KING SAUD UNIVERSITY
COLLEGE OF COMPUTER AND INFORMATION SCIENCES

Payroll Management System
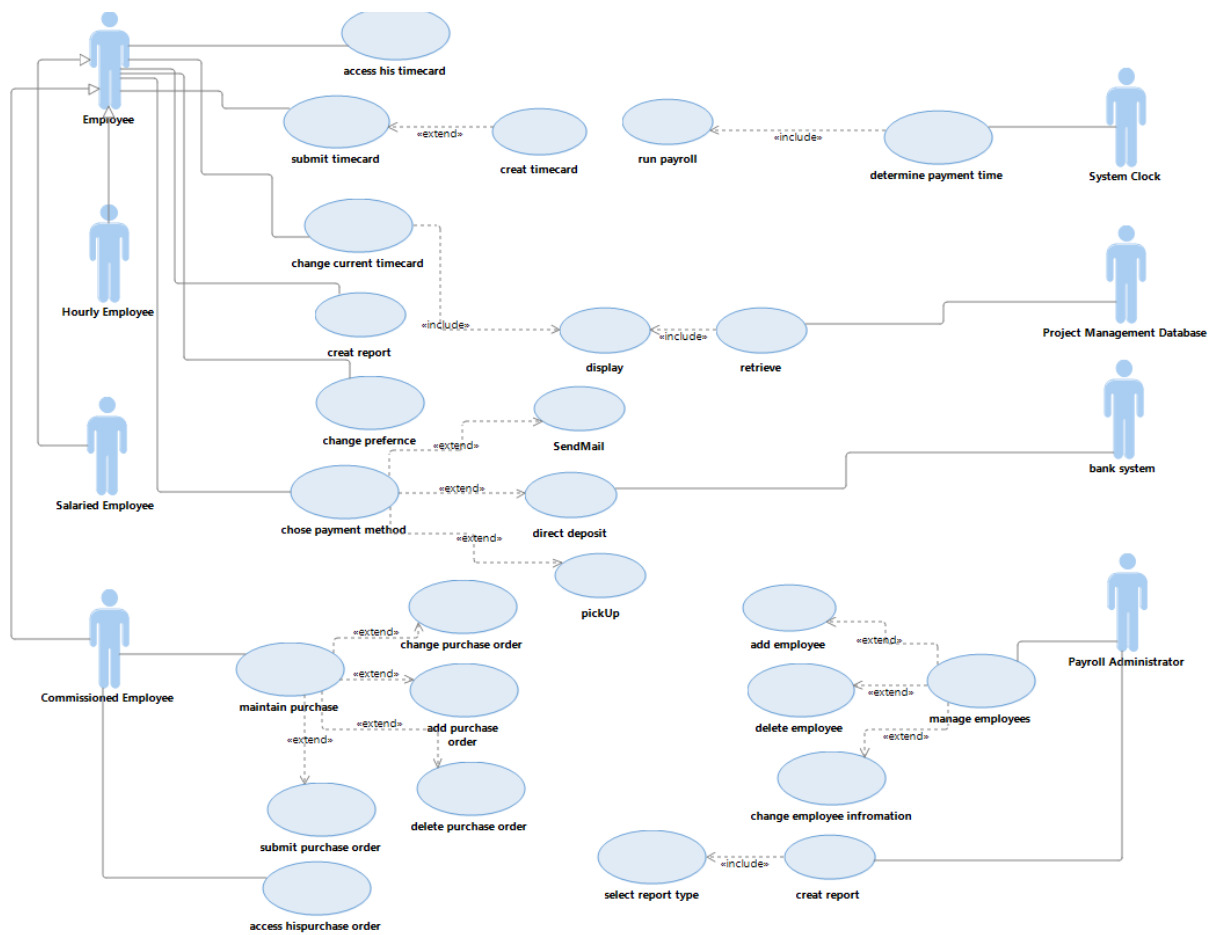
SWE-321 Software Design & Architecture (Phase2)
2nd Semester 1438-1439 H

Name	ID	Section
Deema Naif Alotaibi	436202039	47859
Shahad Abdullah Almuhizi	436201525	47859
Noura Abdullah Alsaadan	436202586	47859

Supervised by: T.Rawan

Use Case Diagram :

(1:use case diagram of all use cases)

