P
S C A N
Y

-Architecture description-

Omer Gokdere

TABLE OF CONTENTS

1. INTRODUCTION

The aim of this document is help developers who is going to work on ScanPay software for maintaining or updating. This document demonstrates an overview of the project, its architecture and explain the decision made. In this document, we present the problematic of the software, followed by the marketecture and showing how the system works. Later on, we will present the utility tree of the project, this diagram presents the tradeoff points of the projects and the stakeholder concerned by them. Then we will introduce an approach analysis presenting to solutions for resolving the assets and their comparison and the elements of design.

2. PROBLEM CONTEXT

This is an industry suited project, initiated on our own idea for leading shopping marts in Poland.

The system that we are developing allows people to do shopping easily by reducing the hustle which is required in the billing procedure and there will be no queues. The system makes use of scanning QR code where a product that is stocked on the shelf of a shopping mall. Scanning an item will automatically display all information's about the product on customer's device with adding to cart option. The use of this system will make the billing easy and packaging of items are going to done by workers.

Moreover, this product allow the shop manager to obtain a lot of data about each costumer: the time used for shopping, the path taken in the shop, his consumption pattern… A lot of valuable indicators could then be used to manage more efficiently the business.

3. MARKETECTURE

This is a software design specification for a shopping system that will be built around the QR Code technology.

The system includes a mobile application that will be used to manage the NFC interactions and desktop applications that will mainly be used to manage product inventory and for shopping Bill generation corresponding to the items in the customer's cart on the mobile application.

This system is meant to help the end users by saving them a lot of time at the cash counters and making payment. It consists of following three faces:

MOBILE APPLICATION

The customer will be able to scanning QR codes for each product that he/she wants to buy and that product will be added to the cart in application. The customer will also be able to:

- ❖ Add Item
- ❖ Delete item
- ❖ Specify quantity

This cart will also be maintained simultaneously on the server database so that when the customer is done shopping, the cart is already available at the cashier's system after the customer ID is read off from server.

The app will also contain a recommendation engine that will provide the customer with appropriate offers and products that are relevant to the customers purchase history and the season. and give cashier number according to product quantities

ADMIN APPLICATION

This is where the product inventory will be managed. Each new product to be added will be assigned a unique QR code and the quantity stated QR codes. The real-time communication between the mobile app and the server database will allow employees to view the status of the payment This is where the product inventory will be managed. Each new product to be added will be assigned a unique QR code and the quantity stated QR codes.

The real-time communication between the mobile app and the server database will allow employees to view the status of the payment and give cashier's number to customers according to product quantity in cart to reduce waiting time in queues and saving customers a lot of time at the cash counters and making payment.

APPLICATION ON CASHIER'S SYSTEM

This application will only fetch the cart from the server corresponding to the customer ID and generate the final bill.

To conclude, with an increasingly consumerist society, the crowds and queues at the markets and stores using the conventional shopping processes, are inevitable, our aim is to change that by making the shopping at a store by application and customer receive bill while products are packaging.

Cashier is going to check the product(s) which is/are in the list of products in system while packaging the cart items and giving the bill to the customer.

4. UTILITY TREE
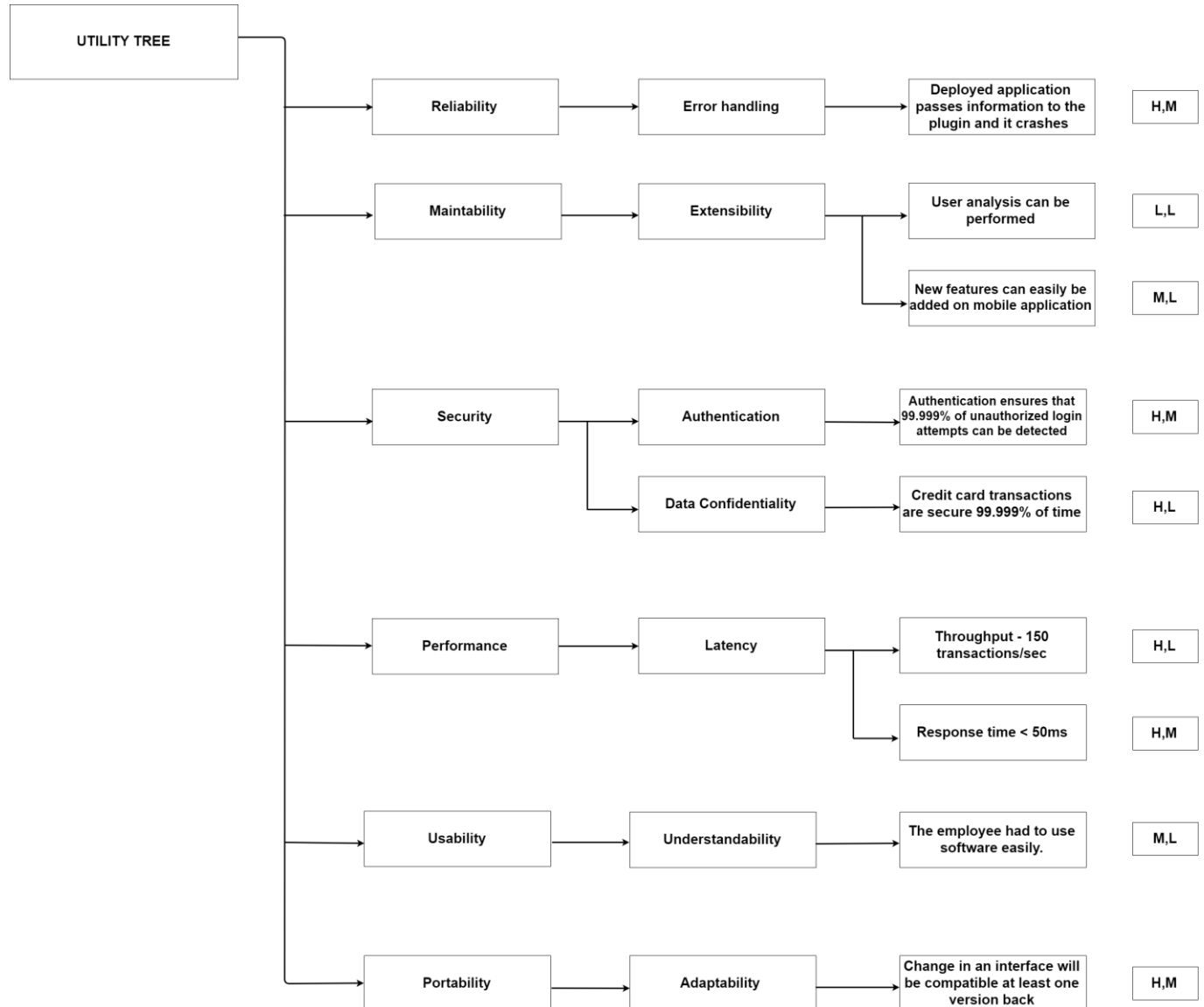
Figure 1 : Utility Tree

5. APPROACHES ANALYSIS

5.1. FIRST APPROACH

The system includes a mobile application that will be used to manage product inventory for shopping. Bill generation corresponding to the items in the customer's cart on the mobile application. Users will have mobile phones in which they will have an application where they can buy products. The customer will be able to: Add Item, Remove item and Specify quantity.

	Helpful	**Harmful**
Internal Origin	Strengths • Helps customers to save a lot of time at the cash counters and making payment • Easy in usage • Free mobile application	Weaknesses • Application is required highly amounted advertising plan • Hard to manage how people will behave against the application
External Origin	Opportunities • Facilitate process of buying and selling	Threats • Budget planning is required some stakeholders. • People may not want to resign "old school" form of buying products. • Everybody has not smart phones to use

Table 1 : First Approach by SWOT Analysis

5.2. SECOND APPROACH

This approach is made for people who don't have smartphones. New technology tablets or devices with the application will be put in some places in Commercial Center. Users will be able to use this device to buy products. The customer will be able to use their accounts by logging in the application.

	Helpful	**Harmful**
Internal Origin	Strengths • Helps customers to save a lot of time at the cash counters and making payment • Easy in usage • Free device service in Commercial center • Free mobile application	Weaknesses • Application is required highly amounted advertising plan • Hard to manage how people will behave against the application
External Origin	Opportunities • Facilitate process of buying and selling	Threats • Budget planning is required some stakeholders. • People may not want to resign "old school" form of buying products. • Devices may not be enough to customer who does not use smart phones

Table 2 : Second Approach by SWOT Analysis

6. DESIGN DECISIONS

6.1. DESIGN DECISIONS FOLLOWED BY CHOSEN TECHNOLOGIES

- **Identifier: D1**

- Name: easy in use

- Description: application should be easy in usage for Customer

- Source: To Try

- Rationale: Customer shouldn't have any problems with using application. If the user had problems he wouldn't use this application.

- **Identifier: D2**

- Name: free

- Description: application should be available without paying for it

- Source: Made

- Rationale: Customer wouldn't like to pay for the application. He would prefer to do "old school" payment rather than paying for application.

