
Software Requirements and Design Document

for

Restaurant Management System

Prepared by Alishba Asif(20i-0582) and Ahmad Maqbool(20i-0702)



Vision 8

27/11/2022

Table of Contents

Table of Contents	ii
1. Introduction.....	1
1.1 Purpose	1
1.2 Product Scope.....	1
1.3 Title.....	1
1.4 Objectives	1
1.5 Problem Statement.....	1
2. Overall Description.....	2
2.1 Product Perspective	2
2.2 Product Functions.....	2
2.3 List of Use Cases	2
2.4 Extended Use Cases.....	2
2.5 Use Case Diagram	Error! Bookmark not defined.
3. Other Nonfunctional Requirements.....	Error! Bookmark not defined.
3.1 Performance Requirements.....	Error! Bookmark not defined.
3.2 Safety Requirements.....	Error! Bookmark not defined.
3.3 Security Requirements.....	Error! Bookmark not defined.
3.4 Software Quality Attributes.....	Error! Bookmark not defined.
3.5 Business Rules.....	Error! Bookmark not defined.
3.6 Operating Environment	Error! Bookmark not defined.
3.7 User Interfaces.....	Error! Bookmark not defined.
4. Domain Model	Error! Bookmark not defined.
5. System Sequence Diagram	Error! Bookmark not defined.
6. Sequence Diagram	Error! Bookmark not defined.
7. Class Diagram	Error! Bookmark not defined.
8. Package Diagram	Error! Bookmark not defined.
9. Deployment Diagram.....	Error! Bookmark not defined.

1. Introduction

1.1 Purpose

The purpose of this application is to automate the current work being done in restaurants manually by employees. All the records will be kept on our automated system which will remove chance of human error in records.

1.2 Product Scope

The scope of our project will be the restaurant industry.

1.3 Title

The aim of this project is to help manage the restaurant more efficiently and effectively by automating the restaurant's billing, employee details, daily reports and inventory which are currently being handled manually.

1.4 Objectives

The main objectives we hope to achieve with our project are to automate and track the following things:

- 1) The inventory so that the menu can be updated accordingly to conduct hassle free order acceptance
- 2) Daily sales reports in order to keep track of the ordering and revenue generation
- 3) Employee management which includes their timely pays and scheduling
- 4) Schedule Reservation which schedules a reservation for a customer
- 5) Monthly reports to keep track of revenue generated over the time period of a month
- 6) Menu is updated and can be viewed by customers to place their order.
- 7) Payment is generated correctly based on the order that was placed.
- 8) Employee can record and place customer's order.

1.5 Problem Statement

The reason we have chosen this project is because we have felt its need first hand. We have observed how many restaurants are still keeping all their records on paper which not only requires extra manpower but also creates a higher chance of error in accounts and records of the restaurant. It is very important to make sure that the records of a restaurant, like any other business, are handled carefully and efficiently. Thus, with our project we aim to provide the restaurant admin and employees with an easy to use software that allows them to keep track of their daily records,

accounts and generate precise sales reports as well as give the owners a good detailed overview of how their restaurant is performing.

Apart from that, our inventory management feature will automatically inform the employees when a product is running low in inventory and needs to be restocked. This will help the restaurant avoid running out of products and not being able to serve their customers.

2. Overall Description

2.1 Product Perspective

It is a replacement for the manual restaurant system where orders are taken on paper and all the records of the restaurant including the accounts details and payment registers are kept on paper in physical form. This system will allow all these actions to be automated.

2.2 Product Functions

- Management of Employees
- Place Order
- Updating Menu

2.3 List of Use Cases

2.4 Extended Use Cases

<u>UC1</u>																					
By:	Alishba Asif – 20i-0582																				
Use Case Name:	Manage Employees																				
Scope:	Restaurant Management System																				
Level:	User-goal																				
Primary Actor:																					
Stake Holders and Interests:	<ul style="list-style-type: none"> - Manager: Wants to register new employees, update employee details and remove employees conveniently, efficiently and keep the system updated to avoid any hindrances. - Employee: Wants to make sure that their details are up to date at all times and that they are registered within the system. 																				
Main Success Scenario:	<table> <tr> <th><u>Actor Action</u></th><th><u>System Response</u></th></tr> <tr> <td>1. New employee arrives and provides details to the manager.</td><td></td></tr> <tr> <td>2. Manager enters login details.</td><td></td></tr> <tr> <td></td><td>3. Authenticates the manager.</td></tr> <tr> <td>4. Manger enters employee details into the system until all the details have been entered.</td><td></td></tr> <tr> <td></td><td>5. Records each employee detail.</td></tr> <tr> <td></td><td>6. Provides final description of all employee details.</td></tr> <tr> <td>7. Manager enters employee shift date and time into the system.</td><td></td></tr> <tr> <td></td><td>8. Records the employee shift details.</td></tr> <tr> <td>9. Manager tells the employee their schedule.</td><td></td></tr> </table>	<u>Actor Action</u>	<u>System Response</u>	1. New employee arrives and provides details to the manager.		2. Manager enters login details.			3. Authenticates the manager.	4. Manger enters employee details into the system until all the details have been entered.			5. Records each employee detail.		6. Provides final description of all employee details.	7. Manager enters employee shift date and time into the system.			8. Records the employee shift details.	9. Manager tells the employee their schedule.	
<u>Actor Action</u>	<u>System Response</u>																				
1. New employee arrives and provides details to the manager.																					
2. Manager enters login details.																					
	3. Authenticates the manager.																				
4. Manger enters employee details into the system until all the details have been entered.																					
	5. Records each employee detail.																				
	6. Provides final description of all employee details.																				
7. Manager enters employee shift date and time into the system.																					
	8. Records the employee shift details.																				
9. Manager tells the employee their schedule.																					
Extensions:	<p>*a. At any time, System fails:</p> <ol style="list-style-type: none"> 1. Manager restarts the system, logs in and requests recovery of previous details. 2. System reconstructs prior details. 4a. Manager resumes entering employee details. 																				

	<ol style="list-style-type: none"> 1a. Previous details not found. <ol style="list-style-type: none"> 1. System signals error to manager. 2. Manager probably starts a new document and re-enters details. 4b. Employee is not a new employee and already exists within the system. <ol style="list-style-type: none"> 1. Manager enters employee's account info into the system. <ol style="list-style-type: none"> 1a. Details not found. <ol style="list-style-type: none"> 1. Manager re-enters account info of employee. 2. System verifies that employee exists within the system. <p>*b. Employee wants to update his/her information.</p> <ol style="list-style-type: none"> 1. Manager enters employee ID into the system. 2. Manager enters the updated information in the system. 3. Manager prompts system to display the updated information. 4. Manager confirms the updated information. <p>*c. Employee wants to be removed.</p> <ol style="list-style-type: none"> 1. Manager enters employee ID into the system. 2. Manager confirms that the displayed information corresponds with Employee. 3. Employee confirms. 4. Manager removes all the information related to the employee. 5. Manager re-enters employee ID to confirm that employee has been removed.
Pre-Conditions:	Manager is identified and authenticated.
Post-Conditions:	<ul style="list-style-type: none"> - New employee is registered - The employee's details are saved in the system - The employee is scheduled for the weekly shifts - Employee has been notified of the updates.

UC2																						
By:	Alishba Asif – 20i-0582																					
Use Case Name:	Payment																					
Scope:	Restaurant Management System																					
Level:	User-goal																					
Primary Actor:	Employee																					
Stake Holders and Interests:	<ul style="list-style-type: none">- Employee: Wants accurate and fast entry of the bill into the system without any errors because the cash shortage is deducted from his/her salary.- Customer: Wants bill and fast service with minimal effort. Wants easily visible display of the price of each item that they ordered and wants proof of the transaction in the form of a receipt.																					
Main Success Scenario:	<table><tr><th>Actor Action</th><th>System Response</th></tr><tr><td>1. Customer services employee to receive the bill.</td><td></td></tr><tr><td>2. Employee starts a new transaction.</td><td></td></tr><tr><td>3. Employee enters each item identifier. <i>Employee repeats steps 3-4 until items are complete.</i></td><td></td></tr><tr><td></td><td>4. Records each item and presents the items price and running total.</td></tr><tr><td></td><td>5. Presents the total amount of bill along with the tax included in it. System presents a receipt.</td></tr><tr><td>6. Employee hands customer the receipt and signals for payment.</td><td></td></tr><tr><td>7. Customer pays.</td><td></td></tr><tr><td></td><td>8. Handles payment</td></tr><tr><td></td><td>9. Saves the completed transaction and the information is sent to external account for all revenues, transactions and inventory items to be updated.</td></tr></table>		Actor Action	System Response	1. Customer services employee to receive the bill.		2. Employee starts a new transaction.		3. Employee enters each item identifier. <i>Employee repeats steps 3-4 until items are complete.</i>			4. Records each item and presents the items price and running total.		5. Presents the total amount of bill along with the tax included in it. System presents a receipt.	6. Employee hands customer the receipt and signals for payment.		7. Customer pays.			8. Handles payment		9. Saves the completed transaction and the information is sent to external account for all revenues, transactions and inventory items to be updated.
Actor Action	System Response																					
1. Customer services employee to receive the bill.																						
2. Employee starts a new transaction.																						
3. Employee enters each item identifier. <i>Employee repeats steps 3-4 until items are complete.</i>																						
	4. Records each item and presents the items price and running total.																					
	5. Presents the total amount of bill along with the tax included in it. System presents a receipt.																					
6. Employee hands customer the receipt and signals for payment.																						
7. Customer pays.																						
	8. Handles payment																					
	9. Saves the completed transaction and the information is sent to external account for all revenues, transactions and inventory items to be updated.																					
Extensions:	*a. At any time, Manager requests an override operation:																					

	<ol style="list-style-type: none"> 1. Manager authorized mode is entered by the system. 2. Manager performs a manager-mode operation such as generate a report, add employee, remove employee or update inventory etc. 3. System reverts back to Employee-authorized mode. <p>*b. At any time, the system fails:</p> <ol style="list-style-type: none"> 1. Employee restarts the system, logs in and requests the system for the recovery of the previous state. 2. System reconstructs the previous state. <p>1a. Employee indicates the suspended transaction to be resumed.</p> <ol style="list-style-type: none"> 1. Employee resumes operation and enters the transaction ID to the retrieve the sale. 2. The system displays the transaction with the subtotal. <ol style="list-style-type: none"> 2a. The transaction is not found. <ol style="list-style-type: none"> 1. The System signals the error to the employee. 2. The employee starts a new transaction and re-enters all the items. 3. The employee continues the transaction by adding more items if needed. <ol style="list-style-type: none"> 3a. Invalid menu item ID (it is not in the system) <ol style="list-style-type: none"> 1. The system signals error and rejects the item's entry. 2. The employee responds to the error <ol style="list-style-type: none"> 2a. The error is readable by Employee (numeric or name error) <ol style="list-style-type: none"> 1. The employee enters the menu item ID 2. The system displays the price and name of the item (description of the item) <ol style="list-style-type: none"> 2a. The item ID is invalid 2b. Item ID is not recognized by the system. <ol style="list-style-type: none"> 1. Employee asks Manager to intervene. System enters manager-authorized mode. 2. Manager checks update menu. <ol style="list-style-type: none"> 2a. Item ID is separate in the menu. Manager gives that item ID to the Employee to enter in the system. 2b. Item ID is the same but system gives error. <ol style="list-style-type: none"> 1. Manager changes the item ID. 3. Employee manually enters item id and price into and requests standard taxation for the item.
Pre-Conditions:	Employee is identified and authenticated.
Post-Conditions:	<ul style="list-style-type: none"> - Transaction is saved. - The tax is correctly calculated and added to receipt.

	<ul style="list-style-type: none">- A receipt is generated.- Payment is recorded.
--	--

UC3																	
By:	Alishba Asif – 20i-0582																
Use Case Name:	Schedule Reservations																
Scope:	Restaurant Management System																
Level:	User-goal																
Primary Actor:	Employee																
Stake Holders and Interests:	<ul style="list-style-type: none"> - Employee: Wants accurate, fast and without error reservation in order to keep the customer satisfied. - Customer: Wants to make sure the reservation is fast and accurate with minimal effort. Wants proof of the reservation. 																
Main Success Scenario:	<table> <tr> <th><u>Actor Action</u></th><th><u>System Response</u></th></tr> <tr> <td>1. Customer calls the restaurant and asks employee to make a reservation.</td><td></td></tr> <tr> <td>2. Employee starts a new reservation.</td><td></td></tr> <tr> <td>3. Employee enters the customer's details into the reservation.</td><td></td></tr> <tr> <td></td><td>4. Records each customer detail</td></tr> <tr> <td></td><td>5. Presents the scheduled reservation along with time, date and customer details.</td></tr> <tr> <td>6. Employee confirms reservation details with customer.</td><td></td></tr> <tr> <td></td><td>7. Reservation is recorded into the system.</td></tr> </table>	<u>Actor Action</u>	<u>System Response</u>	1. Customer calls the restaurant and asks employee to make a reservation.		2. Employee starts a new reservation.		3. Employee enters the customer's details into the reservation.			4. Records each customer detail		5. Presents the scheduled reservation along with time, date and customer details.	6. Employee confirms reservation details with customer.			7. Reservation is recorded into the system.
<u>Actor Action</u>	<u>System Response</u>																
1. Customer calls the restaurant and asks employee to make a reservation.																	
2. Employee starts a new reservation.																	
3. Employee enters the customer's details into the reservation.																	
	4. Records each customer detail																
	5. Presents the scheduled reservation along with time, date and customer details.																
6. Employee confirms reservation details with customer.																	
	7. Reservation is recorded into the system.																
Extensions:	<p>*a. At any time, Manager requests an override operation:</p> <ol style="list-style-type: none"> 1. Manager authorized mode is entered by the system. 2. Manager performs a manager-mode operation such as generate a report, add employee, remove employee or update inventory etc. 3. System reverts back to Employee-authorized mode. <p>*b. At any time, the system fails:</p> <ol style="list-style-type: none"> 1. Employee restarts the system, logs in and requests the system for the recovery of the previous state. 2. System reconstructs the previous state. <p>1a. Employee indicates the suspended reservation to be resumed.</p>																

	<ol style="list-style-type: none"> 1. Employee resumes operation and enters the reservation ID to retrieve the reservation scheduling. 2. The system displays the reservation with customer's details. <ol style="list-style-type: none"> 2a. The reservation is not found. <ol style="list-style-type: none"> 1. The System signals the error to the employee. 2. The employee starts a new reservations and re-enters all the customer's info.
Pre-Conditions:	Employee is identified and authenticated.
Post-Conditions:	<ul style="list-style-type: none"> - New reservation is scheduled. - The customer's reservation details are saved in the system. - Reservation does not overlap other reservations. - Customer has been notified of reservation confirmation.