2

Class Diagram :

Report
- Type : rType
- startDate : String
- endDate : String
- empName : String
- newReport ( )
- fillReport ( )
- submitReport ( )

«enumeration»
rType
- totalHours
- payYearToDate

«enumeration»
EmpType
- hourly
- salaried
- Commissioned

«enumeration»
paymentMethods
- dirctDeposit
- pickOffice
- mailed

- report3
- report
- timecard

timecard
- start date : String
- end date : String
- totalHours : int
- submitTimecard ( )

- admin

admin
- name : String
- Id : int
- email : String
- phone : int
- addEmp ( )
- deleteEmp ( )
- changeEmpType ( )
- changeEmpInfo ( )
- changeReportType ( )
- creatReport ( )

Employee
- name : String
- type : EmpType
- mail address : String
- socialSecurityNum : int
- standardTaxDeductions : double
- phoneNum : int
- hoursRate : double
- salary : double
- ID : int
- email : String
- payment : paymentMethods
- extraHours ( )
- CalSalary ( )
- enterTimecard ( )
- changePreferences ( )
- creatReport ( )

- employee
- admin

Address
- Road : string
- street : String
- District : String
- HouseNumber : int
- phoneNumber : int

- address1
- customer

payment
- emp : Employee
- type : paymentMethods
- bankaccount : String
- mailbox : int
- lastPayment : String
- nextPayment : String
- extraHoursAmount : double
- PaySalary ( )
- extraHoursPayAmount ( )

CommissionedEmp
- numberOfPO : int
- commissionRate : double
- AddPurchaseOrder ( )
- DeletePurchaseOrder ( )
- changePurchaseOrder ( )
- calculateComRate ( )

- purchaseOrder

purchaseOrder
- POdate : String
- salesAmount : double
- customer : Customer

- purchaseOrder
- customer2

Customer
- Name : String
- billingAddress : Adress
- phoneNo : int
- email : String

(2:class diagram representing Application Domain)

3

Use case descriptions:
1-Select Payment Method

<table>
<tr><td colspan="2" align="center">Use Case Description</td></tr>
<tr><td>System: Payroll Management System</td><td></td></tr>
<tr><td colspan="2">Use Case name:Select payment method</td></tr>
<tr><td>Primary actor: Employer</td><td>Other actors: none</td></tr>
<tr><td colspan="2">Description: This use case describes how an Employee can select the preferred payment method</td></tr>
<tr><td colspan="2">Relationships
■   Includes: none
■   Extends: none</td></tr>
<tr><td colspan="2">Pre-conditions:
Employee must be added to the Payroll Management System</td></tr>
<tr><td colspan="2">Steps:</td></tr>
<tr><td align="center">Actor</td><td align="center">System</td></tr>
<tr><td>1. This use case begins when the Employee click on the "Select payment method" Button.
3. The employee will choose direct deposit option
5. The employee will fill he form with the required bank account information and submit it</td><td>2. The system displays the payment method options for the employee:
a.   Pick up
b.   Mail
c.   Direct deposit
4. The system will display a bank account information form
6. The system will display a message that the payment method was submitted successfully, and then direct the Employee to the main page.</td></tr>
<tr><td colspan="2">Alternative and exceptional flows:
1: In step 3 the Employee choose Mail option:
a.   The system will display a postal address form
b.   The Employee will fill in the form with the required postal address information and submit it.
c.   step 6 is resumed.
2: In step 3 the Employee choose Pick up option:
a.   step 6 is resumed.
3: The employee quits:
If at any step before step 5 the Employee selects cancel:
a. The use case will end with a failure condition.</td></tr>
<tr><td colspan="2">Post-conditions:
■   Success: The payment method of the employee is updated in the database.
■   Failure: The payment method was not changed.</td></tr>
</table>

(3:use case description of Select Payment Method use case)

2- Create Employee Report

Use Case Description	
System: Payroll Management System	
Use Case name:Create Employee Report	
Primary actor: Employee	**Other actors: none**
Description:This user case describes how the Employee can create a report on the Payroll Management System	
Relationships ▪ **Includes:none** ▪ **Extends:none**	
Pre-conditions: ▪ **Employee must be added to the Payroll Management System**	
Steps:	

