

KING SAUD UNIVERSITY
COLLEGE OF COMPUTER AND INFORMATION SCIENCES
DEPARTMENT OF SOFTWARE ENGINEERING



كلية علوم الحاسب والمعلومات
قسم هندسة البرمجيات

SWE312: Software Requirements Engineering

Project

Course Code / Title: SWE 312

Assessment: Project

Semester / Year: Spring 2023

Submission Date: 4/5/23

Deliverable

Duration:

STUDENT OUTCOMES:

This phase covers/targets the following students outcomes.

Project Tasks	Relevant SOs/Clos	Marks
Functional, non-functional requirements, and design constraints	SO(2)	
Use cases, Use case description	SO(2)	
Use of SWE case tool	SO(2)	
Report organization, ethical and professional issues		
Teamwork and work distribution, Introduction		

STUDENT IDENTIFICATION:

Section: 37680 **Group Number:** Team 4

Name: Suliman Ibrahim Alasim **ID:** 442101493 **Signature:**.....

Name: Khalid Abdullah Alharbi **ID:** 442103477 **Signature:**.....

Name: Abdullah Abdulkarim Al-Dakhil **ID:** : 442106013 **Signature:**.....

Name: Sultan Hammod Al-enzi **ID:** 442106994. **Signature:**.....

Name: Abdulmajeed Mohammed Al-Romaih **ID:** 442101425 **Signature:**.....

By signing this form, the student recognizes that he understands and accepts the exam policy and ethics. He/she recognizes also that if he/she does not respect these ethical rules, the professor will take the appropriate measures including exclusion from the exam.

Payroll Management System

Software Requirements Specification

Final Version

31/5/23

Khalid Abdullah Alharbi

Lead Software Engineer

Prepared for
SWE 312 - Software Requirements Engineering
Spring 2023

Revision History

Date	Description	Author	Comments
<date>	<Version 1>	<Your Name>	<First Revision>

Document Approval

The following Software Requirements Specification has been accepted and approved by the following:

Signature	Printed Name	Title	Date
	Khalid Abdullah Alharbi	Lead Software Eng.	31/5/23
	Suliman Ibrahim Alasim	Software Eng.	31/5/23
	Sultan Hammod Al-enzi	Software Eng.	31/5/23
	Abdulmajeed Mohammed Alromaih	Software Eng.	31/5/23
	Abdullah Abdulkarim Al-Dakhil	Software Eng.	31/5/23

Table of Contents

Revision History	3
Document Approval	3
1. Introduction	6
1.1 Purpose	6
1.2 Scope	6
1.3 Definitions, Acronyms, and Abbreviations	6
1.4 References	7
1.5 Overview	7
2. General Description	7
2.1 Product Functions	7
2.2 User Characteristics	8
3. Specific Requirements	8
3.1 External Interface Requirements	8
3.1.1 System shall interface with a bank system.	8
3.2 Functional Requirements	8
3.2.1 The employee shall be able to login to the system.	8
3.2.2 The Administrator shall be able to login to the system.	8
3.2.3 The employee shall be able to enter a timecard.	8
3.2.4 The performance employee shall be able to add the purchase order.	8
3.2.5 The employee shall be able to modify his timecard.	8
3.2.6 The employee shall be able to choose the payment method.	8
3.2.7 The system shall display the current timecard to employee.	9
3.2.8 The system shall create a new timecard if a timecard does not exist during the pay period.	9
3.2.9 The performance employee shall be able to change the purchase order.	9
3.2.10 The performance employee shall be able to cancel the purchase order.	9
3.2.11 The system shall set the start and end date of the timecard.	9
3.2.12 The system shall calculate the commission rate for the performance employee based on sales.	9
3.2.13 The system shall assign a unique order number to the purchase order.	9
3.2.14 The employee shall be able to create a report.	10
3.2.15 The administrator shall be able to add a new employee.	10

3.2.16 The employee shall be able to print the created report.	10
3.2.17 The administrator shall be able to delete an employee.	10
3.2.18 The administrator shall be able to change employee information.	10
3.2.19 The administrator shall be able to request administrator report.	11
3.3.2 Table of use cases	13
3.4.1 Use Case #1	14
3.4.2 Use Case #2	15
3.4.3 Use Case #3	16
3.4.4 Use Case #4	17
3.4.5 Use Case #5	18
3.6 Non-Functional Requirements	19
3.6.1 Performance	19
3.6.2 Reliability	19
3.6.3 Availability	19
3.6.4 Security	19
3.6.5 Maintainability	19
3.6.6 Portability	19
3.7 Design Constraints	19
3.8 Other Requirements	19
5 Classes / Objects	28
A. Appendices	31
A.1 Appendix 1	31
A.2 Appendix 2	31

1. Introduction

1.1 Purpose

The purpose of this SRS is to make the payroll management system requirement easy to understand for developers, and designers, which makes them implement the right requirements.

1.2 Scope

1) Software product to be produced is PMS (Payroll management system).

2) The system will generate paychecks.

The system will allow employees to:

A- record timecard

B- enter purchase order.

C-change preference.

D-create reports.

The system will not allow employees to change others timecard except theirs.

The system will not allow administrators to change employee's payment delivery method.

The administrator is responsible for adding, delete and editing employees.

The performance employee is in control of purchase order.

3) The software will not need any manual intervention, also it will interface with a bank system to transact the paychecks.

1.3 Definitions, Acronyms, and Abbreviations

- **Bank System:** Any bank to which direct deposit transactions are sent.
- **Employee:** A person who works for the company that owns and operates the payroll system (SuperTech, Inc.)
- **Payroll Administrator:** The person responsible for maintaining employees and employee information in the system.
- **Work Management Database:** The legacy database that contains all information regarding projects and charge numbers.
- **System Clock:** The internal system clock that keeps track of time. The internal clock will automatically run the payroll at the appropriate times.
- **Pay Period:** The amount of time over which an employee is paid.
- **Paycheck:** A record of how much an employee was paid during a specified Pay

Period.

- **Payment Method:** How the employee is paid, either pick-up, or direct deposit.
- **Timecard:** A record of hours worked by the employee during a specified pay period.
- **Purchase Order:** A record of a sale made by an employee.
- **Salaried Employee:** An employee that receives a salary.
- **Performance Employee:** An employee that receives a salary plus commissions.
- **Hourly Employee:** An employee that is paid by the hour.

1.4 References

- 1- IEEE (10 February 1984) *IEEE Guide for Software Requirement Specification*
<https://ieeexplore.ieee.org/document/278253>
- 2- Software engineering 312 file (payroll management system document)

1.5 Overview

The SRS contains a list of requirements specification.

SRS is organized:

- a- **Introduction:** contains (purpose, scope, definition, references).
- b- **General Description:** contains (product function, user characteristics).
- c- **Specific Requirements:** contains (functional, non-functional, use case, use case description).
- d- **Analysis mode:** contains (sequence diagram, .mockup screens, activity diagram)
- e- **Classes:** contains (class diagram)

2. General Description

2.1 Product Functions

The software will provide many functions to the user. In summary, it will provide a login function for each employee and the administrator. Furthermore, the administrator will be able to add, delete, or edit an employee. And the employee can submit, view, or edit a timecard. Also, he/she can choose the preferred payment method, and for performance, the employee will be able to add, delete, or edit purchased orders. Both the employee and the administrator can generate reports from the system, and the employee will get their paychecks automatically at the appropriate time as delivered by the software.

2.2 User Characteristics

We will have two primary users who use the software: the employee and the administrator. We anticipate that both possess strong technical abilities and have prior experience with similar software.