UC4:													
By:	Alishba Asif – 20i-0582												
Use Case Name:	Generate Daily Reports												
Scope:	Restaurant Management System												
Level:	User-goal												
Primary Actor:	Manager												
Stake Holders and Interests:	<ul style="list-style-type: none"> - Manager: Wants accurate, fast, readable and without error generation of daily reports. 												
Main Success Scenario:	<table> <tr> <th><u>Actor Action</u></th><th><u>System Response</u></th></tr> <tr> <td>1. Manager starts a new report of the day.</td><td></td></tr> <tr> <td></td><td>2. Presents report of the daily transactions, expenses and revenue that was generated.</td></tr> <tr> <td></td><td>3. Report saved in the system to update the monthly report and update inventory.</td></tr> <tr> <td></td><td>4. System presents report.</td></tr> <tr> <td>5. Manager reviews the report.</td><td></td></tr> </table>	<u>Actor Action</u>	<u>System Response</u>	1. Manager starts a new report of the day.			2. Presents report of the daily transactions, expenses and revenue that was generated.		3. Report saved in the system to update the monthly report and update inventory.		4. System presents report.	5. Manager reviews the report.	
<u>Actor Action</u>	<u>System Response</u>												
1. Manager starts a new report of the day.													
	2. Presents report of the daily transactions, expenses and revenue that was generated.												
	3. Report saved in the system to update the monthly report and update inventory.												
	4. System presents report.												
5. Manager reviews the report.													
Extensions:	<p>*a. At any time, Employee requests an override operation:</p> <ol style="list-style-type: none"> 1.Employee authorized mode is entered by the system. 2.Employee performs employee-mode operation such as schedule reservation, payment or generate order. 3.System reverts back to Manager-authorized mode. <p>*b. At any time, the system fails:</p> <ol style="list-style-type: none"> 1.Manager restarts the system, logs in and requests the system for the recovery of the previous state. 2.System reconstructs the previous state. <ol style="list-style-type: none"> 1a. Manager indicates the suspended report generation to be resumed. <ol style="list-style-type: none"> 1. Manager resumes operation and enters the report ID to retrieve the report. 2. The system displays the daily report which consists of expenses, revenue and transaction details. 2a. The report is not found. <ol style="list-style-type: none"> 1. The System signals the error to the Manager. 2. The manager prompts system to generate another daily report of the day. 												

Pre-Conditions:	Manager is identified and authenticated.
Post-Conditions:	<ul style="list-style-type: none">- New report is created.- The details of the report are saved in system.

UC5:													
By:	Ahmad Maqbool – 20i-0702												
Use Case Name:	Generate Monthly Reports												
Scope:	Restaurant Management System												
Level:	User-goal												
Primary Actor:	Manager												
Stake Holders and Interests:	<ul style="list-style-type: none"> - Manager: Wants accurate, fast, readable and without error generation of monthly reports. 												
Main Success Scenario:	<table> <tr> <th><u>Actor Action</u></th><th><u>System Response</u></th></tr> <tr> <td>1. Manager starts a new report of the month.</td><td></td></tr> <tr> <td></td><td>2. Presents report of the monthly expenses such as electricity and gas bills and revenue that was generated.</td></tr> <tr> <td></td><td>3. Presents report of the inventory that was at the start of the month compared to the end of the month.</td></tr> <tr> <td></td><td>4. System presents report.</td></tr> <tr> <td>5. Manager reviews the report.</td><td></td></tr> </table>	<u>Actor Action</u>	<u>System Response</u>	1. Manager starts a new report of the month.			2. Presents report of the monthly expenses such as electricity and gas bills and revenue that was generated.		3. Presents report of the inventory that was at the start of the month compared to the end of the month.		4. System presents report.	5. Manager reviews the report.	
<u>Actor Action</u>	<u>System Response</u>												
1. Manager starts a new report of the month.													
	2. Presents report of the monthly expenses such as electricity and gas bills and revenue that was generated.												
	3. Presents report of the inventory that was at the start of the month compared to the end of the month.												
	4. System presents report.												
5. Manager reviews the report.													
Extensions:	<p>*a. At any time, Employee requests an override operation:</p> <ol style="list-style-type: none"> 1.Employee authorized mode is entered by the system. 2.Employee performs employee-mode operation such as schedule reservation, payment or generate order. 3.System reverts back to Manager-authorized mode. <p>*b. At any time, the system fails:</p> <ol style="list-style-type: none"> 1.Manager restarts the system, logs in and requests the system for the recovery of the previous state. 2.System reconstructs the previous state. <ol style="list-style-type: none"> 1a. Manager indicates the suspended report generation to be resumed. <ol style="list-style-type: none"> 1. Manager resumes operation and enters the report ID to retrieve the report. 2. The system displays the daily report which consists of expenses (electricity, gas etc.), revenue and inventory details. 2a. The report is not found. <ol style="list-style-type: none"> 1. The System signals the error to the Manager. 												

	2. The manager prompts system to generate another monthly report.
Pre-Conditions:	Manager is identified and authenticated.
Post-Conditions:	<ul style="list-style-type: none"> - New report is created. - The details of the report are saved in system.

UC6																	
By:	Ahmad Maqbool - 20i-0702																
Use Case Name:	Manage Inventory																
Scope:	Restaurant Management System																
Level:	User – Goal																
Primary Actor:	Manager																
Stake Holders and Interests:	<ul style="list-style-type: none"> - Admin: Wants to be able to update items in the inventory by restocking items and removing items when they are used. 																
Main Success Scenario:	<table> <tr> <th><u>Actor Action</u></th><th><u>System Response</u></th></tr> <tr> <td>1-Manager Enters login details</td><td></td></tr> <tr> <td></td><td>2-Authenticates the manager</td></tr> <tr> <td>3-Manager enters the name of the item he wants to update in the inventory.</td><td></td></tr> <tr> <td></td><td>4-System provides description of the item and its current quantity in the inventory.</td></tr> <tr> <td>5-Admin adds or removes from quantity of item.</td><td></td></tr> <tr> <td></td><td>6- System updates the item quantity in inventory and displays new quantity.</td></tr> <tr> <td>7- Manager reviews the updated inventory.</td><td></td></tr> </table>	<u>Actor Action</u>	<u>System Response</u>	1-Manager Enters login details			2-Authenticates the manager	3-Manager enters the name of the item he wants to update in the inventory.			4-System provides description of the item and its current quantity in the inventory.	5-Admin adds or removes from quantity of item.			6- System updates the item quantity in inventory and displays new quantity.	7- Manager reviews the updated inventory.	
<u>Actor Action</u>	<u>System Response</u>																
1-Manager Enters login details																	
	2-Authenticates the manager																
3-Manager enters the name of the item he wants to update in the inventory.																	
	4-System provides description of the item and its current quantity in the inventory.																
5-Admin adds or removes from quantity of item.																	
	6- System updates the item quantity in inventory and displays new quantity.																
7- Manager reviews the updated inventory.																	
Extensions	*a. At any time, the system fails: 1.Manager restarts the system, logs in and requests the system for the recovery of the previous state.																

2. System reconstructs the previous state.
3. Manager continues performing updates in inventory.

***b. Manager deletes an item from inventory**

1. Manager searches for item.
2. If item does not exist in inventory,
 1. Manager is notified and taken back to home page.
3. Else if found, they are shown the details of the item.
 1. System asks manager to confirm delete of item.
 2. Manager confirms deletion of item and item is deleted from inventory in system

***c. Manager wants to list new item in inventory:**

1. Manager creates a new item in inventory.
2. Manager adds all details of the new item into the system.
3. System updates inventory and stores new item.

1a. Manager enters wrong credentials

1. Message is displayed and user is taken back to login page to reenter login details.

3a. Inventory has no record of the item in inventory.

1. In this case, user creates new item and adds its details into the system.

3b. If inventory has record of item

1. Item details along with quantity are displayed to user
2. Manager enters quantity to update in inventory.
3. System updates the inventory.

5a. Manager decrements quantity of an item that is already finished from inventory.

1. Manager is notified that item is already finished in inventory and cannot be reduced further.
2. Manager is taken back to home page.

5b. Item chosen for update is currently present

1. System asks for quantity of update of item
2. Item quantity is updated in inventory

Pre-Conditions	- Manager is identified and authenticated
Post-Conditions	- Item quantity in inventory is updated and saved into the system

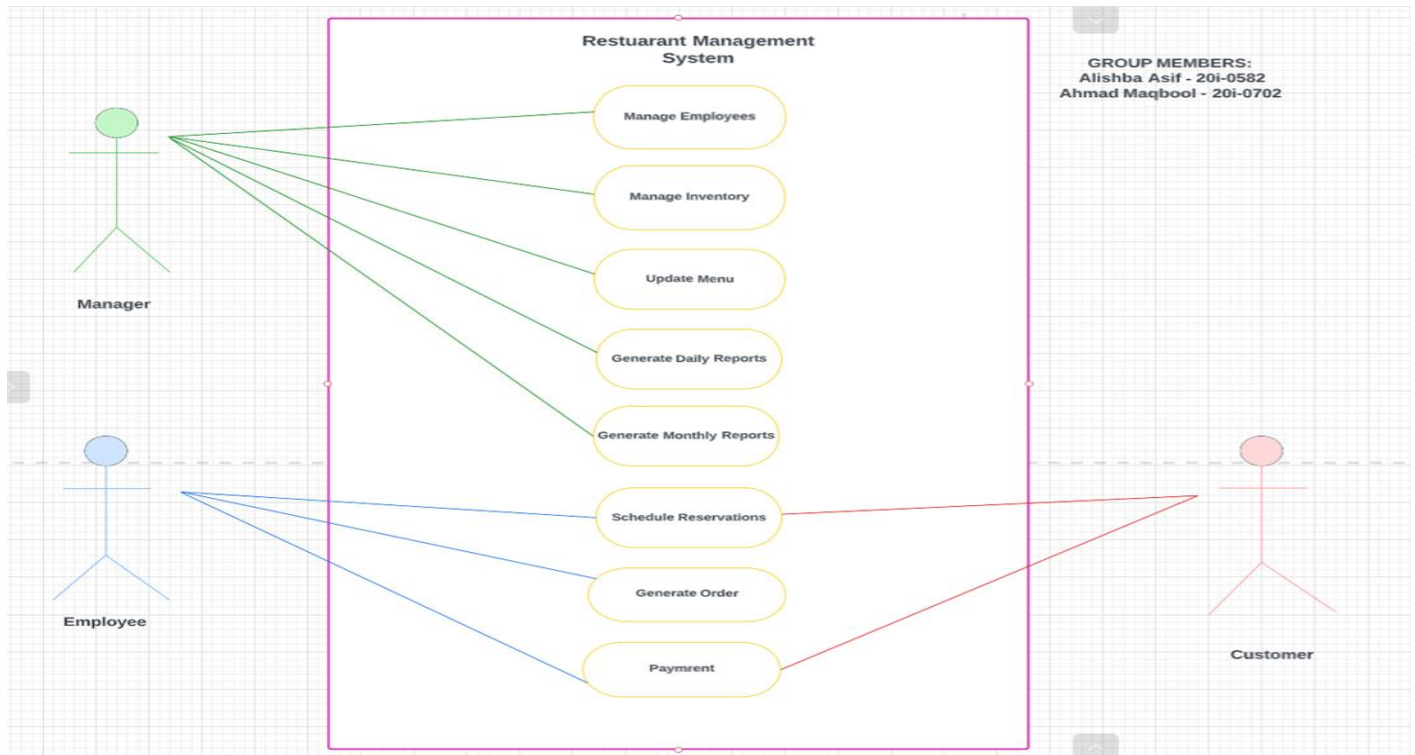
UC7																					
By:	Ahmad Maqbool – 20i-0702																				
Use Case Name:	Generate order																				
Scope:	Restaurant Management System																				
Level:	User – Goal																				
Primary Actor:	Employee																				
Stake Holders and Interests:	<ul style="list-style-type: none"> - Employee: wants accurate fast generation of customer's order for food - Customer: wants to place his order in an easy and efficient way. 																				
Main Success Scenario	<table> <tr> <th><u>Actor Action</u></th><th><u>System Response</u></th></tr> <tr> <td>1. Employee goes to customer to take their order.</td><td></td></tr> <tr> <td>2. Customer names food item they want to order and employee enters each item identifiers <i>(Step 1 -2 are repeated until customer's order is complete)</i></td><td></td></tr> <tr> <td></td><td>3. Records each item and saves it in system</td></tr> <tr> <td></td><td>4. Displays the total order of the customer.</td></tr> <tr> <td>5. Employee repeats order to customer to confirm their order.</td><td></td></tr> <tr> <td>6. Customer confirms order.</td><td></td></tr> <tr> <td></td><td>7. System places order and sets order status to : Preparing</td></tr> <tr> <td>8. Food is prepared in the kitchen and served when ready.</td><td></td></tr> <tr> <td></td><td>9. Order status is changed to : Complete</td></tr> </table>	<u>Actor Action</u>	<u>System Response</u>	1. Employee goes to customer to take their order.		2. Customer names food item they want to order and employee enters each item identifiers <i>(Step 1 -2 are repeated until customer's order is complete)</i>			3. Records each item and saves it in system		4. Displays the total order of the customer.	5. Employee repeats order to customer to confirm their order.		6. Customer confirms order.			7. System places order and sets order status to : Preparing	8. Food is prepared in the kitchen and served when ready.			9. Order status is changed to : Complete
<u>Actor Action</u>	<u>System Response</u>																				
1. Employee goes to customer to take their order.																					
2. Customer names food item they want to order and employee enters each item identifiers <i>(Step 1 -2 are repeated until customer's order is complete)</i>																					
	3. Records each item and saves it in system																				
	4. Displays the total order of the customer.																				
5. Employee repeats order to customer to confirm their order.																					
6. Customer confirms order.																					
	7. System places order and sets order status to : Preparing																				
8. Food is prepared in the kitchen and served when ready.																					
	9. Order status is changed to : Complete																				
Extensions:	<p>*a. At any time, the system fails:</p> <ol style="list-style-type: none"> 1.Employee restarts the system, logs in and requests the system for the recovery of the previous state. 2.System reconstructs the previous state. <ol style="list-style-type: none"> 2a. Employee resumes operation and continues taking order from the customer. 2b. Previous state cannot be retrieved 																				

	<p>1.Employee creates new order and starts to take order of customer from the start.</p> <p>5a. Customer wants to change their order</p> <ol style="list-style-type: none"> 1.Employee asks customer what they want to add or remove from order. 2.Employee adds or removes the item from order and system updates order. 3.Employee repeats the customer's order to confirm once again.
Pre-Conditions	<ul style="list-style-type: none"> - Employee is identified and authenticated. - Customer has a reservation.
Post-Conditions	<ul style="list-style-type: none"> - Correct order is made by customer - Order is noted into the system and sent to kitchen to be prepared - Order is served to customer after being prepared.

