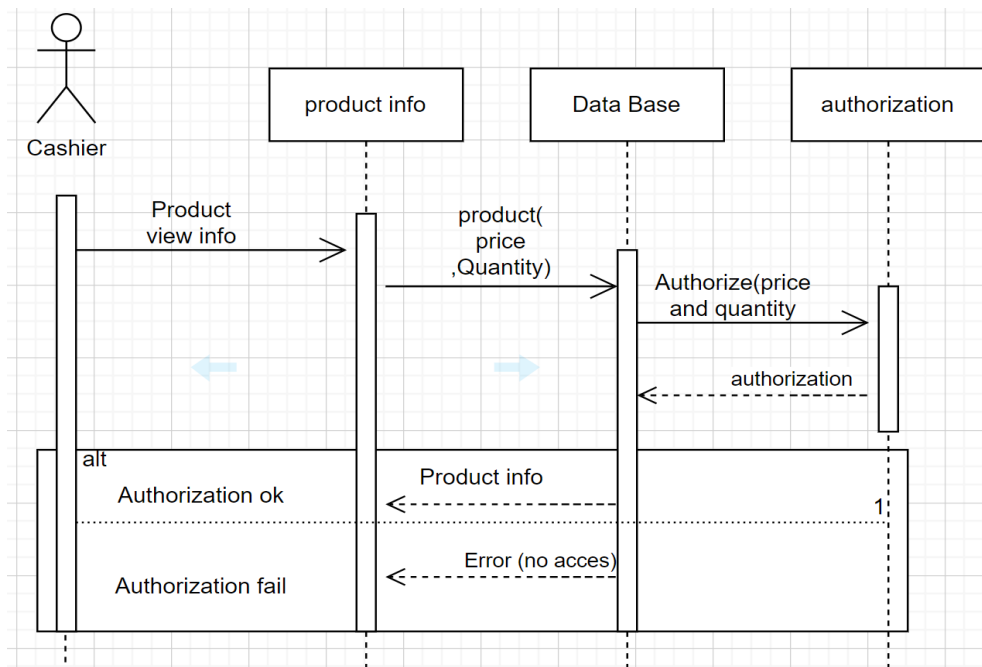
In this model, rectangle shape represents an object both physical or software component whereas rounded rectangle represents an activity taking place.



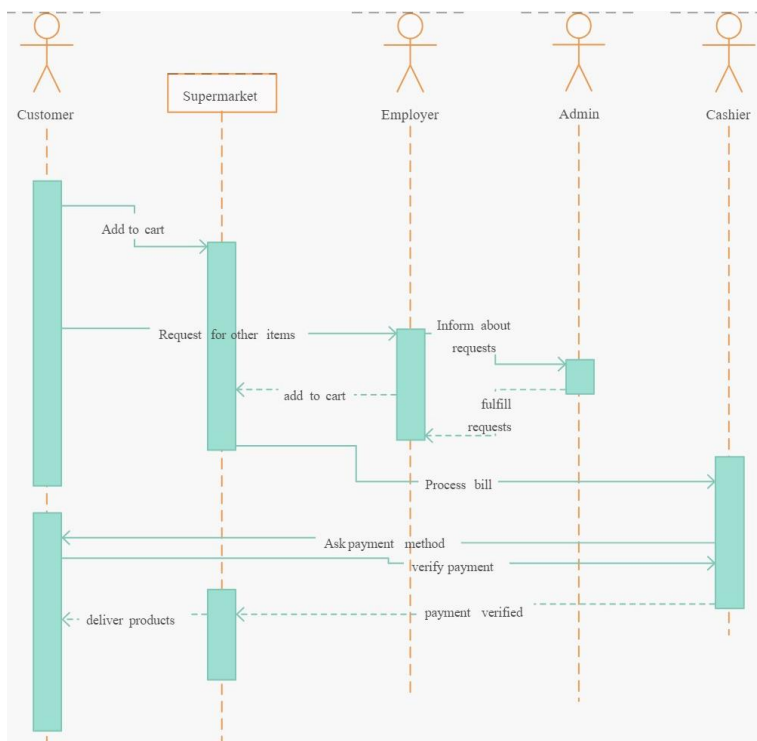
Sequence Diagram:

In data driven modelling we can also make a sequence diagram.