3. Specific Requirements

3.1 External Interface Requirements

3.1.1 System shall interface with a bank system.

3.1.2 System shall work with the existing Work Management database.

3.2 Functional Requirements

3.2.1 The employee shall be able to login to the system.

3.2.1.1 introduction: none.

3.2.1.2 input: email address and password.

3.2.1.3 output: none

3.2.2 The Administrator shall be able to login to the system.

3.2.2.1 introduction: Since there is ambiguity in the login process for the administrator, we assume that he/she will login using the email address and password.

3.2.2.2 input: email address and password.

3.2.2.3 output: none.

3.2.3 The employee shall be able to enter a timecard.

3.2.3.1 introduction: none

3.2.3.2 input: number of hours worked.

3.2.3.3 output: none.

3.2.4 The performance employee shall be able to add the purchase order.

3.2.4.1 introduction: none

3.2.4.2 input: purchase date, purchase products, amount of sale, customer point of contact, customer billing address.

3.2.4.3 output: none.

3.2.5 The employee shall be able to modify his timecard.

3.2.5.1 introduction: the modification must be before submission and in the current pay period.

3.2.5.2 input: number of hours worked.

3.2.5.3 output: none.

3.2.6 The employee shall be able to choose the payment method.

3.2.6.1 introduction: There are two types of payment method (direct deposit / direct pickup at the office).

3.2.6.2 input: the chosen payment method.

3.2.6.3 output: none.

3.2.7 The system shall display the current timecard to employees.

3.2.7.1 introduction: System will retrieve the timecard for the current pay period.

3.2.7.2 input: none.

3.2.7.3 output: number of hours worked and date.

3.2.8 The system shall create a new timecard if a timecard does not exist during the pay period.

3.2.8.1 introduction: the creation of a new timecard will occur if the employee doesn't have one.

3.2.8.2 input: none.

3.2.8.3 output: Timecard.

3.2.9 The performance employee shall be able to change the purchase order.

3.2.9.1 introduction: none

3.2.9.2 input: unique purchase order number, date, purchased products, amount of the sale, customer point of contact, customer billing address.

3.2.9.3 output: none.

3.2.10 The performance employee shall be able to cancel the purchase order.

3.2.10.1 introduction: none.

3.2.10.2 input: unique purchase order number.

3.2.10.3 output: none.

3.2.11 The system shall set the start and end date of the timecard.

3.2.11.1 introduction: The start and end dates of the timecard are set by the system and cannot be changed by the employees.

3.2.11.2 input: none.

3.2.11.3 output: none.

3.2.12 The system shall calculate the commission rate for the performance employee based on sales.

3.2.12.1 introduction: The system calculates the commission rate from a range of (10%,15%,25%,35%).

3.2.12.2 input: none.

3.2.12.3 output: Commission rate.

3.2.13 The system shall assign a unique order number to the purchase order.

3.2.13.1 introduction: The Unique purchase order number used for two things:

A- edit purchase order.

B- cancel purchase order.

3.2.13.2 input: none.

3.2.13.3 output: a unique purchase order number.

3.2.14 The employee shall be able to create a report.

3.2.14.1 Introduction: The report has four types, and employees shall be able to request one of these types:

- A- Total hours worked.
- B- Total hours worked for project.
- C- The total pay received year-to-date.
- D- remaining vacation time (vacation / sick time).

to create these reports the Employees need to indicate the type of the report, begin and end date of the report.

3.2.14.2 Inputs:

- A- Type of report
- B- Begin and end date of the report.

3.2.14.3 Outputs: report on specified criteria.

3.2.15 The administrator shall be able to add a new employee.

3.2.15.1 introduction: The administrator must login to the system if he wants to add a new employee.

3.2.15.2 input: name, social security number, phone number, payment classification (hourly, salaried, performance), salary (for salaried and performance employees), hourly rate (for hourly employees), tax deductions.

3.2.15.3 output: none.

3.2.16 The employee shall be able to print the created report.

3.2.16.1 introduction: After creating a report the employee can request to print a report.

3.2.16.2 input: none.

3.2.16.3 output: none.

3.2.17 The administrator shall be able to delete an employee.

3.2.17.1 introduction: The administrator will use the social security number to find the desired employee.

3.2.17.2 input: social security number.

3.2.17.3 output: none.

3.2.18 The administrator shall be able to change employee information.

3.2.18.1 introduction: none.

3.2.18.2 input: employee name, social security number, phone number, payment classification (hourly, salaried, performance), salary (for salaried and performance employees), hourly rate (for hourly employees), tax deductions and other deductions.

3.2.18.3 output: none.

3.2.19 The administrator shall be able to request administrator report.

3.2.19.1 Introduction: The report type can be:

A-Total Hours Worked

B-Pay Year-to-Date report.

3.2.19.2 Inputs: The report type, Employee name, Begin and end dates for the report.

3.2.19.3 Outputs: report on specified criteria.

3.2.20 The system shall make the timecard read-only after submitting.

3.2.20.1 introduction: None.

3.2.20.2 input: none.

3.2.20.3 output: none.

3.2.21 The system shall calculate payment for performance employee.

3.2.21.1 introduction: none.

3.2.21.2 input: Timecard, purchase orders, employee information, and all legal deductions.

3.2.21.3 output: a calculated payment.

3.2.22 The system shall generate a paycheck for hourly employees every Saturday.

3.2.22.1 introduction: For the system to pay the hourly employees, they must first submit a timecard.

3.2.22.2 input: none

3.2.22.3 output: none

3.2.23 The system shall generate a paycheck for the performance & salaried employee on the first workday of the next month.

3.2.23.1 introduction: In order for the system to pay the salaried and performance employees they must first submit a timecard.

3.2.23.2 input: none

3.2.23.3 output: none

3.3 Use Cases

3.3.1 System boundary:

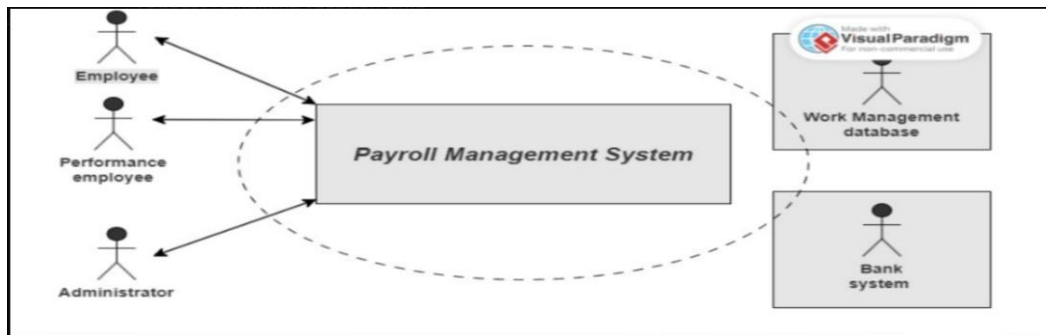


Figure 1: the diagram defines who will interact with the system.