<u>UC8</u>															
By:	Ahmad Maqbool – 20i-0702														
Use Case Name:	Update Menu														
Scope:	Restaurant level system														
Level:	User - Goal														
Primary Actor:	Manager														
Stake Holders and Interests:	<ul style="list-style-type: none"> - Manager: Wants to be able to add items to menu and remove items from menu in case items are finished in inventory. - Customer: Wants to view the menu to place their order. 														
Main Success Scenario	<table> <tr> <th><u>Actor Action</u></th><th><u>System Response</u></th></tr> <tr> <td>1. Manager Enters login details</td><td></td></tr> <tr> <td></td><td>2. Authenticates the manager</td></tr> <tr> <td>3. Manager enters the name of the item he wants to update in the menu.</td><td></td></tr> <tr> <td></td><td>4. System provides relevant description of the item.</td></tr> <tr> <td>5. Admin adds or removes item from the menu.</td><td></td></tr> <tr> <td></td><td>6. System Updates the menu.</td></tr> </table>	<u>Actor Action</u>	<u>System Response</u>	1. Manager Enters login details			2. Authenticates the manager	3. Manager enters the name of the item he wants to update in the menu.			4. System provides relevant description of the item.	5. Admin adds or removes item from the menu.			6. System Updates the menu.
<u>Actor Action</u>	<u>System Response</u>														
1. Manager Enters login details															
	2. Authenticates the manager														
3. Manager enters the name of the item he wants to update in the menu.															
	4. System provides relevant description of the item.														
5. Admin adds or removes item from the menu.															
	6. System Updates the menu.														

	7- Manager verifies the update.	
Extensions	<p>*a. At any time, the system fails:</p> <ol style="list-style-type: none"> 1.Manager restarts the system, logs in and requests the system for the recovery of the previous state. 2.System reconstructs the previous state. 3.Manager continues performing updates in menu. <p>*b. Manager removes an item from menu</p> <ol style="list-style-type: none"> 1.Manager searches for item. 2.If item does not exist in menu, <ol style="list-style-type: none"> 1. Manager is notified and taken back to home page. 3.Else if found, they are shown the details of the item. <ol style="list-style-type: none"> 1. System asks manager to confirm delete of item. 2. Manager confirms deletion of item and item is deleted from menu in system <p>*c. Manager wants to list new item to menu:</p> <ol style="list-style-type: none"> 1.Manager creates a new item in menu. 2.Manager adds all details of the new item into the system. 3.System updates menu and stores new item. <p>1a. Manager enters wrong credentials</p> <ol style="list-style-type: none"> 1.Message is displayed and user is taken back to login page to reenter login details. <p>3a. Menu has no record of the item.</p> <ol style="list-style-type: none"> 1. In this case, user creates new item and adds its details into the system. <p>5a. Manager adds an item that is already present in menu.</p> <ol style="list-style-type: none"> 1. Manager is notified that item is already present in menu and cannot be placed again. 2. Manager is taken back to home page. 	
Pre-Conditions	- Manager is identified and authenticated	
Post-Conditions	- Menu is updated and only shows items available in restaurant.	

2.5 Use Case Diagram



3. Other Nonfunctional Requirements

3.1 Performance Requirements

Since the data is being saved into a database, the storage capacity can be increased and set to as much as the user requires. Execution time and speed of response is approximately null as data is received and inserted into the database at a very quick speed.

3.2 Safety Requirements

The database in which the data is being saved should not be deleted or else all data will be lost.

3.3 Security Requirements

No outsider except the staff of the restaurant should have access to the system or to the passwords of the system.

3.4 Software Quality Attributes

Easy to use user interface which can even be used by untrained and unskilled employees. There are prompts on every page of the User interface which guides the user and makes it very easy for the user to work with the application.

There is maintainability as adding a new item into the UI easily adds the item into the database and it is also a reliable system as the data is mapped correctly onto the database.

3.5 Business Rules

The managers can manage and update employee records and schedules. They can also update menu and generate daily and monthly reports. The employees can generate order, generate payment and schedule reservations for customers.

3.6 Operating Environment

The software will operate on either a computer or a laptop which has a windows system. It should have mysql workbench and scenebuilder along with java, javafx and javafx converters.

3.7 User Interfaces

Error message is prompted when user forgets to fill information into a textfield. Back button on every page takes the user back to the last page that he was on. There are create buttons and view buttons to create objects and data into the database and then view them respectively.

Login


—□×

Close


Welcome Back

Please enter your details

Invalid details!



