SECP3204: Software Engineering WBL

System Requirements Specification (SRS)

VENDING MACHINE INVENTORY MANAGEMENT SYSTEM

Version 1

22 May 2022

School of Computing, Faculty of Engineering

Prepared by: Garrison Tech

Table of Contents

1. Introduction

1.1 **Purpose**

The system documentation describes the system's expected behavior and functionality, with the goal of assisting the client's business logic as defined through various types of diagrams. Our client, MFA Solutions is a vending machine company actively operating in Johor, Malaysia. We were asked to develop a stock inventory system which will assist our client in managing the stock inventory and to replace their old manual system which was not very efficient and effective. As a result, we expect that by developing this SD, we will be able to present our client, MFA Solutions, with more precise plans for the development of the vending machine inventory management system.

1.2 **Scope**

Our group has decided on creating a vending machine inventory management system. There will be two main users in this system: the employer and the employee. There are also several modules in this system which are manage users, manage stocks, manage vending machine sales, manage reports and dashboard module. The employer has full access to all modules, whereas the employee only has access to a few, such as managing stocks and vending machine sales. Vending machines, stocks, staff, and sales charts will all be displayed on the dashboard. This will provide the employer with a general summary of his business. This system's primary goal is to allow users to execute CRUD operations on stock and vending machine sales. Reports are also an important aspect of this system, but they can only be generated by the employer because they contain confidential business information. On a daily, monthly, or annual basis, reports can be generated. In addition, if necessary, the business can add new employees and manage their information. For user management, the employer may also do basic CRUD operations. We feel that this system will help our client's business run more efficiently and effectively. Not only that, but this system will assist in reducing any human errors that existed in the previous system.

1.3 **Definitions, Acronyms and Abbreviation**

Abbreviation/Acronyms	Definition
CRUD	Create,Read,Update,Delete
DB	Database
TXN	Transaction

1.4 **References**

1. (n.d.). Vensoft - A World Class IT Solutions and Services Provider. Retrieved June 19, 2022, from https://www.vensoft.com/

2. "IEEE Recommended Practice for Software Requirements Specifications," in IEEE Std 830-1998 , vol., no., pp.1-40, 20 Oct. 1998, doi: 10.1109/IEEESTD.1998.88286.

1.5 **Overview**

Using diagrams such as the Entity Relationship Diagram, Use Case Diagram, State Diagram, and Activity Diagram, the System Documentation continues to define the system's features, functionality, and intended behavior for each user. Both Employer and Employee will be explained about User Interfaces, Hardware Interfaces, Software Interfaces, and Communication Interfaces. All type of users will explain about External Interfaces Requirements.

The document also includes a detailed explanation for our use cases. Our system consists of Manage Stock , Manage Stock History, Manage Employee and Manage Report. Each of these use cases contain its own extensions and inclusions.

The document also describes the System Features, Performance and Design Constraints. Any limits set by the customer where our system will be used must adhere to a specified standard as well as other non-functional criteria, as detailed in the Design Constraints section while, Software System Attributes will ensure ease of use and attractiveness of our system to the users.

2. Specific Requirements

2.1 External Interface Requirements

2.1.1 User Interfaces

The web-based system will create a responsive user interface that provides a nice user experience on both PC displays and mobile devices. Despite interacting differently, the user interface should allow both PC and mobile users to accomplish all actions on the website (PC users interact using peripherals such as Mouse and Keyboard, while Mobile users interact via tapping through a touch-screen).

By using common design patterns that provide efficient communication that follows common sense, the user interface should be self-explanatory and intuitive for all sorts of users (Admin and Family users). To put it another way, the user interface should allow users to browse and operate the system without the need for a manual or guide.

Admin, on the other hand, has more elaborated use cases, hence a thorough manual with instructions and explanations is provided under the section Ease of Use in order to properly manage the system.

Login Interface

All the employees and employers can login to the vending machine inventory management system by entering the correct password and username. An error message will be displayed which implies the wrong password or username entered and the user will need to enter again.

Dashboard Interface

Dashboard is the first landing page for the employers after they have logged in successfully to the system.Dashboards contain various graphs as well as widgets to gain some fast overall insight like employers can view the top 5 best sellers for the drinks in each of the vending machines. The employer will have an overall insight of his business.

Read Interface

- Manage Stock Module

In the stock managing module interface, the system will display a list of details ranging from stock name, stock quantity, stock cost price as well as the stock selling price. All system users can access this interface to view the list of details being displayed. In this interface,it will be displaying the edit and delete stock button as well.Employers can also be searching the stock by entering the name of the stock in the search bar and the system will display accordingly.

- Manage User Module

All users including employee and employer of this inventory system can view their own profile personal details but one thing different is the employers can view all the users personal details but not the password.

- Manage Vending Machine Sales Module

All users are allowed to read the details showing in the vending machine sales module like stock name,quantity of the stock sold ,cumulative of the quantity sold .They can also view the net profit per each can of drink they sold,total net profit for the respective stock and the percentage of net profit. Users can also search to only display certain vending machine sales details in case there might be more than 1 vending machine for not causing any confusion.

- Manage Report Module

Employer is the only user that can get access to this module interface.The employer can view the report once it has been generated.

Create Interface

- Manage Stock Module

All users can create new stocks to the inventory system by first landing on the view stock interface and selecting the add button. The users will need to enter some of the details like stock name,stock quantity, stock cost and the stock selling price and click on the create button. A message will be displayed once the stock is created successfully to the system.

- Manage Employee Module

Employers of this inventory system can create new users by entering the personal details like name,address, email and a default password for the users first.After done filling the details, the employer only needs to click the register button and a message will be displayed which indicates the successful registration process.

- Manage Vending Machine Sales Module

All users can create a new sale product in any of the vending machines by first landing on the view interface and selecting the add button. The users will need to enter some of the details like product name,product quantity, product cost and the selling price and click on the create button. A message will be displayed once the item is created in the vending machine successfully to the system.

- Manage Report Module

Employer is the only user that can get access to this module interface.The employer can download the report generated in the csv,excel as well as pdf format. Moreover , employers can also print the report generated for any additional usage.

Update Interface

- Manage Stock Module

All the users are allowed to update the stocks in terms of the stock name,cost price,quantity or even selling price. Most of the time,when certain drinks get restocked,they will need to update the stock quantity or adjust the selling price due to inflation. After updating the stocks, users need to click the submit button and a message will be displayed.

- Manage Employee Module

Employees can only update their personal details including password,address or email whereas the employers get access to edit the profiles of all users except for the password. “Submit” button will be provided for employers and employees to save the updated information. A message will be displayed if the personal details of the user are updated successfully.

- Manage Vending Machine Sales Module

All the users are allowed to update the products in the vending machine in terms of the name, price, quantity or even selling price. Most of the time,when certain drinks get sold or added in there will be a need for an update of the quantity or adjust the selling price due to inflation. After updating the products, users need to click the submit button and a message will be displayed.

Delete Interface

- Manage Stock Module

All users are eligible to delete the unwanted stocks in the system. Users will need to click on the confirm button when the delete message pops up. A message will be displayed if the particular selected stock is being deleted successfully.

- Manage Employee Module

Employer is the only user that can perform delete operation on a particular employee .Employer will need to confirm the delete operation once the message pops up by clicking the confirm button. A message will be displayed if the particular selected employee is deleted successfully.

- Manage Vending Machine Sales Module

All users are eligible to delete the unwanted products in the system. Users will need to click on the confirm button when the delete message pops up. A message will be displayed if the particular selected products are being deleted successfully.

2.1.2 **Hardware Interfaces**

The user must interact with this system using a device that can run a browser with Javascript enabled. Users can interact using a PC or a mobile device, for example.

A database server and other backend servers for various utilities are also required by the system. While a physical on-premise server is possible, a remote server or cloud solution is more realistic in terms of cost, maintenance, and administrative labor.

A data center-based remote development server is now being used for development purposes, including hosting the database, providing web pages, and running backend servers.

2.1.3 **Software Interfaces**

Name	Version	Source	Description	Purpose
ASP.NET Core Runtime	Asp .Net Core 6.0	https://dotnet.microsoft.com/en-us/download/dotnet	The ASP.Net Core is being set up by default when starting a new project in Visual Studio 2022 with the aid of template.	Serve the system's webpage. During development the template has enabled docker support which will package the application to be deployed in server environment.
Visual Studio 2022	Version 2022	https://visualstudio.microsoft.com/downloads/	A fully-featured, extensible, free IDE for creating modern applications for Android, iOS, Windows, as well as web applications and cloud services.	A comprehensive IDE used for development. It has complete tools installed such as NuGet Package Manager.

SQL Server Management Studio 18	Version 18.12	https://docs.microsoft.com/en-us/sql/ssms/download-sql-server-management-studio-ssms?view=sql-server-ver16	SQL Server Management Studio (SSMS) is an integrated environment for managing any SQL infrastructure, from SQL Server to Azure SQL Database. SSMS provides tools to configure, monitor, and administer instances of SQL Server and databases.	A DBMS used to manage the database to carry out operations such as create table, manually insert rows.

2.1.4 **Communication Interfaces**

Web Page to Client

The HTTP/HTTPS protocol is used by the client to connect to the website. For convenience of development, the HTTP protocol is utilized; however, for better security, it will be removed and only enable HTTPS connections. During development, the website is served locally at localhost.

Connection to a database

TCP/IP is used to connect to the database. The database is connected to the ASP.NET server through Entity Framework, which is configured during development. For convenience, a local database is utilized in the development phase, but a remote database can be established for production builds.

Backend Integration

REST API is used by other backend services like email notification, report generating, and payment gateway services, which communicate via HTTP or HTTPS protocol.

2.2 **System Features**

The system features include stock management functions for MFA Solutions.