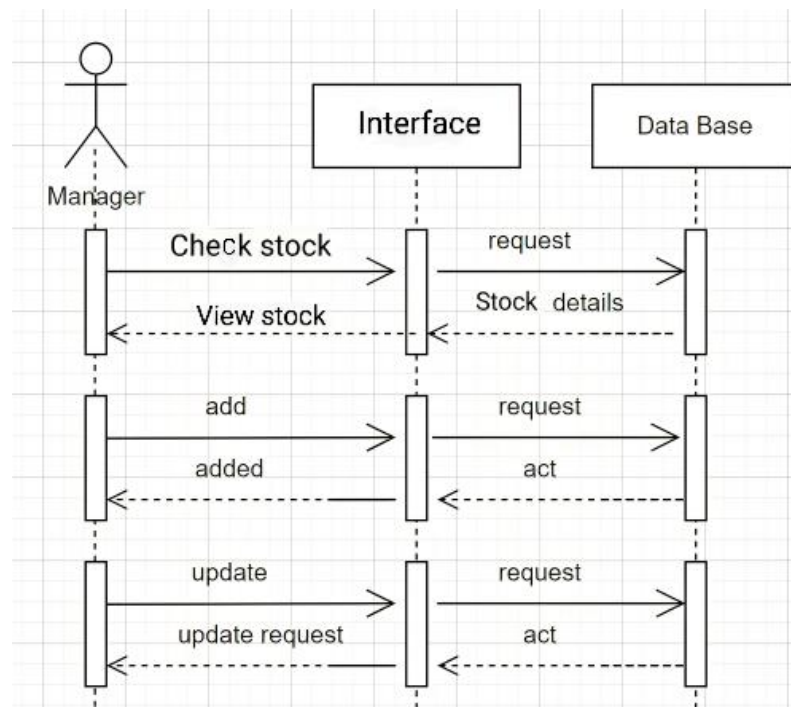
Cashier gets product details for customers: by Talha Ramzan



Customer Add product to cart: by Adil Khan



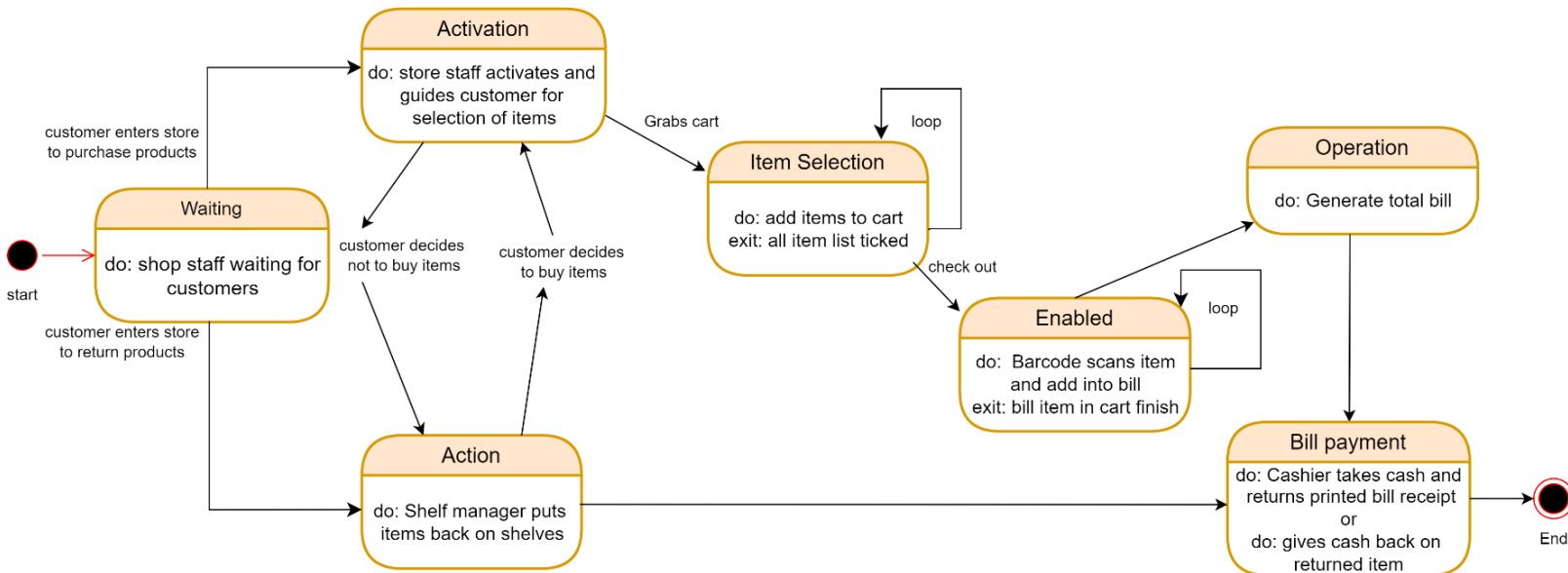
Manager Add product to inventory: by Talha