Username



Password


☐

Manager

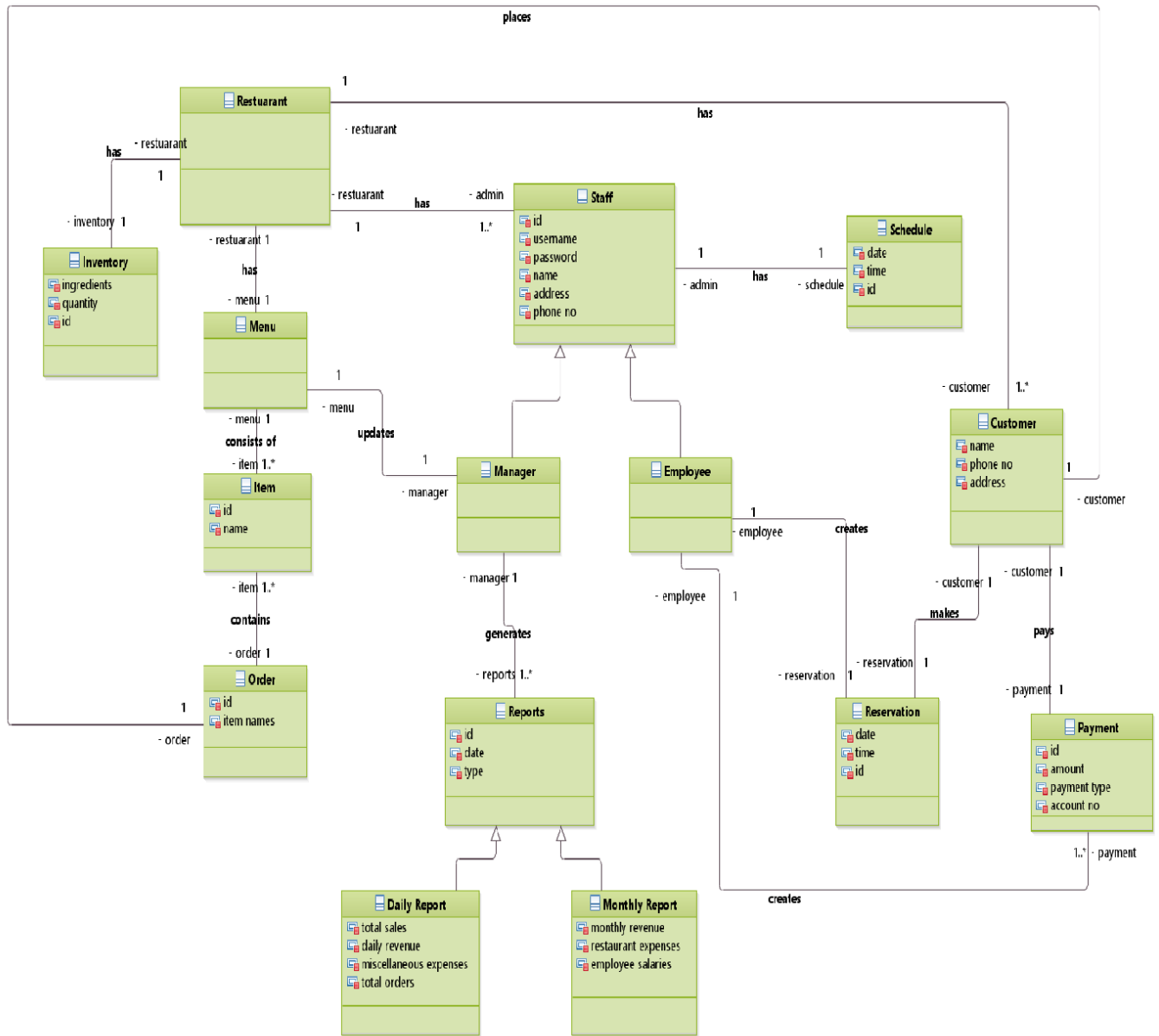
☐

Employee

Login


Vision8

4 Domain Model

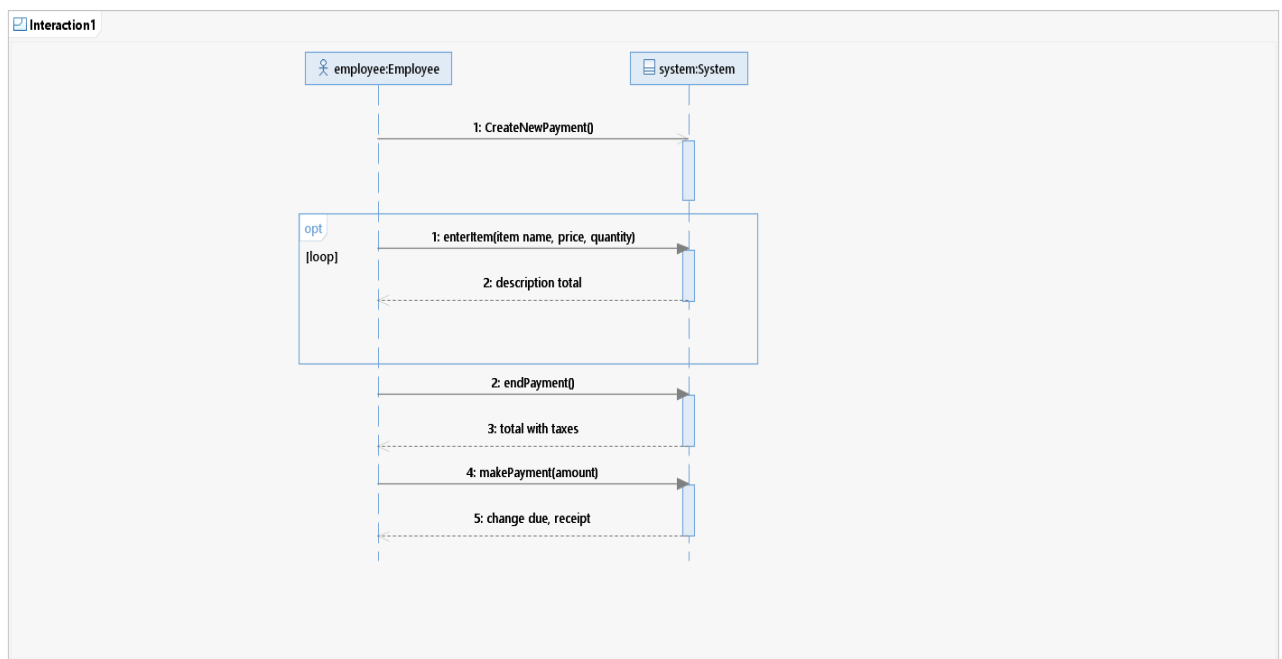


5 System Sequence Diagram

1. Manage Employees



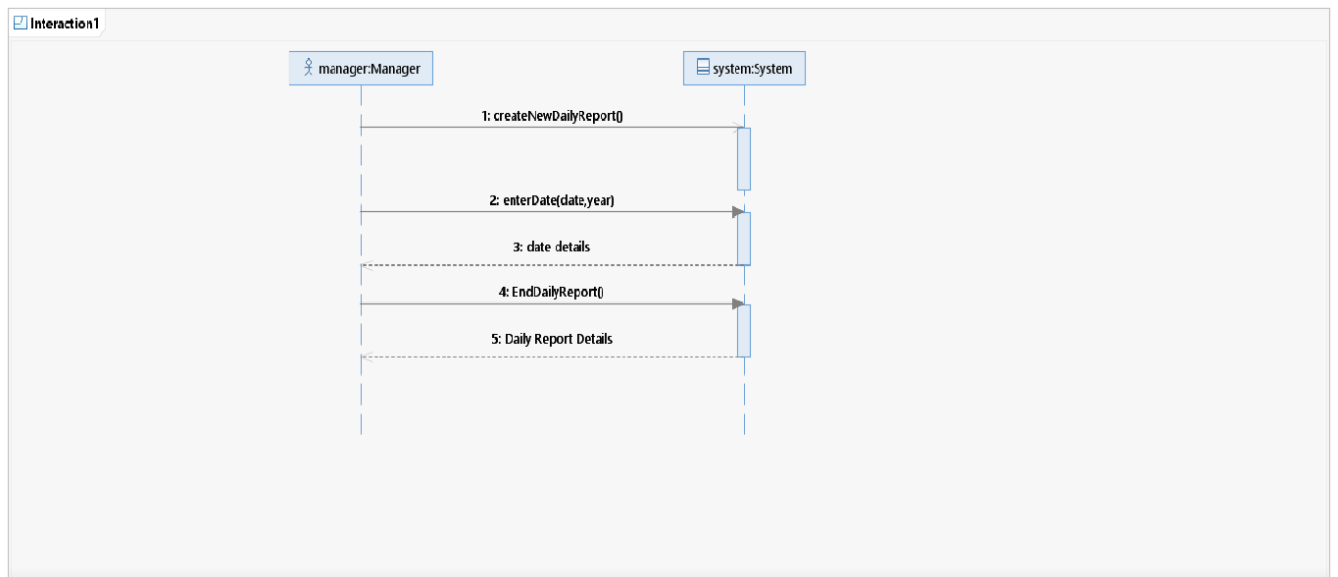
1. Generate Payment



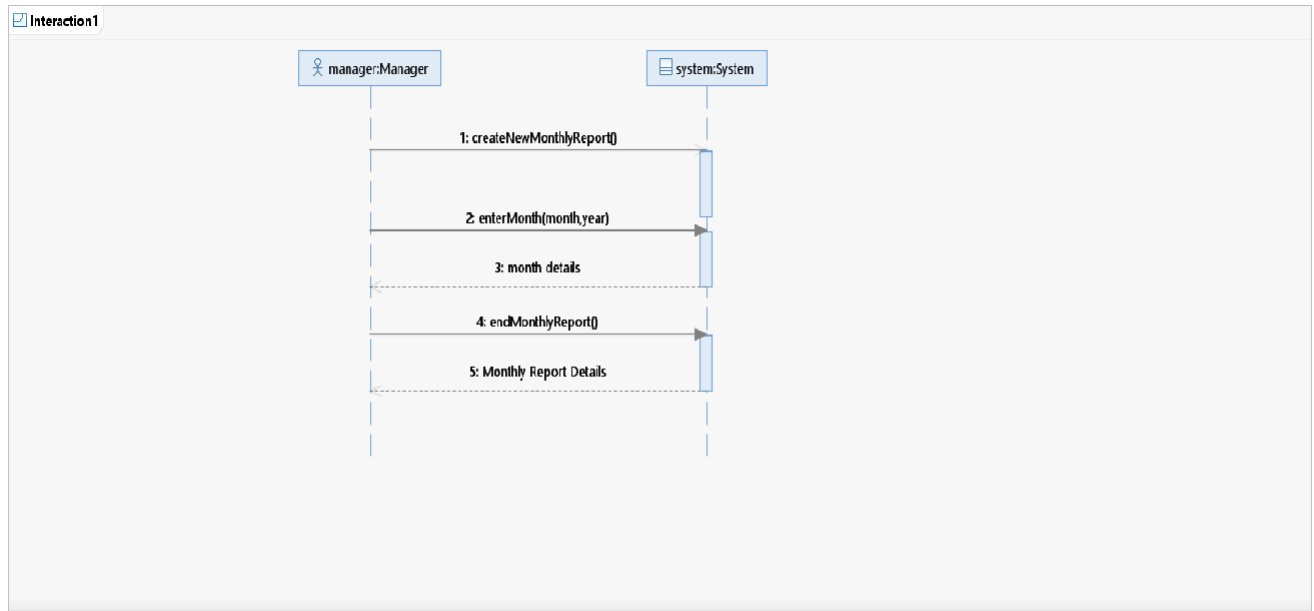
2. Schedule Reservations



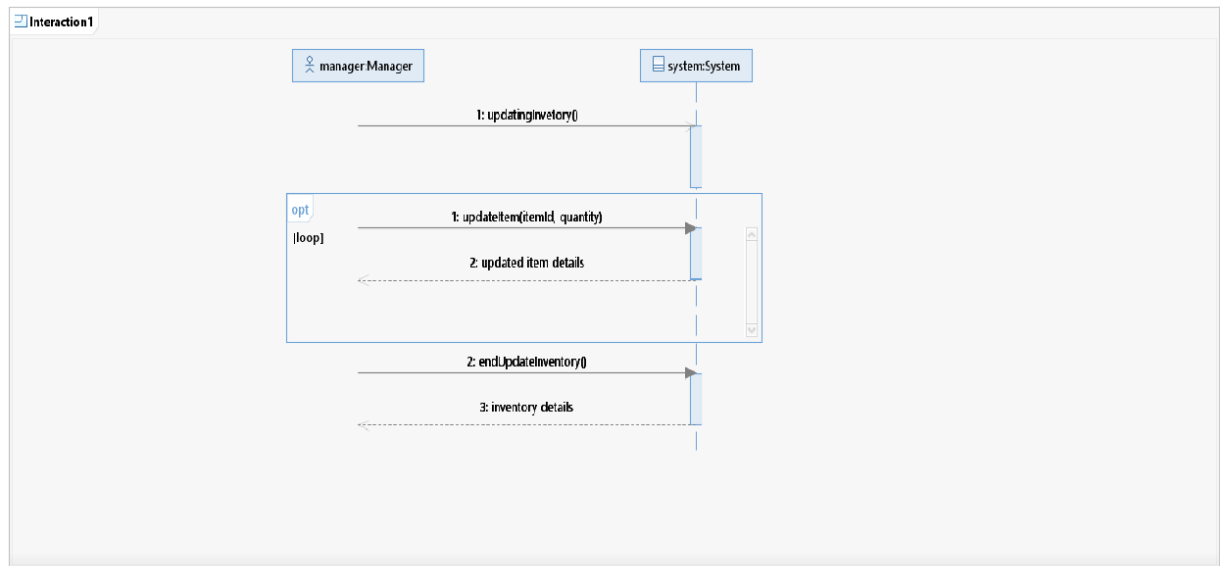