Actor	System
1. This use case begins when the Employee click on the "Create Employee Report" button **3. The employee choose "Remaining vacation time" option and then clicks "Next"** **6. The employee clicks on "Home page" option**	**2. The system displays a list with the information available about the Employee to choose to view.** **4. The system display the remaining vacation time to the Employee.** **5. The system will display two options, "Back" option and "Home page" option.** **7. The system will direct the user to the home page.**

Alternative and exceptional flows:
1: In step 6, the Employee clicks on "Back" option:
a. step 2 is resumed.
2: The employee quits:
If at any step, the employee selects "Cancel":
a. The use case will end with failure condition

Post-conditions:
▪ **Success: The employee will view any information in the report successfully**
▪ **Failure: The Employee will not view the report.**

(4:use case description of Create Employee Report use case)

5

3 -Submit Timecard

Use Case Description	
System: Payroll Management System.	
Use Case name: Submit Timecard.	
Primary actor:Employee	Other actors:None
Description: This use case describes how the Hourly Employee can submit timecard on the Payroll Management System.	
Relationships ▪ **Includes:None** ▪ **Extends: Make changes to timecard.**	
Pre-conditions: The employee must be added to the Payroll Management System and logged in.	

Actor	System
1.This use case begins when the Hourly Employee click on the "Submit Timecard" button. 3.The Hourly Employee will select continue. 5.The Hourly Employee will select quit.	2. The system will display a message that the timecard cannot be changed after submitting with an option whether he wants to continue or cancel. 4.The system will validat the timecard information: •Start and end dates of the timecard. •Number of hours worked. 6. The system will direct the Hourly Employee to main page

Alternative and exceptional flows:

Alternative Flow:

1. The hours more than 8:

If in step 2 the worked hours are more than 8:

1. The system will display a message that Acme will pay them 1.5 times their normal rate for those extra hours and will display the new amount that Acme will pay them.

2. Step3isresumed.

3. Step5isresumed.

2. Hourly Employee quits:

If in step 3 the Employee selects cancel
1. The use case ends with a failure condition.

Post-conditions:

If the use case was successful, the Employee timecard information is saved to the system.

Otherwise, the system state is unchanged.

(5:use case description of Submit Timecard use case)

4 -Maintain Timecard

Use Case Description	
System: Payroll Management System	
Use Case name: Maintain Timecard	
Primary actor: Employee	**Other actors:**
Description:This use case allows the Employee to update and submit timecard information.	
Relationships ▪ **Includes: None** ▪ **Extends: None**	
Pre-conditions **The Employee must be logged onto the system before this use case begins**	
Steps:	

Actor	System
1. use case begins when the employee click Maintain Timecard .	**2. The system retrieves and displays the current timecard for the Employee. If a timecard does not exist , the system creates a new one and the start and end dates of the timecard are set by the system.**
3. employee enters the hours worked.	**4. system saves the timecard**
5. employee select submit timecard.	**6. system change timecard status to read only .**

Alternative and exceptional flows:

1.Timecard Already Submitted
if step 2 in the Employee's current timecard status is read only:
1.The system displays a read-only copy of the timecard.
2.The system will display a message to the employee that the timecard has
 been already submitted.
3.Employee will confirm the message.
4.The use case will end with a failure condition.

2.Invalid Number of Hours
if step 3 in the employee enter invalid number of hours:
1.The system will display an error message to the employee.
2. Employee confirm the message.
3.Step 3 is resumed which will allow the employee to re-enter the number
 of hours.

3. Employee Quit
The user press "Cancel" button before step 4 the use case ends with a
failure condition

Post-conditions:
**Successful condition: payment method for a certain employee will be
chosen**
Failure condition: nothing will change

(7:use case description of Maintain Timecard use case)