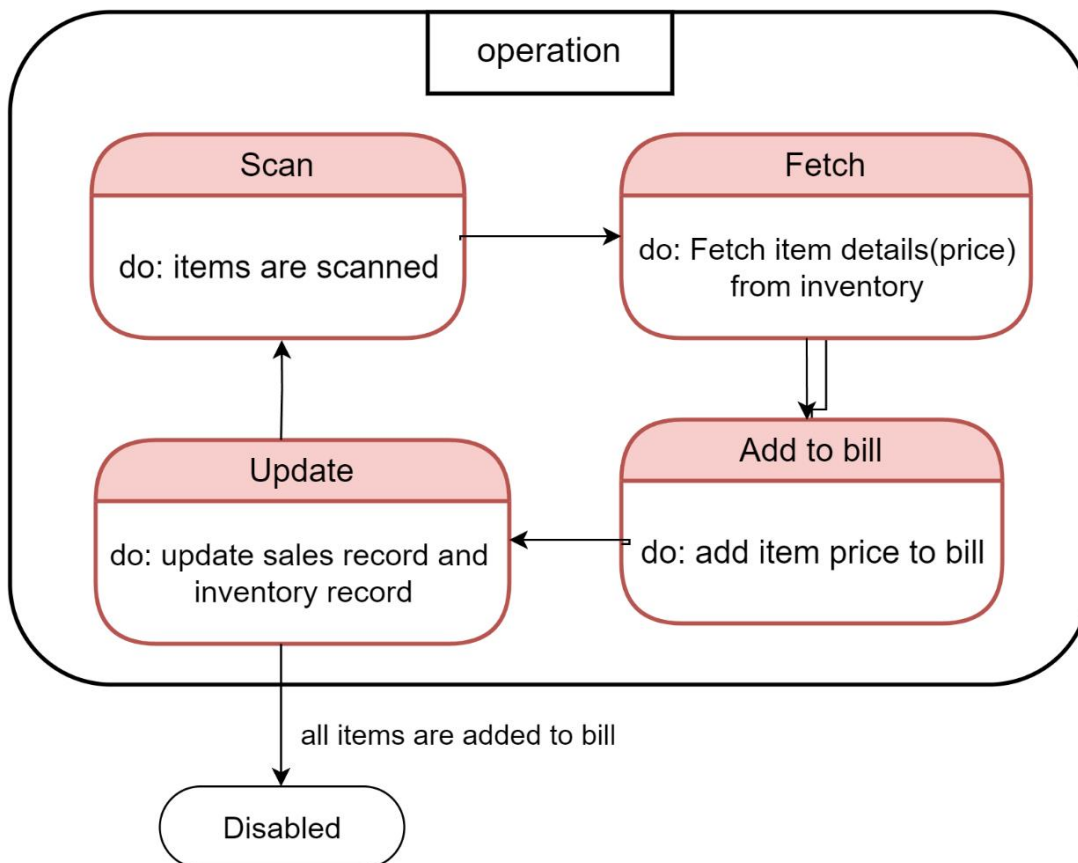
Event-driven modeling

State machine model of Karyana Store:

State model of karyana store is a part of event driven modeling. In this model we have states written in rounded rectangles and stimulus described with arrows.



Generate bill operation state:



States and Description Table of Karyana store state model:

States	Description
Waiting	Shop staff waiting for customers to enter store
Activation	Store staff activates and guides customer for selection of items
Instruction	Shelf manager puts items back on shelves
Item selection	Add items to cart until all item list ticked.
Enabled	Barcode scans item and add into bill until bill item in cart finish
Bill Payment	Cashier takes cash and returns printed bill receipt Or gives cash back on returned item
Operation	Generate total bill

Index

Barcode Scanning.....	19
FMCG (Fast-Moving Consumer Goods).....	15
Inventory Management	25
Point of Sale (POS)	31
Test cases	35
User Story.....	24
UML (Unified Modeling Language)	37
User Interface Design.....	34