- **Identifier: D3**

- Name: available

- Description: application should be available for all smartphones

- Source: Idea
- Rationale: All customers would like to use the application. Not only the customers who are in use of smart phones.

6.2. DECISION MATRIX

Skill Levels	1	2	3	4	5	6	7	8	9	10
Software / Hardware Installation								X		
Software / Hardware Maintanance								X		
HTML										X
PHP										X
OOP Concepts and Programming										X
Android Development Kit									X	
Mobile Developer									X	

Table 3 : Decision Matrix

7. ARCHITECTURAL VIEWS

7.1. USE CASES

7.1.1. USE CASE DIAGRAM: COSTUMER
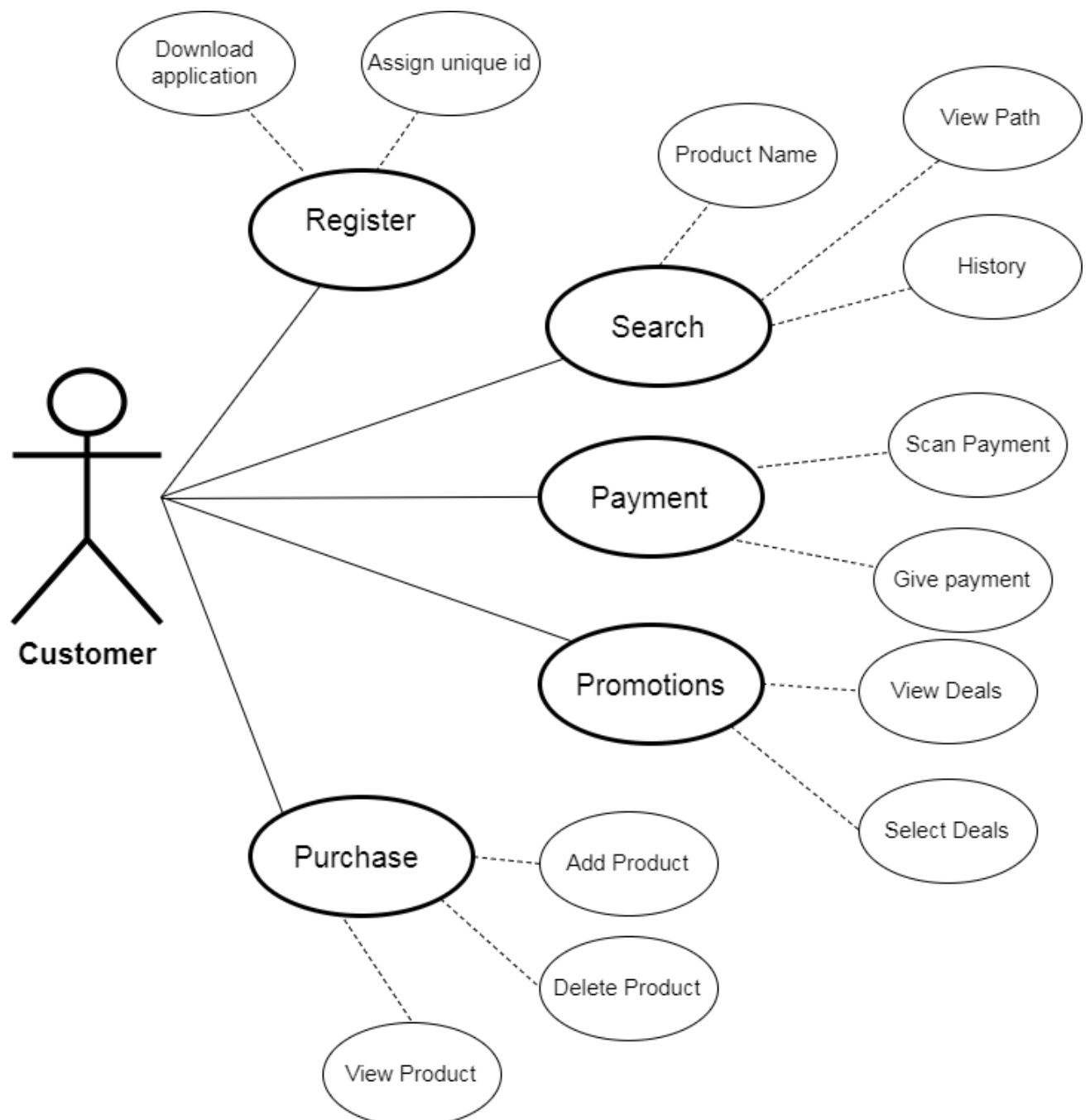
Figure 2 : Customer Use Case Diagram

7.1.1.1 USE CASE DESCRIPTION: Customer to Register

<Use Case 1.1: Register>		
Use case Id:	**1**	
Actors:	**Customers**	
Feature:	**Customers that come for shopping will register themselves**	
Pre-condition:	**Customer must have a working camera on smart phone**	
Scenarios		
Step #	**Action**	**Software Reaction**
1.	**Scan QR code to download app**	**App start downloading into the device**
2.	**Install the application**	**Unique ID is assigned to the customer**
Alternate Scenarios: Write additional, optional, branching or iterative steps. Refer to specific action number to ensure understandability.		
1a: **Device is not near proximately to scan QR code, so the app does not install.** **1b:** **Bring the mobile device close enough and scan QR code again**		
Post Conditions		
Step #	**Description**	
	Customer have an app on its phone	
	Customer have unique id	
	Customer can now purchase items	
Use Case Cross referenced	**1**	

Table 4 : Customer to Register Use Case Description

7.1.1.2 USE CASE DESCRIPTION: Customer to Purchase

<Use Case 1.2: purchase>		
Use case Id:	**2**	
Actors:	**Customers**	
Feature:	**Customers can add, delete or view items**	
Pre-condition:	**Customer should have register him/herself**	
Scenarios		
Step#	**Action**	**Software Reaction**
1.	**Scan the QR code on product**	**Product appear on screen to add/view**
2.	**Click on add button**	**Product added into the cart**
3	**View cart**	**List of purchased items will display**
Alternate Scenarios:		
1a: **Device is not near proximately to scan the QR code , so the product will not be able to view in application** **1b:** **Bring the mobile device close enough and scan QR code again**		
Post Conditions		
Step#	**Description**	
	Customer can now see its total items and bill	
Use Case Cross referenced	**1**	

Table 5 : Customer to Purchase Use Case Description

7.1.1.3 USE CASE DESCRIPTION: Customer to Search

<table>
<tr><td colspan="3" align="center"><Use Case 1.3: search></td></tr>
<tr><td>Use case Id:</td><td colspan="2">3</td></tr>
<tr><td>Actors:</td><td colspan="2" align="center">Customers</td></tr>
<tr><td>Feature:</td><td colspan="2" align="center">Customers can find the item shelf where the is place</td></tr>
<tr><td>Pre-condition:</td><td colspan="2">Customer should have register him/her self</td></tr>
<tr><td colspan="3">Scenarios</td></tr>
<tr><td>Step#</td><td>Action</td><td>Software Reaction</td></tr>
<tr><td>1.</td><td>Enter the name of the item</td><td>Display list of related items</td></tr>
<tr><td>2.</td><td>Click on desired item</td><td>Display product in map with shelf no</td></tr>
<tr><td></td><td></td><td></td></tr>
<tr><td colspan="3">Alternate Scenarios:</td></tr>
<tr><td colspan="3">

1a:

Product name is not entered correctly

1b:

Re-enter the name of the product

1c:

Item is not available
</td></tr>
<tr><td colspan="3">Post Conditions</td></tr>
<tr><td>Step#</td><td colspan="2">Description</td></tr>
<tr><td></td><td colspan="2">Customer can go to the desired shelf and scan the product</td></tr>
<tr><td colspan="2">Use Case Cross referenced</td><td>1</td></tr>
</table>

Table 6 : Customer to Search Use Case Description

7.1.1.4 USE CASE DESCRIPTION: Customer to Promotions

<table>
<tr><td colspan="3" align="center"><Use Case 1.4: promotions ></td></tr>
<tr><td>Use case Id:</td><td colspan="2">4</td></tr>
<tr><td>Actors:</td><td colspan="2" align="center">Customers</td></tr>
<tr><td>Feature:
related offers</td><td colspan="2" align="center">People/customers can see the general deals and product</td></tr>
<tr><td>Pre-condition:</td><td colspan="2" align="center">Customer should have register him/her self</td></tr>
<tr><td colspan="3">Scenarios</td></tr>
<tr><td>Step#</td><td>Action</td><td>Software Reaction</td></tr>
<tr><td>1.</td><td>Click on Promotions</td><td>List of promotions displayed</td></tr>
<tr><td>2.</td><td>Select the desire promotions</td><td>Promotion activates</td></tr>
<tr><td></td><td></td><td></td></tr>
<tr><td colspan="3">Alternate Scenarios:</td></tr>
<tr><td colspan="3">Post Conditions</td></tr>
<tr><td>Step#</td><td colspan="2">Description</td></tr>
<tr><td></td><td colspan="2">Customer can now get discount on the bases of selected promtion</td></tr>
<tr><td colspan="3">Use Case Cross referenced 1 , 2</td></tr>
</table>

