KING SAUD UNIVERSITY

COLLEGE OF COMPUTER AND INFORMATION SCIENCES
DEPARTMENT OF SOFTWARE ENGINEERING

SWE312: Software Requirements Engineering

Project

Course Code / Title: SWE 312
Assessment: Project **Deliverable**
Semester / Year: Spring 2023 **Submission Date**: **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: 32735 – 88365 **Group Number:** 14

Name: Abdurrahman Alzomea	**ID**: 444101078	**Signature:**…………………
Name: Sulaiman Mokhaniq	**ID:** 444100089	**Signature:…………………**
Name: Abdullah Alyousef	**ID:** 444102160	**Signature:…………………**
Name: Waleed alharbi	**ID:** 444102405	**Signature:…………………**
Name: Abdullah Al ammar	**ID:** 444100076	**Signature:…………………**
Name: Abdulmalik Alzeer	**ID:** 444100765	**Signature:…………………**

.

Payroll Management System

Software Requirements Specification

Version 1

24/10/2024

Group 14
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
	<Your Name>	Lead Software Eng.	

Table of Contents

1. Introduction

1.1 Purpose

The purpose of this SRS document is to provide an extensive description about the Payroll Management System to be developed for PayNow, Inc, to replace the existing system. This document outlines the functional and non-functional requirements, features of the system, use cases, and constrains important for the design, implementation, and maintenance of the new payroll system. It serves as a fundamental reference for software engineers, stakeholders, and project manager to gain a clear understanding of the new system's objectives and functionalities.

1.2 Scope

Software Products
The Payroll Management System will produce the following software products:

- Mobile Application
- Payroll Processing Module
- Employee Reporting
- Administrative Controls

Software Functionality Will Do:

- Mobile application: An interface for employees to enter and manage their timecard 's information's, purchase orders, change preferences (such as payment method), and generate reports on their mobile devices (phones and tablets).

- Payroll Processing Module: : Automated system responsible for calculating and generating paychecks based on entered timecards, sales performance, and applicable deductions.

- Employee Reporting: A tool for generating detailed reports related to hours worked, pay received, remaining vacation time, project data, and other relevant information's.

- Administrator Controls: A way for Payroll Administrators to manage employee information, manage employee data, run reports, and oversee system operations.

Application of the Software:

- Mobile application: provides Accessibility by Enabling employees to manage their payroll-related information anytime and anywhere through a mobile interface, increasing convenience and user satisfaction.

- Payroll Processing Module: provides Efficiency by Automating payroll calculations and paycheck generation to reduce processing time and eliminate manual errors. And provides accuracy by Ensuring precise calculations of pay based on hours worked, sales performance, and applicable deductions, minimizing mistakes.

• Employee Reporting: achieves Reporting by providing comprehensive and customizable reporting tools that allow employees and administrators to generate detailed insights into payroll data.

• Administrative Controls: provides Reporting and efficiency by allowing the administrator to generate reports and manage employee's information

1.3 Definitions, Acronyms, and Abbreviations

- PayNow, Inc.: The company for which the Payroll Management System is being developed.
- MySQL: An open-source relational database management system.
- Commission Rate: The percentage of sales that a performance employee earns as commission.
- Payroll Administrator: The individual responsible for managing employee information and overseeing the payroll system.
- Pay Period: is the amount of time over which an employee is paid.
- Timecard: A record of the hours worked by an employee during a pay period
- Performance Employee: An employee that receives a salary plus commissions.
- Salaried Employee: An employee that receives a salary.
- Hourly Employee: An employee that is paid by the hour.
- Paycheck: A record of how much an employee was paid during a specified Pay Period.
- Employee: A person who works for the company that owns and operates the payroll system (PayNow, Inc.)

1.4 References

- The SW312 Project document: system description: Payroll Management System.
- The IEEE Guide to Software Requirements Specification (ANSI/IEEE Std. 830-1984) Used in the SRS file.

1.5 Overview

This SRS document is organized into the following sections:
Introduction: Provides the purpose, scope, definitions, references, and an overview of the document structure.
General Description: Gives a high-level view of the Payroll Management System, covering its product functions and the characteristics of its users.
Specific Requirements: Outlines detailed requirements, including: External Interface Requirements Functional Requirements Use Cases Non-Functional Requirements Design Constraints Analysis Models: Presents the sequence diagrams, mockups, and activity diagrams used in developing the system requirements.

2. General Description

2.1 Product Functions

The system will allow employees to enter timecard information, submit timecards, change desired method of payment, enter purchase orders for performance employees. The system will allow administrators to add or delete employees, change employee's information. The system will generate paychecks automatically based on hours worked and amount of sales (for performance employees) every Sunday for hourly employees and every first day of the month for salaried employees. The system will also allow employees and administrators to generate various reports based on specified criteria.

2.2 User Characteristics

The Payroll Management System will provide to the following user groups: Employees (Hourly, Salaried, and Performance-based): Employees will use the system to submit timecards, manage purchase orders (for performance-based employees), and view reports such as total hours worked, year-to-date pay, and remaining vacation time. They will vary in technical experience, so the system must be intuitive and easy to navigate, with clear instructions. Payroll Administrator: The Payroll Administrator will have advanced access to managing employee data, including adding, deleting, and updating employee information such as salary, hourly rates, commission rates, and tax deductions. This user will also be able to generate administrative reports.

3. Specific Requirements

3.1 External Interface Requirements

Mobile Interface:
The Mobile Interface shall allow Employees to enter, edit, and submit their timecard information via the mobile interface.
The Mobile Interface shall allow employees to only access and edit their own timecards.
The Mobile Interface shall allow Performance employees to add, modify, and cancel purchase orders through the mobile interface.
The Mobile Interface shall assign a unique purchase order number for each order.
The Mobile Interface shall allow employees to create various reports on total hours worked, hours worked for specific projects, year-to-date pay, and remaining vacation time.
Software Interface:
This Interface shall be linked with the existing MySQL Work Management Database.
This Interface shall Ensure read-only access to prevent modifications to the legacy database.
Banking systems Interface:
This interface shall Integrate with existing banking systems via electronic transactions to complete the deposits of the generated paychecks.

3.2 Functional Requirements

3.2.1 The system shall allow employees to login to the system using the employee's phone number and password.

Comment: none.
Input: phone number and password.
Output: none.

3.2.2 The system shall allow the employee to enter timecard information: (number of hours worked for a particular workload) for the current pay period. Changes can only be made before the timecard is submitted.

Comment: hourly workers cannot enter more than 8 hours per day.
Inputs: number of hours worked.
Outputs: none.

3.2.3 The system shall allow the employee to submit their own timecard. Upon submission, the system shall make the timecard read-only.

Comment:
Input: timecard information:(number of hours worked)**.**
Output: none.

3.2.4 The system shall generate paycheck for Hourly workers every Sunday.

Comment: the system will use the System Clock to determine the time for the payment, Hourly workers must submit their timecard to record date and number of hours worked.
Input:
Output: paycheck.

3.2.5 The System shall generate paycheck for salaried employees on the second working day of the month

Comment: the system will use the System Clock to determine the time for the payment, even though Salaried employees are paid flat salary, they submit their timecard to record date and hours worked.
Input:
Output: paycheck.

3.2.6 The system shall generate paychecks for performance-based employees based on their salary and commission rates.

Comment: the system will use the System Clock to determine the time for the payment, Performance employees submit purchase orders with recorded dates, sales amounts, and product details. The system calculates the commission and adds it to their base salary.

Input:
Output: Paycheck (with salary and calculated commission)

3.2.7 The system shall create a timecard if it doesn't exist for a pay period.

Comment: if the employee hasn't created and submitted his timecard in a pay period, the system shall create a new timecard.
Input: none.
Output: timecard

3.2.8 The system shall assign the start and end dates of the timecard.

Comment: Employees cannot change the start or end date of the timecard
Input: none.
Output: none.

3.2.9 The system shall display current timecard to the employee.

Comment: none
Input: none
Output: timecard .

3.2.10 The system shall allow performance employees to add a purchase order, they must provide (the date, purchased products, amount of sale and the customer billing address).

Comment: none.
Input: date, purchased products, the amount of sale and customer billing address.
Output: none.

3.2.11 The system shall automatically assign a unique purchase order number to each purchase order.

Comment: it can be used later by the performance employees to update their orders' information, as well as to cancel it.
Input: none.
Output: none.

3.2.12 The system shall allow performance employees to change purchase order information (the date, purchased products, amount of sale and the customer billing address).

Comment: using the purchase order number (3.2.11 output) to identify the purchase order.
Input: purchase order number, the new date, new purchased products, new amounts of sale, new customer billing address.
Output: none.

3.2.13 The system shall allow performance employees to cancel purchase order.

Comment: using the purchase order number (3.2.11 output) to identify the purchase order.
Input: purchase order number.
Output: none

3.2.14 The system shall determine the commission rate for each performance employee based on their total sales using one of the following rates: 10%, 15%, 25%, or 35%.