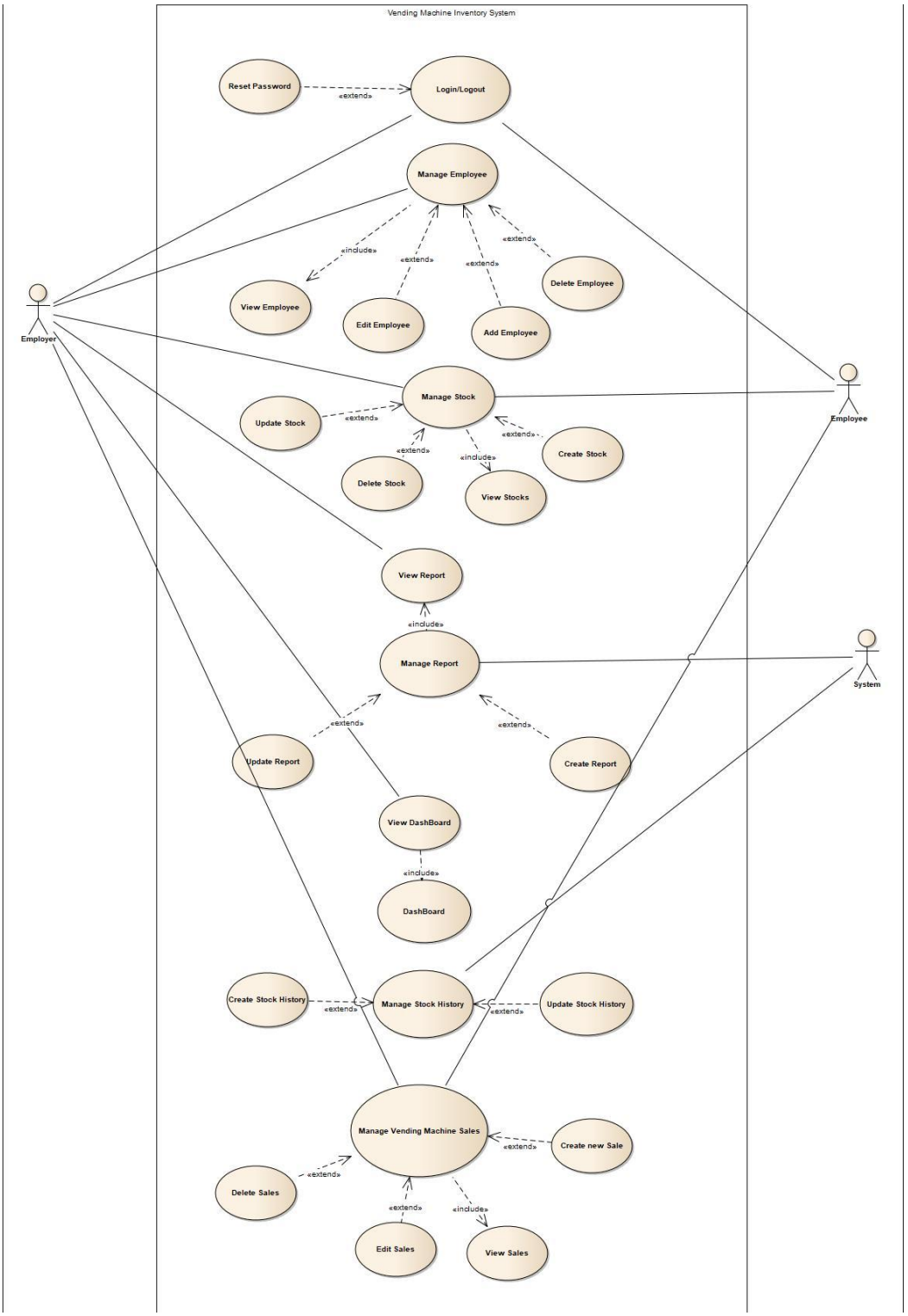

Figure 1: Use Case Diagram for Vending Machine Inventory Management System

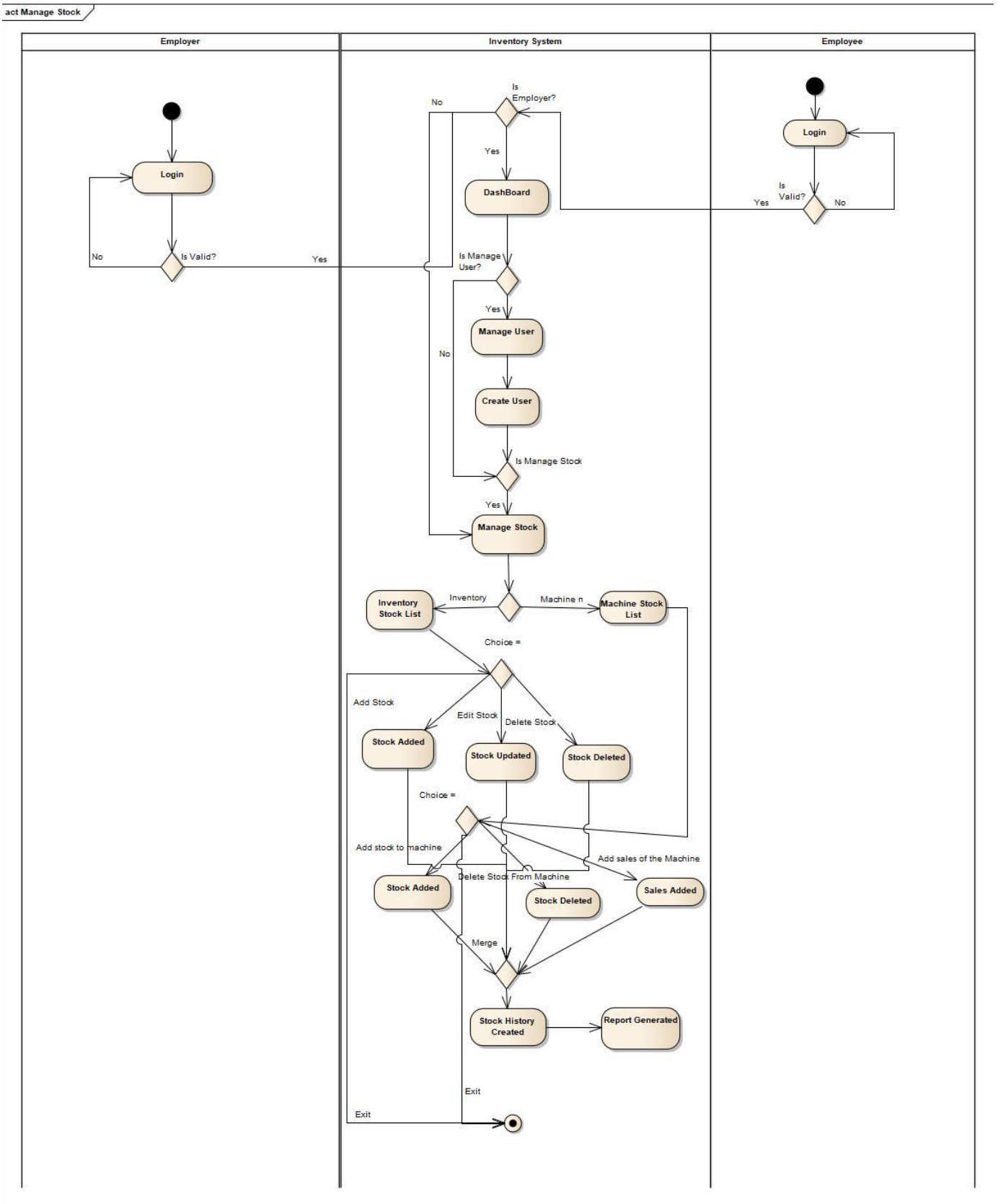

Figure 2: Activity Diagram for Vending Machine Inventory Management System

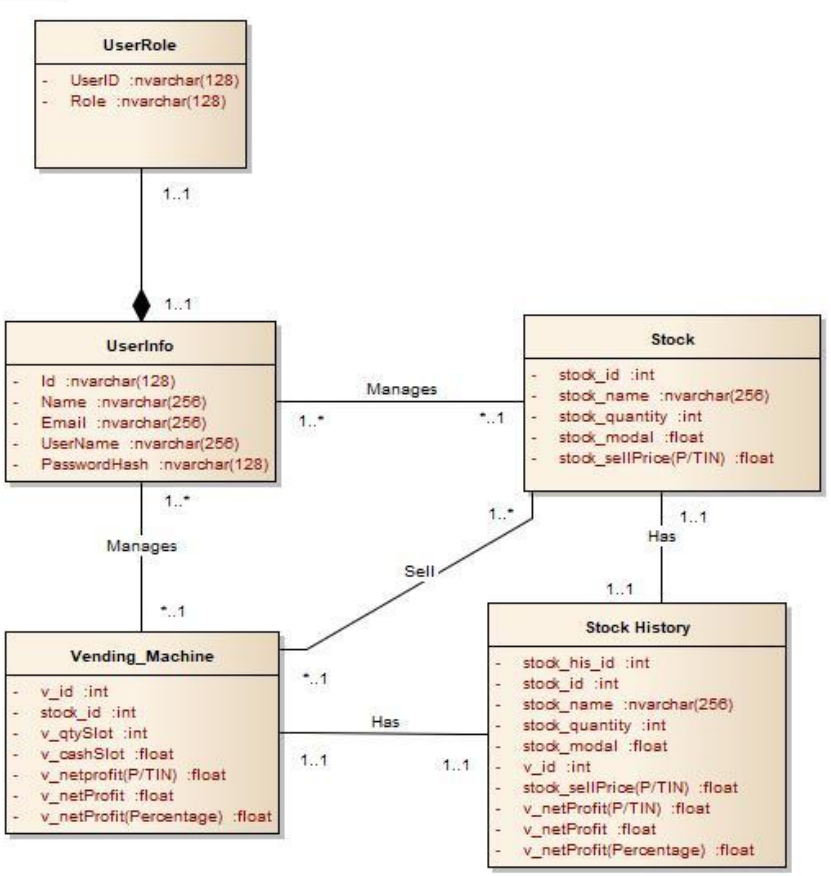

Figure 3: Domain Model for Vending Machine Inventory Management System

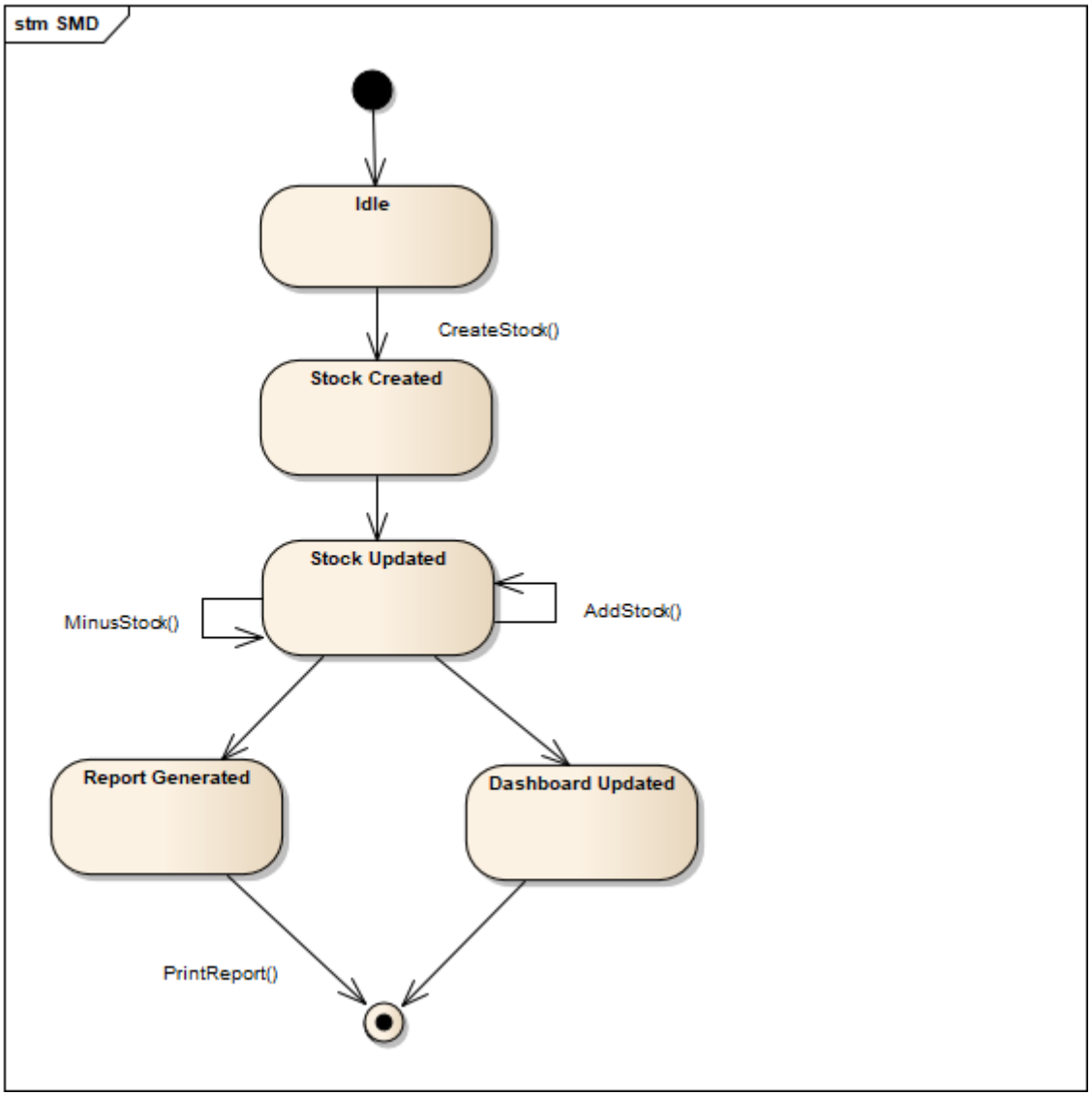

Figure 4: State Machine Diagram for Stock Inventory System

2.2.1 UC001: Use Case <Module Login/Logout>

Table 1: Use Case Description for <Module Login/Logout>

Use case: < Module Login/Logout>
ID: UC001
Actors: Employer & Employee
Preconditions: 1. User must already have an existing account (For Login) 2. User must be already logged in (For logout)
Flow of events: 1. System displays the interface of the login page 2. User inserts their login credentials 3. System checks whether the user credentials are valid 4. System displays message : “Successfully logged in” and user is logged in
Postconditions: 1. User is logged in into the system
Alternate flow 1: 1. User can then opt to logout of system 2. User clicks on logout button 3. User gets logged out and a popup message is displayed: “Successfully logged out”. 4. System displays the login page again
Postconditions: 1. User is logged out of the system
Exception flow (if any): 1. If login credentials are invalid, user will not be able to login into the system

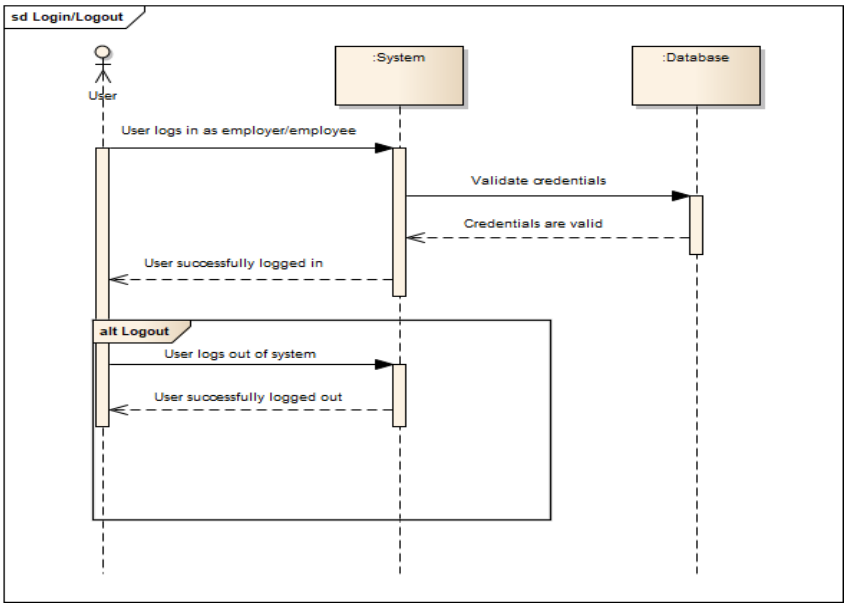

Figure 5: Sequence Diagram for Module Login/Logout

2.2.2 UC002: Extension Use Case <Create Employee>

Table 2: Use Case Description for <Create Employee>

Use case: < Create Employee >
ID: UC002
Actors: Employer
Preconditions: 1. Employer is logged in
Flow of events: 1. Employer navigates to Manage Employee tab 2. Clicks on “Add Employee” button 3. Fills in employee details 4. Employee is successfully added and a popup message is displayed: “Employee has been added successfully”
Postconditions: 1. Employee is added to list of existing employees 2. Employee profile is created

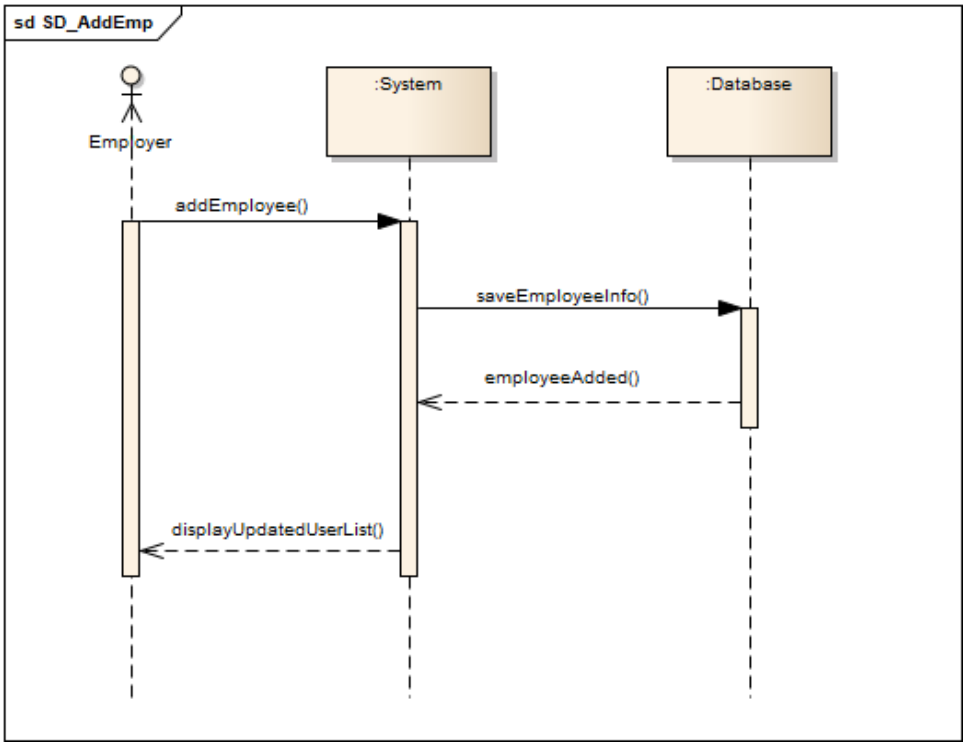

Figure 6: Sequence Diagram for Create Employee

2.2.3 UC003: Extension Use Case <Edit Employee>

Table 3: Use Case Description for < Edit Employee>

Use case: < Edit Employee >
ID: UC003
Actors: Employer & Employee
Preconditions: 1. User is logged in 2. Employee account already exists
Flow of events: 1. User clicks on “Edit” button 2. User updates desired information 3. User double checks updated information and clicks the “Save” button.
Postconditions: 1. Details of user is updated on the system