Table 7 : Customer to Promotions Use Case Description

7.1.1.5 USE CASE DESCRIPTION: Customer to Payment

<table>
<tr><td colspan="3" align="center"><Use Case 1.5 payment></td></tr>
<tr><td>Use case Id:</td><td colspan="2" align="center">5</td></tr>
<tr><td>Actors:</td><td colspan="2" align="center">Customers, cashier</td></tr>
<tr><td>Feature:</td><td colspan="2" align="center">Customers can pay the amount</td></tr>
<tr><td>Pre-condition:</td><td colspan="2" align="center">Customer should have purchased products</td></tr>
<tr><td colspan="3">Scenarios</td></tr>
<tr><td>Step#</td><td>Action</td><td>Software Reaction</td></tr>
<tr><td>1.</td><td>Complete payment - in app</td><td>Generate Invoice and complete shopping</td></tr>
<tr><td>2.</td><td>Move to the cashier to package</td><td>Update data base inventory</td></tr>
<tr><td></td><td></td><td></td></tr>
<tr><td colspan="3">Alternate Scenarios:</td></tr>
<tr><td colspan="3">1a:

Cashier can scan items to add customer's cart in application so customer can use application to complete payment instead of using credit card or cash money

Post Conditions</td></tr>
<tr><td colspan="3">Step#   Description</td></tr>
<tr><td></td><td colspan="2">Inventory is updated</td></tr>
<tr><td colspan="3">Use Case Cross referenced 1 ,2</td></tr>
</table>

Table 8 : Customer to Payment Use Case Description

7.1.2. USE CASE DIAGRAM: EMPLOYEE

Figure 3 : Employee Use Case

7.1.2.1 USE CASE DESCRIPTION: Employee to Order

<table>
<tr><td colspan="3" align="center"><Use Case 2.1: order></td></tr>
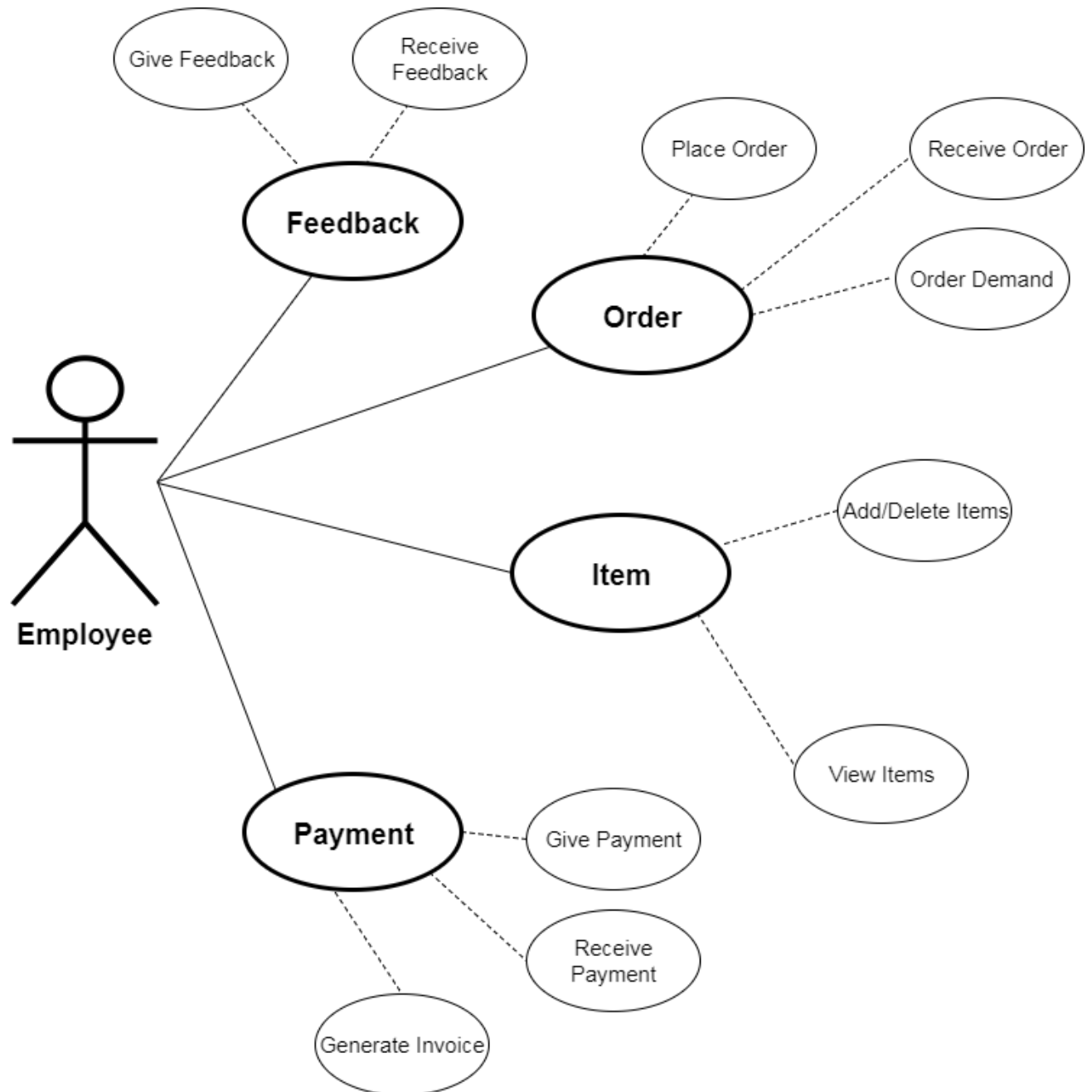<tr><td>Use case Id:</td><td colspan="2">6</td></tr>
<tr><td>Actors:</td><td colspan="2" align="center">Customers , Employee</td></tr>
<tr><td>Feature:</td><td colspan="2" align="center">Employee can place and receive order</td></tr>
<tr><td>Pre-condition:</td><td colspan="2">Employee must be logged in</td></tr>
<tr><td colspan="3">Scenarios</td></tr>
<tr><td>Step#</td><td>Action</td><td>Software Reaction</td></tr>
<tr><td>1.</td><td>Order to the supplier</td><td>Send notification to supplier</td></tr>
<tr><td>2.</td><td>Receive order from customer</td><td>Customer's cart is on purchased list</td></tr>
<tr><td></td><td></td><td></td></tr>
<tr><td colspan="3">Alternate Scenarios:</td></tr>
<tr><td colspan="3">1a:

Did not receive the Acknowledgment

1b:

Resend the order</td></tr>
<tr><td colspan="3">Post Conditions</td></tr>
<tr><td>Step#</td><td colspan="2">Description</td></tr>
<tr><td></td><td colspan="2">Supplier can provide items according to the order</td></tr>
<tr><td></td><td colspan="2">Cashier perform action with respect to the owner's order</td></tr>
<tr><td>Use Case Cross referenced</td><td colspan="2">1 ,2, 5</td></tr>
</table>

Table 9 : Employee to Order Use Case Description

7.1.2.2 USE CASE DESCRIPTION: Employee to Feedback

<Use Case 2.2: feedback>		
Use case Id:	7	
Actors:	**Employee, Owner**	
Feature:	**Employee can send and receive the feedback**	
Pre-condition:	**Employee must be logged in**	
Scenarios		
Step#	Action	Software Reaction
1.	**Cashier send the feedback to the owner by going to the feedback portion**	**Sent to the owner**
2.	**Receive feedback from owner**	**acknowledgment**
Alternate Scenarios:		
1a: **Did not receive the Acknowledgment** 1b: **Resend the feedback**		
Post Conditions		
Step#	Description	
	Perform actions according to feedback	
Use Case Cross referenced	1 ,2, 5	

Table 10 : Employee to feedback Use Case Description

7.1.2.3 USE CASE DESCRIPTION: Employee to Payment

<Use Case 2.3: payment>		
Use case Id:	**8**	
Actors:		**Customers , Employee**
Feature:		**Employee can give and receive payments**
Pre-condition:		**Employee must be logged in and customer have purchased items**
Scenarios		
Step#	**Action**	**Software Reaction**
1.	**Customer can scan QR code that employee generated**	**Total bill displayed in front of cashier**
2.	**Receive amount**	**Generate invoice**
Alternate Scenarios:		
1a:		
Device is not near proximately to scan QR code , so the payment will not complete		
1b:		
Bring the mobile device close enough and scan QR code again		
Post Conditions		
Step#	**Description**	
	Inventory must be update	
Use Case Cross referenced	**1 ,2, 5**	