3. Generate Daily Reports



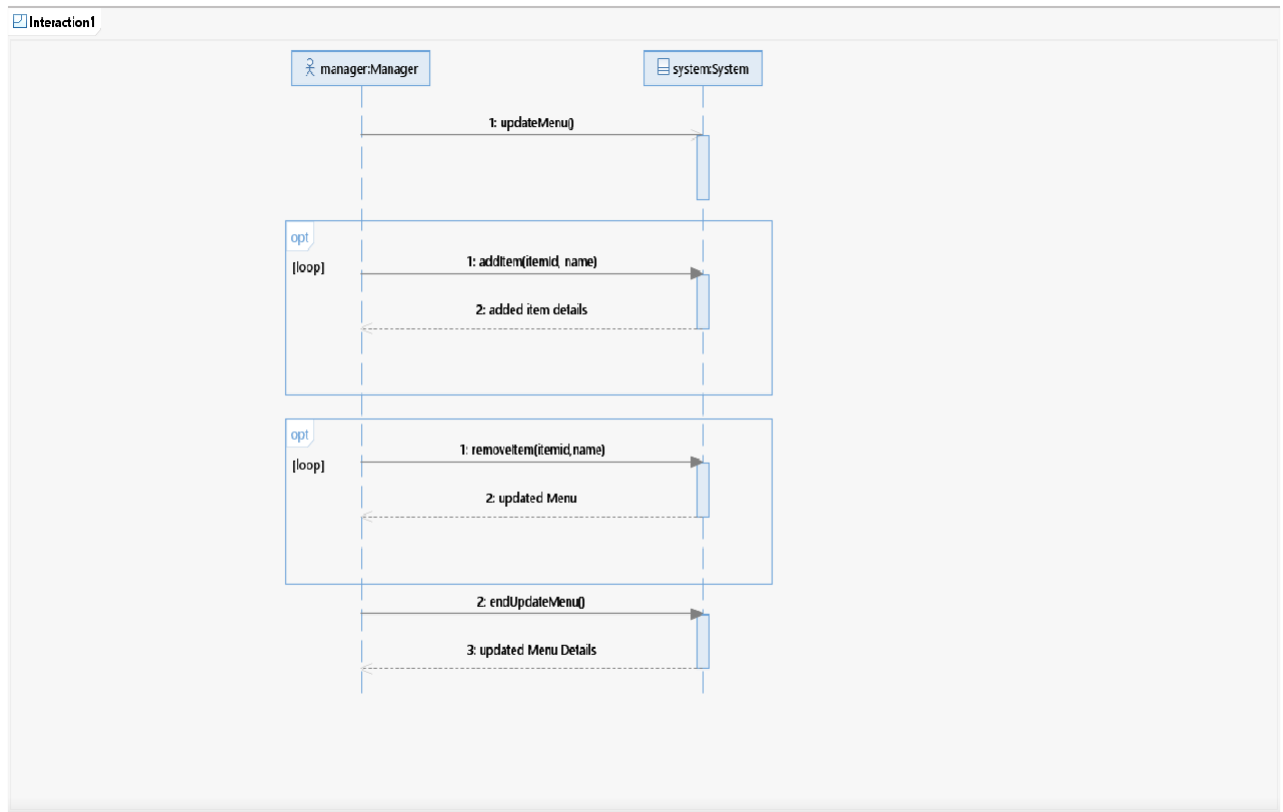
4. Generate Monthly Reports



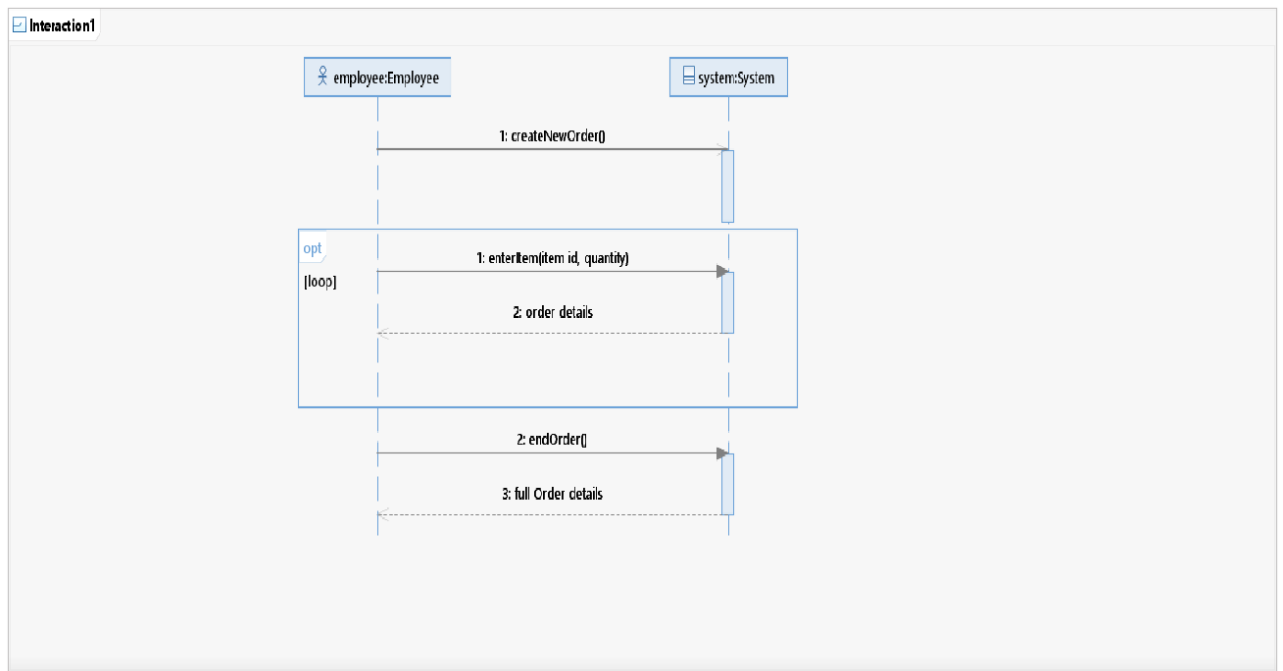
5. Update Inventory



6. Update Menu

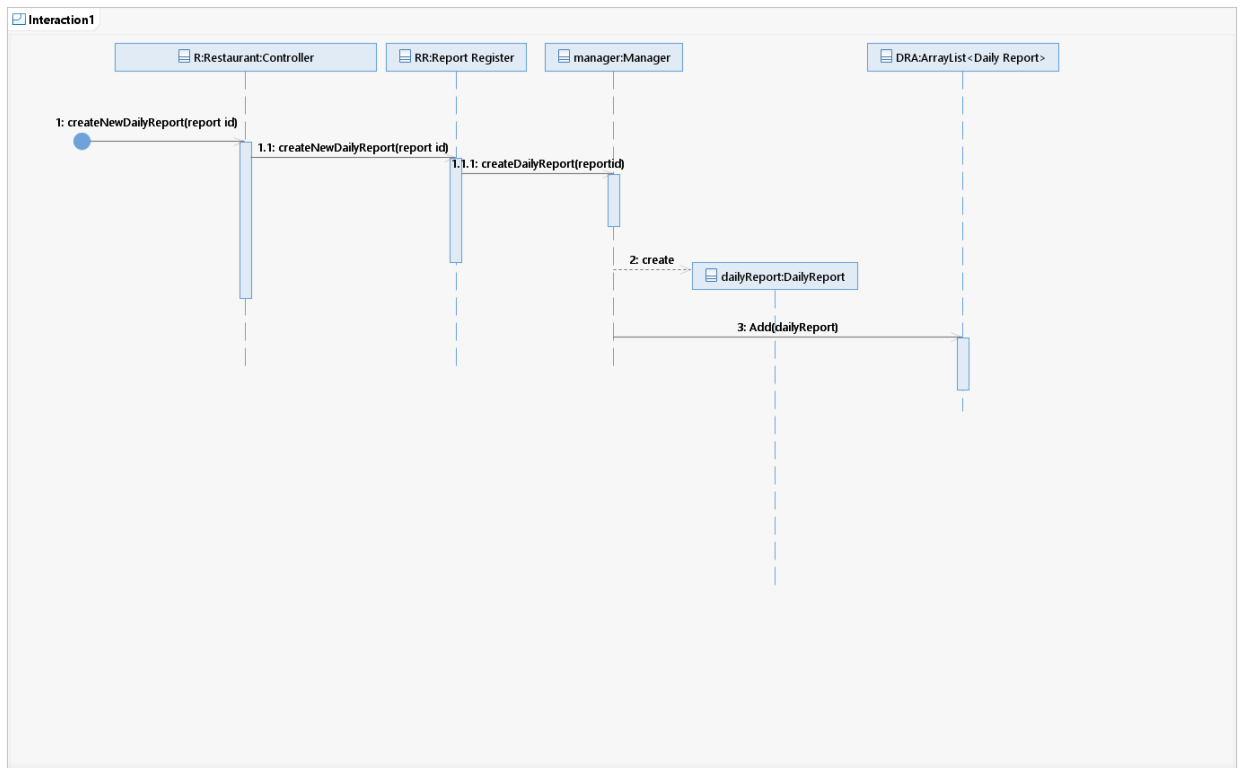


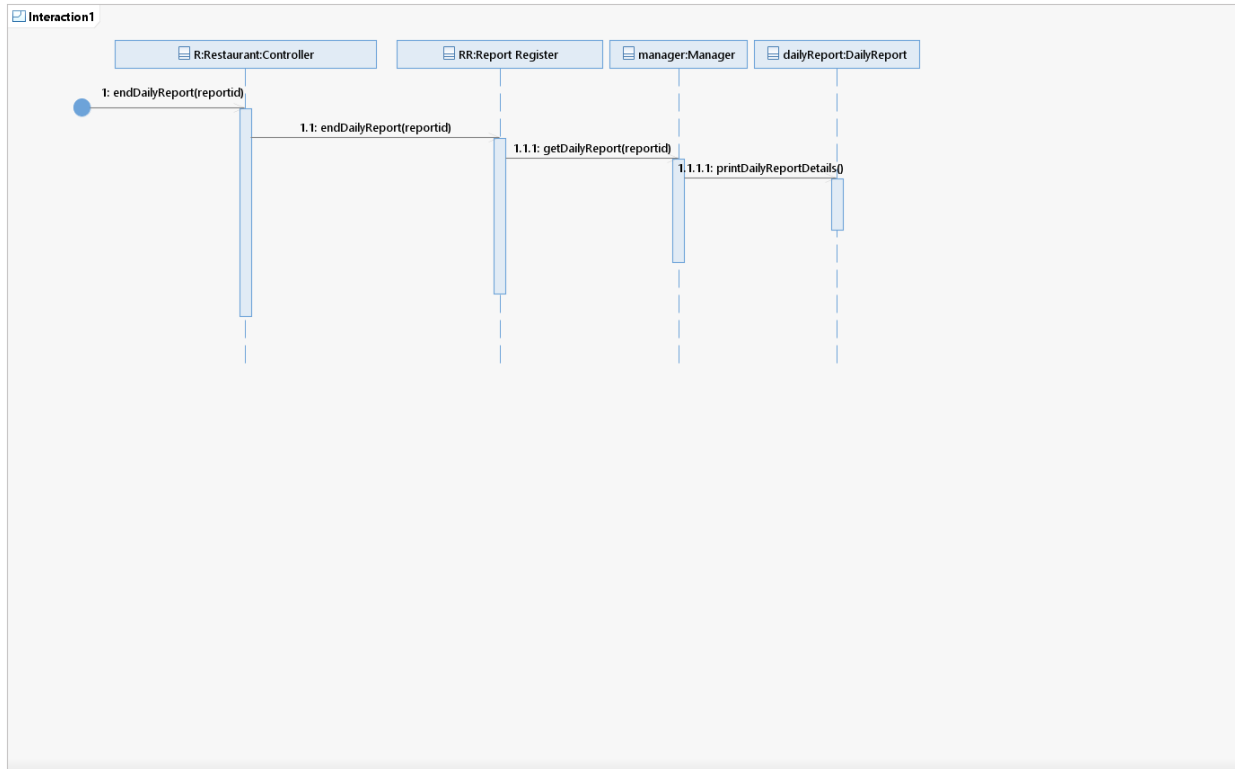
7. Generate Order



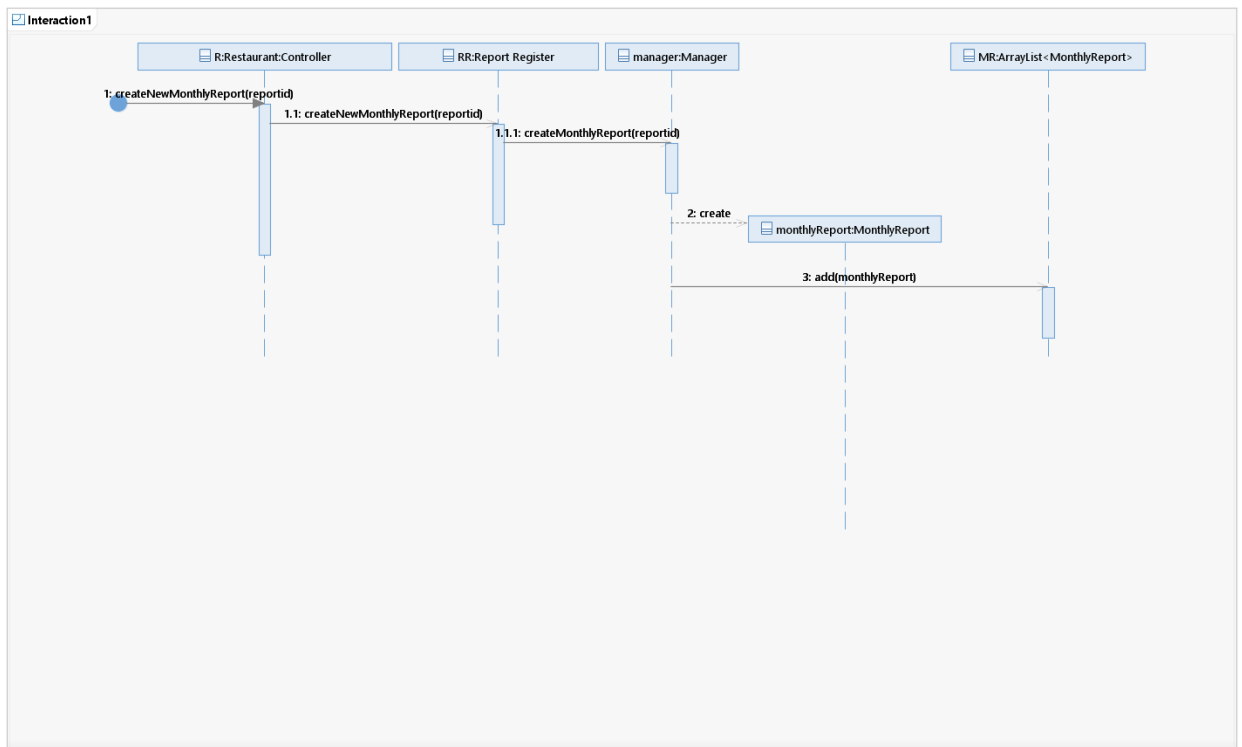
6 System Sequence Diagram

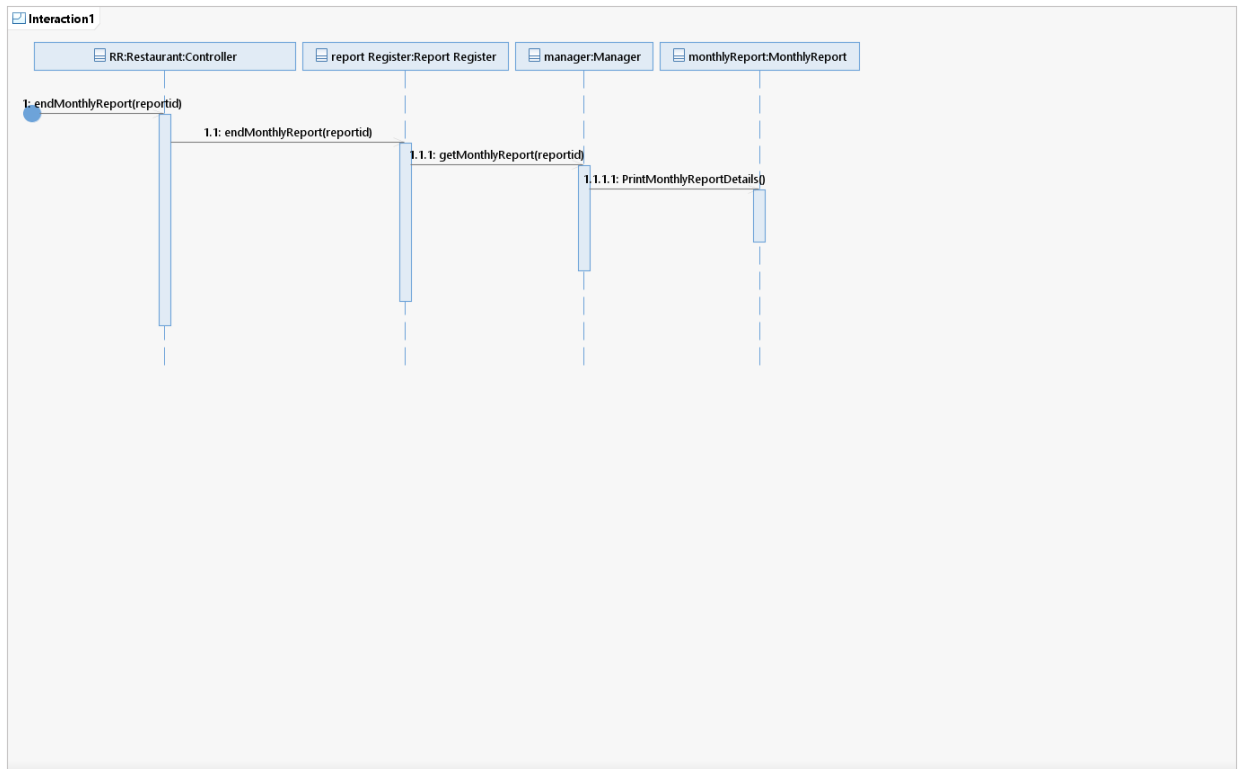