3.3.2 Use case diagram:

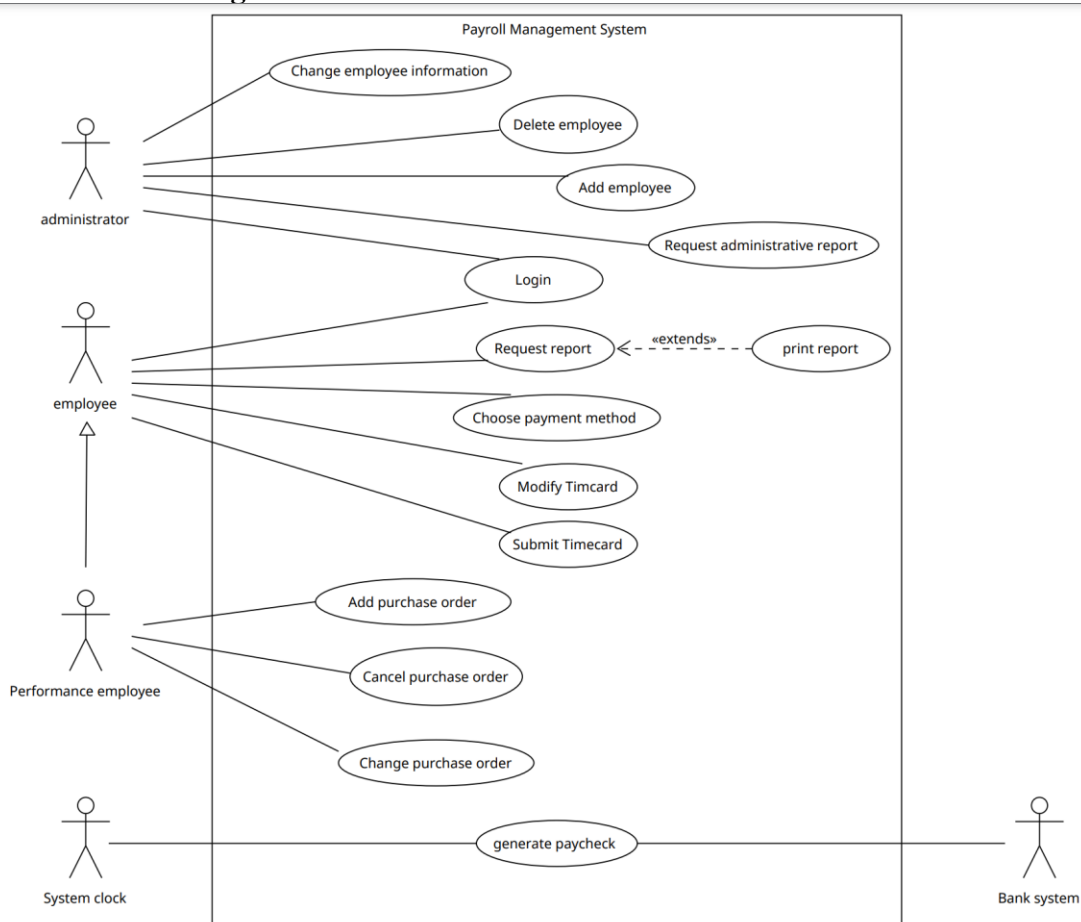


Figure 2: Use case diagram.

3.3.2 Table of use cases

Use case	Actor	Relationships
Login	Administrator, Employee, Performance employee	none
Change employee information	Administrator	none
Delete employee	Administrator	none
Add employee	Administrator	none
Request report	Employee, Performance employee	none
Request administrative report	Administrator	none
Request report	Employee, Performance employee	none
Choose payment method	Employee, Performance employee	none
Submit timecard	Employee, Performance employee	none
Modify timecard	Employee, Performance employee	none
Print report	Employee, Performance employee	Extends (Request employee report) UC
Add purchase order	Performance employee	none
Cancel purchase order	Performance employee	none
Change purchase order	Performance employee	none
generate paycheck	Primary actor: System clock Secondary actor: Bank System	none

3.4 Use Cases Description

Write description for **five use cases**.

3.4.1 Use Case #1

3.4.1.1 Name of Use case: Add employee.

3.4.1.2 Description This use case describes the scenario where the administrator adds an employee to the system.

3.4.1.3 Actor(s): Administrator.

3.4.1.4 Relationships:

3.4.1.5 Precondition: Administrator is logged in.

3.4.1.6 Basic Flow:

Administrator	System
<p>1. The Administrator asks the system to add an employee.</p> <p>3. The Administrator enters the employee information:</p> <ul style="list-style-type: none"> • Name • Social security number • Phone number • Payment classification • Salary • Hourly rate • Commission rate • Tax deductions • Other deductions 	<p>2.The system prompts the employee information (name, social security number, phone number , payment classification (hourly, salaried, performance), salary (for salaried and performance employee), hourly rate (for hourly employee), commission rate (for performance employee), tax deductions , and other deductions).</p> <p>4. The system validates the given information. (* validates means check if the employee has been added before) if the employee hasn't been added before and the administrator filled in all the information correctly.</p> <p>5. the system stores the employee information.</p> <p>6. the system displays a success message to the Administrator.</p>

3.4.1.7 Alternative Flow:

4a. The system finds out that the Administrator has entered an employee information who has been added before via work management database, the administrator enters invalid information.

4a1. The system displays an error message.

4a2. The system redirects the Administrator to step 2.

3.4.2 Use Case #2

3.4.2.1 Name of Use case: Delete employee.

3.4.2.2 Description: This use case describes the scenario where the administrator deletes an employee from the system.

3.4.2.3 Actor(s): Administrator

3.4.2.4 Relationships: none.

3.4.2.5 Precondition: Administrator is logged in.

3.4.2.6 Basic Flow:

Administrator	System
1. the Administrator asks the system to delete an employee.	2. the system prompts employee social security number.
3. the Administrator enters the employee social security number.	
	4. The system validates the given information. (* validates means that the system will check for a match in the database.) the system finds a matched employee.
	5 the system will show the employee information and asks the administrator if he/she wants to delete the employee.
6. the administrator confirms that he/she would delete the employee.	7. the system displays a success message that the employee has been deleted to the administrator

3.4.2.7 Alternative Flow:

4a the system didn't find a match for the desired employee.

4a1. The system displays an error message indicating that there is no employee with the given social security number.

4a2. Flow goes back to step 2.

6a If the administrator does not want to delete the employee.

6a1. Flow goes back to step 2.

3.4.3 Use Case #3

3.4.3.1 Name of Use case: submit timecard.

3.4.3.2 Description: This use case describes the scenario where the employee submits the timecard.

3.4.3.3 Actor(s): Employee, Performance employee

3.4.3.4 Relationships: none.

3.4.3.5 Precondition: Employee is logged in.

3.4.3.6 Basic Flow:

Employee	System
1. the employee asks the system to submit a timecard.	2. the system prompts the timecard information (number of hours worked).
3. the employee enters the timecard information (number of hours worked).	
	4. The system validates the timecard information (*validates means that the system will check if the date is for the current pay period and will not allow the number of hours worked to exceed 8 hours daily).
	5. the system stores the timecard.
	6. the system displays a success message.
	7. the system makes the timecard read-only.

3.4.3.7 Alternative Flow:

4a. The employee submits a timecard that is not for the current pay period, the worked hours exceed 8 hours daily.

4a1. The system displays an error message explaining the cause.

4a2. The system redirects the employee to step 2.

3.4.4 Use Case #4

3.4.4.1 Name of Use case: Add purchase order.

3.4.4.2 Description: This use case describes the scenario where performance employee adds a purchase order.

3.4.4.3 Actor(s): Performance employee

3.4.4.4 Relationships

3.4.4.5 Precondition: Performance employee is logged in.

3.4.4.6 Basic Flow:

Performance employee	System
1. the Performance employee asks the system to add a purchase order.	2. the system prompts the purchase order information (date, purchased products, amount of sale, customer point of contact, customer billing address) .
3. the Performance employee enters the purchase order information: <ul style="list-style-type: none"> • Date • The purchased products • Amount of sale • Customer point of contact • Customer billing address 	4. the system validates the purchase order information; it will check if the employee has completed the information 5. the system assigns a unique purchase order number to the purchase order. 6. the system stores the purchase order. 7. the system displays a success message to the performance employee.