Comment: The exact criteria for determining the commission rate are not specified in the description; we assume the rates are 10% for sales below $5,000, 15% for $5,000–$10,000, 25% for $10,000–$20,000, and 35% for sales above $20,000.
Input: .
Output: none.

3.2.15 The system shall allow employees to request reports, specifying the type of the report (total hours worked- total hours worked for a particular project-total pay received year-to-date- remaining vacation time (vacation days – sick

leave)), information related to each type (hours worked-paychecks received-vacation dates), begin and end date of the report.

Comments: Projects refer to specific work assignments that employees log hours against. The system must retrieve hours worked by employees for the specified project during the requested time period.
Input: type of the report- information related to each type- begin and end dates.
Output: Report.

3.2.16 The system shall allow employee to print the requested report.

Comment: by providing the report type (total hours worked- total hours worked for a particular project- total pay received year-to-date- remaining vacation time), information related to each type (hours worked-paychecks received- vacation dates), begin and end date of the report.
Input: none.
Output: none.

3.2.17 The system shall allow employees to choose their method of payment.

Comment: payment methods are direct deposit (paycheck will be deposited into a specific bank account of the employee's choosing), or directly pick paycheck (from the office).
Input: payment method.
Output: none.

3.2.18 The system shall allow the Administrator to add new employees Administrator inserts the 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..

Comment:
Input: name, social security number, phone number, payment classification, salary or hourly rate, all deductions.
Output: none.

3.2.19 The system shall allow the Administrator to change employee's information Information includes name, social security number, phone

number, salary (for salaried and performance employees), hourly rate (for hourly employees), tax deductions and other deductions.

Comment: It is not specified in the text which information specifically can be changed, but we assume the Administrator can modify an employee's information as listed above. Changes to sensitive fields, such as social security number, must require additional validation.
Input: name, social security number, phone number, salary or hourly rate, all deductions.
Output: none.

3.2.20 The system shall allow the administrator to delete employees.

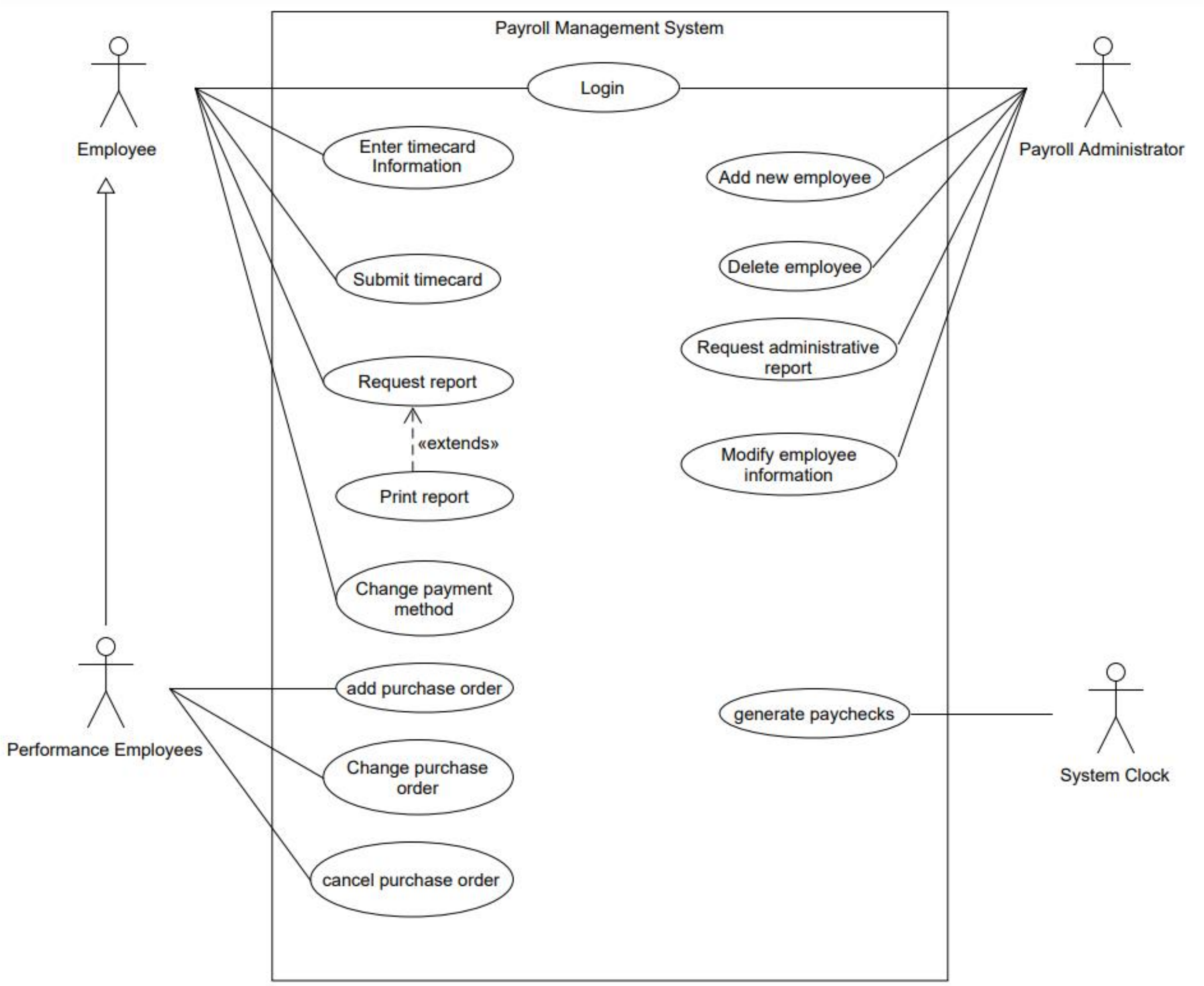
Comment: It is not specified in the text which information is used to delete an employee, but we assume that the social security number is used as the unique identifier to delete an employee.
Input: social security number
Output: none.

3.2.21 The system shall allow the Administrator to create administrative reports.

Comment: Administrative reports are either a "Total Hours Worked" or "Pay Year-to-Date" type report.
Input: type of administrative report, employee's name, begin and end dates of the report.
Output: Administrative report.

3.3 Use Cases

3.3.1 Use case diagram



3.3.2 Table of use cases

Use case	Actor	Relationships
Login	Employee / performance Employee/Administrator	none
Submit Timecard	Employee / performance Employee	none
Enter Timecard information	Employee / performance Employee	none
Add purchase order	Performance Employee	none
Change purchase order	Performance Employee	none
Cancel purchase order	Performance Employee	none

Change Payment Method	Employee / performance Employee	none
Request report	Employee / performance Employee	none
Add employee	Administrator	none
Delete employee	Administrator	none
Print report	Employee / performance Employee	Extends request report
Change employee’s information	Administrator	none
Request administrative report	Administrator	none

3.4 Use Cases Description

3.4.1 Use Case #1

Use Case Description	
System: Payroll Management System	
Use Case name: Add Purchase Order	
Primary actor: Performance Employee	**Other actors:** none
Stakeholders: none	
Description: This use case allows a performance employee to add a new purchase order.	
Relationships ▪**Includes:** none ▪**Extends:** none	
Input: date, purchased products, the amount of sale and customer billing address.	
Pre-conditions: Performance Employee is logged in the system.	
Steps:	
Performance employee	**System**
1-The employee requests the system to add a purchase order 3-The employee enters the requested info.	2-The system asks the employee to provide: date, purchased products, the amount of sale and customer billing address. 4-The system validates the entered information. (validate: ensures that the data entered by the employee meets the requirements for creating a valid purchase order) 5-The system assigns a new unique purchase order number. 6-The system adds a new purchase order. 7- The system displays a success message.
Alternative and exceptional flows: none	
Post-conditions: The new purchase order is created.	

3.4.2 Use Case #2

Use Case Description	
System: Payroll Management System	
Use Case name: Change payment method	
Primary actor: Employee	**Other actors:** none
Stakeholders: none	
Description: This use case allows an employee to change their preferred method of payment (either direct deposit or paycheck pickup) within the Payroll Management System.	
Relationships ▪**Includes:** none ▪**Extends:** none	
Input: Payment method.	
Pre-conditions: **1-**The employee is logged in to the system.	
Steps:	
Employee	**System**
1-The employee asks the system to change the payment method. 3- The employee provides the preferred payment method (either Direct Deposit or Paycheck Pickup) and provides the required bank information (bank account IBAN number for direct deposit).	2-The system asks the employee to provide the payment method (Direct Deposit or Paycheck Pickup). 4-The System updates the employee's payment method. 5-The system displays a success message.
Alternative and exceptional flows: none	
Post-conditions: the payment method of the employee is updated.	