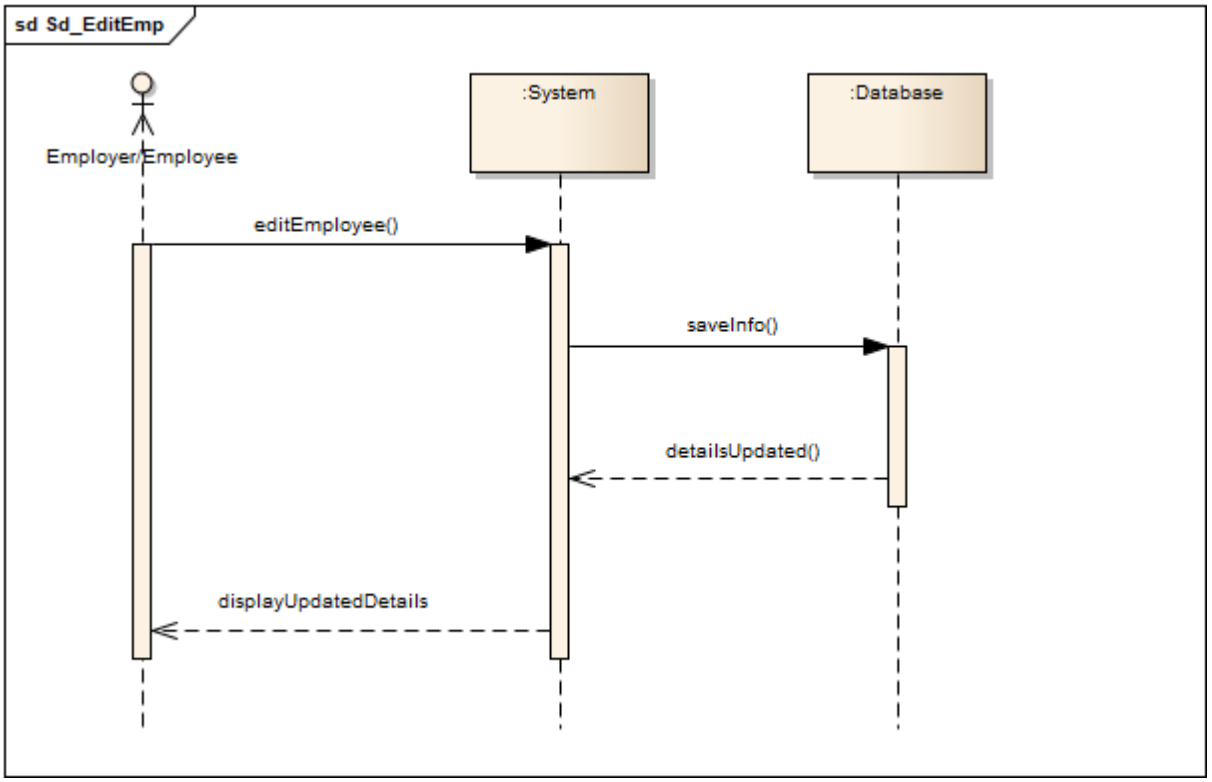

Figure 7: Sequence Diagram for Edit Employee

2.2.4 UC*004*: Extension Use Case < Delete Employee>

Table 4: Use Case Description for < Delete Employee>

Use case: < Delete Employee >
ID: UC004
Actors: Employer
Preconditions: 1. Employee is logged in
Flow of events: 1. Employer navigates to Manage Employee tab 2. Employer clicks on “Delete” button on desired employee 3. A prompt is displayed: “Are you sure you want to delete this user?” 4. Employer clicks on “Yes”
Postconditions: 1. Selected employee is deleted from the system

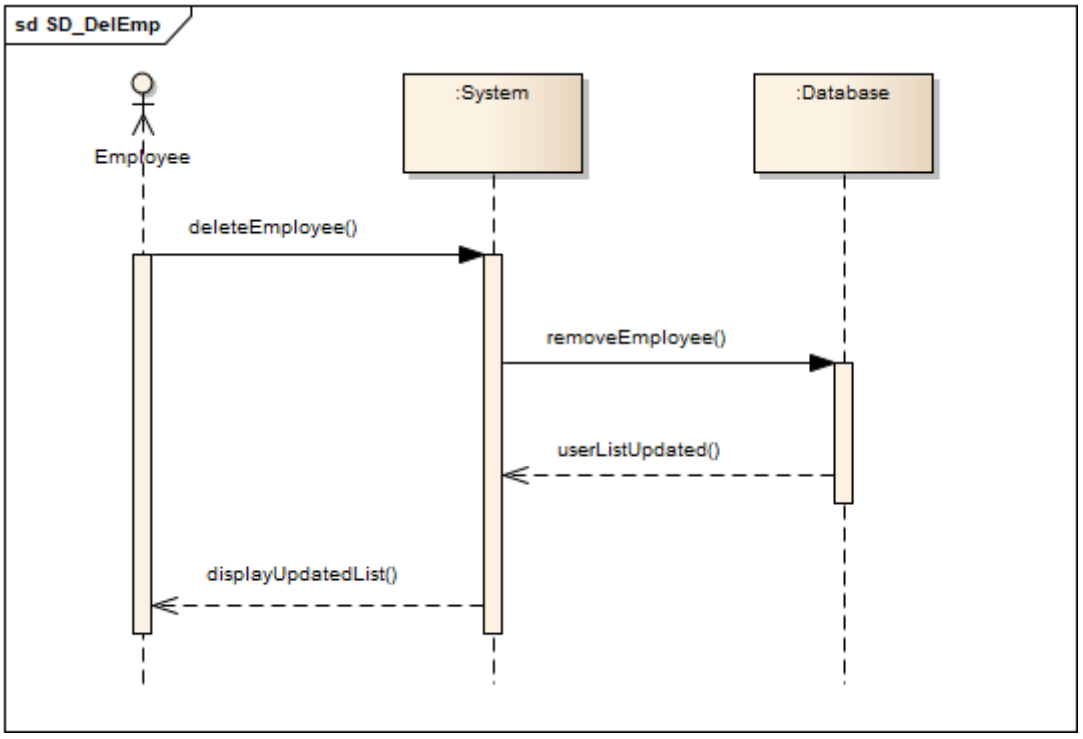

Figure 8: Sequence Diagram for Delete Employee

2.2.5 **UC*005*: Use Case < View Employee>**

Table 5: Use Case Description for < View Employee>

Use case: < View Employee >
ID: UC005
Actors: Employer
Preconditions: 1. Employer is logged in
Flow of events: 1. Employer navigates to Manage Employee tab 2. List of employees are displayed 3. Employer may click on desired employee to view more detailed information regarding that particular employee
Postconditions: 1. Employer will be able to see list of employees 2. Employer will be able to see detailed information of a certain employee

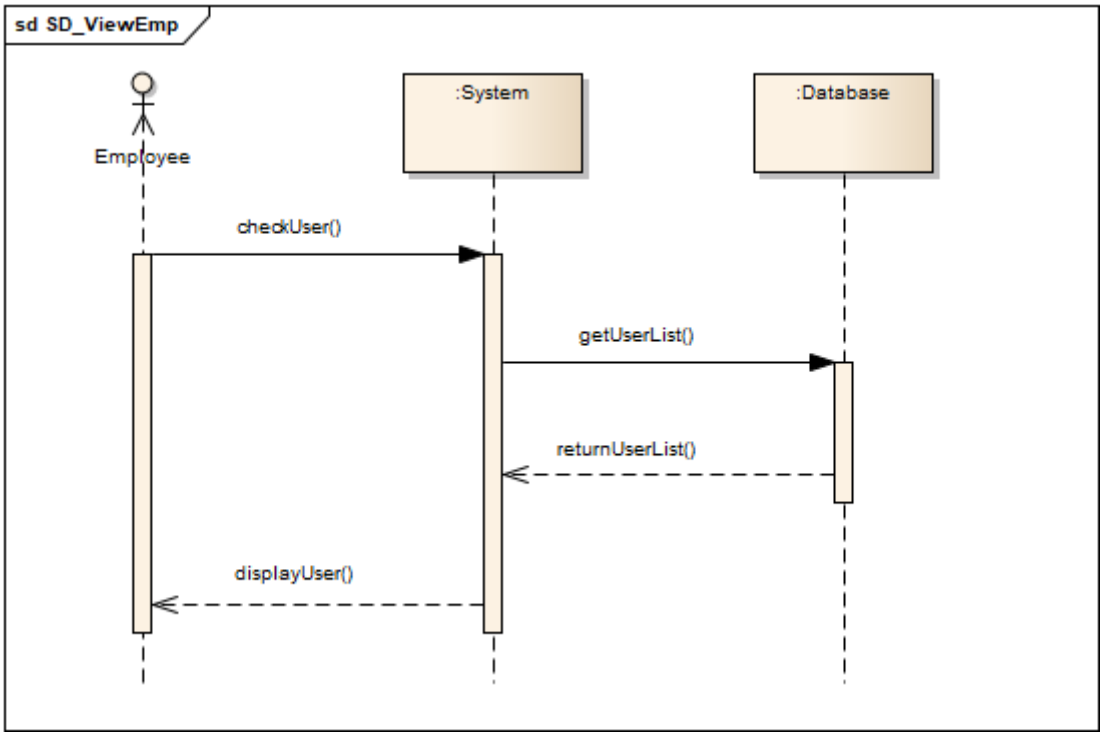

Figure 9: Sequence Diagram for View Employee

2.2.6 **UC*006*: Use Case < Manage Employee>**

Table 6: Use Case Description for <Manage Employee>

Use case: < Manage Employee >
ID: UC006
Actors: Employer & Employee
Includes: UC005 View Employee
Extension points: <Create employee> <Update employee> <Delete employee>
Preconditions: 1. User must be logged into the system
Flow of events: 1. User is logged into the system 2. Employer selects Manage employee tab 3. include (View Employee) 4. If employer clicks on “Add Employee” <Add Employee> 5. After employee is created, employer is redirected to updated list of employees 6. include (View Employee) 7. If employer/employee chooses to update personal information <Edit Employee> 8. Once changes are saved, user can view changes 9. include (View Employee) 10. If employer wants to delete employee account <Delete Employee> 11. After employee account is deleted, system redirects employer to updated list of employees 12. include (View Employee)
Postconditions: 1. Employer manages to create,read,update and delete employee information. 2. Employee manages to update personal information

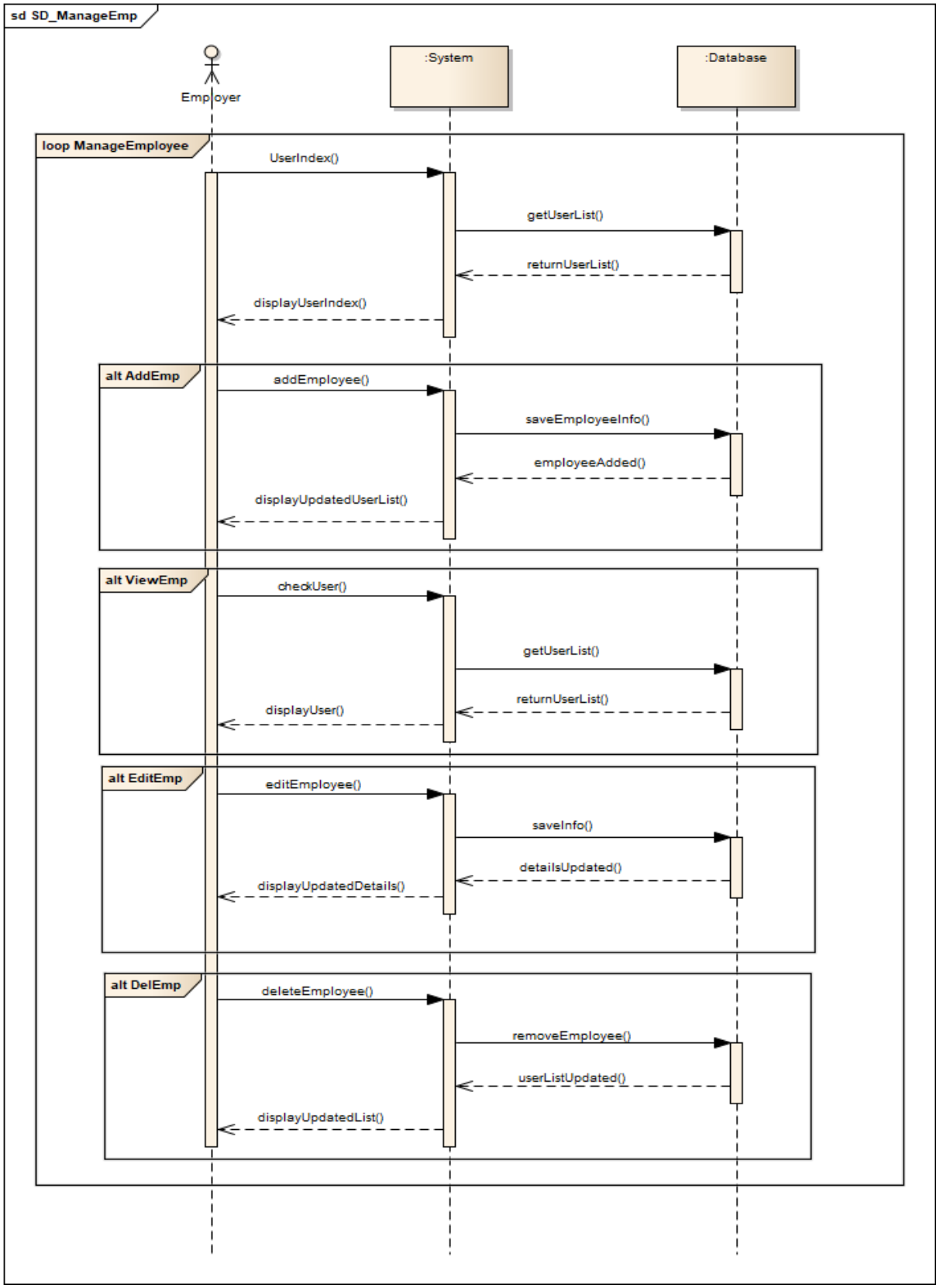

Figure 10: Sequence Diagram for Manage Employee

2.2.7 **UC007: Use Case <View Stock>**

Table 7: Use Case Description for <View Stock>

Use case: <View Stock >
ID: UC007
Actors: 1. Employee 2. Employer
Preconditions: 1. An Employer is logged on to the system with correct credentials. 2. An Employee is logged on to the system with correct credentials.
Flow of events: 1. If the Employer or Employee selects the “Inventory”: 1.1. System will display the overall stock information in inventory. 2. If Employee or Employer selects “Machine n ”,n=machine number : 2.1. System will display the specific stock information inside the machine .
Postconditions: The Employer or Employee can view the stock details.
Alternative flow: At any point the Employee or Employer may leave the stock view page.