Table 11 : Employee to payment Use Case Description

7.1.2.4 USE CASE DESCRIPTION: Employee to Item

<Use Case 2.4: item>		
Use case Id: **9**		
Actors:	**Employee**	
Feature:	**Employee can add ,delete and view items**	
Pre-condition:	**Employee must be logged in**	
Scenarios		
Step#	**Action**	**Software Reaction**
1.	**Employee enter the customer ID**	**Products list displayed**
2.	**Select the item and press delete**	**Item will delete and total bill upgrade**
3.	**Employee can add the item**	**Item will add and total bill upgrade**
Alternate Scenarios:		
1a:		
Customer ID did not found		
1b: **Display error** **1c:** **Re enter iD**		
Post Conditions		
Step#	**Description**	
	Invoice generate and update the inventory	
Use Case Cross referenced	**1 ,2, 5 ,8**	

Table 12 : Employee to Item Use Case Description

7.1.3 USE CASE DIAGRAM: Owner

Figure 4 : Owner Use Case Diagram

7.1.3.1 USE CASE DESCRIPTION: Owner to Manage Users

<table>
<tr><td colspan="3" align="center"><Use Case 3.1: manage users></td></tr>
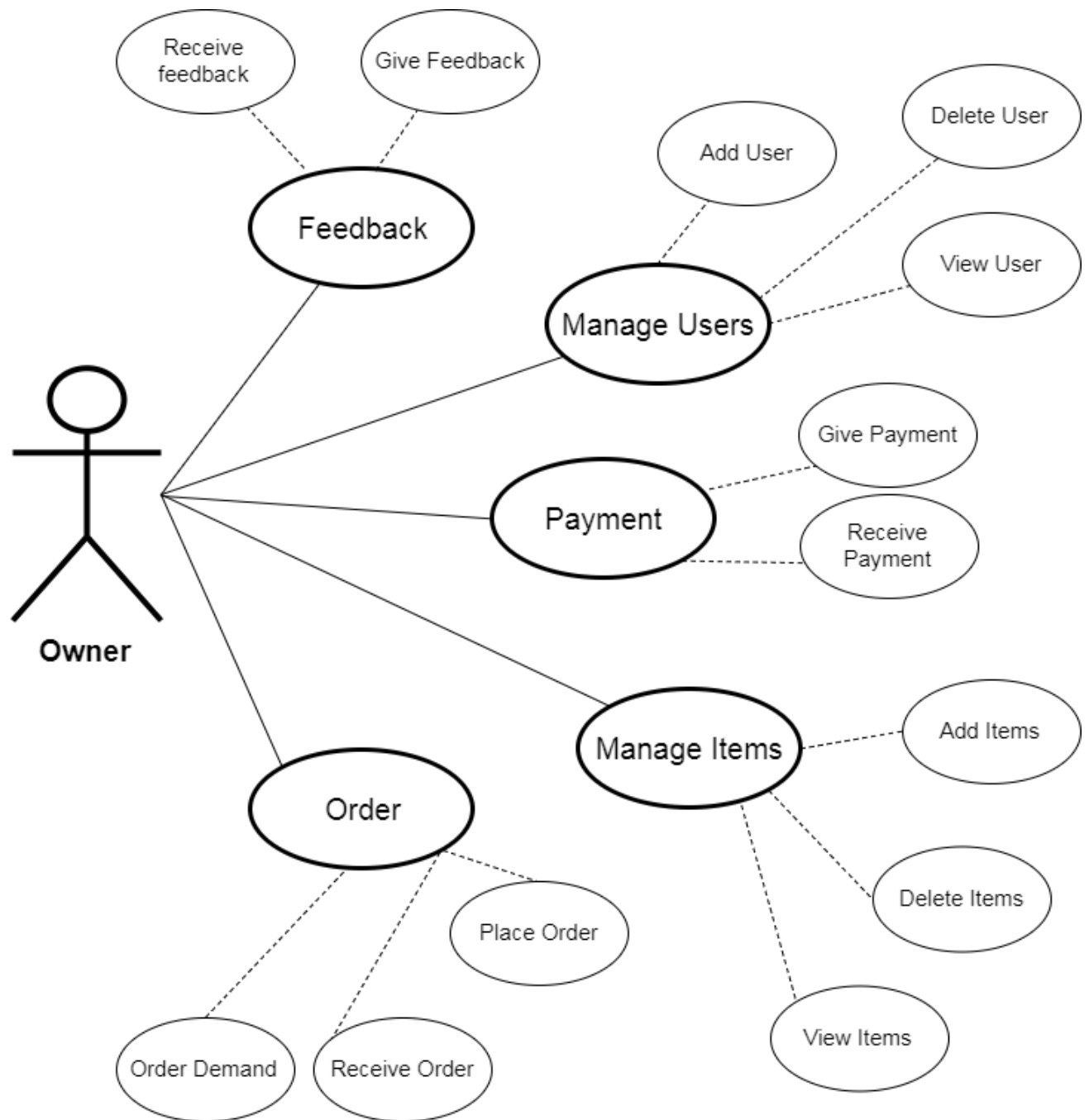<tr><td>Use case Id:</td><td colspan="2">11</td></tr>
<tr><td>Actors:</td><td colspan="2" align="center">owner</td></tr>
<tr><td>Feature:</td><td colspan="2" align="center">owner can add ,delete and view the employees</td></tr>
<tr><td>Pre-condition:</td><td colspan="2">owner must be logged in</td></tr>
<tr><td colspan="3">Scenarios</td></tr>
<tr><td>Step#</td><td>Action</td><td>Software Reaction</td></tr>
<tr><td>1.</td><td>Click on the add employee</td><td>Employee portal open</td></tr>
<tr><td>2.</td><td>Enter the details of the employee</td><td>Details are saved</td></tr>
<tr><td>3.</td><td>Click on view employees</td><td>Employees list displayed</td></tr>
<tr><td>4.</td><td>Select the employee and click on delete</td><td>Sytem will remove the employee and its information</td></tr>
<tr><td colspan="3">Alternate Scenarios:</td></tr>
<tr><td colspan="3">1a: Log in ID did not found

1b: Display error

1c: Re enter iD

2a: In adding an employee if employee of same name is already present then display error
Post Conditions</td></tr>
<tr><td>Step#</td><td colspan="2">Description</td></tr>
<tr><td></td><td colspan="2">Assign task to the employee</td></tr>
<tr><td colspan="3">Use Case Cross referenced</td></tr>
</table>

Table 13 : Owner to Manage Users Use Case Description

7.1.3.2 USE CASE DESCRIPTION: Owner to Manage Items

<Use Case 3.2: manage items>		
Use case Id: 12		
Actors:	Employee	
Feature:	owner can add, delete and view items	
Pre-condition:	owner must be logged in	
Scenarios		
Step#	**Action**	**Software Reaction**
1.	Owner click on display items button	Products list displayed
2.	Select the item and press delete	Item will delete the item from inventory
3.	owner can add the item	Item will add into inventory
Alternate Scenarios:		
1a:		
Supplier don't have the desired items		
1b:		
Find another supplier		
Post Conditions		
Step#	**Description**	
	update the inventory	
Use Case Cross referenced	1 ,2, 5 ,8	

Table 14 : Owner to Manage Items Use Case Description

7.1.3.3 USE CASE DESCRIPTION: Owner to Feedback

<Use case 3.3: feedback>		
Use case Id:	13	
Actors:	owner	
Feature:	owner can send and receive the feedback	
Pre-condition:	owner must be logged in	
Scenarios		
Step#	**Action**	**Software Reaction**
1.	owner send the feedback to the owner by going to the feedback portion	Sent to the employee
2.	Receive feedback	acknowledgment
Alternate Scenarios:		
1a: Did not receive the Acknowledgment **1b:** Resend the feedback		
Post Conditions		
Step#	**Description**	
	Perform actions according to feedback	
Use Case Cross referenced	1 ,2, 5	

Table 15 : Owner to Feedback Use Case Description

7.1.3.4 USE CASE DESCRIPTION: Owner to Payment

<Use Case 3.4: payment>		
Use case Id:	**14**	
Actors:	owner	
Feature:	**owner can give and receive payments**	
Pre-condition:	**Owner must be logged in and customer have purchased items**	
Scenarios		
Step#	**Action**	**Software Reaction**
1.	**Customer scan the QR code that generated by employee**	**Total bill displayed in front of cashier**
2.	**Receive amount**	**Generate invoice**
Alternate Scenarios:		
1a:		
Mobile is not near proximately to read the tag, so the invoice will not generate 1b:		
Bring the mobile close enough and read the tag again		
Post Conditions		
Step#	**Description**	
	Inventory must be update	
Use Case Cross referenced	**1 ,2, 5**	