1.Generate Daily Reports – SDs



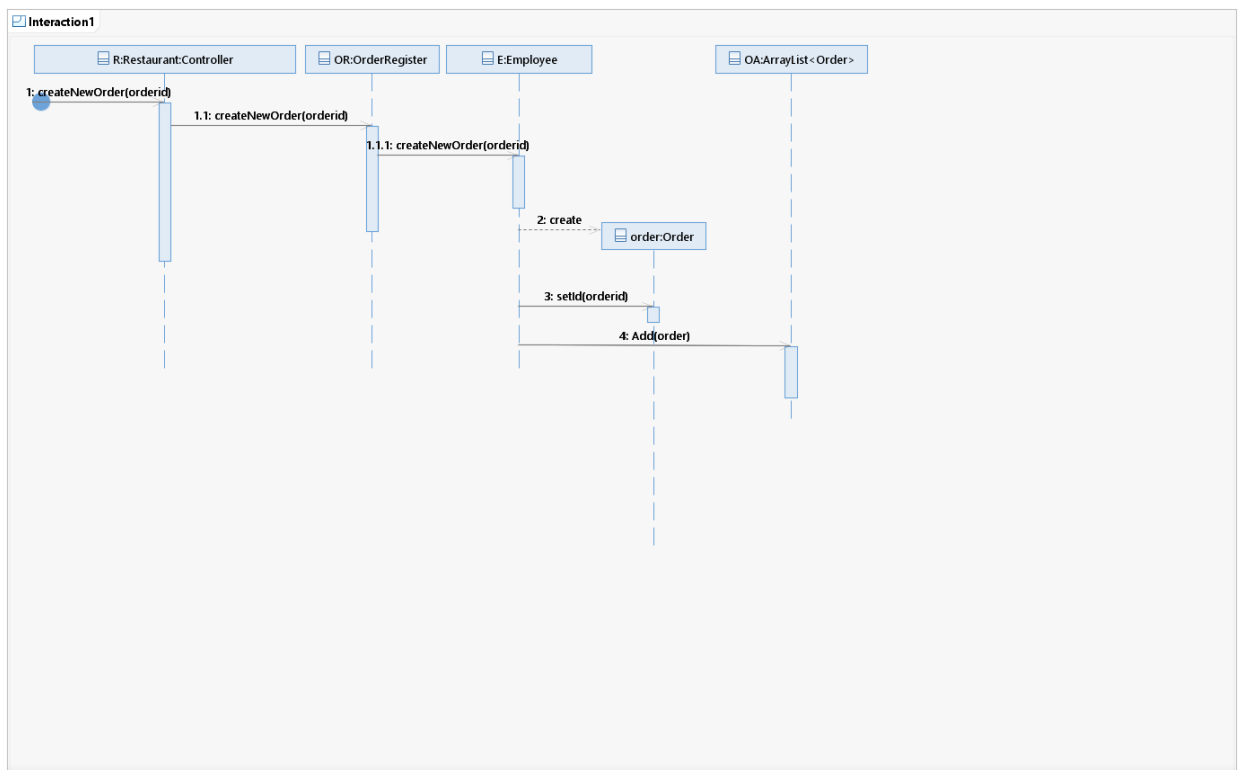


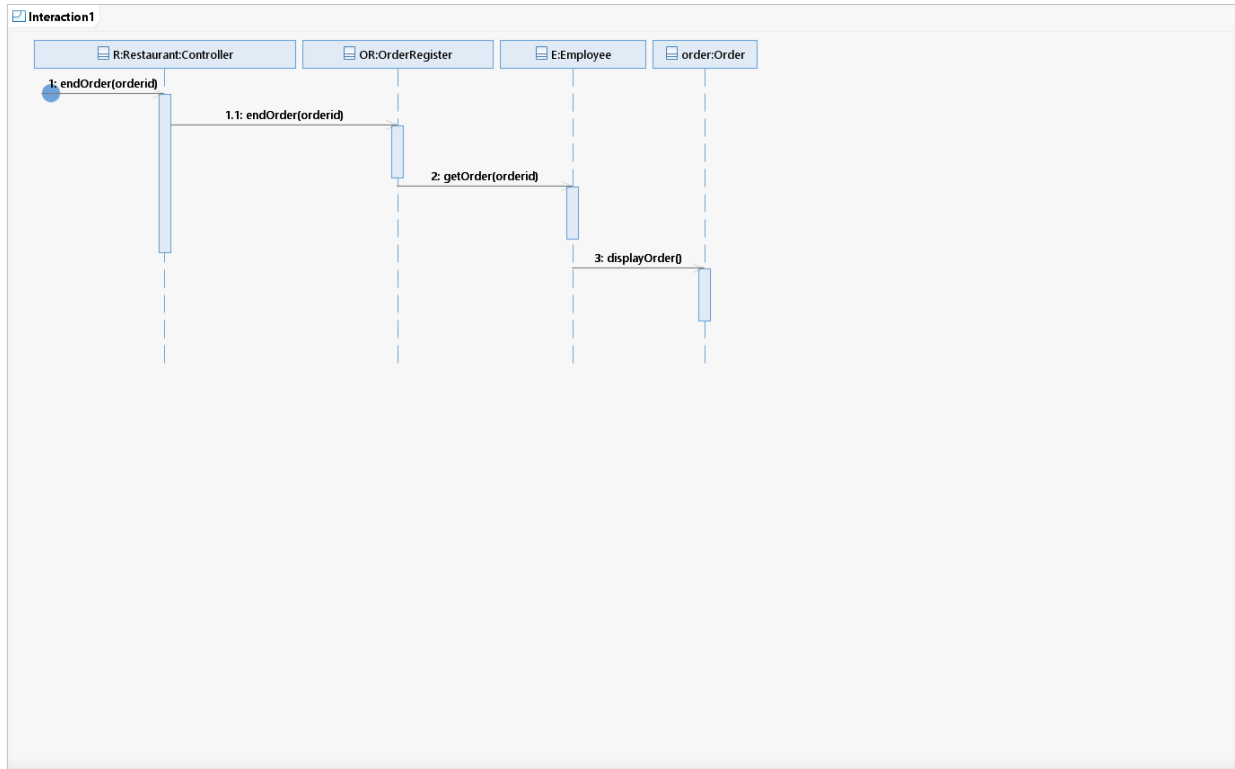
1. Generate Monthly Report – SDs



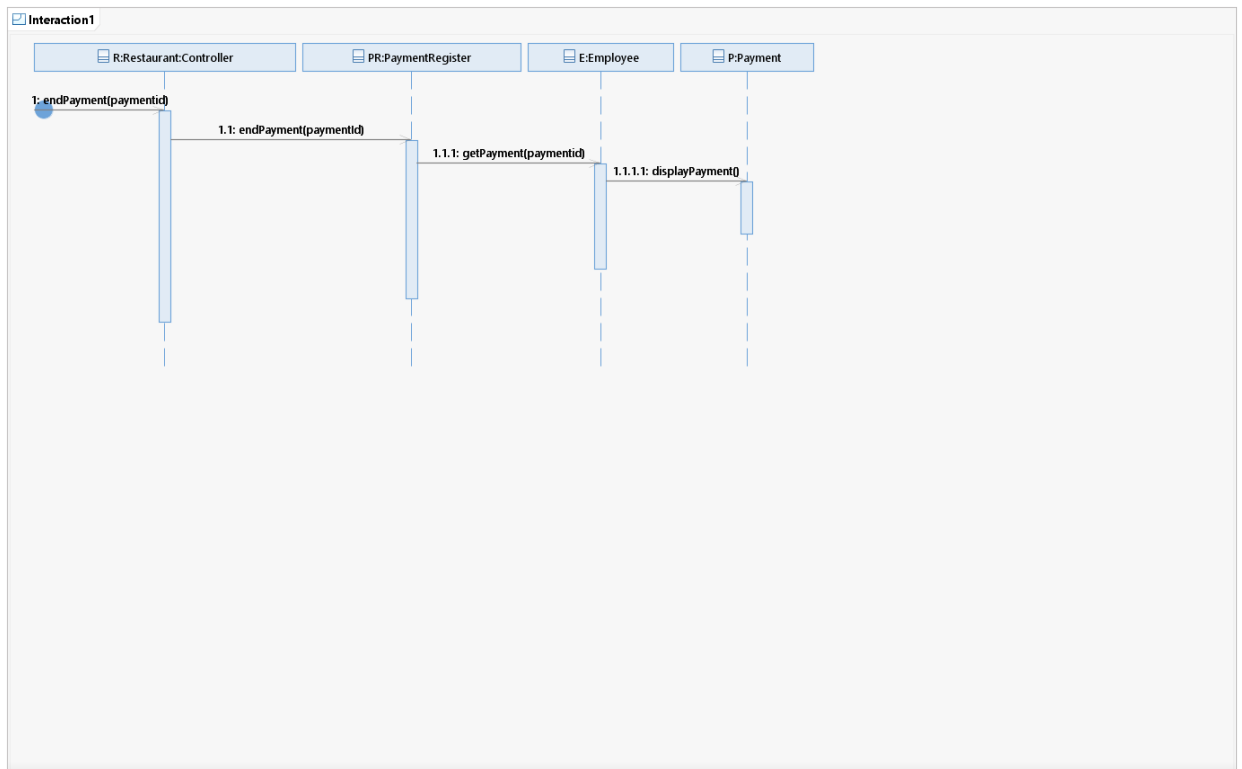


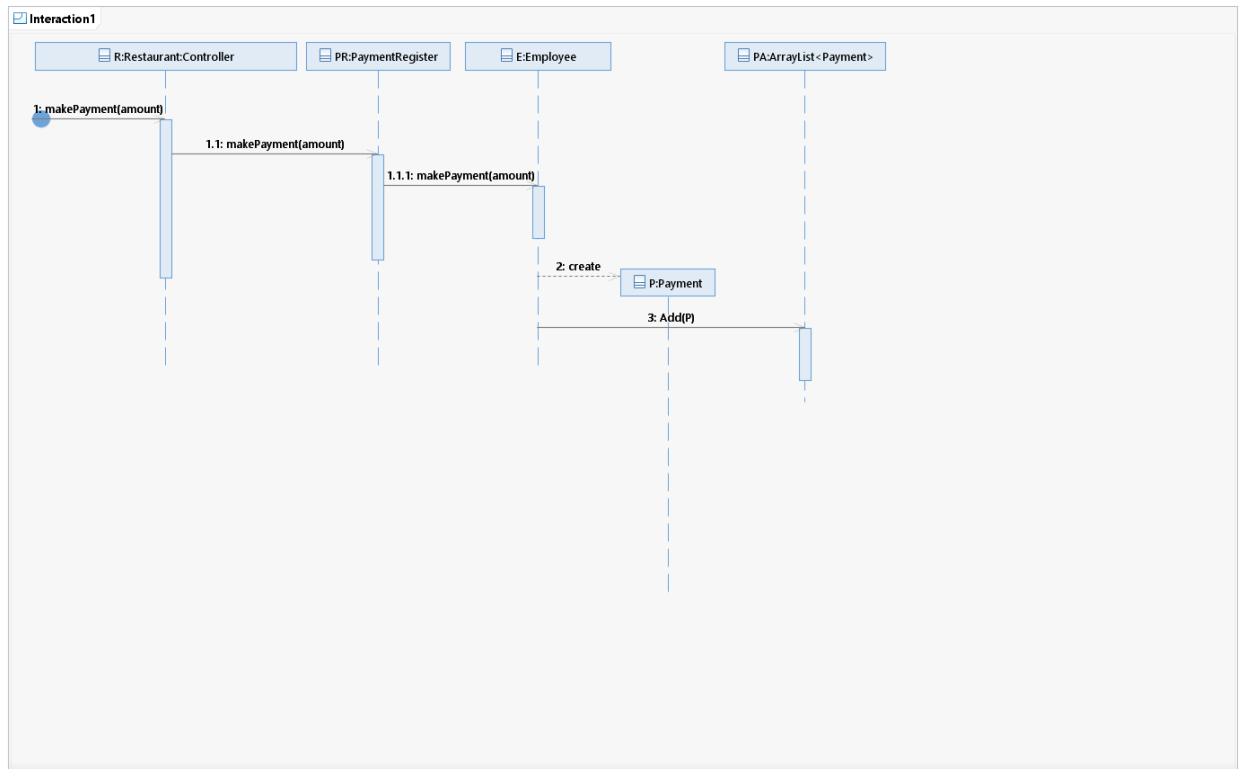
2. Generate Order – SDs



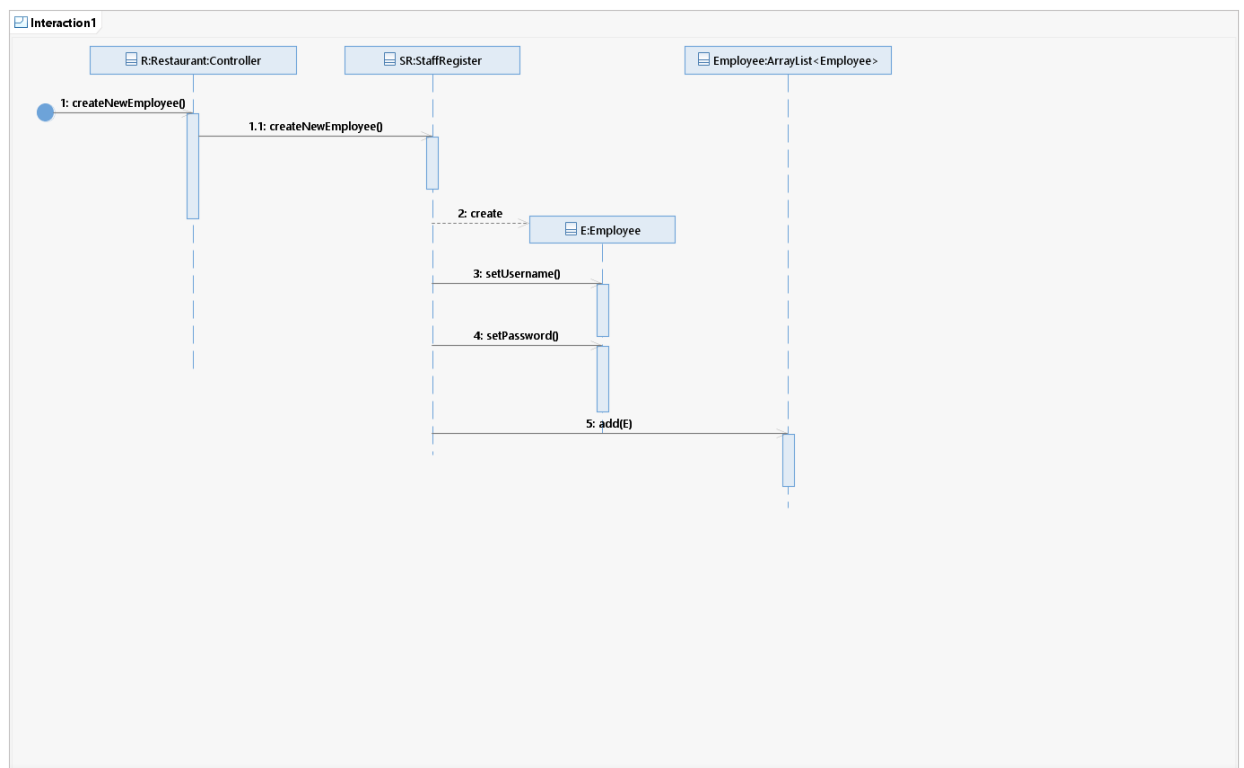


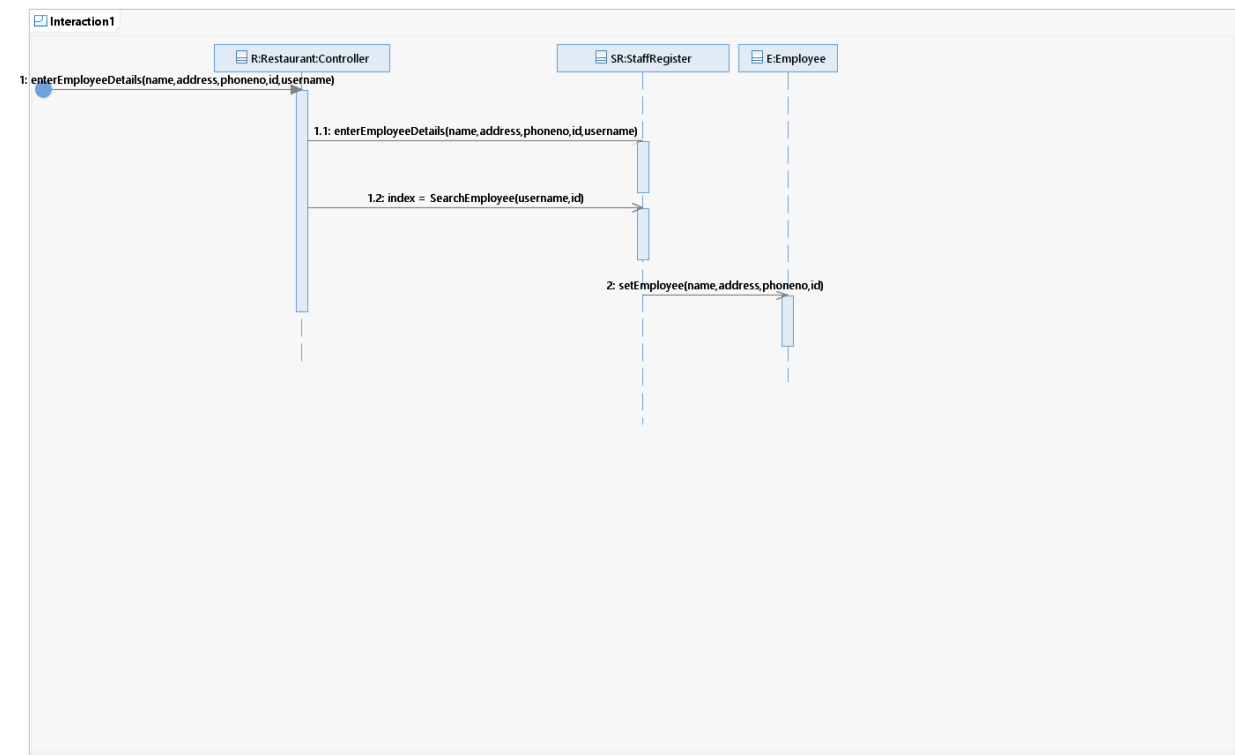
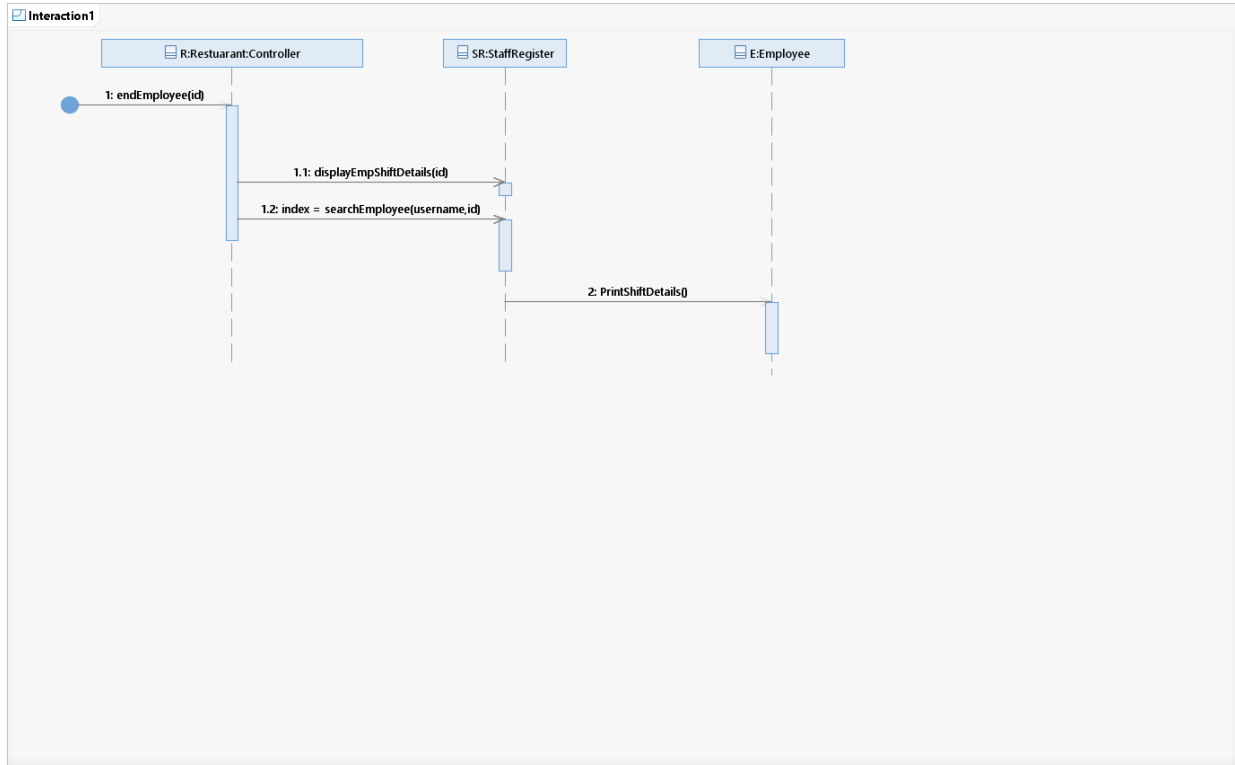
3. Generate Payment -SDs