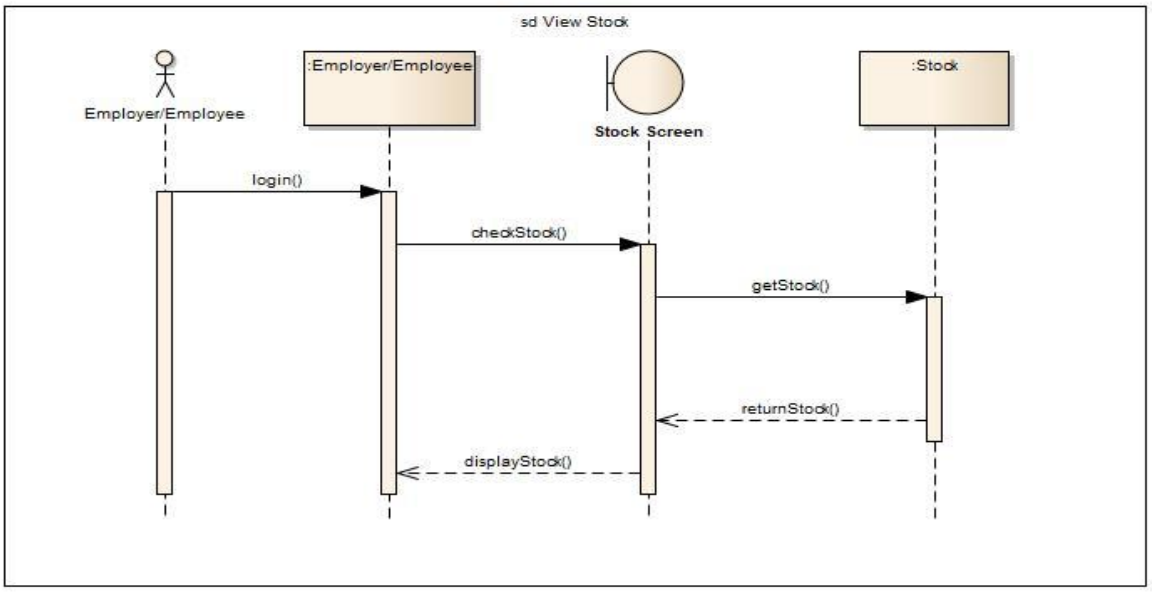

Figure 11: Sequence Diagram for View Stocks

2.2.8 **UC008: Extension Use Case <Create Stock>**

Table 8:Extension Use Case Description for <Create Stock>

Extension Use case: <Create Stock >
ID: UC008
Actors: 1. Employee 2. Employer
Preconditions: 1. An Employer is logged on to the system with correct credentials. 2. An Employee is logged on to the system with correct credentials.
Flow of events: 1. The system will display a form to create a new stock. 2. The Employer or Employee must fill up the form and submit it. 3. If the system successfully added the stock to database: 3.1. A “Successfully Created” message will be displayed 4. I got some error: 4.1. A ”error” message will be displayed.
Postconditions: The database is updated with the new stock.
Alternative flow: At any point the Employee or Employer may leave the create stock page.

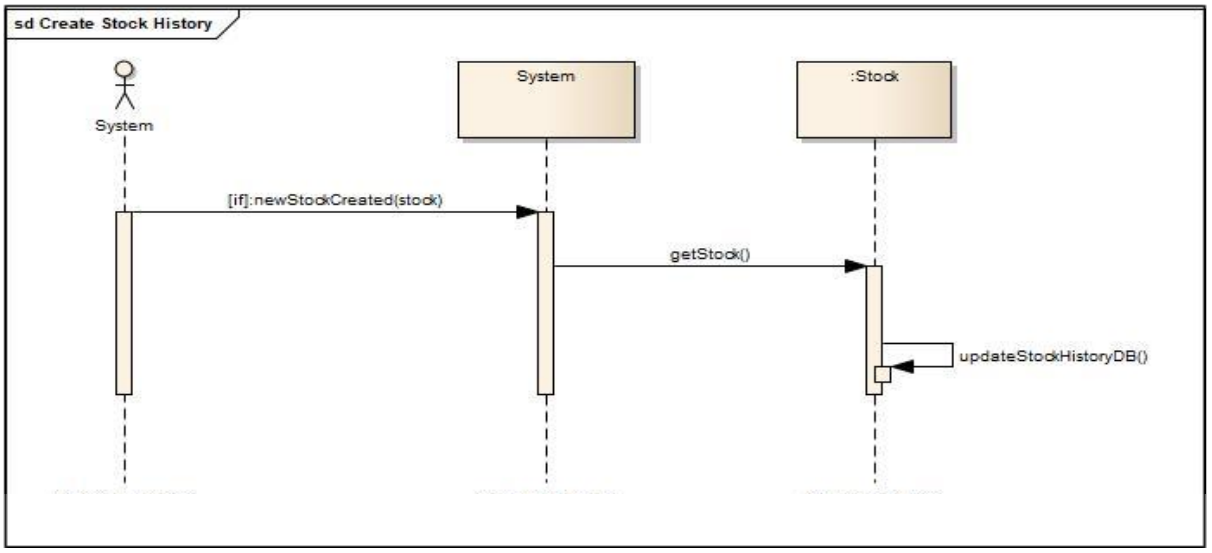

Figure 12: Sequence Diagram for Create Stocks

2.2.9 **UC009: Extension Use Case <Update Stock>**

Table 9:Extension Use Case Description for <Update Stock>

Extension Use case: <Update Stock >
ID: UC009
Actors: 1. Employee 2. Employer
Preconditions: 1. An Employer is logged on to the system with correct credentials. 2. An Employee is logged on to the system with correct credentials.
Flow of events: 1. If the Employer or Employee selects the “Inventory”: 1.1. The Employer or Employee will able to add more stocks to inventory 1.2. The Employer or Employee will able to change the stock information by filling out the form 2. If Employee or Employer selects “Machine n ”,n=machine number : 2.1. The Employer or Employee will be able to input how much stock sales ini the machine and how much put in the slot. 3. If the system successfully updated the stock to database: 3.1. A “Successfully Updated ” message will be displayed 4. I got some error: 4.1. A ”error” message will be displayed.
Postconditions: The database is updated with the updated stock details.
Alternative flow: At any point the Employee or Employer may leave the update stock page.

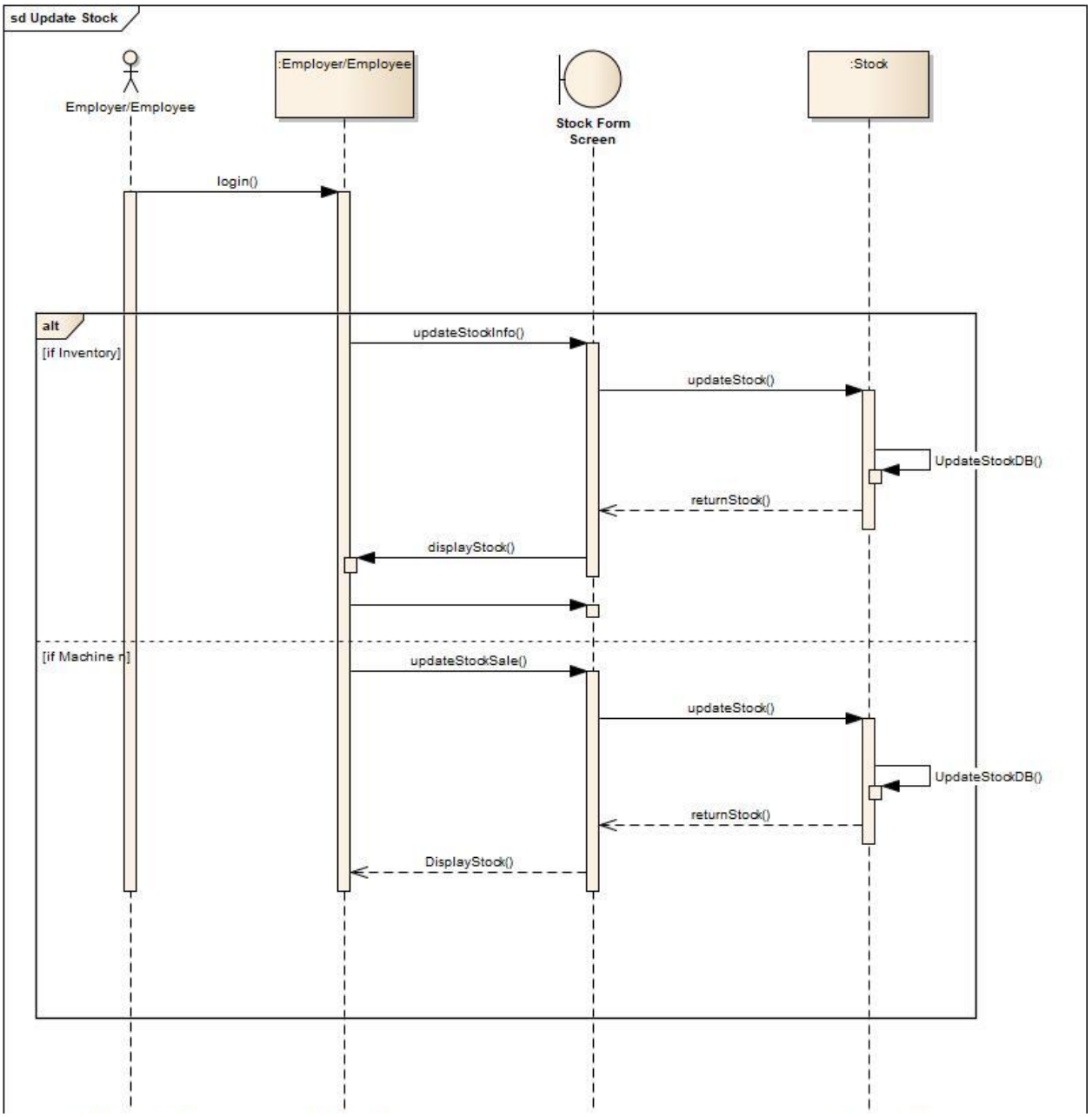

Figure 13: Sequence Diagram for Update Stocks

2.2.10 **UC010: Extension Use Case < Delete Stock>**

Table 10:Extension Use Case Description for <Delete Stock>

Extension Use case: <Delete Stock >
ID: UC010
Actors: 1. Employee 2. Employer
Preconditions: 1. An Employer is logged on to the system with correct credentials. 2. An Employee is logged on to the system with correct credentials.
Flow of events: 1. After the Employee or Employer clicks the delete button,it will display the delete confirmation message. 2. The Employee or Employer will need to click the “Yes” button to delete the stock. 3. If the system successfully deleted the stock from database: 3.1. A “Successfully Deleted” message will be displayed 4. I got some error: 4.1. A ”error” message will be displayed.
Postconditions: The database is updated with the updated stock details.
Alternative flow: At any point the Employee or Employer may leave the delete stock page.