3.4.3 Use Case #3

Use Case Description	
System: Payroll Management System	
Use Case name: Request report	
Primary actor: Employee	**Other actors:** none.
Stakeholders: none	
Description: This use case allows an employee to request reports of different types **(**total hours worked- total hours worked for a particular project- total pay received year-to-date- remaining vacation time).	
Relationships ▪**Includes:** none ▪**Extends:** none	
Input: type of report, begin and end dates of the report.	
Pre-conditions: Employee is logged in the system.	
Steps:	
Employee:	**System:**
1-Employee requests the system to display report 3-Employee provides the requested info.	2-System asks the employee to provide the type of report for a particular project, begin and end dates of the report, the project 4-System generates the requested report 5-The system displays the requested report, The report content will vary based on the employee's selection (total hours worked, hours for a specific project, total pay, or remaining vacation time).
Alternative and exceptional flows: A.6: Employee request the system to print the report. A.6.1: The Employee requests the system to print the report. A.6.2: The system sends the report to the configured printer.	
Post-conditions: Report generated and displayed.	

3.4.4 Use Case #4

Use Case Description	
System: Payroll Management System	
Use Case name: Add Employee	
Primary actor: Administrator	**Other actors:** none
Stakeholders: Employee	
Description: This use case allows the Payroll Administrator to add a new employee to the system.	
Relationships ▪**Includes:** none ▪**Extends:** none	
Input: name, social security number, phone number, payment classification (hourly, salaried, performance), salary (for salaried and performance employees), hourly rate (for hourly employees), all deductions.	
Pre-conditions: 1-Administrator is logged into the system.	
Steps:	
Administrator	**System**
1-The administrator asks the system to add a new employee 3- The administrator provides the requested input.	2-The system asks the user to provide name, social security number, phone number, payment classification, salary or hourly rate, all deductions. 4-The system adds the new employee. 5- the system displays a success message.
Alternative and exceptional flows: none	
Post-conditions: the new employee is added to the system.	

3.4.5 Use Case #5

Use Case Description	
System: Payroll management system	
Use Case name: Cancel purchase order	
Primary actor: Performance Employee	**Other actors:** none
Stakeholders: none	
Description: This use case allows a performance employee to cancel a purchase order.	
Relationships ▪**Includes:** None. ▪**Extends:** None.	
Input: purchase order number.	
Pre-conditions: Performance Employee is logged in the system.	
Steps:	
Performance Employee:	**System:**
1. the Performance Employee asks the system to cancel a purchase order. 3. the Performance Employee provides the system the requested info.	2.the system asks the Performance employee to provide the purchase order number. 4.the system validates the provided info 5. the system deletes the specified purchase order. 6. the system displays a success message. *validate: System compares the provided purchase order number with the purchase order number stored in the system/Database.
Alternative and exceptional flows: A.4: Performance employee provides an invalid purchase order number. A.4.1: the Performance employee provides an invalid purchase order number. A.4.2: The system validates the provided info. A.4.3: The system displays an "error" message. A.4.4: The system returns to step 2 of the basic flow.	
Post-conditions: purchase order is deleted from the system.	

3.6 Non-Functional Requirements

3.6.1 Performance

The system shall simultaneously support a large number of users against the central database at any given time. (this is ambiguous (a large number) we should have an exact number of employees)

3.6.2 Reliability

The system shall pay each employee the correct amount based on their employee type, on time, and by the payment method, the timing is specified as the following: -Hourly employees shall be paid every Sunday -Salaried employees and Performance employees shall be paid on the second working day of the month.

3.6.3 Availability

1-The main system must be running most of the time. (not testable we must say an exact number, for example 99%).
2-The system shall run during the times the payroll is run (every Sunday and the second working day of the month).

3.6.4 Security

1-The system shall not allow employees to access and edit other employees' timecards, they can only edit and access their own.
2-The system shall not allow performance employees to access and edit other performance employees' purchase orders, they can only edit and access their own.

3.6.5 Maintainability

3.6.6 Portability

The system shall run on individual employees' phones and tablets.

3.7 Design Constraints

1- The system will have a mobile phone interface that allows employees to submit timecard information, manage purchase orders, change payment methods, and create reports, ensuring that the system can run on employee phones and tablets.

2-The system shall work with the existing Work Management Database, the system will access but not update the information stored in the old database.

3-The system shall interface with the existing bank systems via an electronic transaction to deposit the generated paychecks.

3.8 Other Requirements

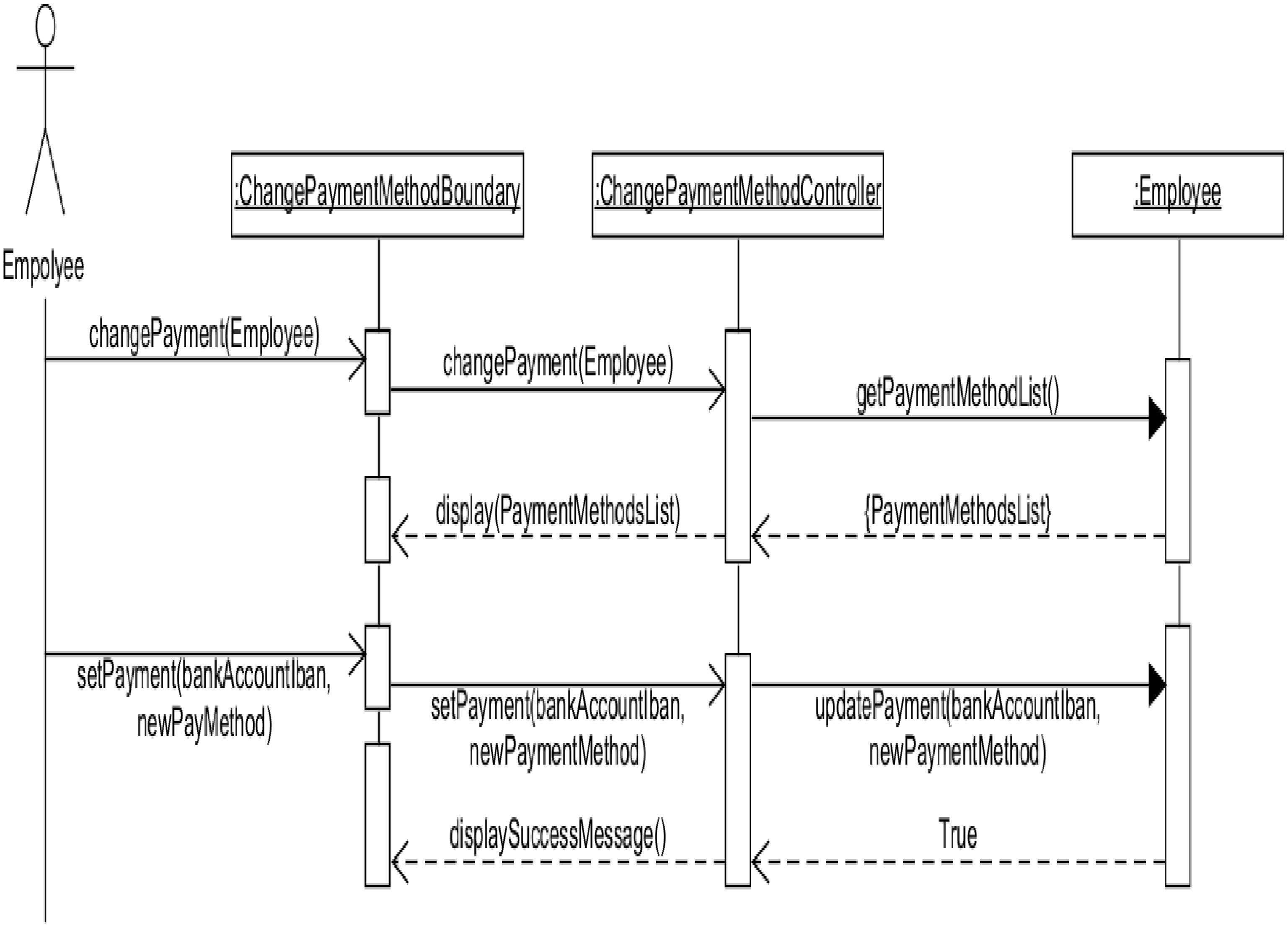
None

4. Analysis Models

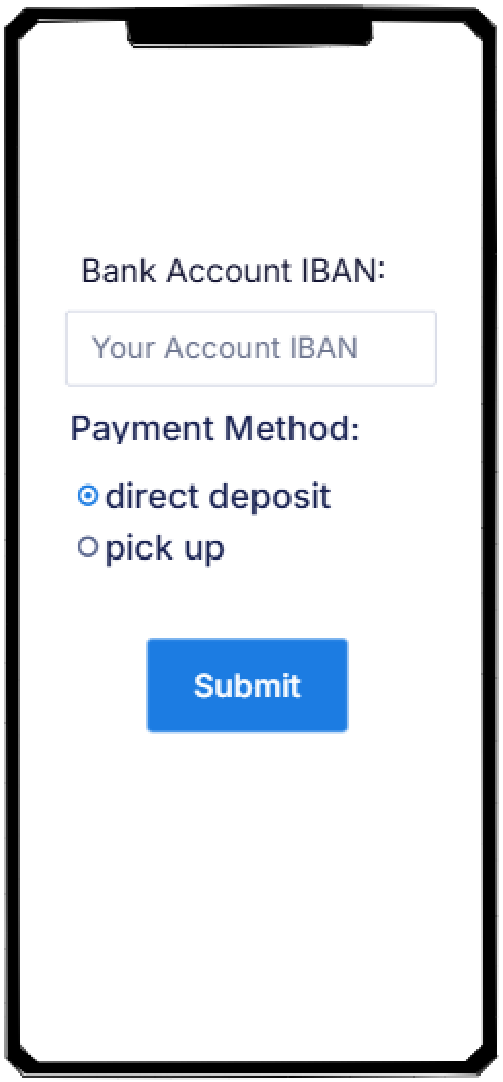
4.1 Sequence Diagrams and Mockup Screens

4.1.1 Change Payment Sequence Diagram:

This sequence diagram shows how an employee can change a payment method.