3.4.4.7 Alternative Flow:

4a. the performance employee has entered invalid purchase order information, the performance employee hasn't completed the purchase order information.

4a1. system displays an error message explaining the cause.

4a2. Flow goes back to step 2.

3.4.5 Use Case #5

3.4.5.1 Name of Use case: Cancel purchase order.

3.4.5.2 Description: This use case describes the scenario where the performance employee cancels a purchase order.

3.4.5.3 Actor(s): performance employee

3.4.5.4 Relationships: none.

3.4.5.5 Precondition: Performance employee is logged in.

3.4.5.6 Basic Flow:

Performance employee	System
1. The performance employee asks the system to cancel a purchase order.	2. The system prompts for a purchase order number.
3. The performance employee enters the purchase order number.	4. The system validates the purchase order number, (* validates means that it will check if there is a purchase order assigned to the entered purchase order number on the database), if there are a purchase order
	5. The system cancels the purchase order
	6. The system displays a success message to the performance employee.

3.4.5.7 Alternative Flow:

4a the performance employee has not completed the purchase order number, there is no match for the purchase order number.

4a1. System displays an error message showing the cause.

4a2 System redirects the performance employee to step 2.

3.6 Non-Functional Requirements

3.6.1 Performance

3.6.1.1 The system shall accommodate 7,000 employees worldwide.

3.6.2 Reliability

3.6.2.1 The system shall pay appropriate employees on those days (hourly employees every Saturday, salaried **employees** on the first workday of the month).

3.6.3 Availability

3.6.3.1 The system shall be available every Saturday and on the first working day of the month.

3.6.3.2 The system shall be available 99% of the time.

3.6.4 Security

3.6.4.1 The system shall prevent employees from changing any timecard other than their own.

3.6.4.2 The system shall grant access to the administrator to change the employee information with the exception of the payment delivery method.

3.6.5 Maintainability

3.6.6 Portability

3.6.6.1 the system shall run on all employee phones and tablets.

3.7 Design Constraints

3.7.1 The system shall have a mobile phone interface.

3.7.2 The system shall work with the existing (Work Management Database) which is an Oracle database .

3.8 Other Requirements

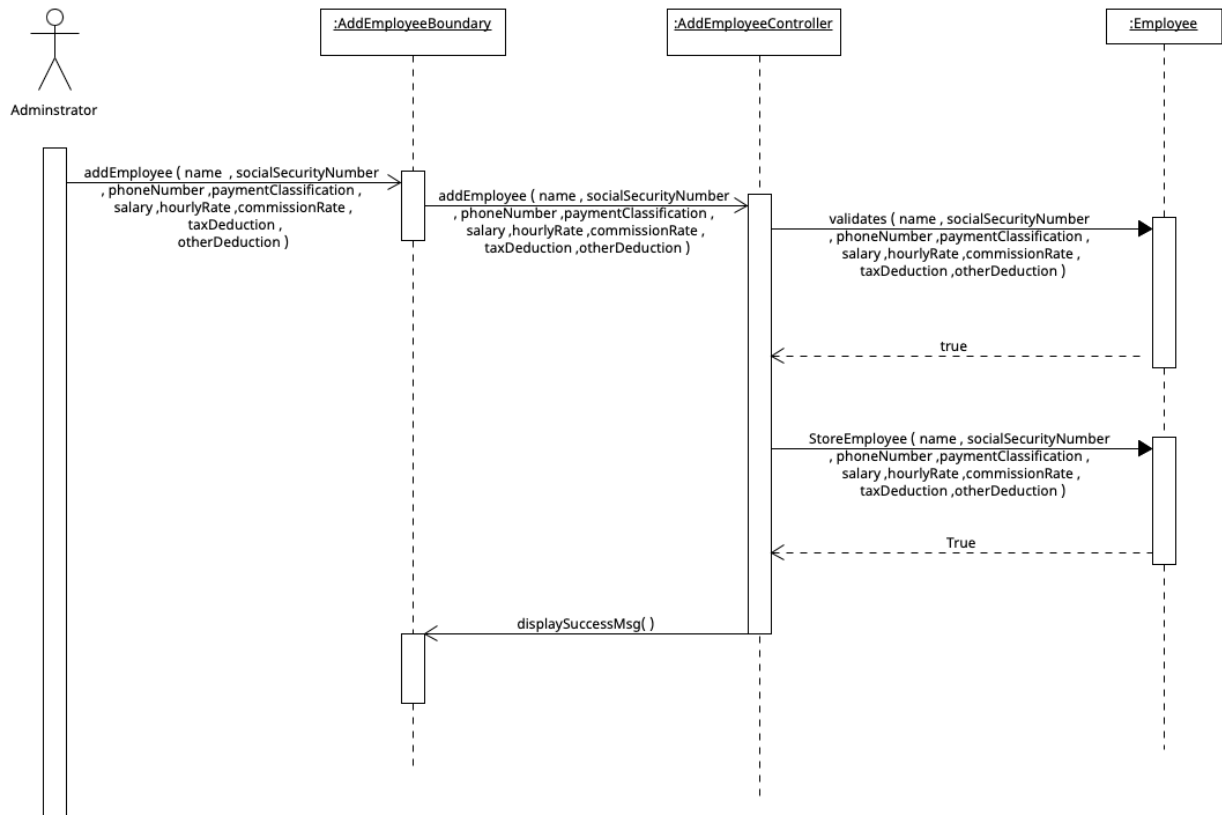
/ none /

4. Analysis Models.

4.1 Sequence Diagrams and Mockup Screens

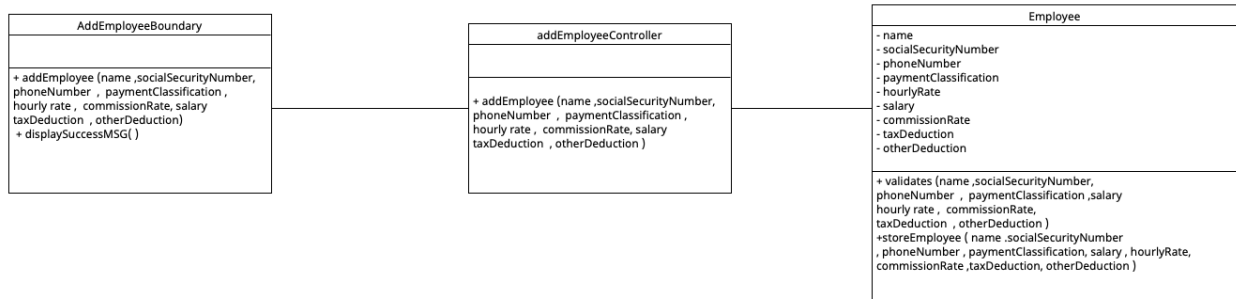
Sequence Diagram : is a UML Dynamic model diagram that is used to show how the object model elements interact with each other to produce a functionality .