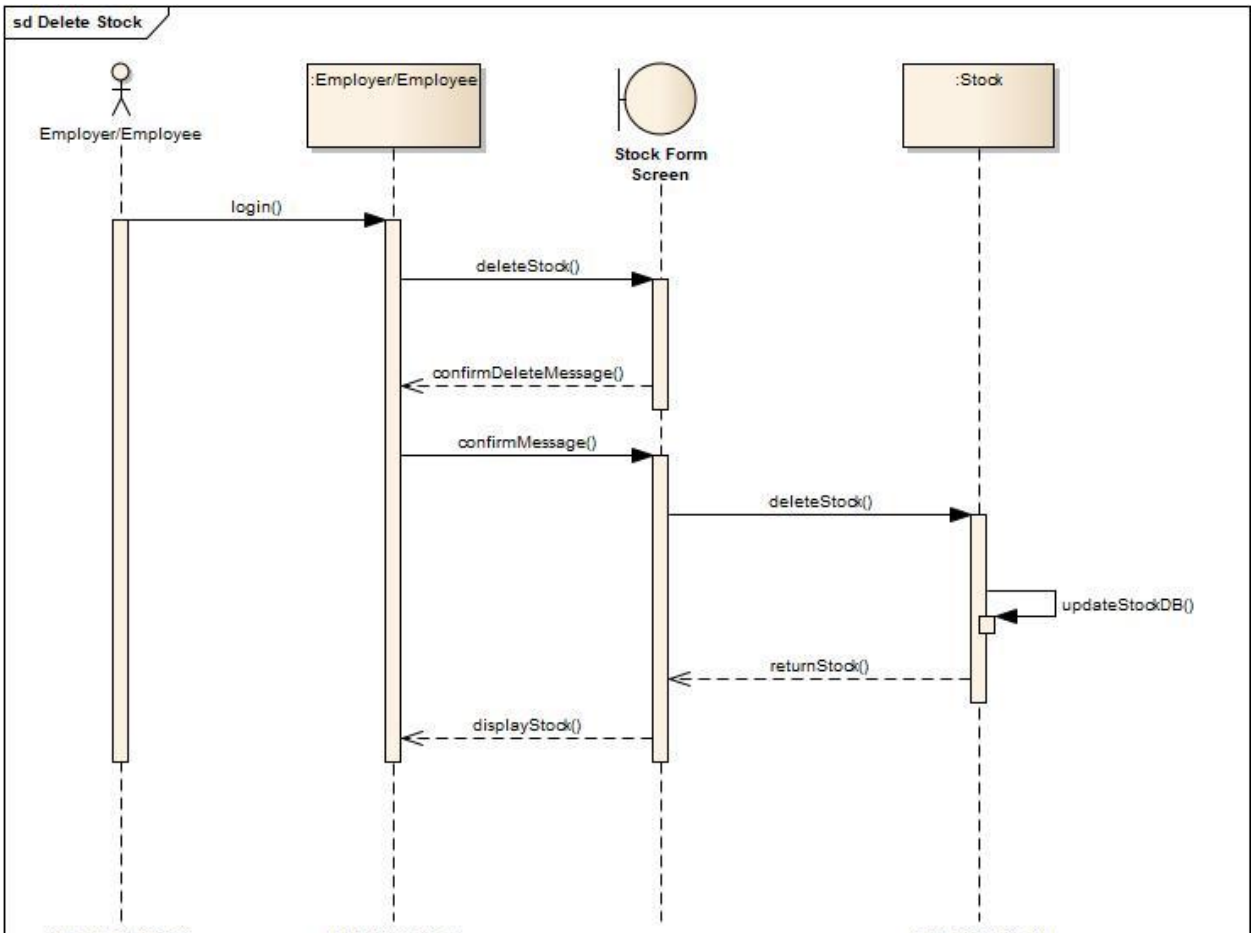

Figure 14: Sequence Diagram for Delete Stocks

2.2.11 **UC011: Use Case < Manage Stock >**

Table 11: Use Case Description for <Manage Stock>

Use case: <Manage Stock >
ID: UC011
Actors: 1. Employee 2. Employer
Includes: UC007<View Stocks>
Extension Points: UC008<Create Stocks> UC009<Update Stocks> UC010<Delete Stocks>
Preconditions: 1. An Employer is logged on to the system with correct credentials. 2. An Employee is logged on to the system with correct credentials.
Flow of events: 1. The Employee or Employer enters the system and selects the ”Manage Stocks” which has 2 options. 2. If Employee or Employer selects “Inventory” 2.1. Include(View Stocks) by overall 2.2. If the employee or employer want to add new overall stocks: <Create Stocks> 2.3. If the employee or employer want to update stock overall information: <Update Stocks> 2.4. If the employee or employer want to delete stock overall information: <Delete Stocks> 3. If Employee or Employer selects “Machine n ”,n=machine number : 3.1. If the employee or employer want to update machine stock information like how much they sell: <Update Stocks> 4. After all the above, the system will go back to view stocks by category. 5. Include (View Stocks)
Postconditions: 1. The employee or employer managed to update the database by create,update,delete stocks
Alternative flow : 1. At any point, the employee or employer may leave the create,update,delete stock page. 2. If the employee or employer leaves the page during the middle of the create,update,delete stock process the database will not be updated with the new information. 3. A log file/copy of the stock information will be created and saved in stock history, include <Manage Stock History>

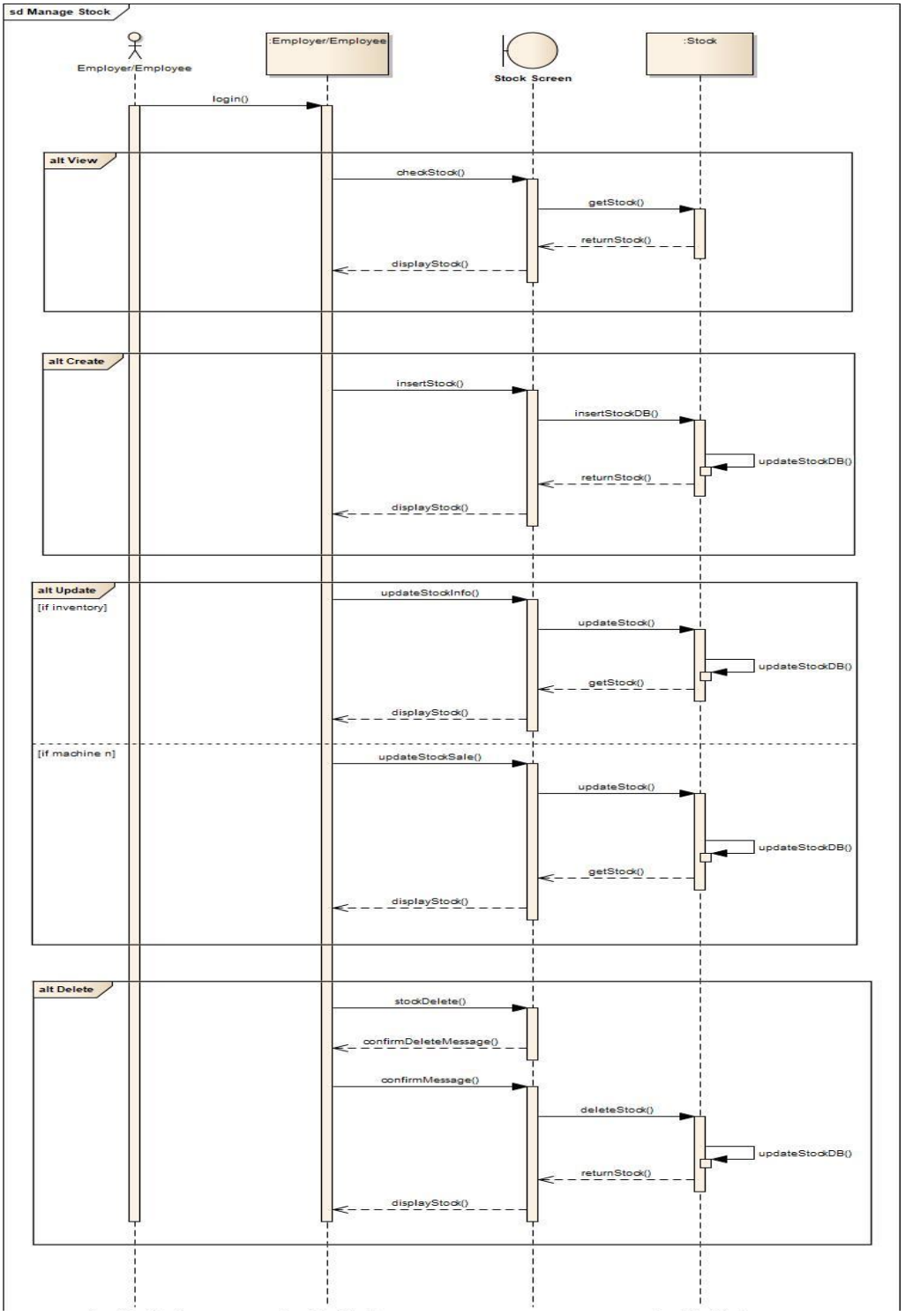

Figure 15: Sequence Diagram for Manage Stock

2.2.12 **UC012: Extension Use Case <Create Stock History>**

Table 12:Extension Use Case Description for <Create Stock History>

Extension Use case: <Create Stock History >
ID: UC012
Actors: System
Preconditions: A stock has created in <Manage Stock> module
Flow of events: 1. After,a stock created , the system will automatically create a same stock here
Postconditions: The database is updated with the new stock history details.

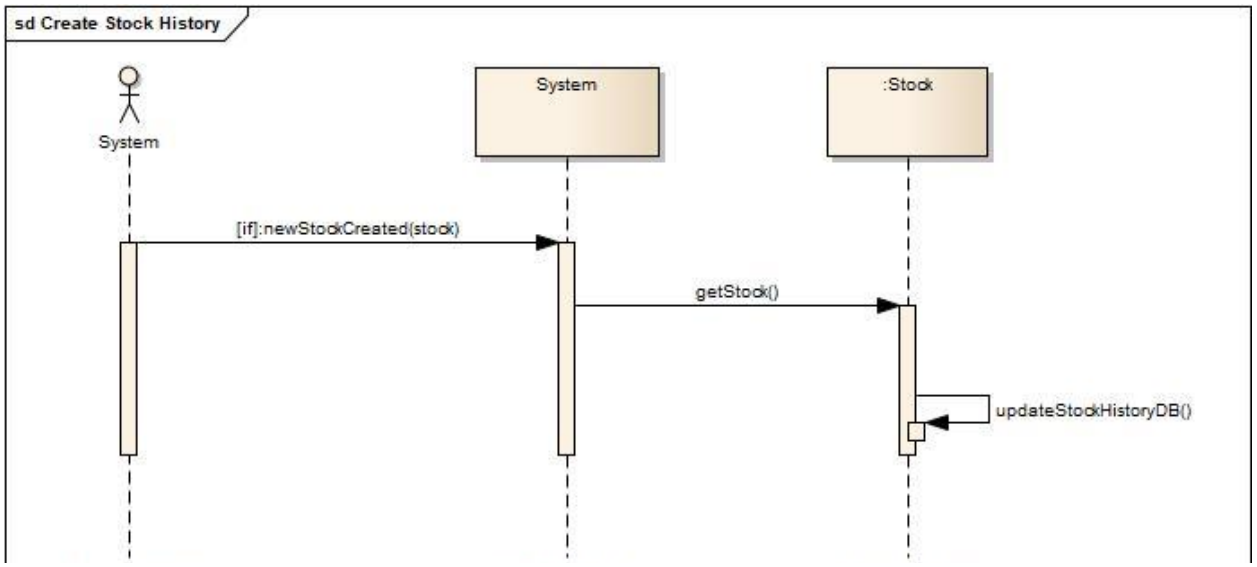

Figure 16: Sequence Diagram for Create Stock History

2.2.13 **UC013: Extension Use Case <Update Stock History>**

Table 13:Extension Use Case Description for <Update Stock History>

Extension Use case: <Update Stock History >
ID: UC014
Actors: System
Preconditions: A stock has updated in <Manage Stock> module
Flow of events: 2. After the stock update, the system will automatically update the same stock here.
Postconditions: The database is updated with the new updated stock history details.

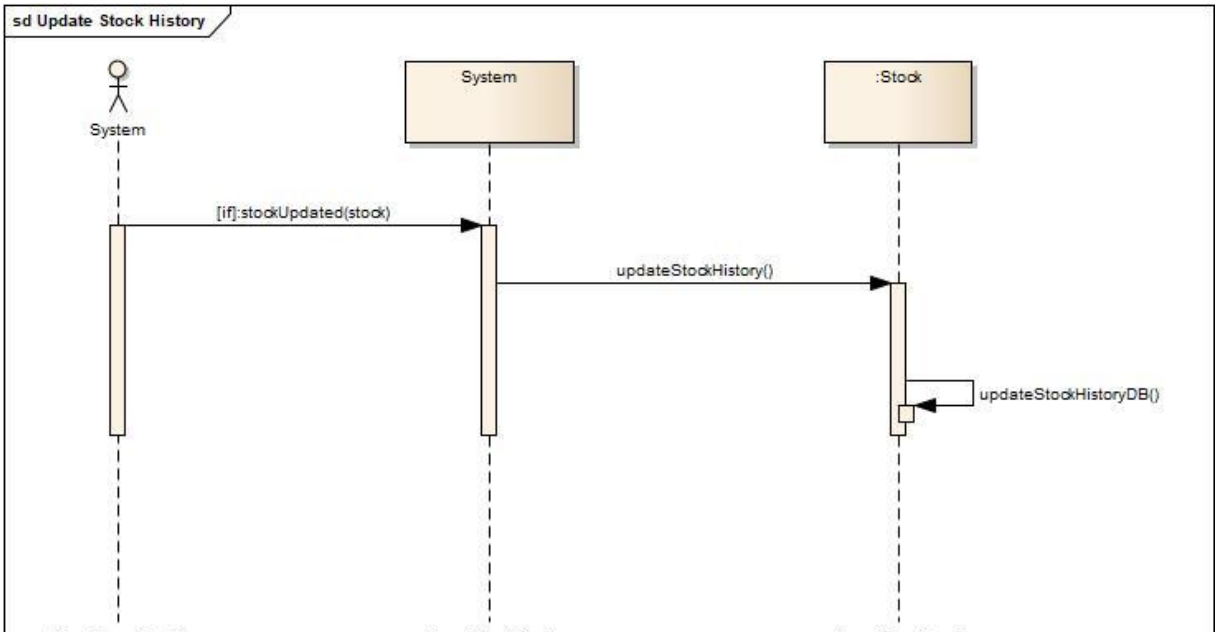

Figure 17: Sequence Diagram for Update Stock History

2.2.14 **UC014: Use Case < Manage Stock History>**

Table 14: Use Case Description for <Manage Stock History>

Use case: <Manage Stock >
ID: UC014
Actors: System
Extension Points: <Create Stocks History> <Update Stocks History>
Preconditions: A stock has created/updated in <Manage Stock> module
Flow of events: 1. The system will receive the data inserted into <Manage Stock> module 2. If new stock created: 2.1. The system will save the data into the database: <Create Stock History > 3. If stock updated: 3.1. The system will save the updated data into the database: <Update Stock History >
Postconditions: The system managed to update the database by create,update stock history.