Empolyee
:ChangePaymentMethodBoundary
:ChangePaymentMethodController
:Employee
changePayment(Employee)
changePayment(Employee)
getPaymentMethodList()
display(PaymentMethodsList)
{PaymentMethodsList}
setPayment(bankAccountIban, newPayMethod)
setPayment(bankAccountIban, newPaymentMethod)
updatePayment(bankAccountIban, newPaymentMethod)
displaySuccessMessage()
True

Mock-up screens:



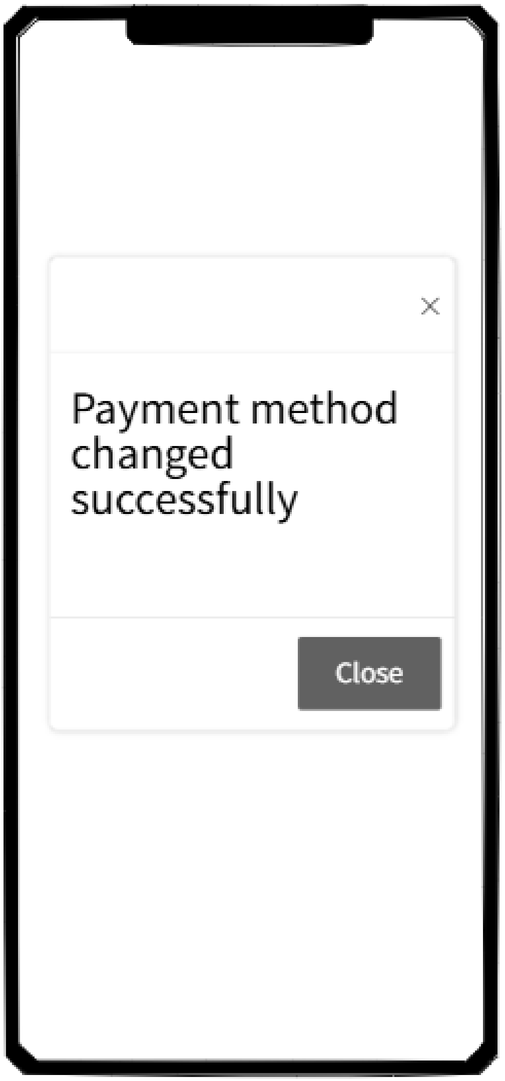
Bank Account IBAN:

Your Account IBAN

Payment Method:

● direct deposit

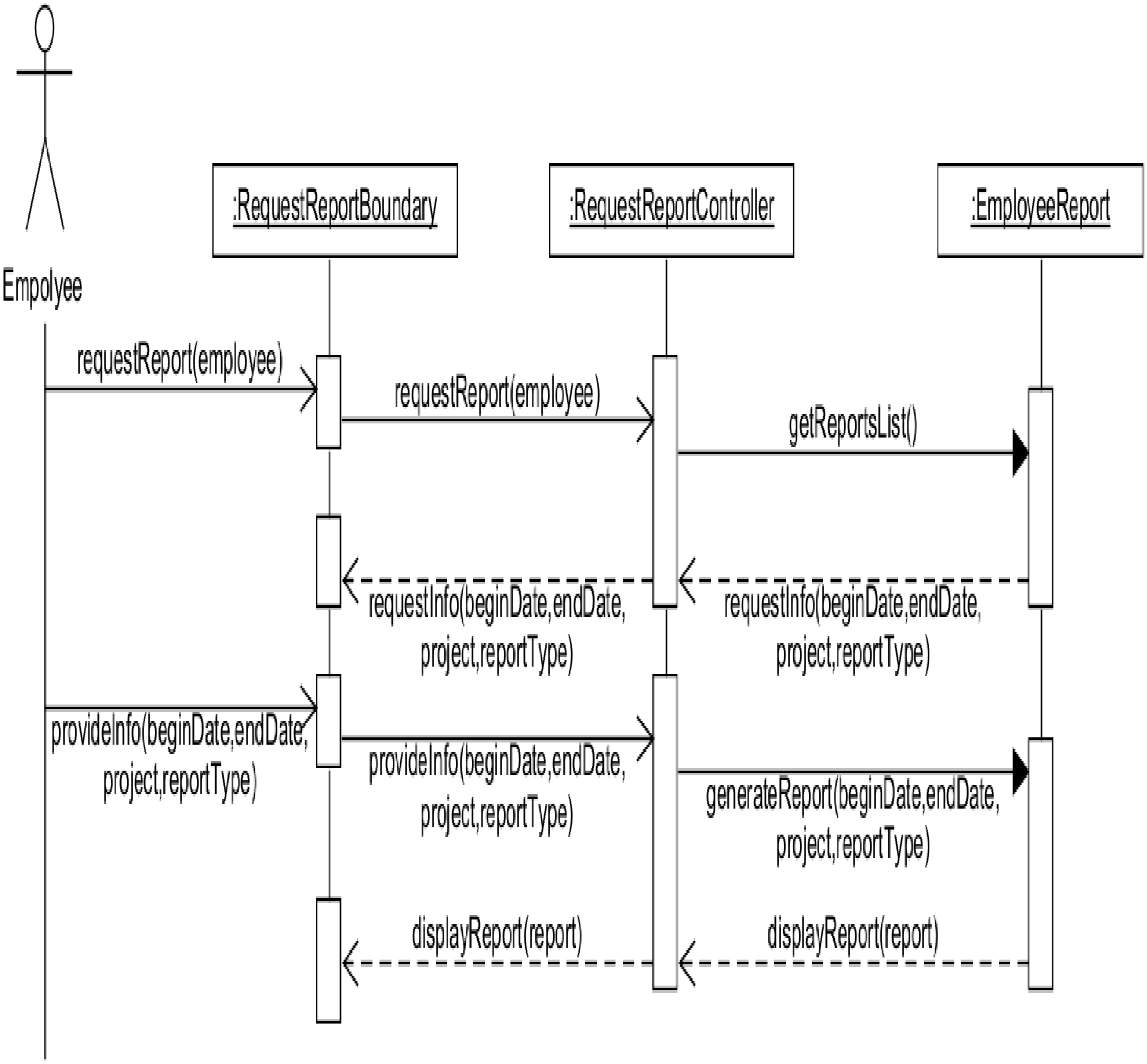
○ pick up



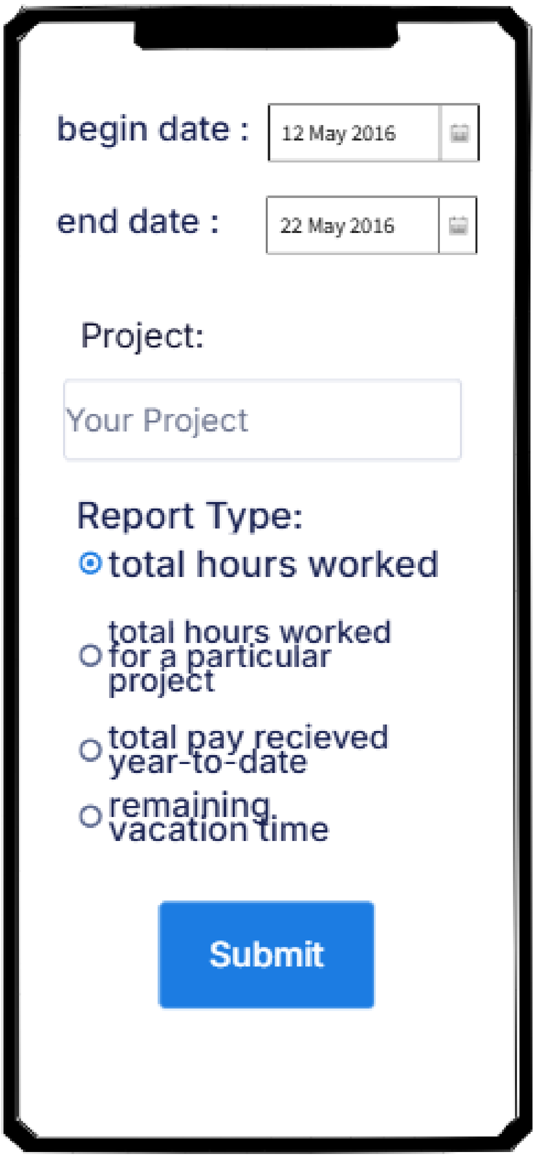
Payment method changed successfully

4.1.2 Request Report Sequence Diagram:

This sequence diagram shows how an employee can request a report.