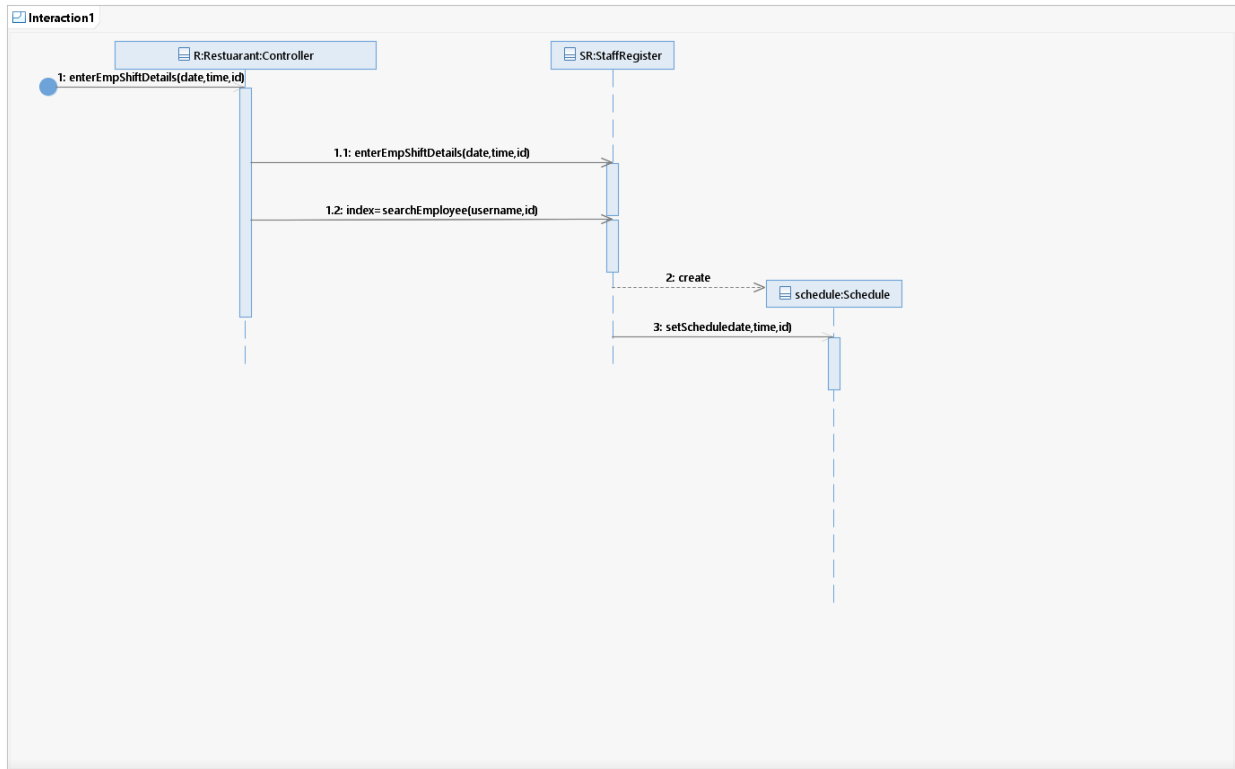




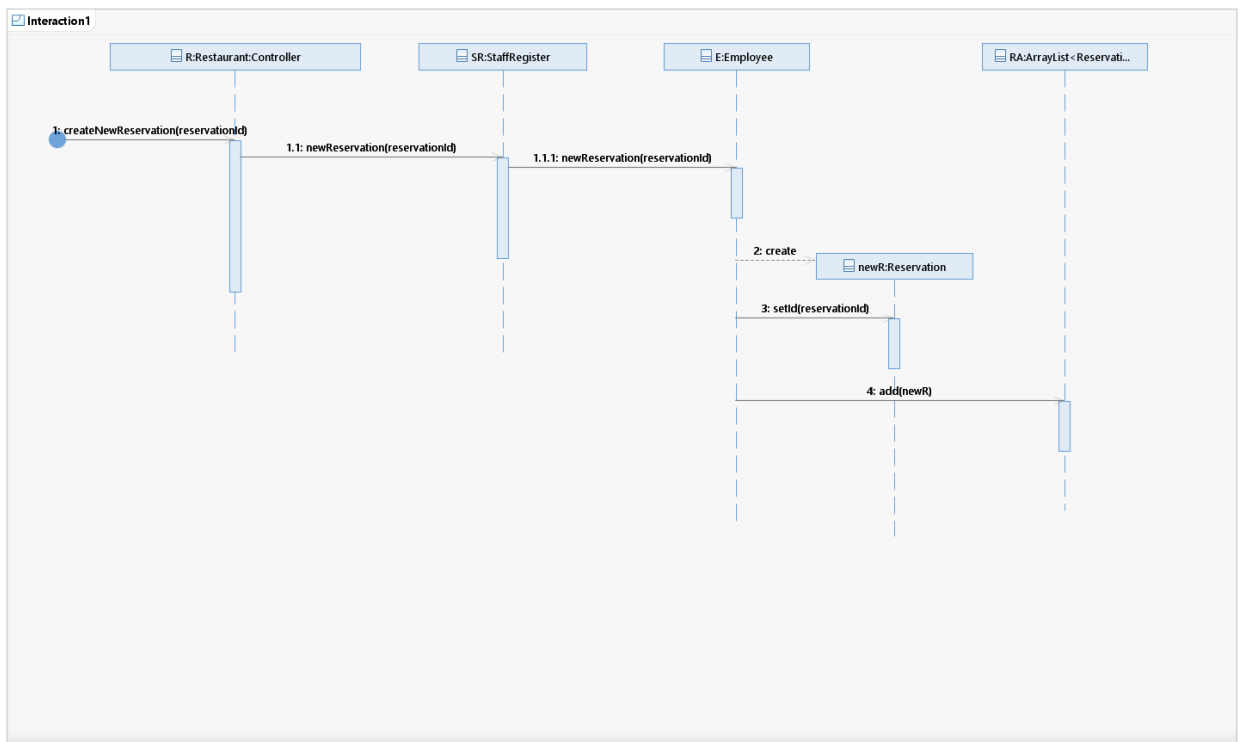
4. Manage Employee-SDs

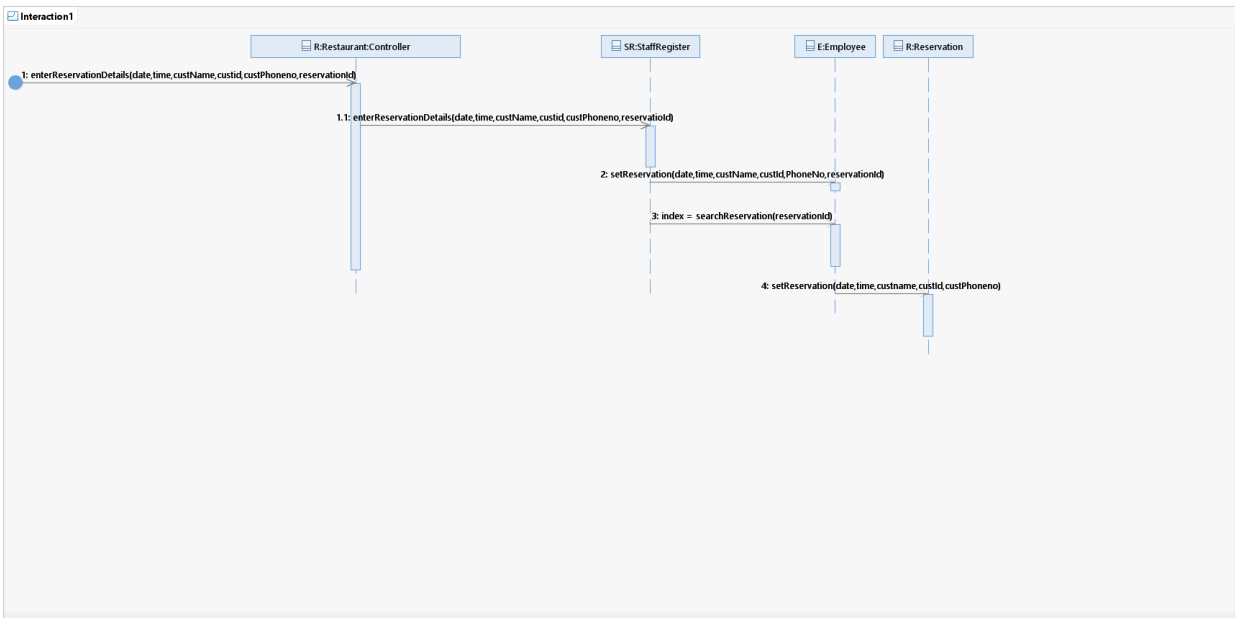
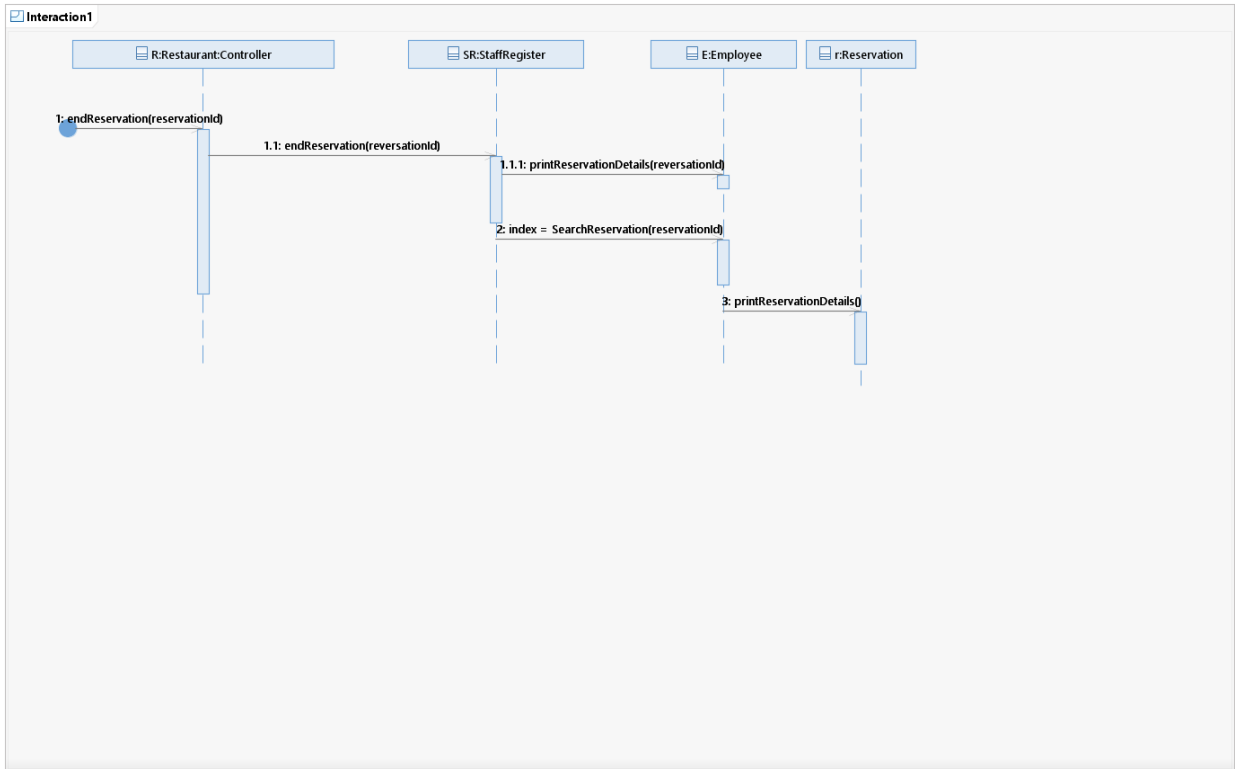