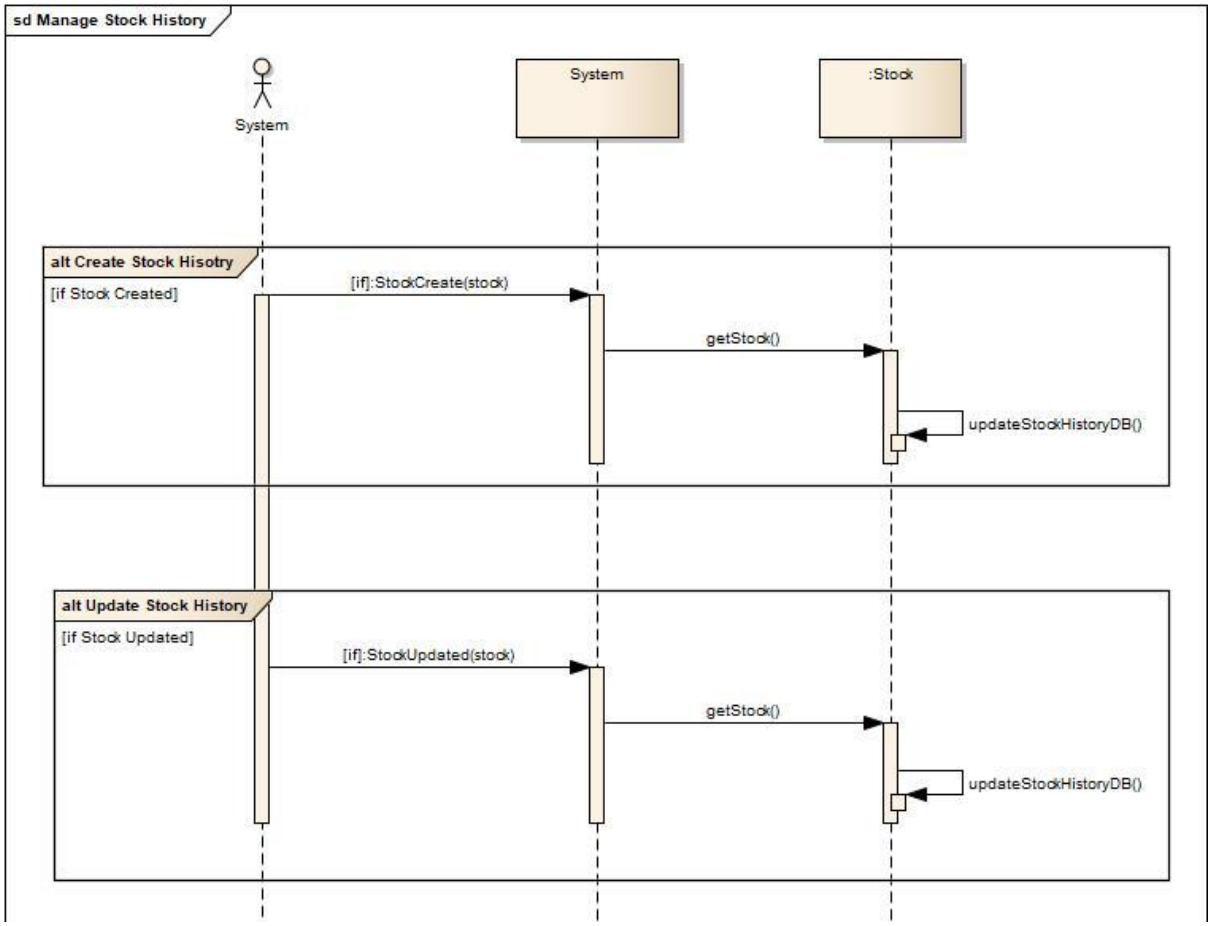

Figure 18: Sequence Diagram for Manage Stock History

2.2.15 **UC*015*: Use Case <View Dashboard>**

Table 15: Use Case Description for <View Dashboard>

Use case: <View Dashboard>
ID: UC015
Actors: Admin
Preconditions: A valid employer user is logged in
Flow of events: 1. Click the dashboard button 2. View the dashboard and get updated to the latest information.
Postconditions: 1. User able to view dashboard

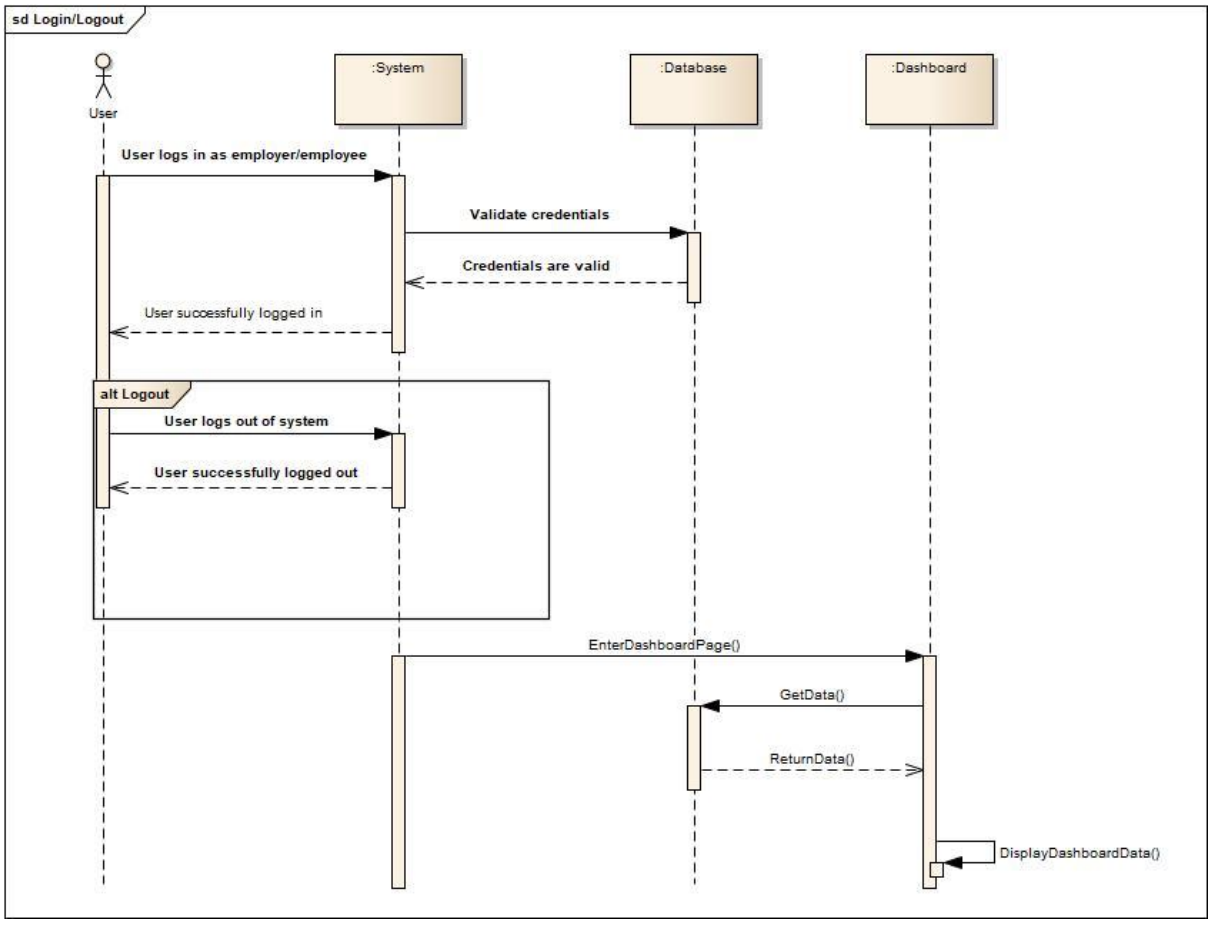

Figure 19: Sequence Diagram for View Dashboard

2.2.16 **UC*016*: Use Case < View Report >**

Table 16: Use Case Description for <View Report>

Use case: < View Report >
ID: UC016
Actors: 1. System 2. Employer
Preconditions: 1. An Employer is logged on to the system with correct credentials.
Flow of events: 1. If the Employer selects the “Report ”: 1.1. System will display the auto generated Report.
Postconditions: The Employer can view the auto generated report.
Alternative flow: At any point the Employer may leave the report view page.

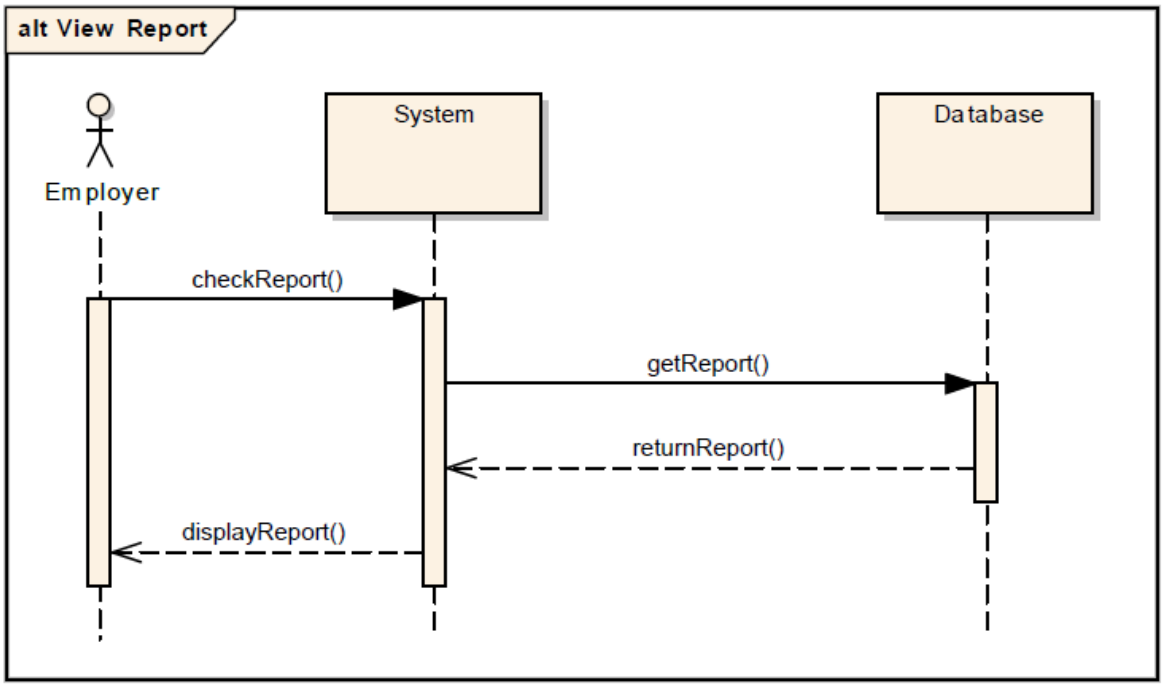

Figure 20: Sequence Diagram for View Report

2.2.17 **UC*017*:Extension Use Case < Create Report >**

Table 17: Use Case Description for < Create Report>

Extension Use case: < Create Report >
ID: UC017
Actors: 1. Employer
Preconditions: 1. An Employer is logged on to the system with correct credentials.
Flow of events: 1. The system creates a new report in the background. 2. If the system successfully created the report to database: 2.1. A “Report Successfully Created” message will be displayed or sent to the Employer 3. If the Report cannot be created : 3.1. A ”error” message will be displayed.
Postconditions: The database is updated with the new report.
Alternative flow:

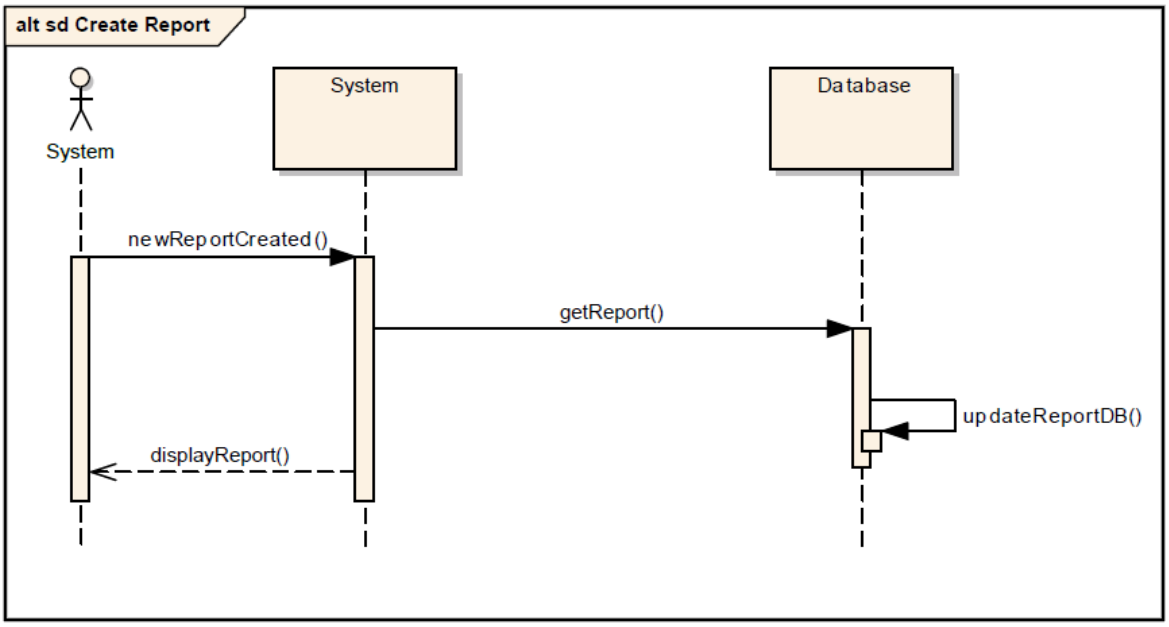

Figure 21: Sequence Diagram for Create Report

2.2.18 **UC*018*: Extension Use Case < Update Report >**

Table 18: Use Case Description for < Update Report>

Extension Use case: < Update Report >
ID: UC018
Actors: 1. Employer
Preconditions: 1. An Employer is logged on to the system with correct credentials.
Flow of events: 1. If the Employer selects the “Report”: 1.1. The Employer will be able to view the auto generated report. 1.2. The Employer or Employee will be able to create a report which will be generated automatically. 1.3. The Employer or Employee will be able to update a report which will be updated automatically 2. If the system successfully updated the report to database: 2.1. A “ Report Successfully Updated ” message will be displayed 3. If the Report cannot be updated: 3.1. A ”error” message will be displayed.
Postconditions: The database is updated with the updated report.
Alternative flow: At any point the Employer may leave the Report page.