Table 16 : Owner to Payment Use Case Description

7.1.3.5 USE CASE DESCRIPTION: Owner to Order

<table>
<tr><td colspan="3" align="center"><Use case 3.5: order></td></tr>
<tr><td>Use case Id:</td><td colspan="2">15</td></tr>
<tr><td>Actors:</td><td colspan="2" align="center">owner</td></tr>
<tr><td>Feature:</td><td colspan="2" align="center">owner can place and receive order</td></tr>
<tr><td>Pre-condition:</td><td colspan="2">owner must be logged in</td></tr>
<tr><td colspan="3">Scenarios</td></tr>
<tr><td>Step#</td><td>Action</td><td>Software Reaction</td></tr>
<tr><td>1.</td><td>Order to the supplier</td><td>Send notification to supplier</td></tr>
<tr><td>2.</td><td>Receive order from Employee</td><td>acknowledgment</td></tr>
<tr><td></td><td></td><td></td></tr>
<tr><td colspan="3">Alternate Scenarios:</td></tr>
<tr><td colspan="3">1a:

Did not receive the Acknowledgment

1b:

Resend the order</td></tr>
<tr><td colspan="3">Post Conditions</td></tr>
<tr><td>Step#</td><td colspan="2">Description</td></tr>
<tr><td></td><td colspan="2">Supplier can provide items according to the order</td></tr>
<tr><td></td><td colspan="2">Cashier perform action with respect to the owners order</td></tr>
<tr><td colspan="2">Use Case Cross referenced</td><td>1 ,2, 5</td></tr>
</table>

Table 17 : Owner to Order Use Case Description

7.2. LOGICAL VIEW

7.2.1. SEQUENCE DIAGRAM: Customer to register
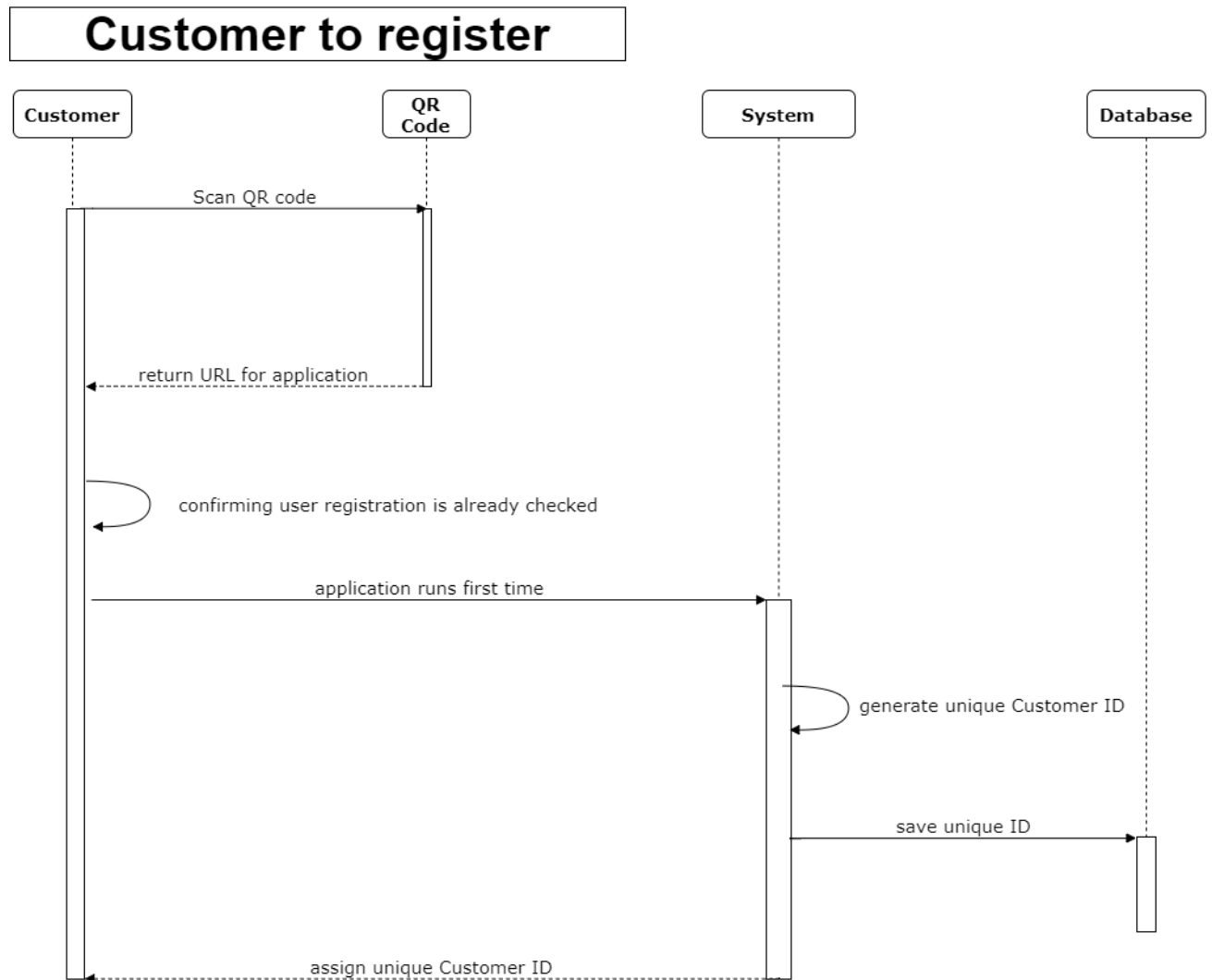
Figure 5 : Customer to Register Sequence Diagram
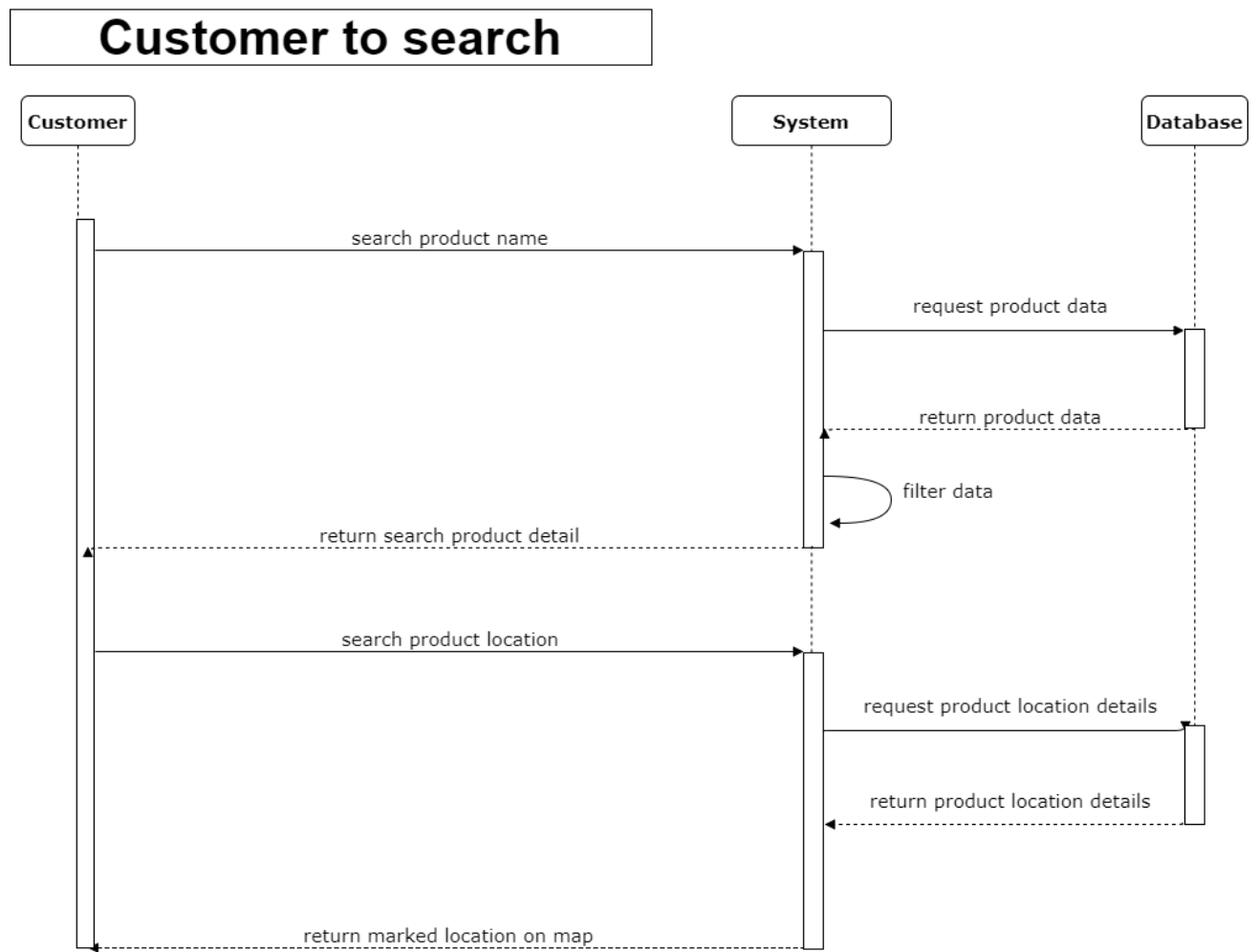
7.2.2. SEQUENCE DIAGRAM: Customer to search