4.1.1 Sequence Diagram for (Add Employee) UC:



4.1.1.1 (VOPC):

View of participating classes



4.1.1.2 Mockup screens (add employee):

Add Employee window

Name : Salary :

Social security number : Commission rate :

Phone number: Tax deduction :

Payment classification : other deduction :

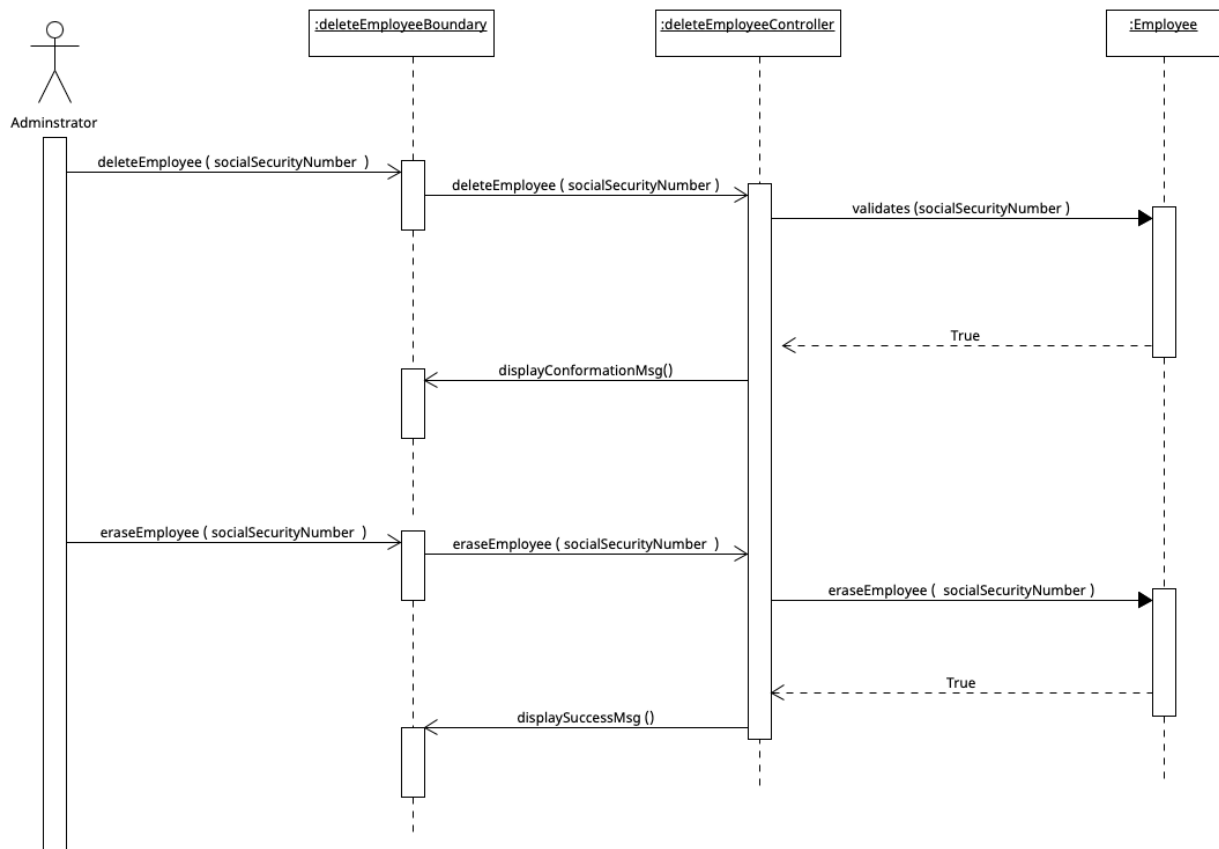
CANCEL **SUBMIT**

Success Message

the employee has been added successfully.

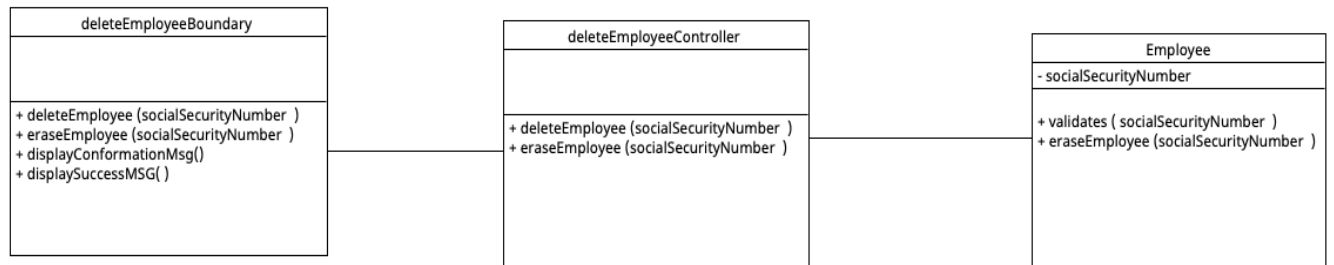
OK

4.1.2 Sequence Diagram for (Delete Employee) UC:



4.1.2.1 (VOPC):

View of participating classes



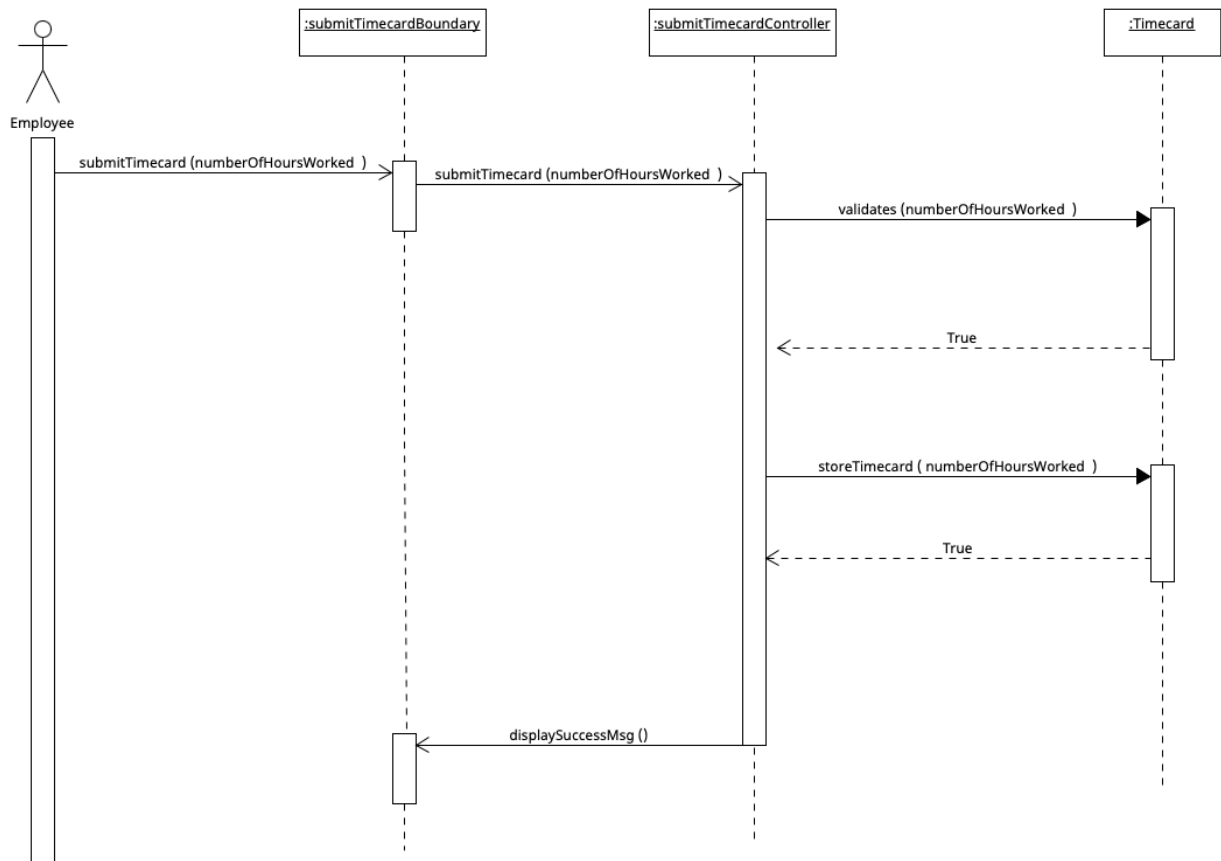
4.1.2.2 Mockup screens (delete employee):

The mockup shows a window titled "Delete Employee". Inside, there is a label "Employee Social Security Number :" followed by a text input field. At the bottom left is a red "CANCEL" button, and at the bottom right is a blue "SUBMIT" button.