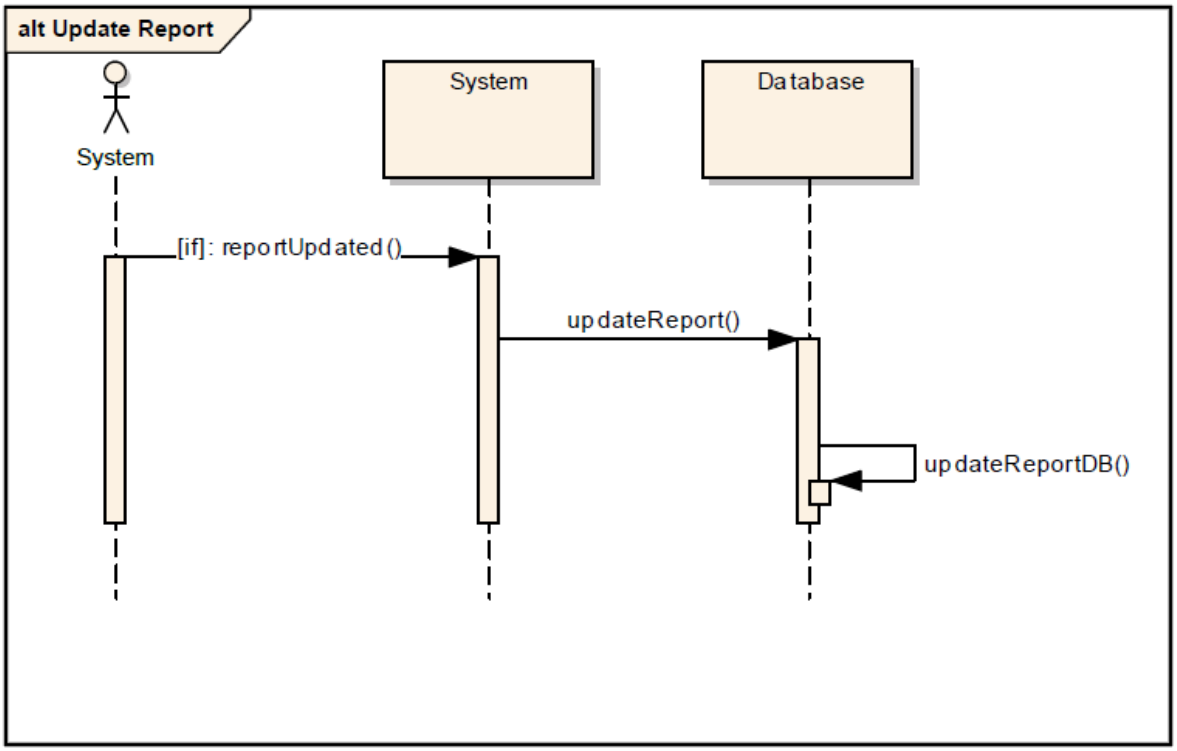

Figure 22: Sequence Diagram for Update Report

2.2.19 **UC*019*: Use Case < Manage Report>**

Table 19: Use Case Description for < Manage Report>

Use case: < Manage Report >
ID: UC019
Actors: 1. System 2. Employer
Extension Points: <View Report> <Create Report> <Update Report>
Preconditions: 1. A stock has created/updated in <Manage Report> module 1. An Employer is logged on to the system with correct credentials
Flow of events: 1. The system will receive the data inserted into <Manage Report> module 2. If new report is generated: 2.1. The auto generated report will be made by the system in: <Create Report > 3. If report updated: 3.1. The system will save the updated report into the database: <Update Stock History > 4. If the Employer selects the “Report ”: 4.1. System will display the auto generated Report.
Postconditions: The system managed to update the database by create,update report.

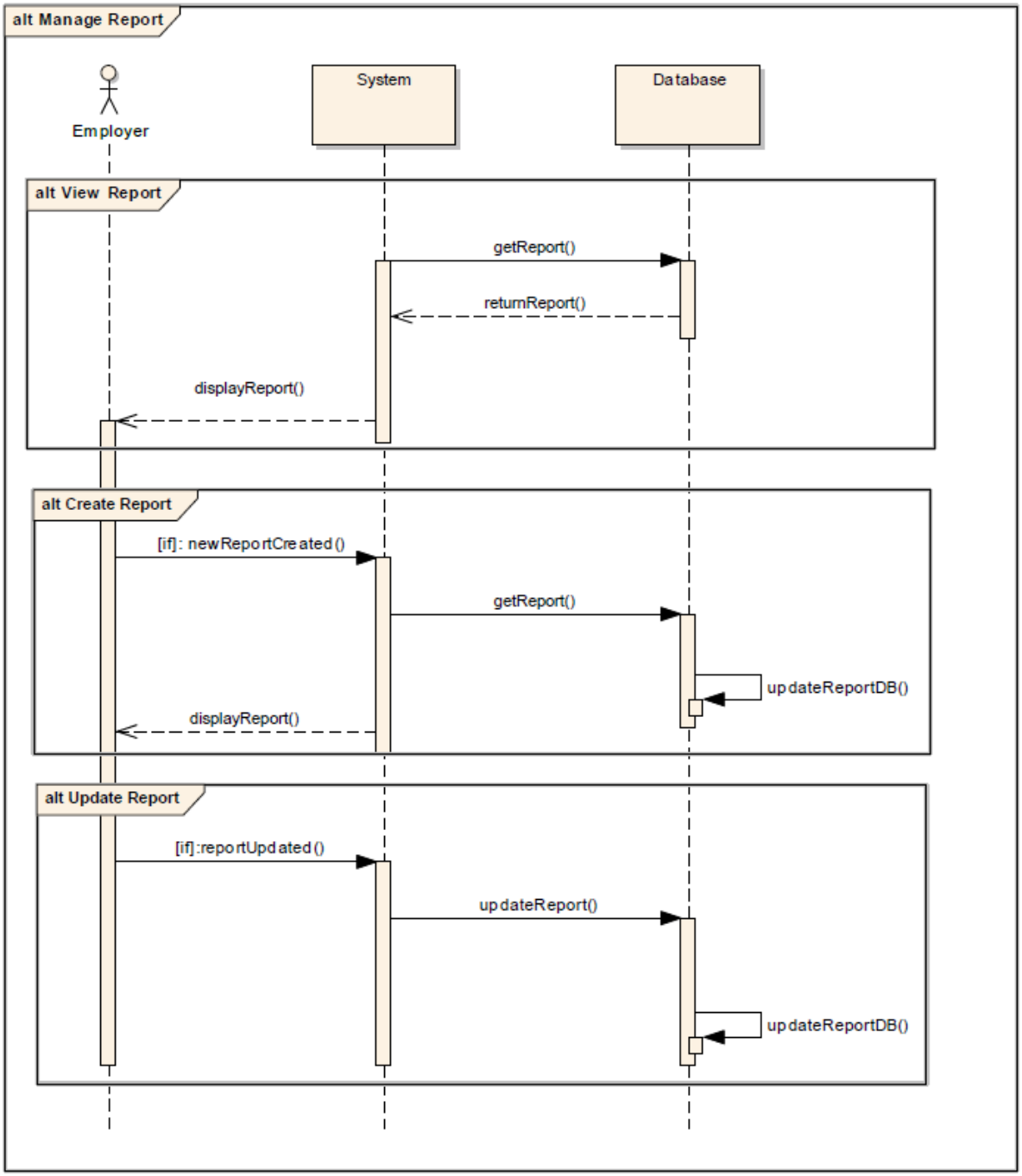

Figure 23: Sequence Diagram for Manage Report

2.2.20 **UC*020*: Use Case < View Vending Machine Sales >**

Table 20: Use Case Description for < View Vending Machine Sales>

Use case: < View Vending Machine Sales >
ID: UC020
Actors: 1. Employer 2. Employee
Preconditions: 1. An Employer/Employee is logged on to the system with correct credentials.
Flow of events: 1. Employer/Employee clicks on “Vending Machine” and selects preferred vending machine 2. System displays list of sales in existing vending machine 3. Employer/Employee can filter sales if needed
Postconditions: Employer/Employee can view sales in a specific vending machine

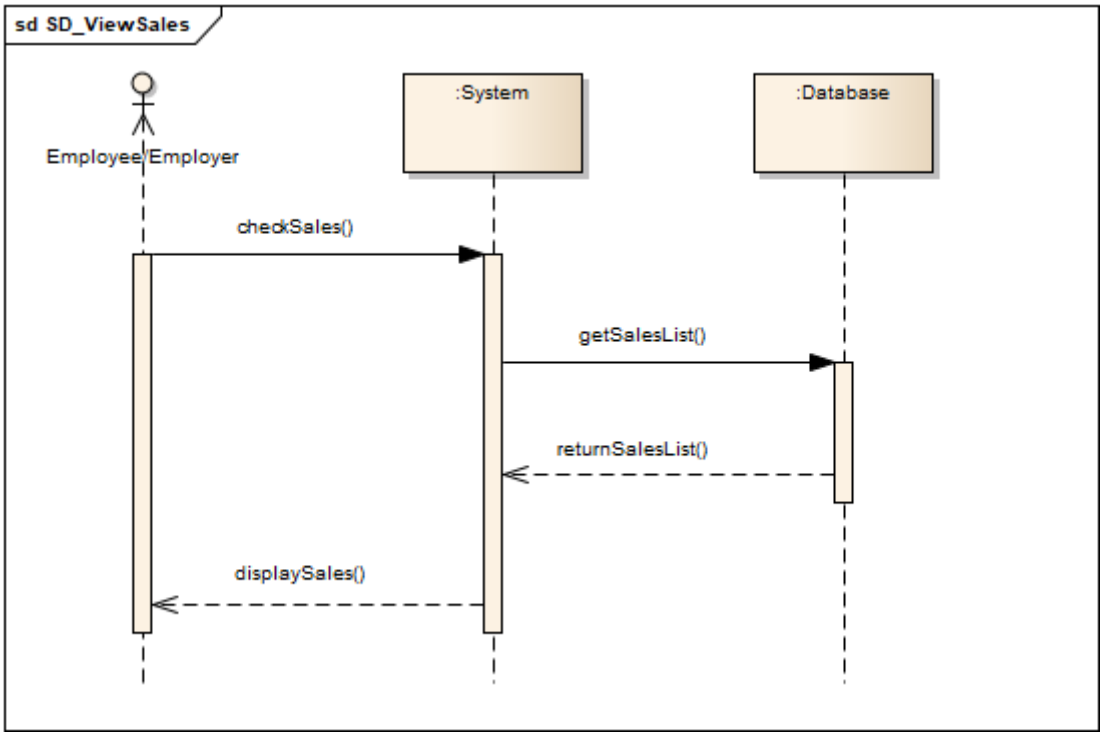

Figure 24: Sequence Diagram for View Vending Machine Sales

2.2.21 **UC*021*: Extension Use Case < Create Vending Machine Sales >**

Table 21: Use Case Description for < Create Vending Machine Sales>

Extension Use case: < Create Vending Machine Sales >
ID: UC021
Actors: 1. Employer 2. Employee
Preconditions: 1. An Employer/Employee is logged on to the system with correct credentials.
Flow of events: 1. Employer/Employee clicks on “Vending Machine” and selects preferred vending machine 2. System displays list of sales in existing vending machine 3. Employer/Employee click on “Add new vending machine sale” 4. Employer/Employee fills up fields and clicks “Create”
Postconditions: New sales is added to specified vending machine

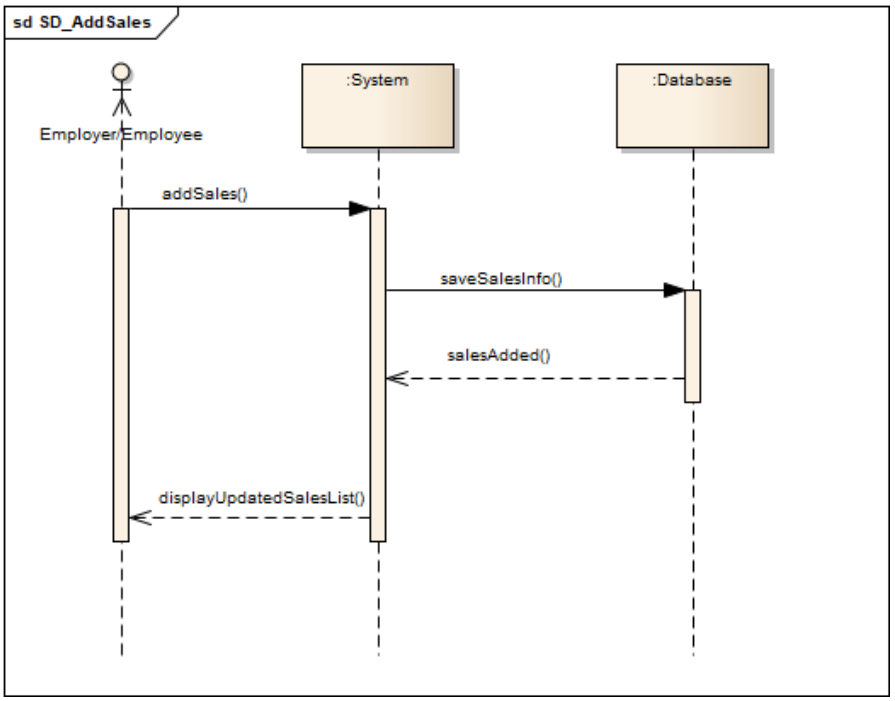

Figure 25: Sequence Diagram for Create Vending Machine Sales

2.2.22 **UC*022*: Extension Use Case < Update Vending Machine Sales >**

Table 22: Use Case Description for < Update Vending Machine Sales>

Extension Use case: < Update Vending Machine Sales >
ID: UC022
Actors: 1. Employer 2. Employee
Preconditions: 1. An Employer/Employee is logged on to the system with correct credentials. 2. Sales must exist in specified vending machine
Flow of events: 1. Employer/Employee clicks on “Vending Machine” and selects preferred vending machine 2. System displays list of sales in existing vending machine 3. Employer/Employee click on “Edit” button on specified sales 4. Employer/Employee edits fields and clicks “Save”
Postconditions: Sale is updated for specified vending machine

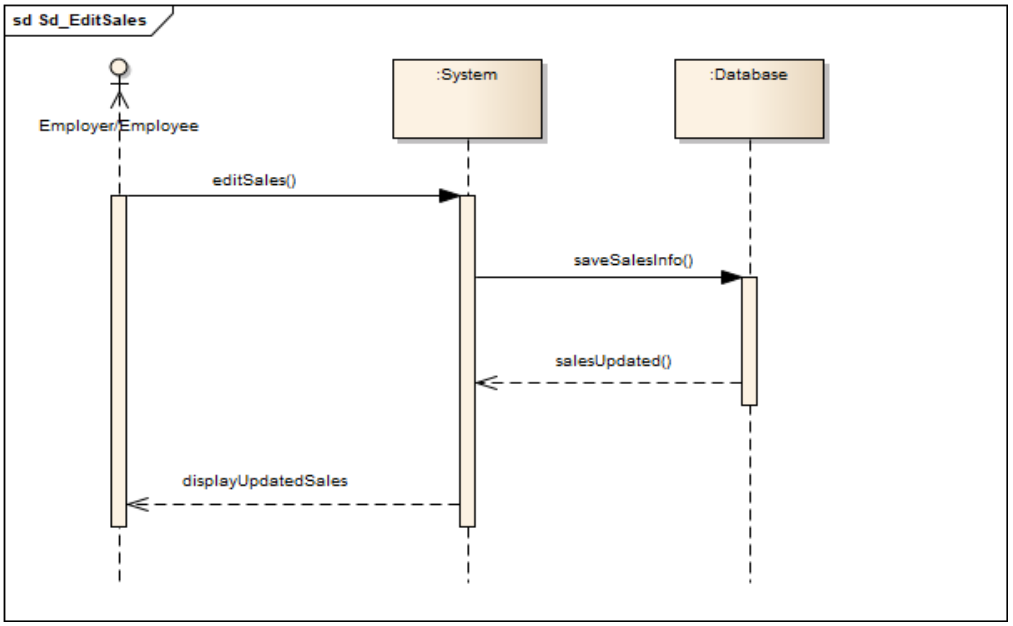

Figure 26: Sequence Diagram for Update Vending Machine Sales

2.2.23 **UC*023*: Extension Use Case < Delete Vending Machine Sales >**

Table 23: Use Case Description for < Delete Vending Machine Sales>

Extension Use case: < Delete Vending Machine Sales >
ID: UC023
Actors: 1. Employer 2. Employee
Preconditions: 1. An Employer/Employee is logged on to the system with correct credentials. 2. Sales must exist in specified vending machine
Flow of events: 1. Employer/Employee clicks on “Vending Machine” and selects preferred vending machine 2. System displays list of sales in existing vending machine 3. Employer/Employee click on “Delete” button on specified sales 4. Confirmation message is shown and user clicks on “Confirm Delete”
Postconditions: Specific sales is deleted from vending machine