Empolyee
:RequestReportBoundary
:RequestReportController
:EmployeeReport
requestReport(employee)
requestReport(employee)
getReportsList()
requestInfo(beginDate,endDate, project,reportType)
requestInfo(beginDate,endDate, project,reportType)
provideInfo(beginDate,endDate, project,reportType)
provideInfo(beginDate,endDate, project,reportType)
generateReport(beginDate,endDate, project,reportType)
displayReport(report)
displayReport(report)

Mock-up screens:



begin date : 12 May 2016

end date : 22 May 2016

Project:

Your Project

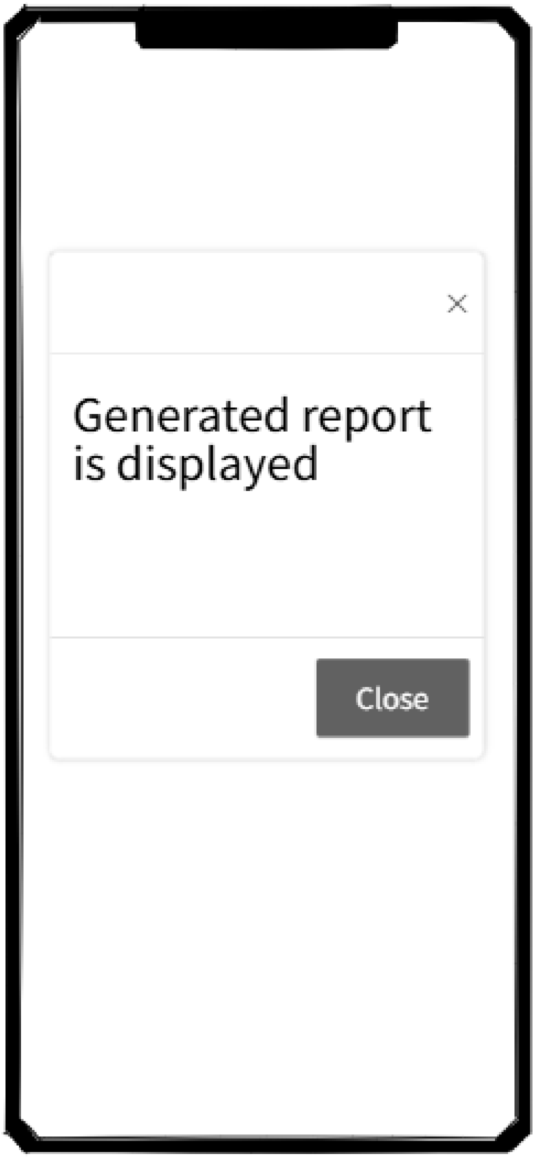
Report Type:

● total hours worked

○ total hours worked for a particular project

○ total pay recieved year-to-date

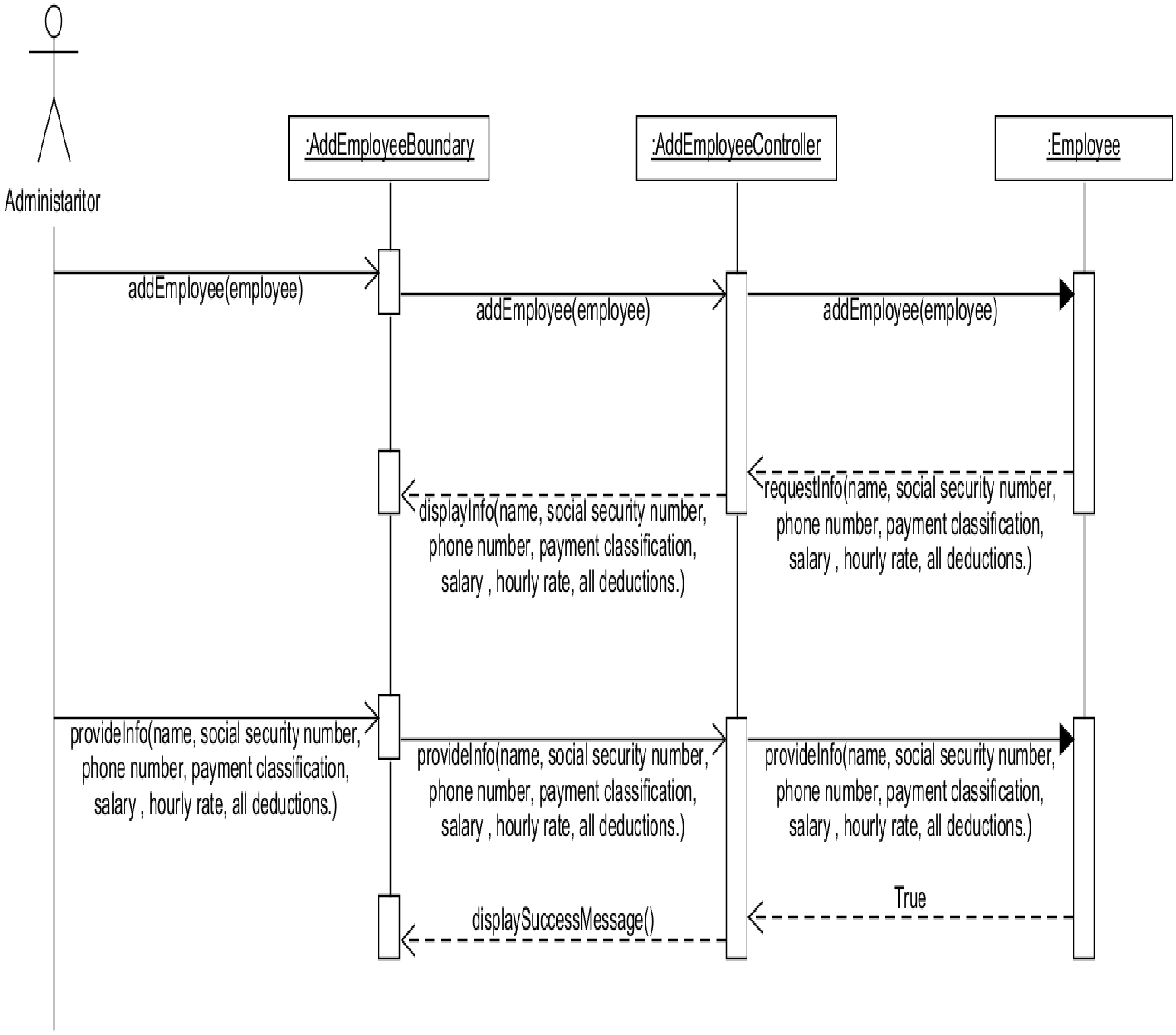
○ remaining vacation time



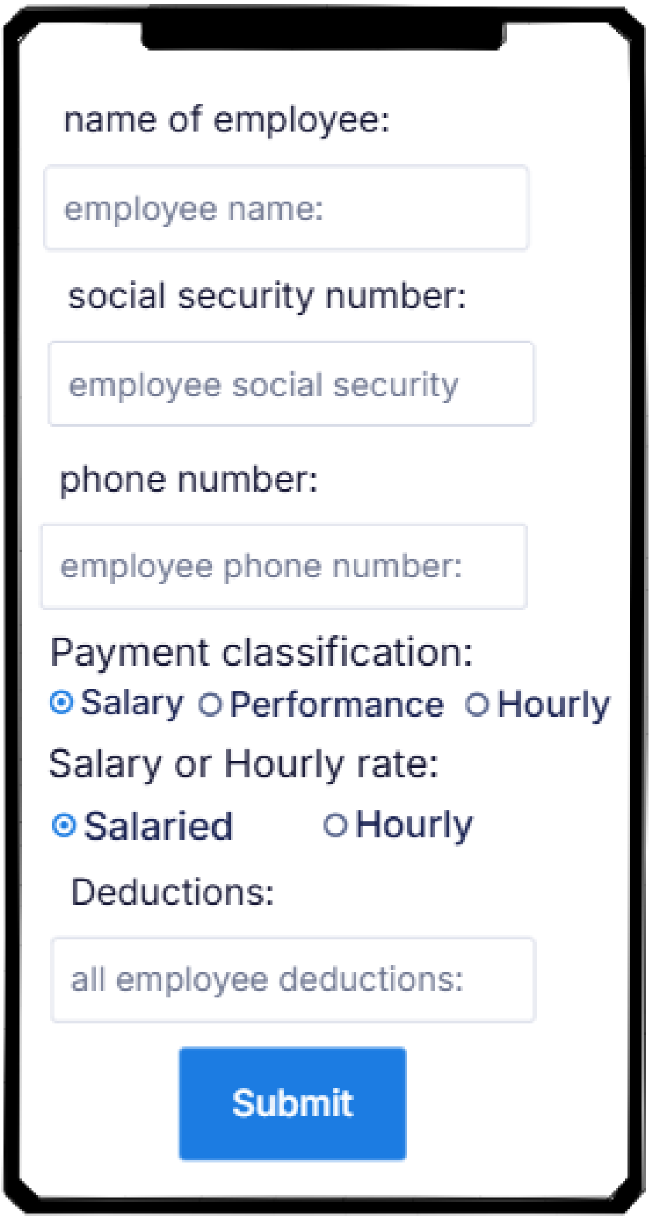
Generated report is displayed

4.1.3 Add Employee Sequence Diagram:

This sequence diagram shows how an administrator can add an employee.