The mockup shows a "Success Message" dialog box with a close button (X) in the top right corner. The message text reads: "The employee has been deleted successfully". At the bottom right of the dialog is a blue "OK" button. Below the dialog, in the main window, are red "CANCEL" and blue "SUBMIT" buttons.

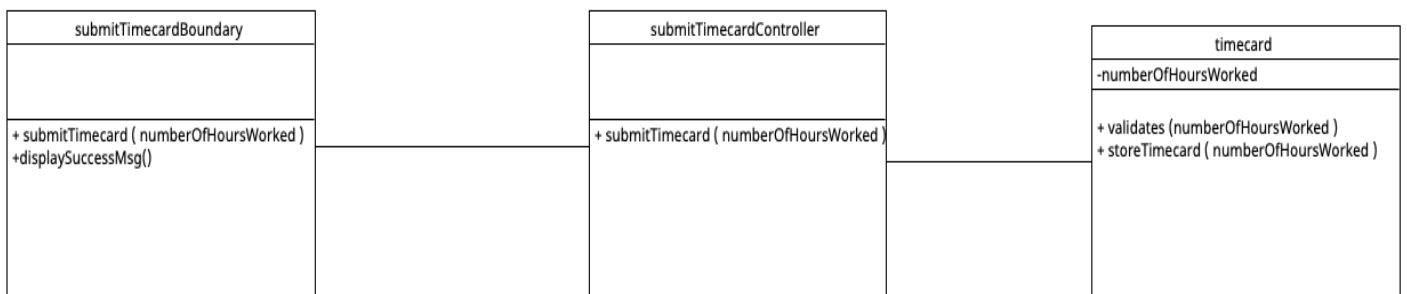
The mockup shows the "Delete Employee" window with a confirmation dialog box open. The dialog box is titled "conformation" and contains the text: "Are you sure want to delete the employee with the following social security number : 356728299111". At the bottom of the dialog are "Cancel" and "yes" buttons. In the background window, the "CANCEL" button is red and the "SUBMIT" button is blue.

4.1.3 Sequence Diagram for (Submit Timecard) UC:



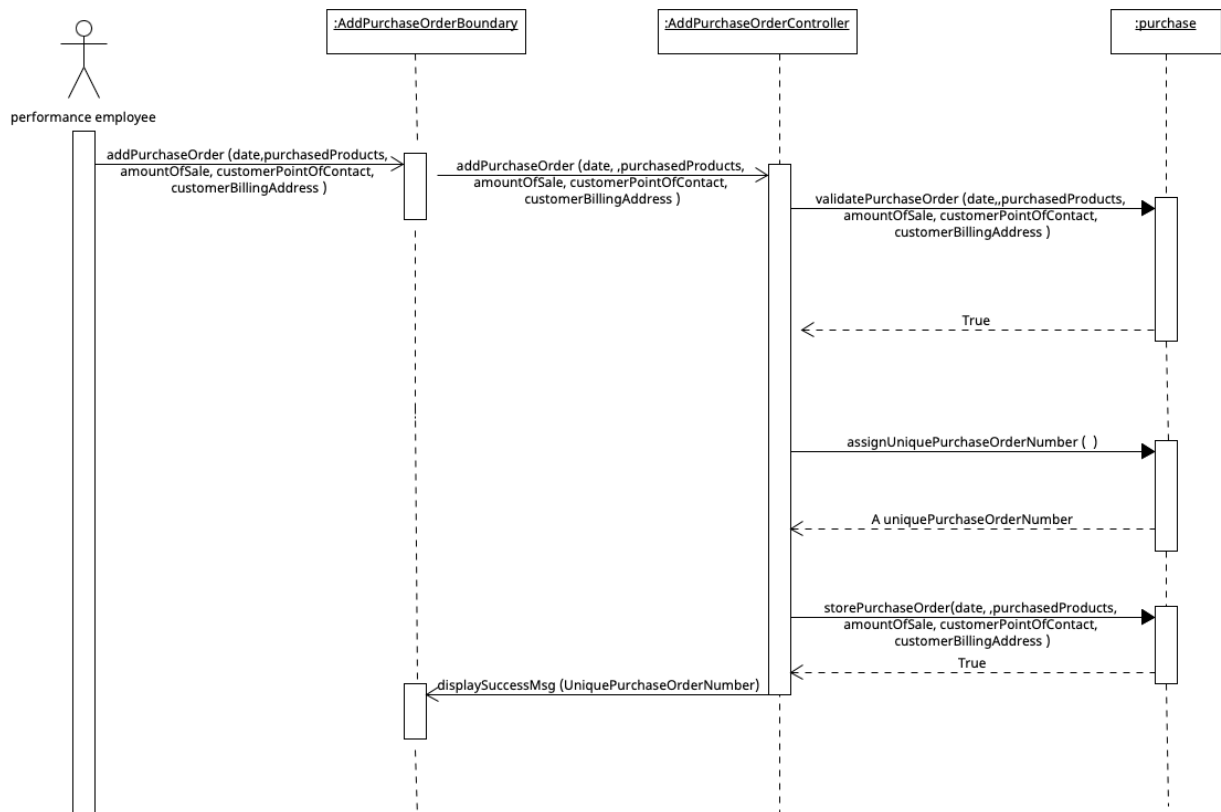
4.1.3.1 (VOPC):

View of participating classes



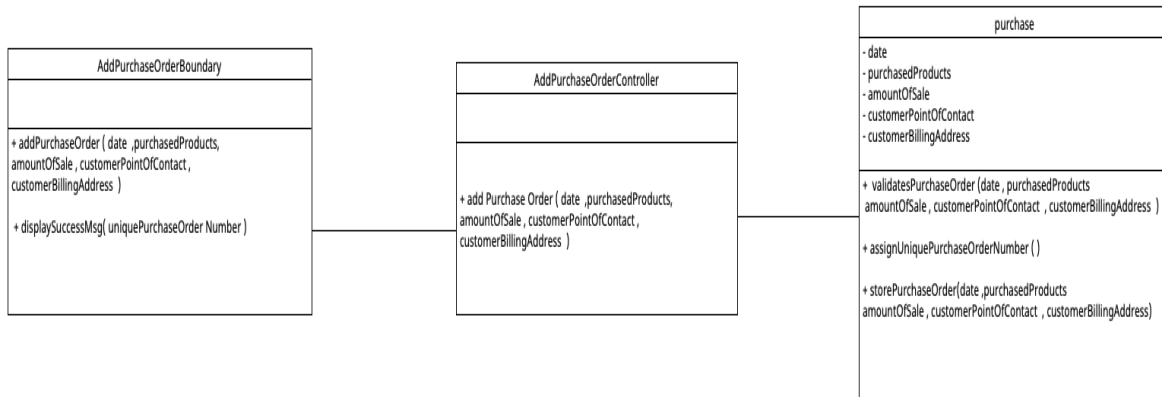
4.1.3.2 Mockup screens (submit timecard):

4.1.4 Sequence Diagram for (Add purchase order) UC:



4.1.4.1 (VOPC):

View of participating classes



4.1.4.2 Mockup screens:

The mockup displays two screens for the 'Add a purchase order' application:

Main Form:

- Title:** Add a purchase order
- Date:** 12 May 2016 (with a calendar icon)
- Purchased Products:** Select (dropdown menu)
- Calendar:** A calendar for January 2017, showing days from 1 to 28.
- point of contact:** Select (dropdown menu)
- Customer phone number:** (966+) (text input)
- Customer Biling address:** (text input)
- Buttons:** CANCEL (red) and SUBMIT (blue)

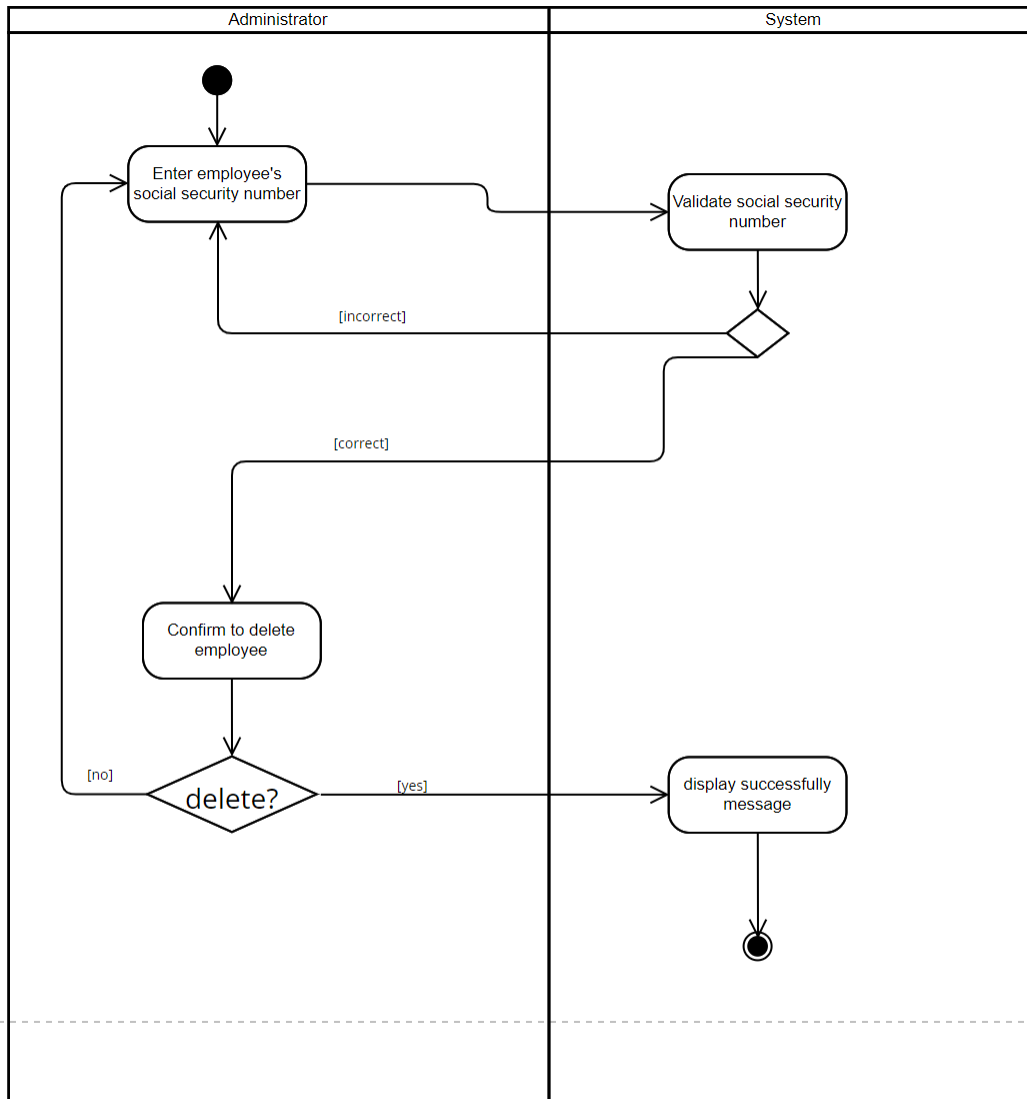
Success Message Dialog:

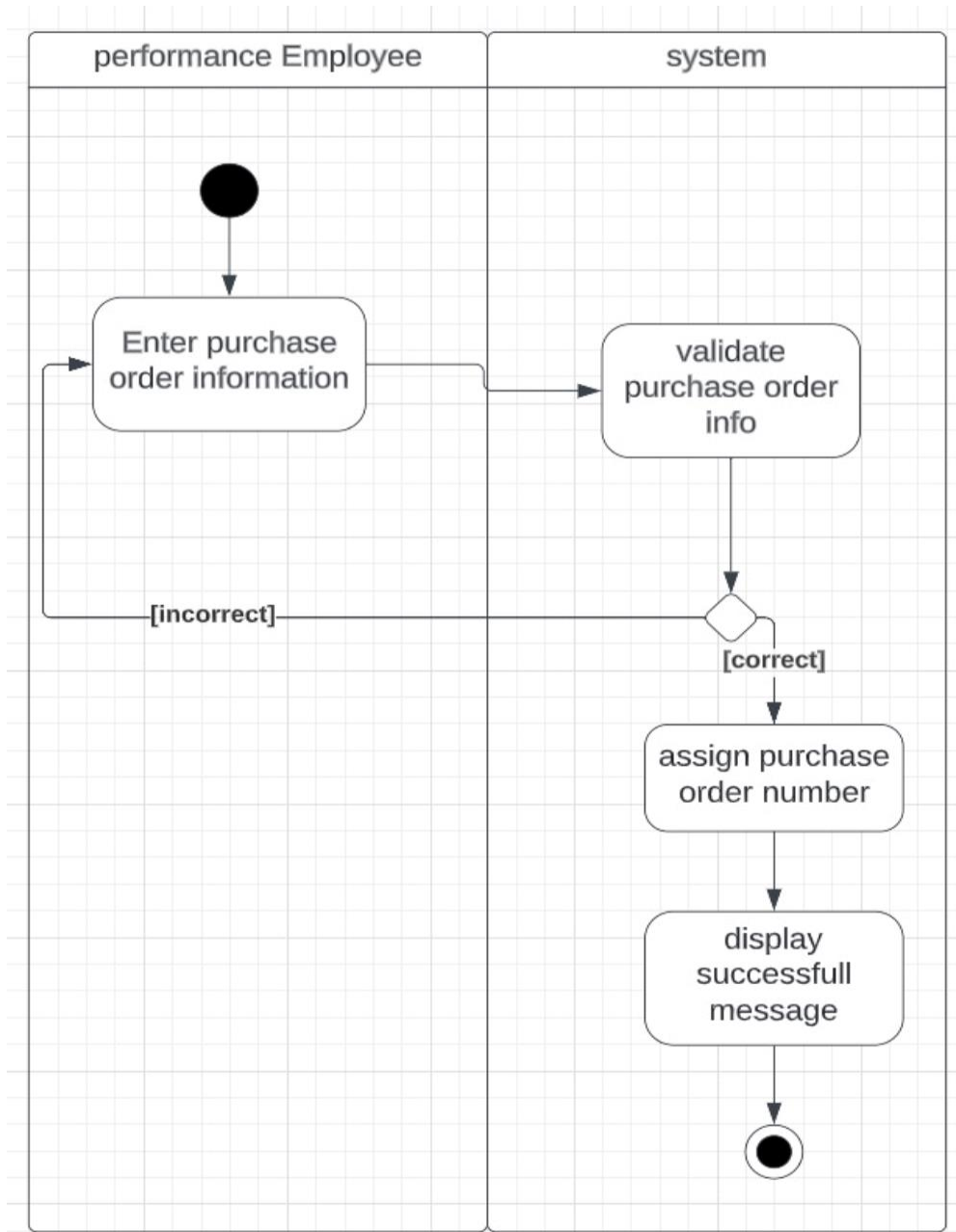
- Title:** Success Message
- Message:** Order has been added
- Buttons:** OK (blue), CANCEL (red), and SUBMIT (blue)

4.2 Activity Diagrams :-

Activity diagram : is a UML diagram used to show the flow control within the system.