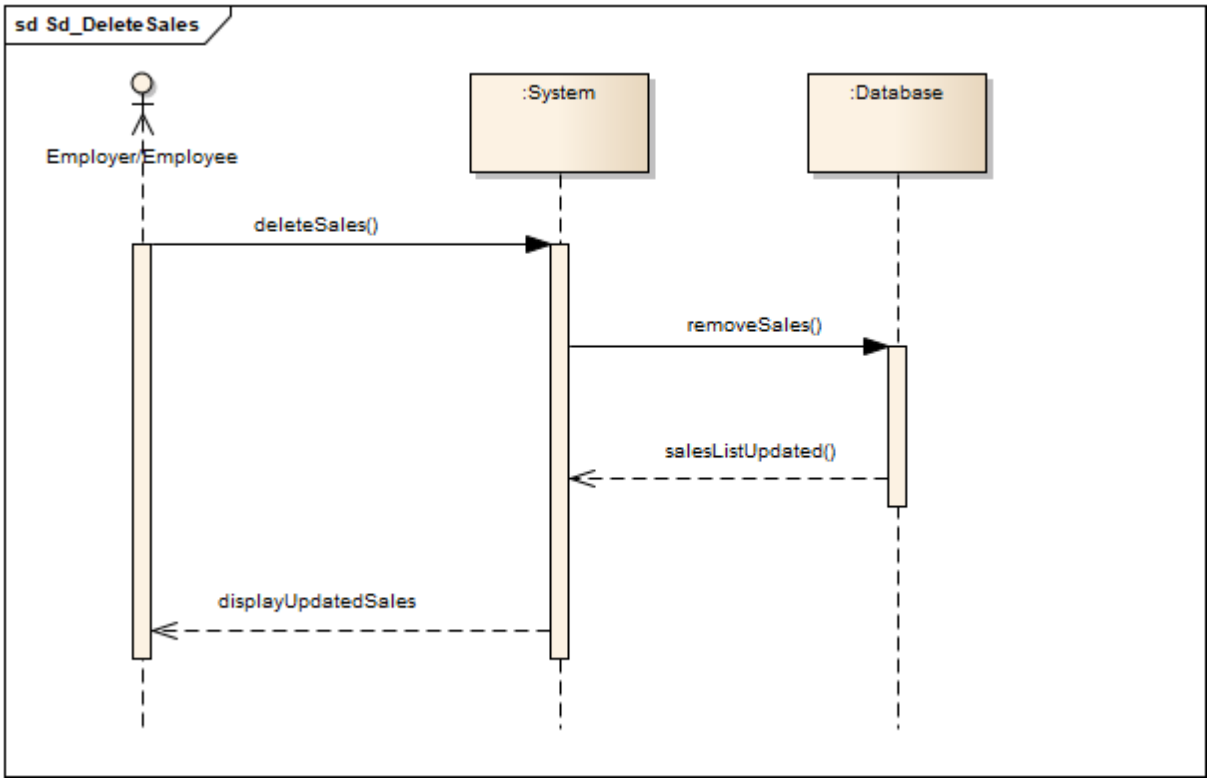

Figure 27: Sequence Diagram for Delete Vending Machine Sales

2.2.24 **UC*024*: Use Case < Manage Vending Machine Sales >**

Table 24: Use Case Description for < Manage Vending Machine Sales>

Use case: < Manage Vending Machine Sales >
ID: UC024
Actors: 1. Employer 2. Employee
Includes: UC005 View Vending Machine Sales
Extension points: <Create Vending Machine Sales> <Update Vending Machine Sales> <Delete Vending Machine Sales>
Preconditions: 1. User must be logged into the system
Flow of events: 1. The user enters the system and selects “Vending Machine” 2. include (View Vending Machine Sales) 3. If user wants to create new sales <Create Vending Machine Sales> 4. After sales is created, user is redirected to updated list of sales 5. include (View Vending Machine Sales) 6. If user chooses to update sales <Edit Vending Machine Sales> 7. Once changes are saved, user can view changes 8. include (View Vending Machine Sales) 9. If user wants to delete sales <Delete Vending Machine Sales> 10. After sales is deleted, system redirects employer to updated list of sales 11. include (View Vending Machine Sales)
Postconditions: 1. Employer manages to create,read,update and delete Vending Machine Sales.

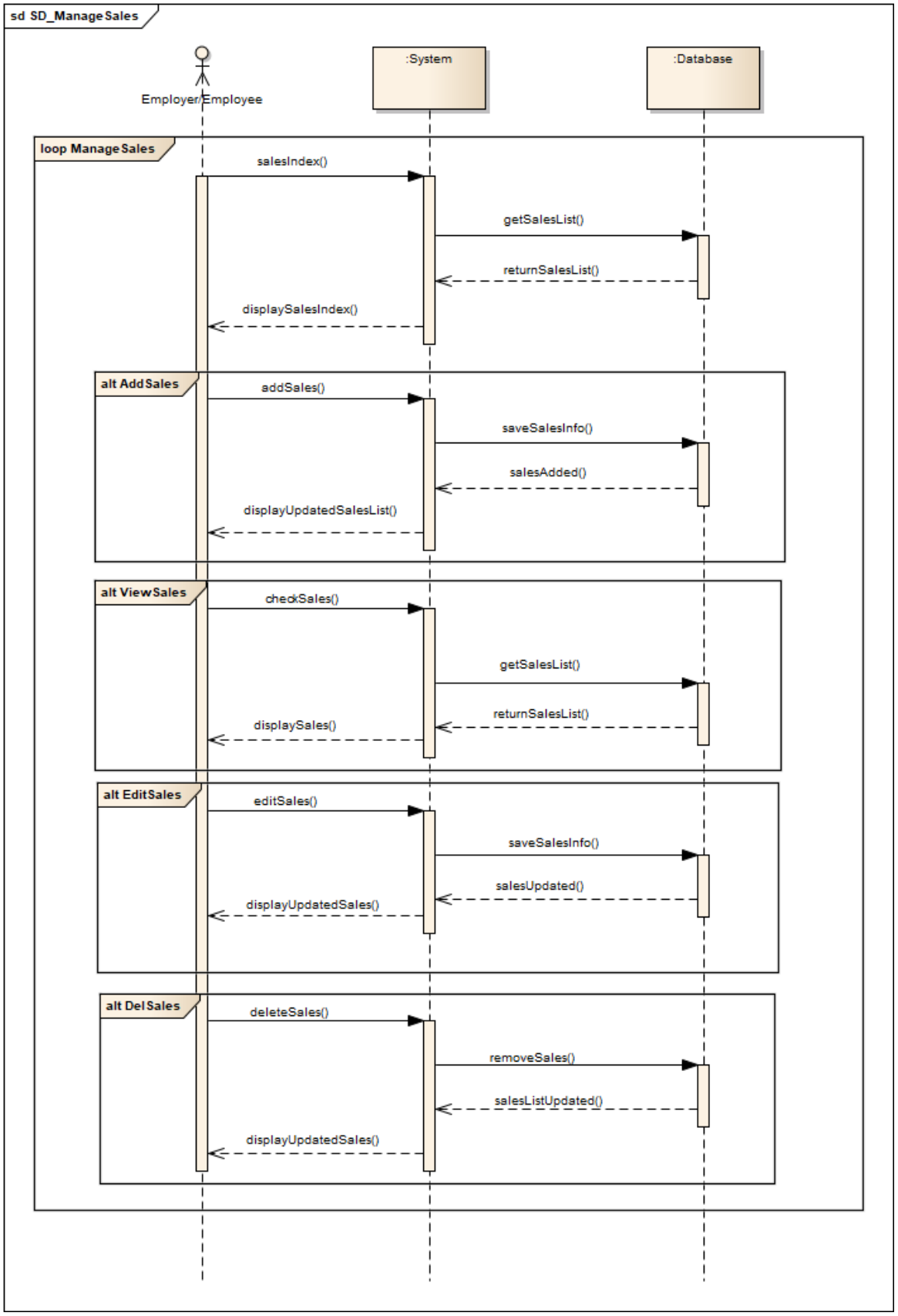

Figure 28: Sequence Diagram for Manage Vending Machine Sales

2.3 **Performance and Other Requirements**

Functional Requirements

Employer

Function 1<Create Stocks>

The employer shall be able to create new stock types if the stock does not exist on the system.

Function 2<Update Stocks>

The employer shall be able to update existing stock on the system.

Function 3<Delete Stocks>

The employer shall be able to delete existing stock on the system.

Function 4<View Stocks>

The employer shall be able to view existing stock information on the system.

Function 5<Create Vending Machine Sales>

The employer shall be able to create new sales for a specific vending machine on the system.

Function 6<Update Vending Machine Sales>

The employer shall be able to update existing sales for a specific vending machine on the system.

Function 7<Delete Vending Machine Sales>

The employer shall be able to delete existing sales for a specific vending machine on the system.

Function 8<View Vending Machine Sales>

The employer shall be able to view existing sales for a specific vending machine on the system.

Function 9<Create User>

The employer shall be able to create new users with proper credentials on the system.

Function 10<Update User>

The employer shall be able to update existing user information on the system.

Function 11<Delete User>

The employer shall be able to delete existing users on the system.

Function 12<View User>

The employer shall be able to view the user list on the system.

Function 13<View Dashboard>

The employer shall be able to view insights regarding business on the system.

Function 14<Create Report>

The employer shall be able to generate monthly and annual reports as various file types on the system.

Function 15<View Report>

The employer shall be able to view monthly and annual reports once it is generated on the system.

Employee

Function 1<Create Stocks>

The employee shall be able to create new stock types if the stock does not exist on the system.

Function 2<Update Stocks>

The employee shall be able to update existing stock on the system.

Function 3<Delete Stocks>

The employee shall be able to delete existing stock on the system.

Function 4<View Stocks>

The employee shall be able to view existing stock information on the system.

Function 5<Create Vending Machine Sales>

The employee shall be able to create new sales for a specific vending machine on the system.

Function 6<Update Vending Machine Sales>

The employee shall be able to update existing sales for a specific vending machine on the system.

Function 7<Delete Vending Machine Sales>

The employee shall be able to delete existing sales for a specific vending machine on the system.

Function 8<View Vending Machine Sales>

The employee shall be able to view existing sales for a specific vending machine on the system.

Non functional Requirements

NFR1 External Requirements- Security Requirements

The system should always ensure that all the data inside the system including password, personal user details are always in a secure condition against malware attacks or any unauthorized access

NFR2 Product Requirements - Efficiency Requirements

The system should calculate revenue, profit and generate a report daily ,weekly or even monthly.

NFR3 Product Requirements - Usability Requirements

The system should be able to notify the system admin when the stock in the vending machine is low in quantity.

NRF4 Product Requirements- Performance Requirements

The system should be able to generate a sales report when it is requested within a few seconds.

NFR5 Organizational Requirements- Operational Requirements

This system shall ensure that admin can place orders of stocks when the quantity of stocks is below the minimum requirements of stocks in order to ensure a better cash flow.

NFR6 Organization Requirements - Development Requirements

The system shall never allow the employees to view the report profit/loss generated by the system.

2.4 **Design Constraints**

1. This will be a web-based application built with the.NET framework.
2. Only registered users have access to this programme (admin, teacher or student).
3. Users will be authenticated using a unique username and password when they log in.
4. Regardless of user type, only users with an "active" status can log in to the system.
5. Users will be verified using a session of their user type upon login, allowing them to distinguish between the interfaces, pages, and functions available to them.

2.5 **Software System Attributes**

1. Functional Suitability

When employed under certain conditions, functional appropriateness is the degree to which a product or system performs functions that meet the stated or implicit requirements. Our System Module can fulfill the client's needs in every aspect of business operations.

2. Reliability

The likelihood that a software system will fulfill a function (given by the specifications) over a certain number of input trials under specified input conditions in a particular time frame is defined as reliability (assuming that hardware and input are free of errors). Create, Read, Update, Delete, Edit (CRUDE) Operations in our Manage Stock,Manage Vending Machine Sales, Manage User, Manage Stock History, Manage Report and Dashboard working well all the time. The Financial calculation is done based on the client's request.

3. Operability

The features of a system that make it work successfully in production are known as operability. Our system is not only providing reliable end-user functionality, but also operates well from the operations team's standpoint. Our system is easy to prepare, operate and interpret inputs. Our interface standard is high and consistent with all the modules as well as any new users can learn to use our system with minimal effort.

4. **Interoperability**

Our system is simple to communicate data or services with other systems thanks to interoperability. All the modules and operational functions are compatible with various operating systems, databases, and protocol circumstances.

5. **Reusability**

Software reuse is an excellent way to save money and effort when developing software. Various code library classes in our system are generic enough to be used in a variety of application modules. We have divided the programme into modules so that modules can be utilized throughout the project.

6. **Testability**

It is simple to test and detect flaws in our system. We can also break our system into different modules for testing if necessary.

7. **Integrity and Security**

Integrity goes hand in hand with safety. Our System integrity or security is sufficient to prevent unauthorized access to system functions, data loss, and virus infection of software, as well as to protect the privacy of data entered the system.

8. **Maintainability**

Lastly, our system is simple to maintain. It is easy to add code to our system in future. We can upgrade our system easily with new features and technologies as they become available.