Figure 6 : Customer to Search Sequence Diagram
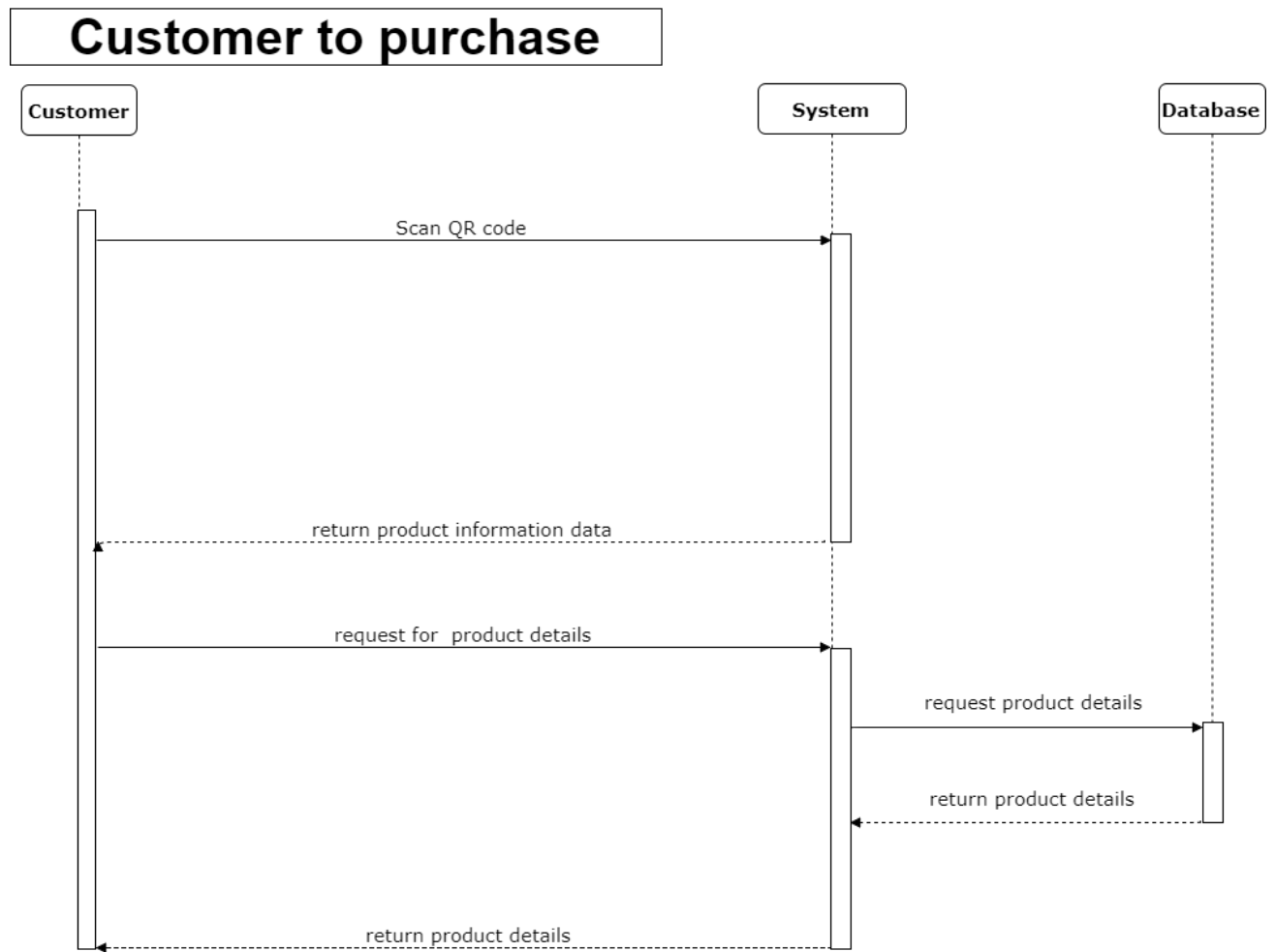
7.2.3. SEQUENCE DIAGRAM: Customer to purchase

Figure 7 : Customer to Purchase Sequence Diagram
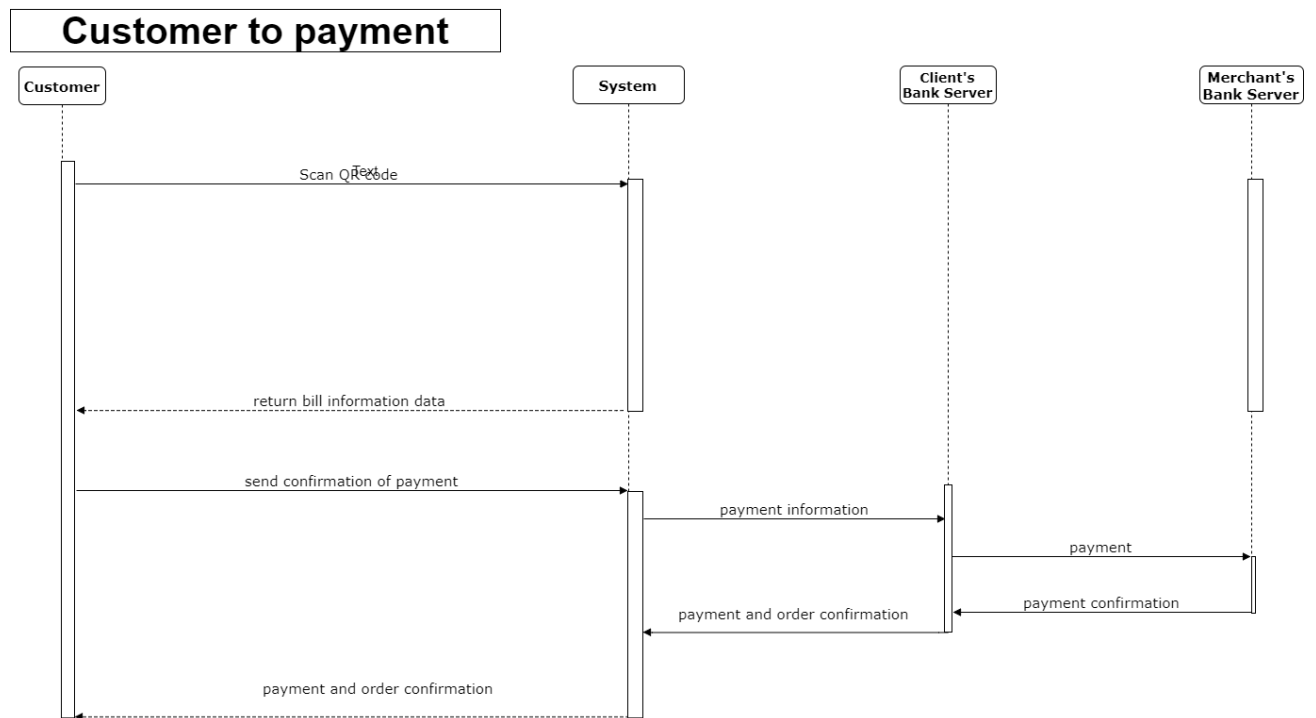
7.2.4. SEQUENCE DIAGRAM: Customer to payment