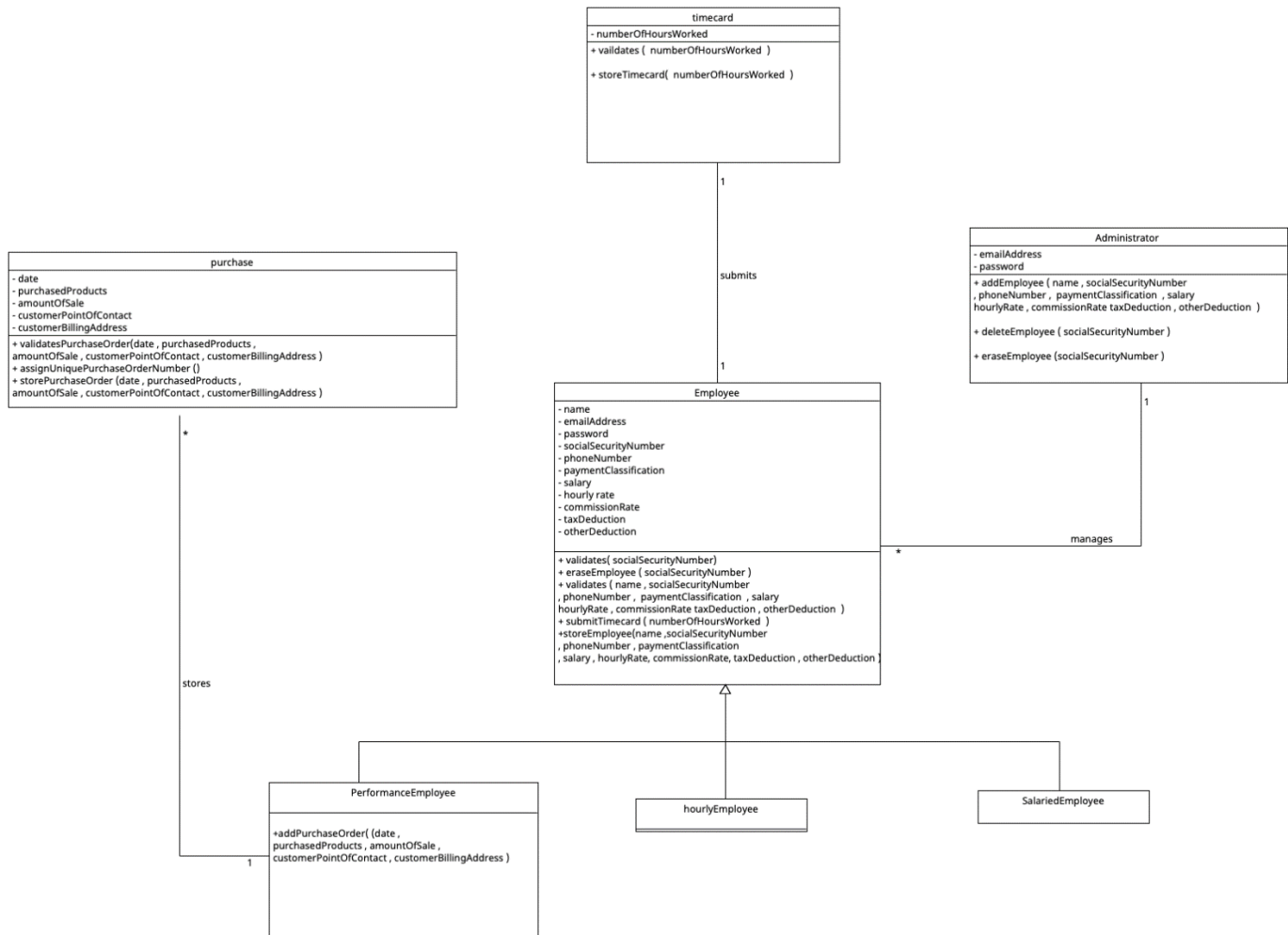
4.2.1 Activity diagram for (delete employee) UC :



4.2.2 Activity diagram for (add purchase order) UC:

5 Classes / Objects

5.1 Class diagram ; is a UML diagram used to present the structure of the system in terms of classes and objects .



5.2 Employee class:

5.2.1 Attributes _:-

name , emailAddress , socialSecurityNumber, password , paymentClassification , hourlyRate , salary , commissionRate , taxDeduction , otherDeduction

5.2.2 Functions:-

1- validates (socialSecurityNumber) reference (delete employee) UC

2- eraseEmployee (socialSecurityNumber) reference (delete employee) UC

3-validates (name , socialSecurityNumber, phoneNumber , paymentClassification , hourlyRate salary , commissionrRate , taxDeduction , otherDeduction) reference (add employee) UC

4- submitTimecard(numberOfhoursWorked) reference (submit timecard) UC

5.3 Administrator Class:

5.3.1 Attributes _:-

email address , password

5.3. 2 Functions:-

1- addEmployee(name , socialSecurityNumber, phoneNumber , paymentClassification , hourly-Rate salary , commissionrRate , taxDeduction , otherDeduction) reference (add employee) UC

2- deleteEmployee (socialSecurityNumber) reference (delete employee) UC

3- eraseEmployee (socialSecurityNumber) reference (delete employee) UC

5.4 Timecard Class:

5.4.1 Attributes _:-

numberOfhoursWorked

5.4. 2 Functions:-

1- validates (numberOfhoursWorked) reference (submit timecard) UC

2- storeTimecard (numberOfhoursWorked) reference (submit timecard) UC

5.5. Purchase Class:

5.5.1 Attributes:-

date , purchasedProducts , amountOfSale , customerPointOfContact ,customerBillingAddress

5.5. 2 Functions:-

1- validatesPurchaseOrder (date , purchasedProducts , amountOfSale , customerPointOfContact ,customerBillingAddress) reference (add purchase order) UC

2- assignUniquePurchaseOrder Number () reference (add purchase order) UC

3- storePurchaseOrder (date , purchasedProducts , amountOfSale , customerPointOfContact ,customerBillingAddress) reference (add purchase order) UC .

5.6 Performance Employee Class:

5.6.1 Attributes _:-

inherited attributes from the parent class

5.6. 2 Functions:-

addPurchaseOrder date , purchasedProducts , amountOfSale , customerPointOfContact ,customerBillingAddress) reference (add purchase order) UC

5.7 Hourly Employee Class:

5.7.1 Attributes _: - inherited attributes from the parent class

5.7. 2 Functions:- No Functions

5.8 Salaried Employee Class:

5.8.1 Attributes _: - inherited attributes from the parent class

5.8. 2 Functions:- No Functions

A. Appendices

Appendices may be used to provide additional (and hopefully helpful) information. If present, the SRS should explicitly state whether the information contained within an appendix is to be considered as a part of the SRS's overall set of requirements.

Example Appendices could include (initial) conceptual documents for the software project, marketing materials, minutes of meetings, etc.

A.1 Appendix 1

A.2 Appendix 2