:AddEmployeeBoundary
:AddEmployeeController
:Employee
Administaritor
addEmployee(employee)
addEmployee(employee)
addEmployee(employee)
requestInfo(name, social security number, phone number, payment classification, salary , hourly rate, all deductions.)
displayInfo(name, social security number, phone number, payment classification, salary , hourly rate, all deductions.)
provideInfo(name, social security number, phone number, payment classification, salary , hourly rate, all deductions.)
provideInfo(name, social security number, phone number, payment classification, salary , hourly rate, all deductions.)
provideInfo(name, social security number, phone number, payment classification, salary , hourly rate, all deductions.)
True
displaySuccessMessage()

Mock-up screens:



name of employee:

employee name:

social security number:

employee social security

phone number:

employee phone number:

Payment classification:

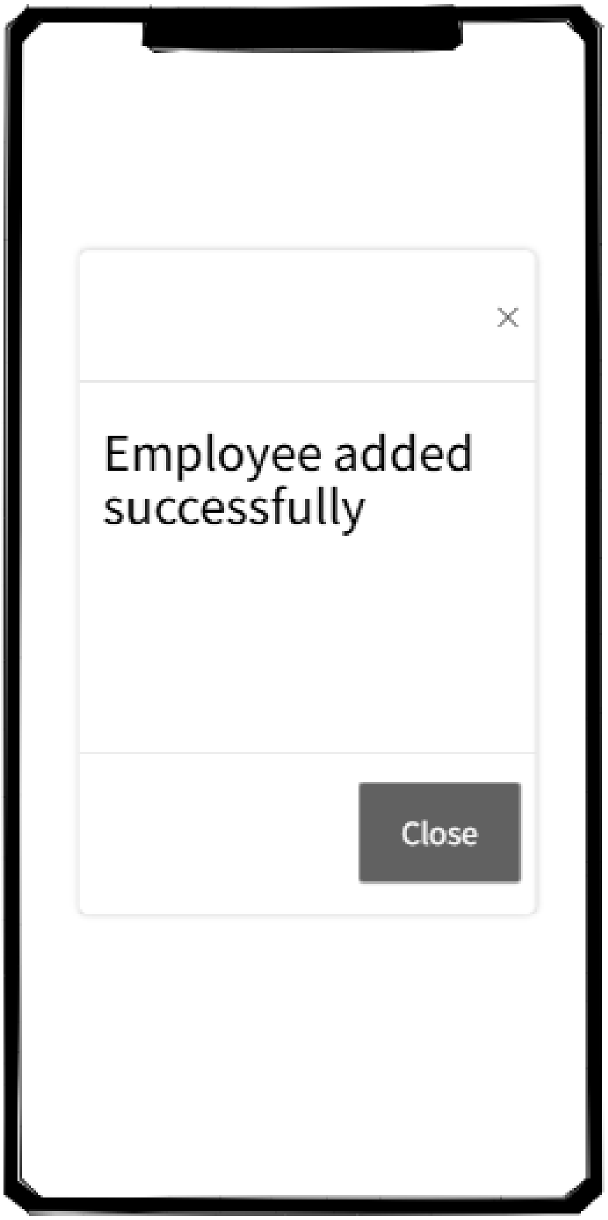
● Salary ○ Performance ○ Hourly

Salary or Hourly rate:

● Salaried ○ Hourly

Deductions:

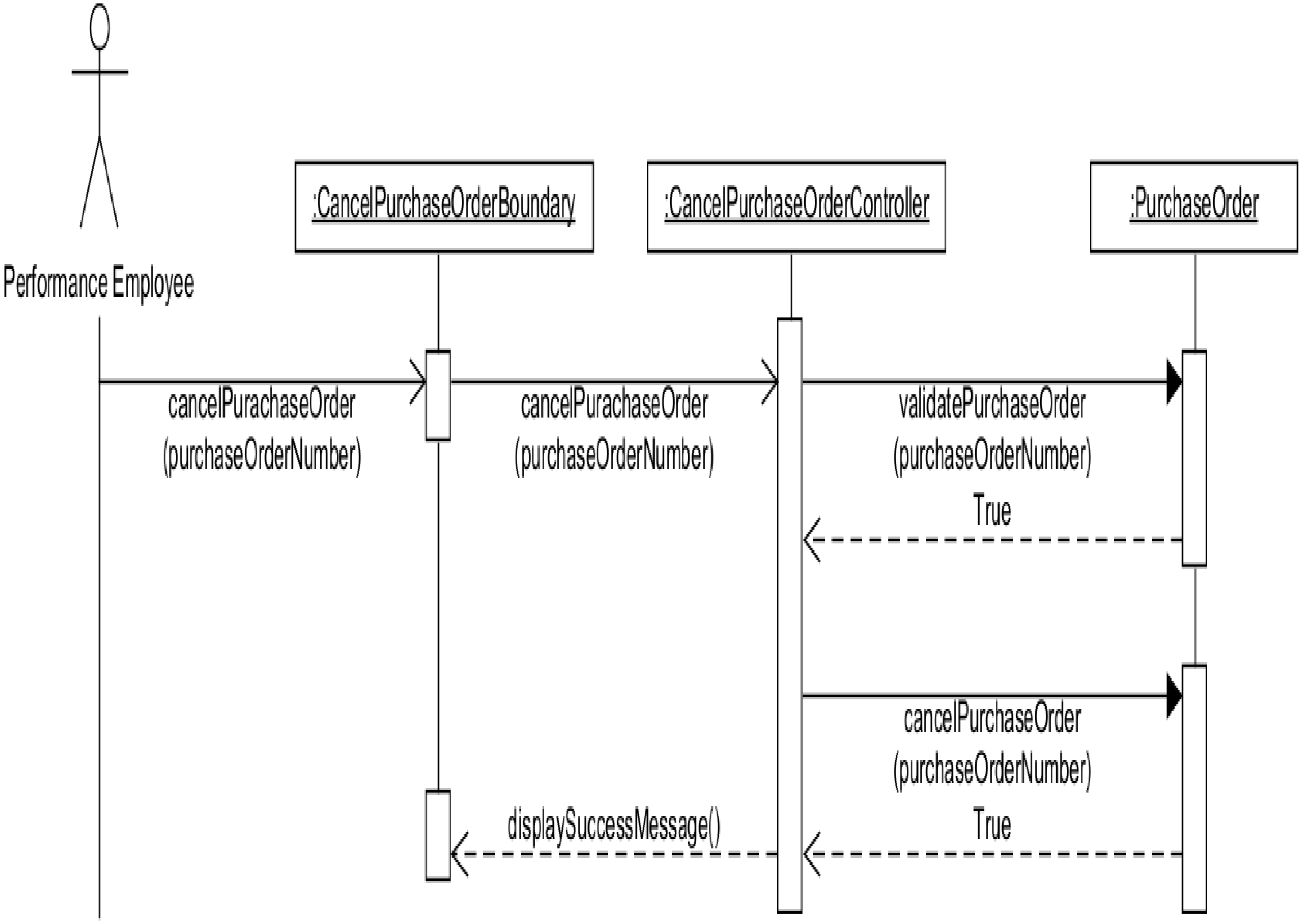
all employee deductions:



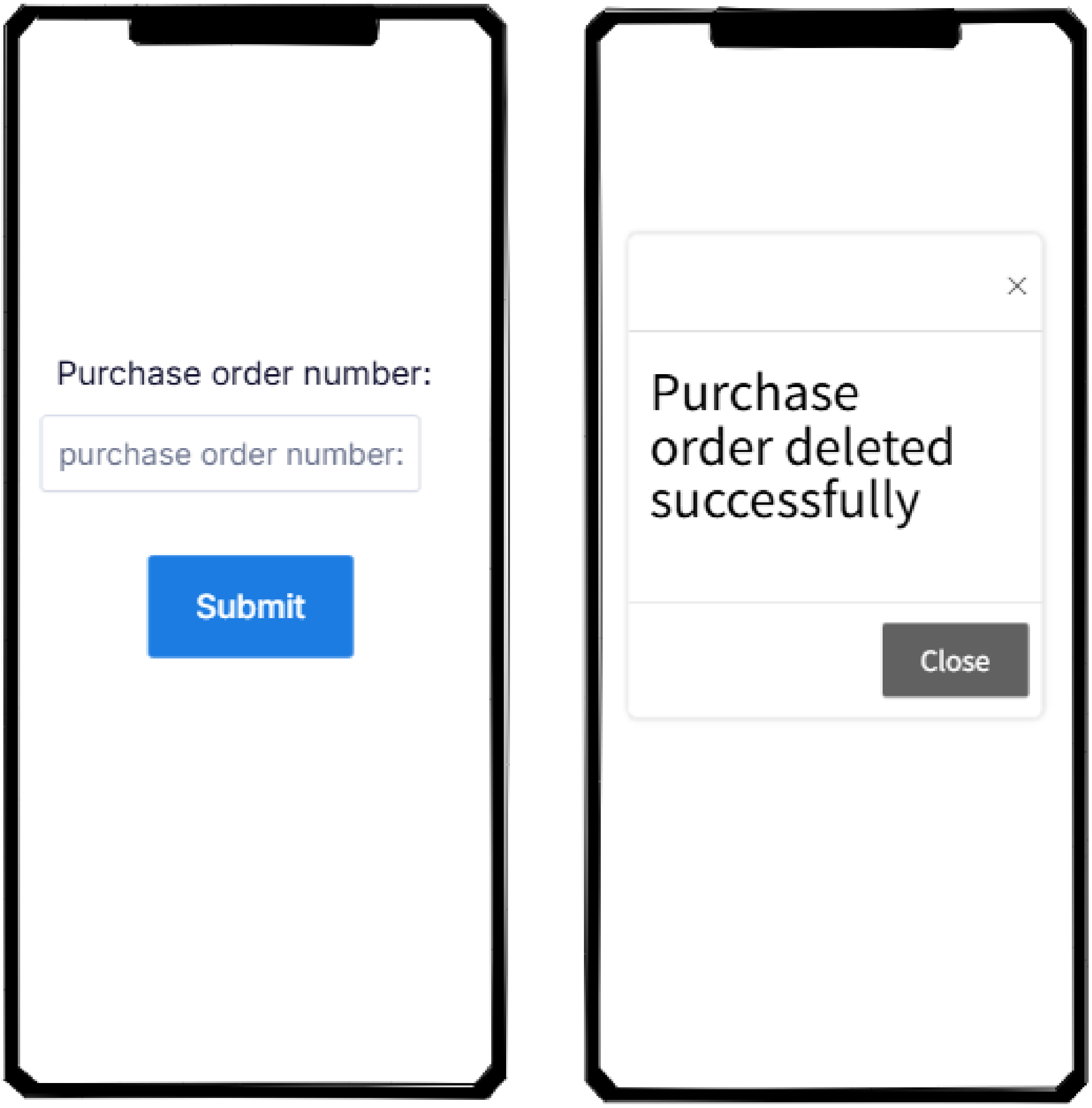
Employee added successfully

4.1.4 cancel purchase order Sequence Diagram:

This sequence diagram shows how a performance employee can cancel a purchase order.



Mock-up screens:

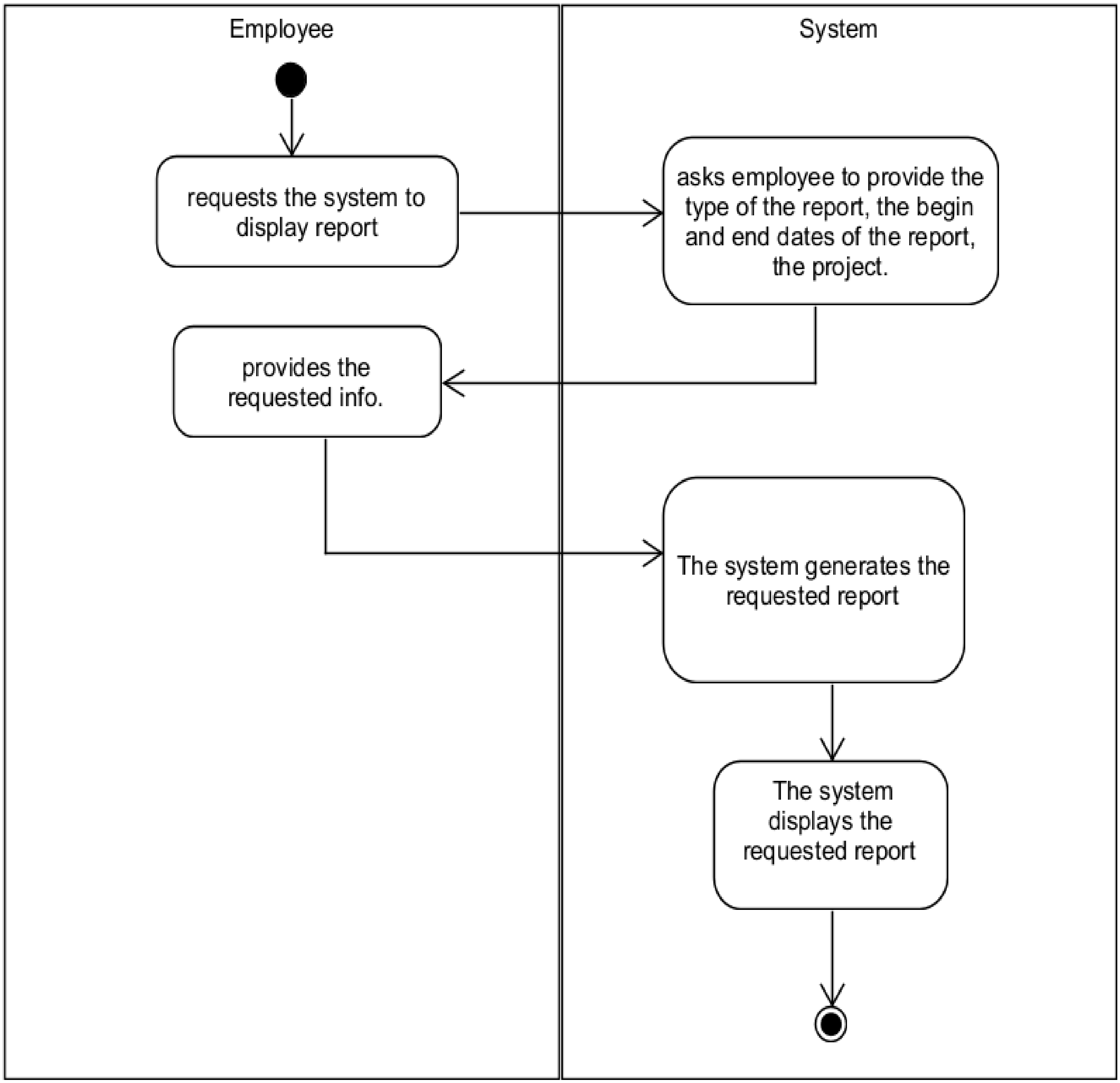


Purchase order deleted successfully

4.2 Activity Diagrams

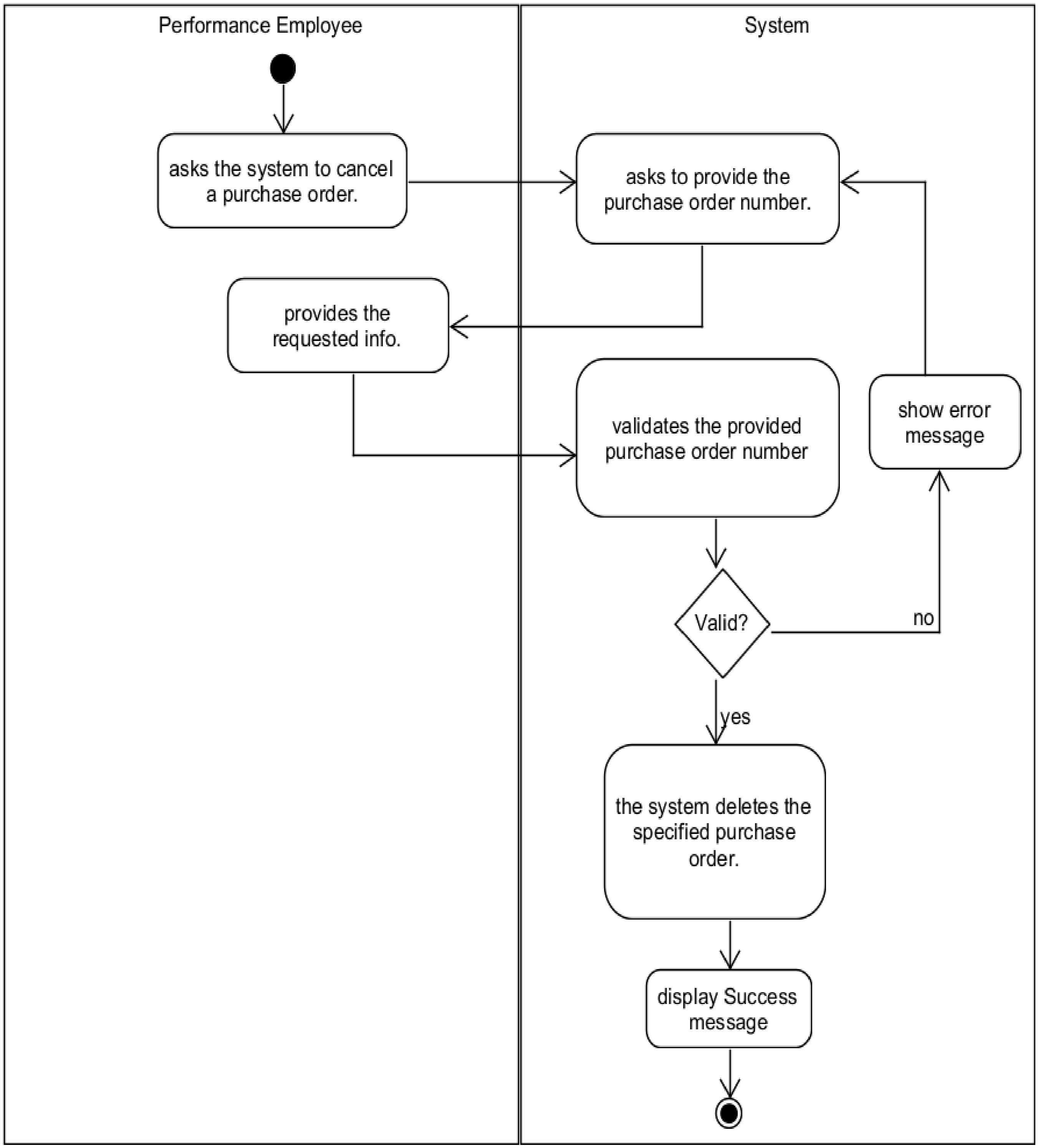
4.2.1 Request Report Activity Diagram:

This activity diagram shows how the employee interacts with the system to request a report.


Employee
System
requests the system to display report
asks employee to provide the type of the report, the begin and end dates of the report, the project.
provides the requested info.
The system generates the requested report
The system displays the requested report

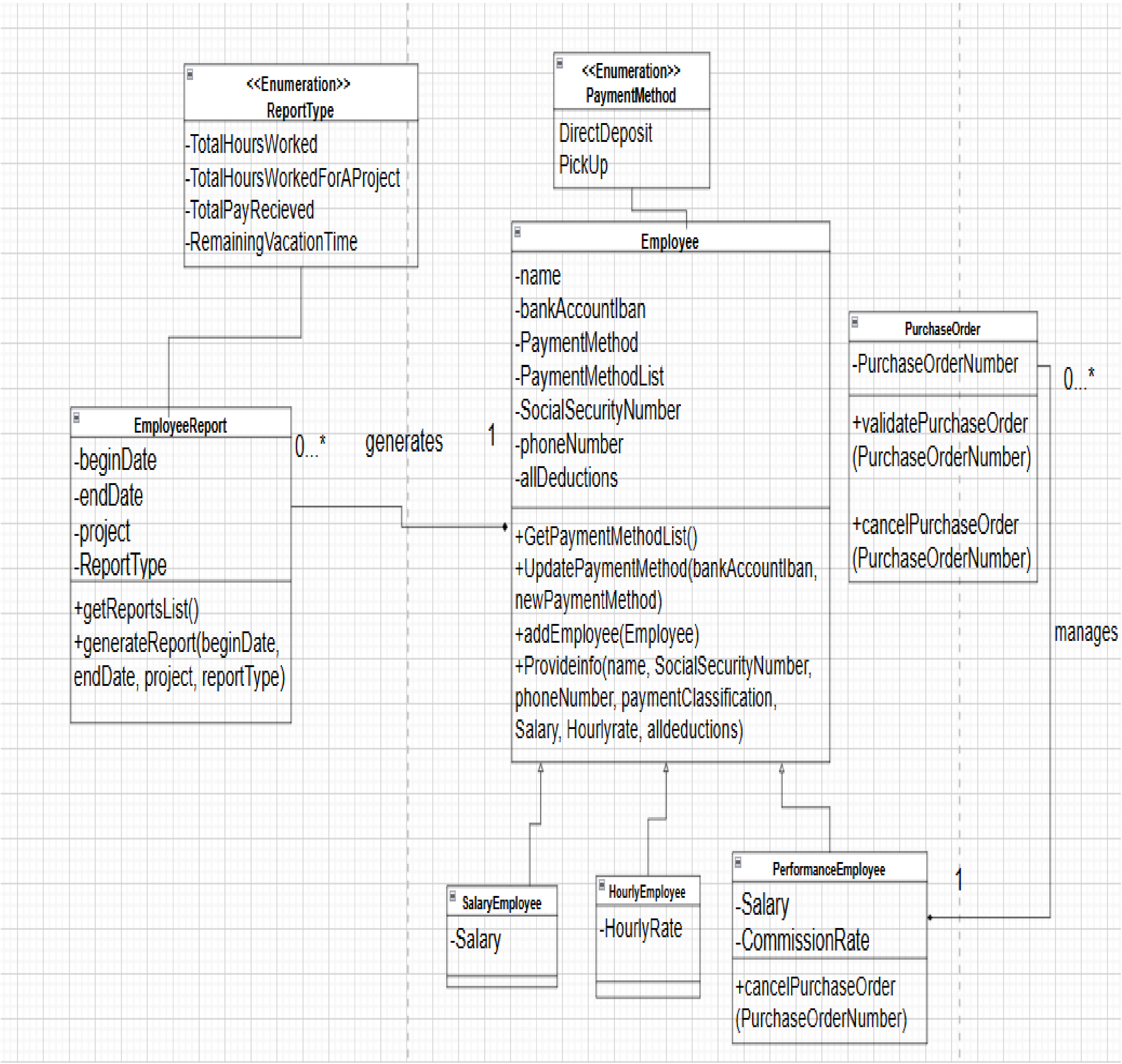
4.2.2 cancel purchase order Activity Diagram:

This activity diagram shows how the performance employee interacts with the system to cancel a purchase order.


Performance Employee
System
asks the system to cancel a purchase order.
asks to provide the purchase order number.
provides the requested info.
validates the provided purchase order number
show error message
Valid?
no
yes
the system deletes the specified purchase order.
display Success message

5 Classes / Objects

5.1 Class diagram



5.2 Employee

5.2.1 Attributes

-name:
- bankAccountIBAN : Use Case #2
-socialSecurityNumber : Use Case #4
-phoneNumber : Use Case #4
-paymentMethod(enumeration object) : Use Case #2
-allDeductions: Use Case #4
-paymentMethodList : Use Case #2

5.2.2 Functions

+GetPaymentMethodList() : Use Case #2
+UpdatePaymentMethod(bankAccountIBAN, newPaymentMethod) : Use Case #2
+addEmployee(Employee) : Use Case #4
+Provideinfo(name, socialSecurityNumber, phoneNumber, paymentClassification, salary, hourlyrate, alldeductions) : Use Case #4

5.3 EmployeeReport

5.3.1 Attributes

-beginDate : Use Case #3
-endDate : Use Case #3
-project : Use Case #3
-reportType (enumeration object) : Use Case #3

5.3.2 Functions

+getReportList() : Use Case #3
+generateReport(beginDate, endDate, project, reportType) : Use Case #3

5.4 PurchaseOrder

5.4.1 Attributes

-PurchaseOrderNumber : Use Case #5

5.4.2 Functions

+validatePurchaseOrder(PurchaseOrderNumber) : Use Case #5
+cancelPurchaseOrder(PurchaseOrderNumber) : Use Case #5

5.5 SalaryEmplyee

5.5.1 Attributes

-Salary : Use Case #4

5.6 HourlyEmplyee

5.6.1 Attributes

-HourlyRate : Use Case #4

5.7 PerformanceEmplyee

5.7.1 Attributes

- -Salary : Use Case #4
- -CommissionRate : Functional Requirement 3.2.14

5.7.2 Functions

- +cancelPurchaseOrder(PurchaseOrderNumber) : Use Case #5

5.8 PaymentMethod (Enum class)

5.8.1 Attributes

- -DirectDeposit : Use Case #2
- -PickUp : Use Case #2

5.9 ReportType(Enum class)

5.9.1 Attributes

- - TotalHoursWorked : Use Case #3
- -TotalHoursWorkedForAProject : Use Case #3
- -TotalPayRecieved : Use Case #3
- -RemainingVacationTime : Use Case #3