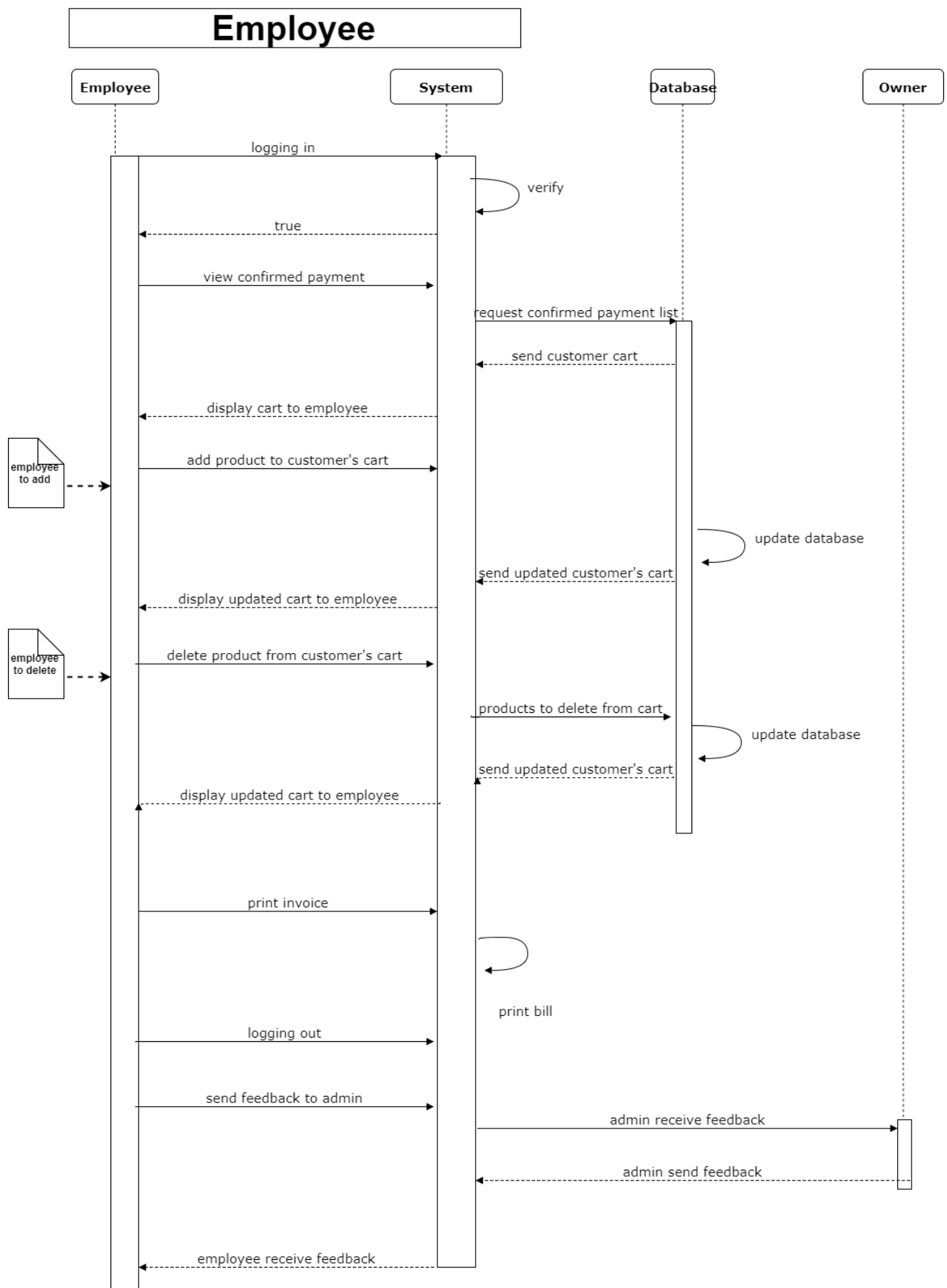*Figure 8 : Customer to Payment Sequence Diagram*