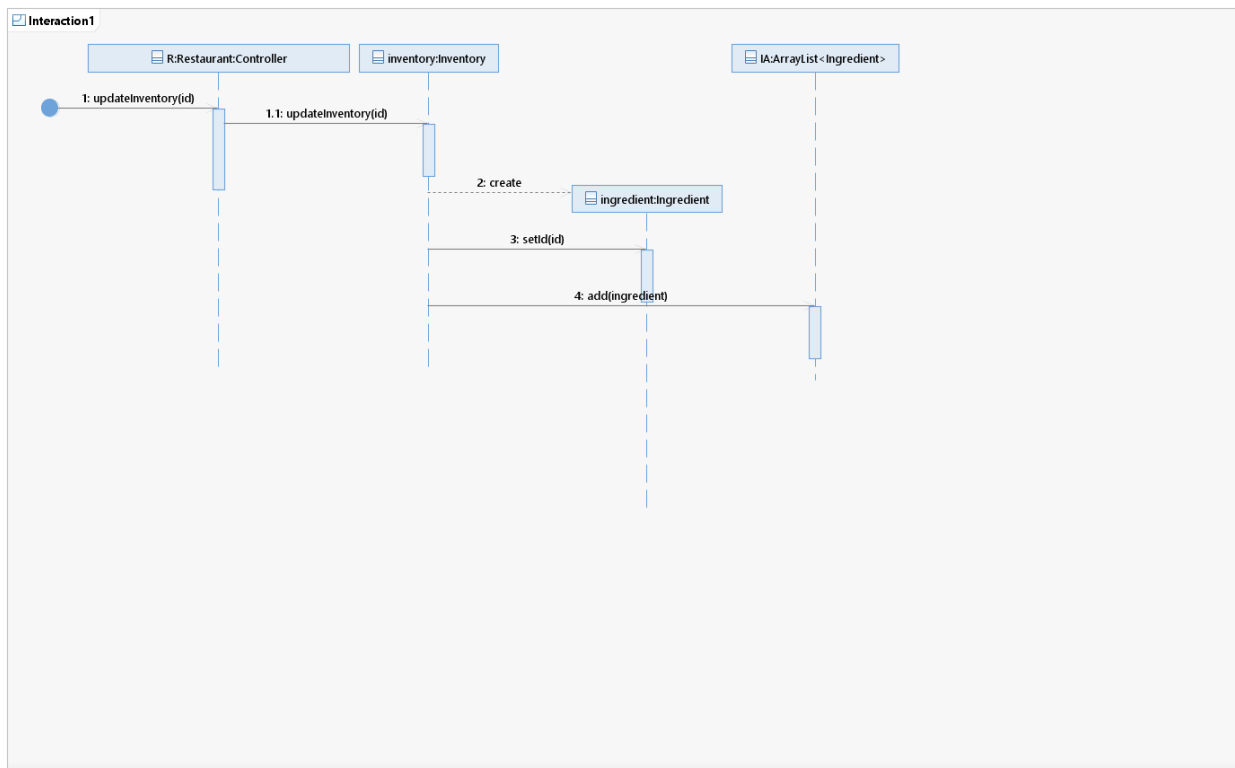
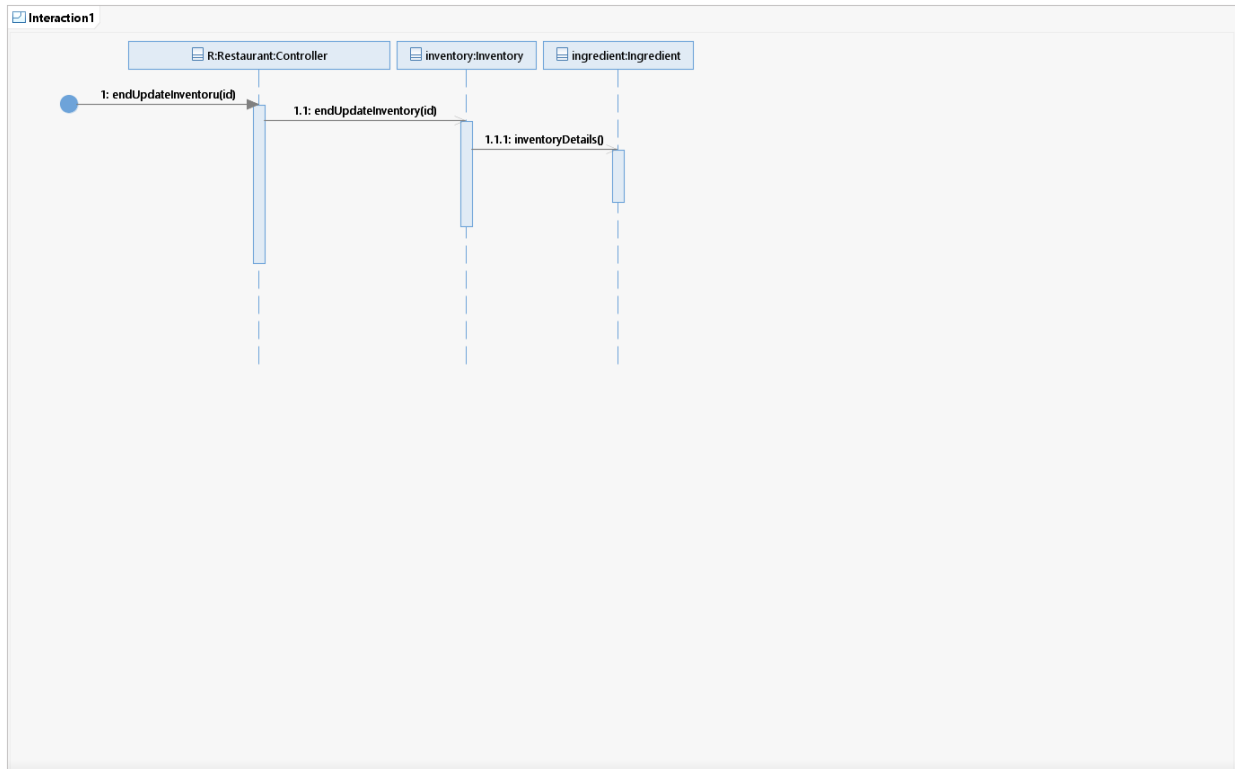


5. Schedule Reservation – SDs

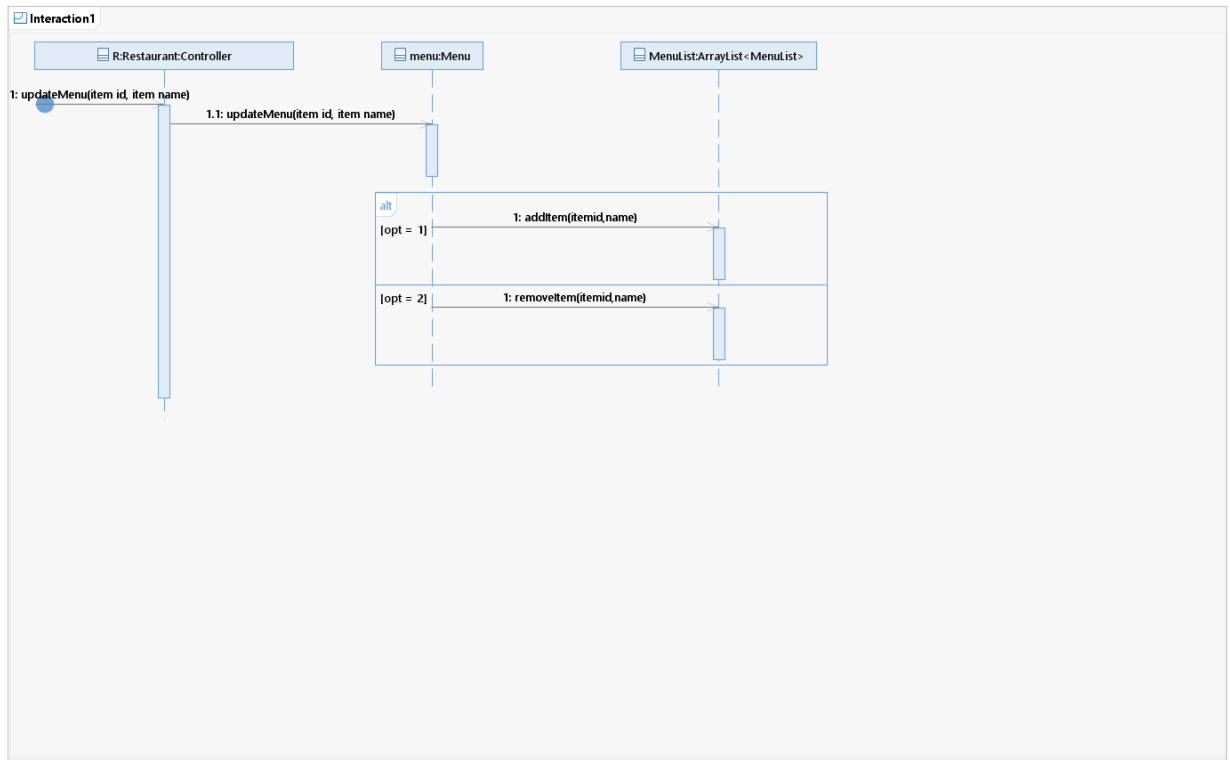
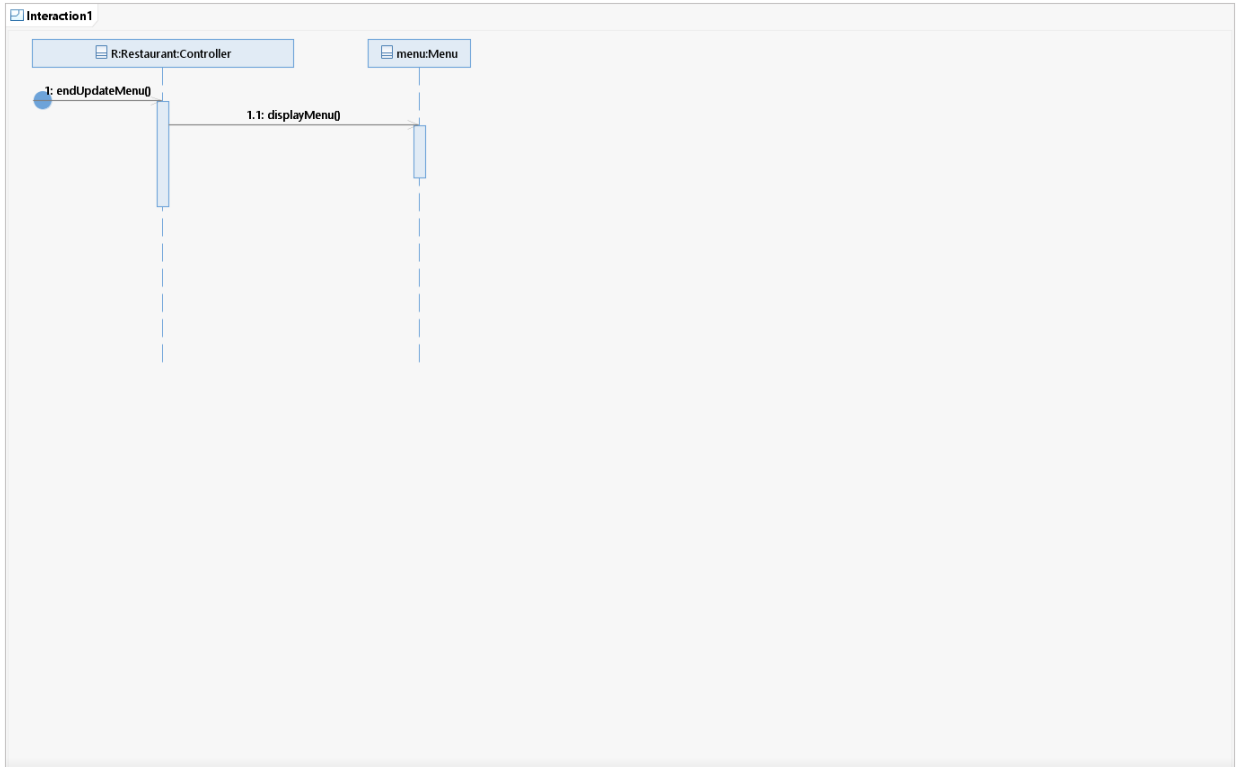




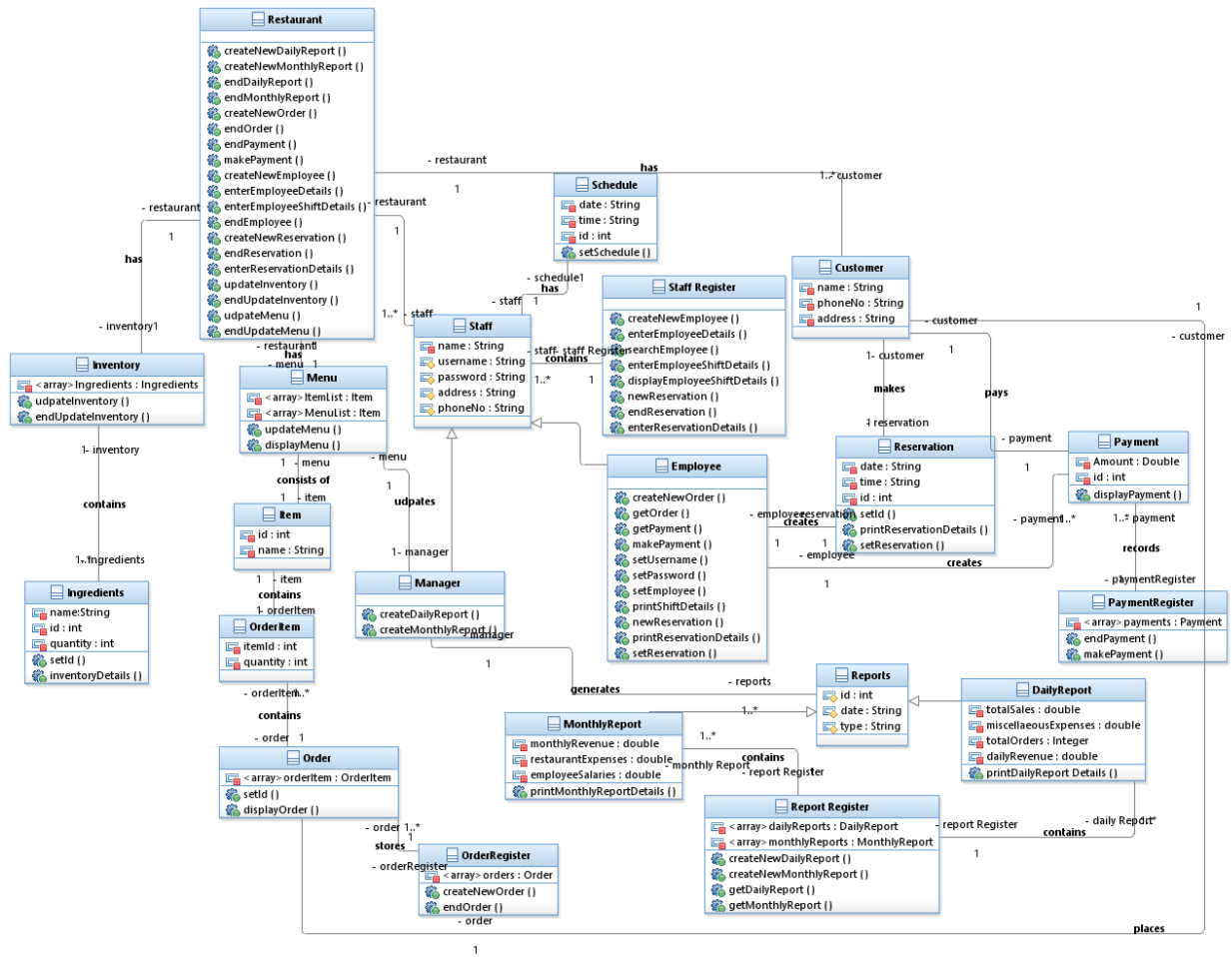
6. Update Inventory – SDs



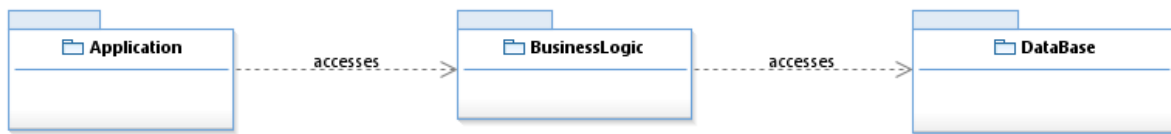
7. Update Menu – SDs



7 Class Diagram



8 Package Diagram



9. Deployment Diagram