7.2.5 SEQUENCE DIAGRAM: Employee

Figure 9 : Employee Sequence Diagram

7.3. PROCESS VIEWS

7.3.1. ACTIVITY DIAGRAM: OWNER
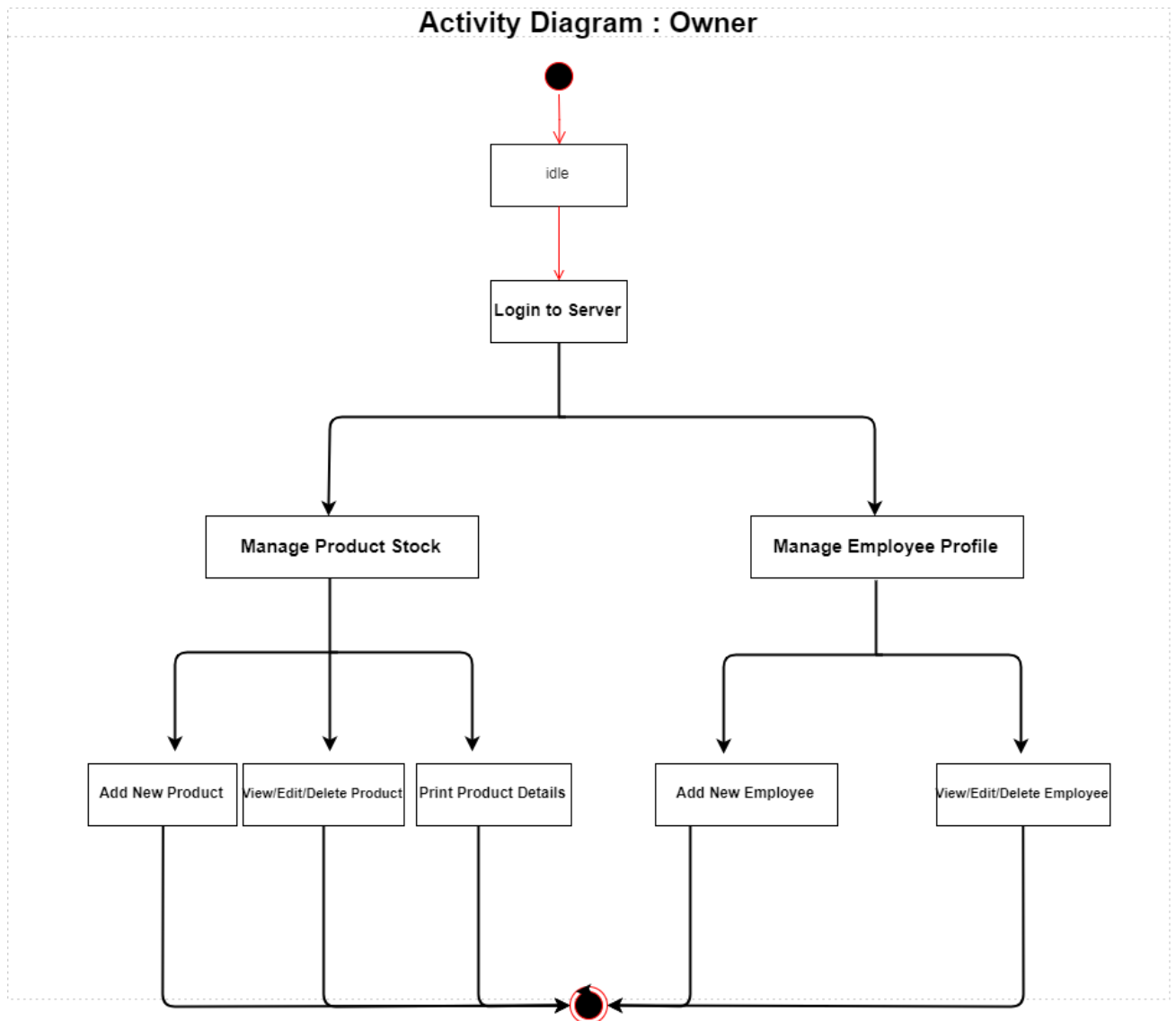
Figure 10 : Owner Activity Diagram

7.3.2. ACTIVITY DIAGRAM: COSTUMER
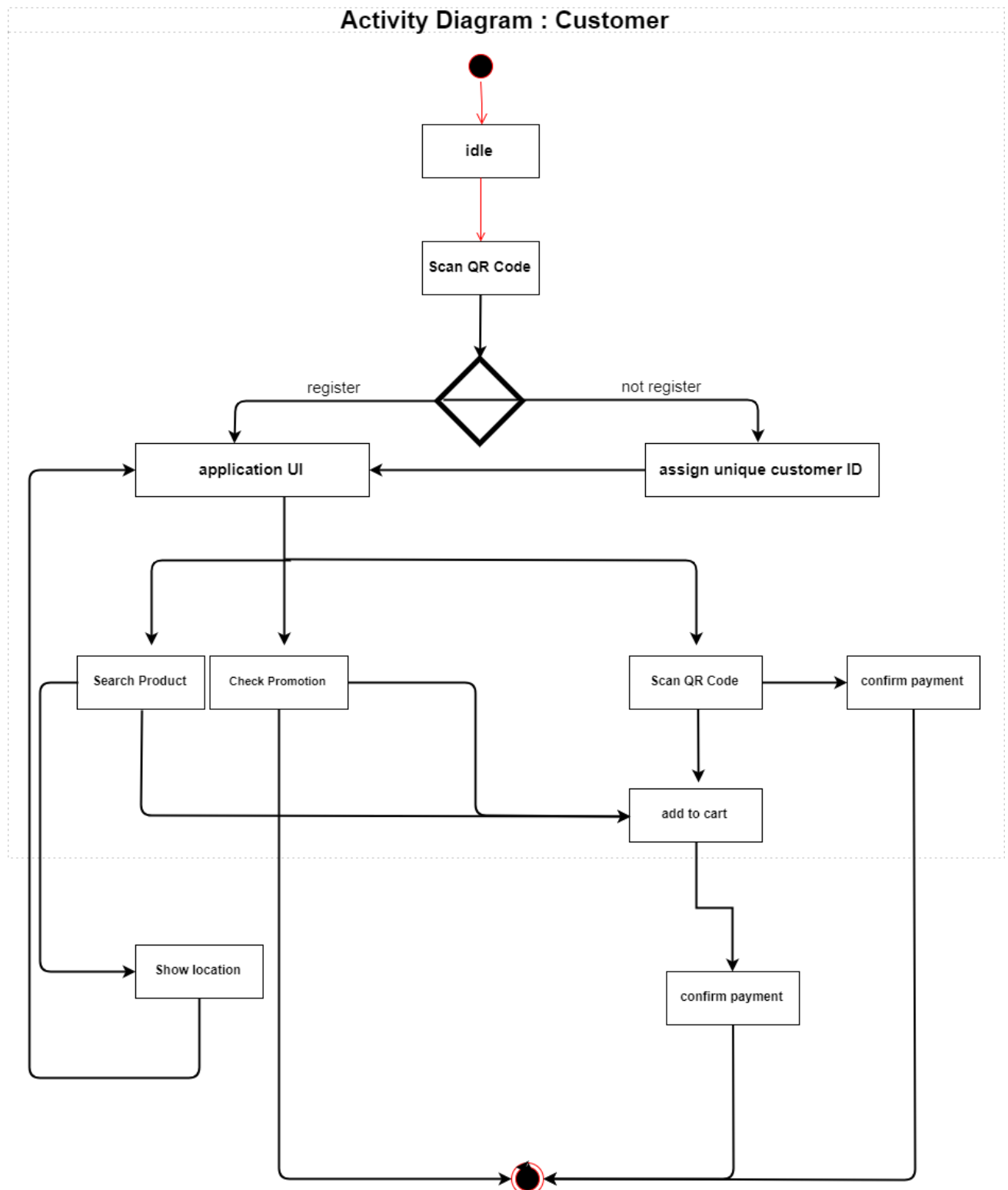
Figure 11 : Customer Activity Diagram
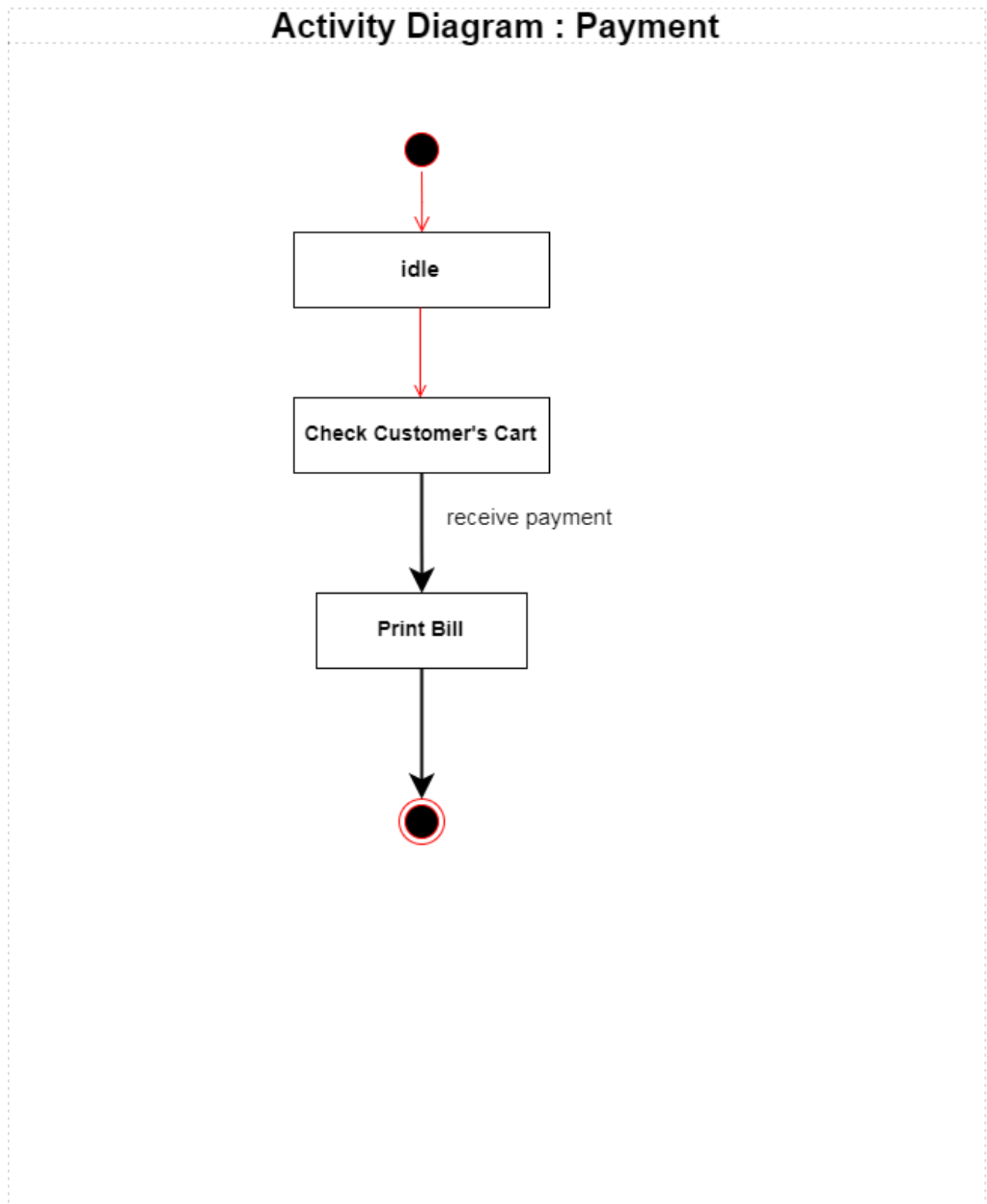
7.3.3. ACTIVITY DIAGRAM: PAYMENT
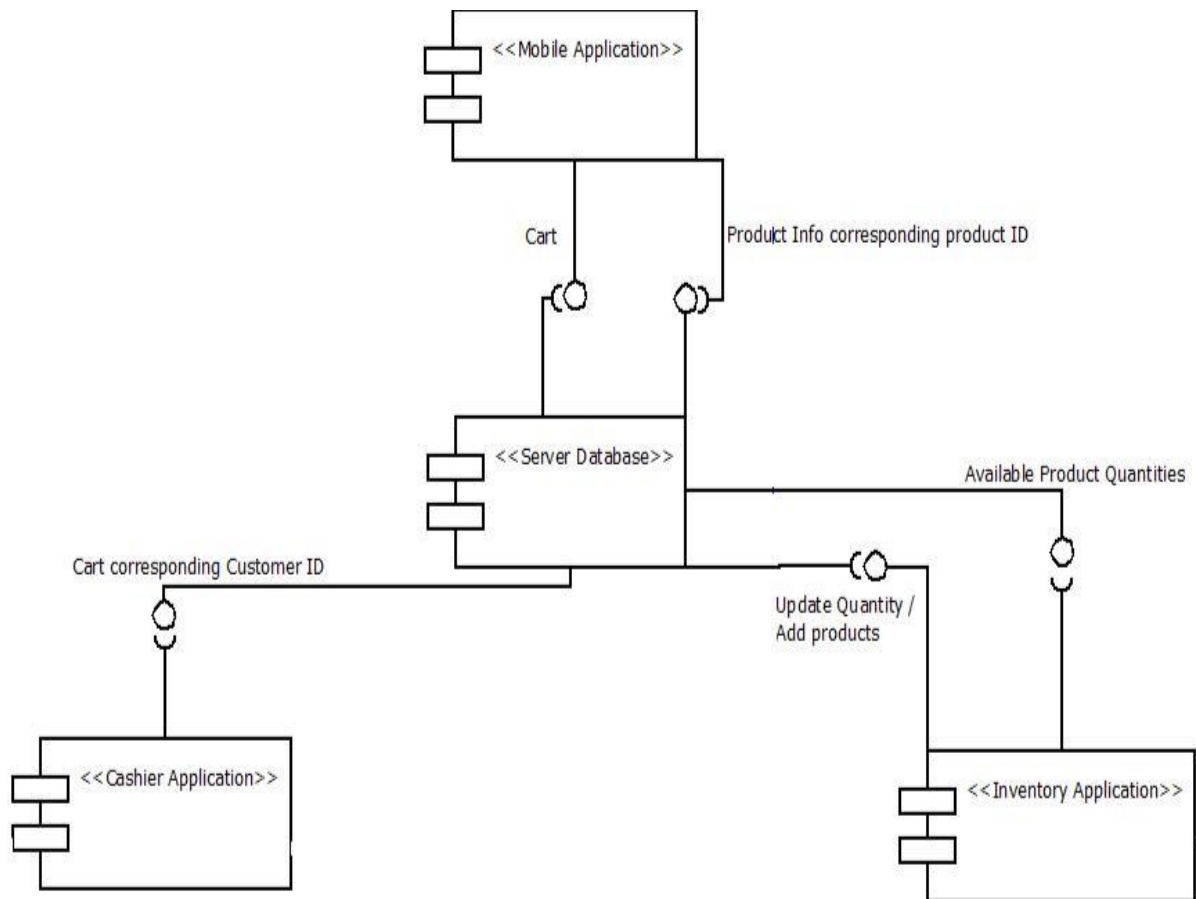

Figure 12 : Payment Activity Diagram

7.4. IMPLEMENTATION VIEW

Figure 13 : Implementation View