Mockups: 1- Create Employee Report:
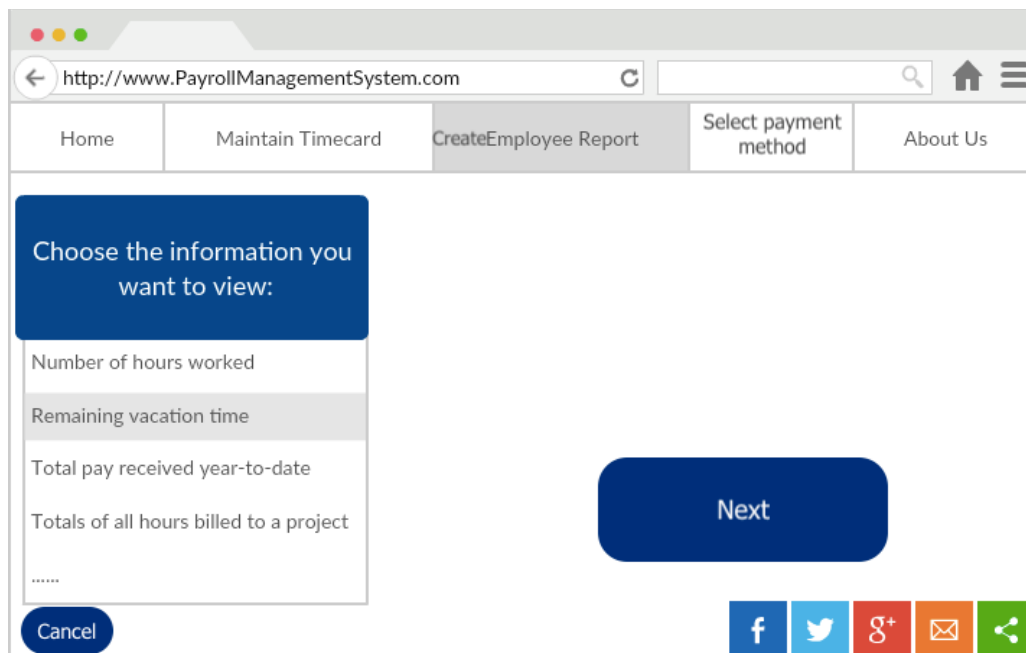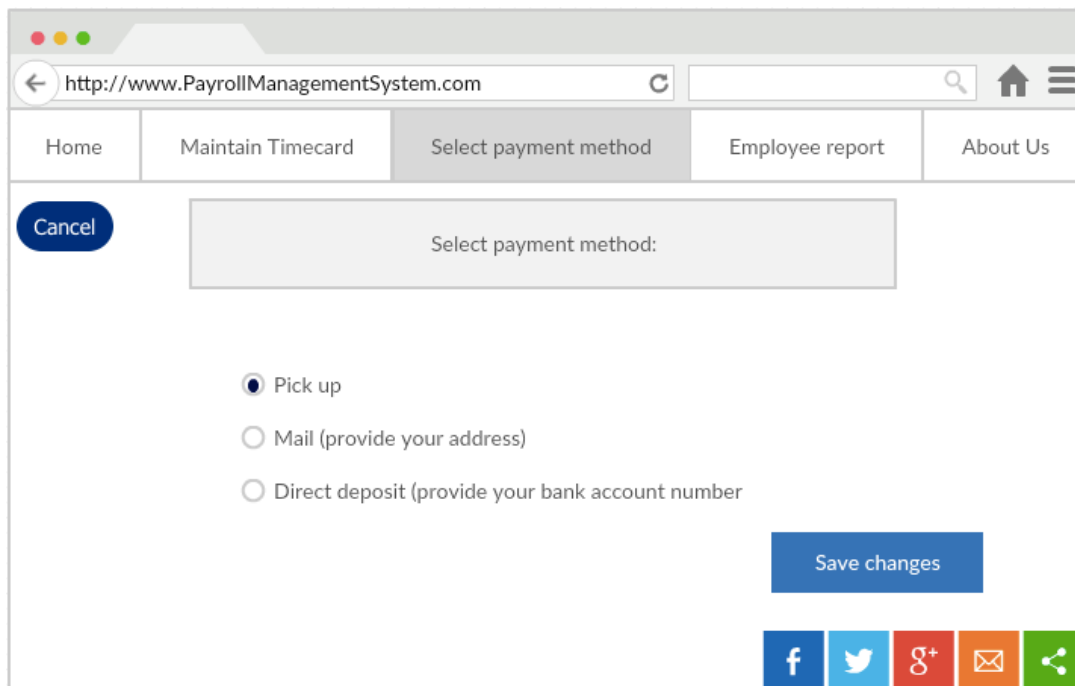
(8: Screen mock-up corresponding to Create Employee Report use case)
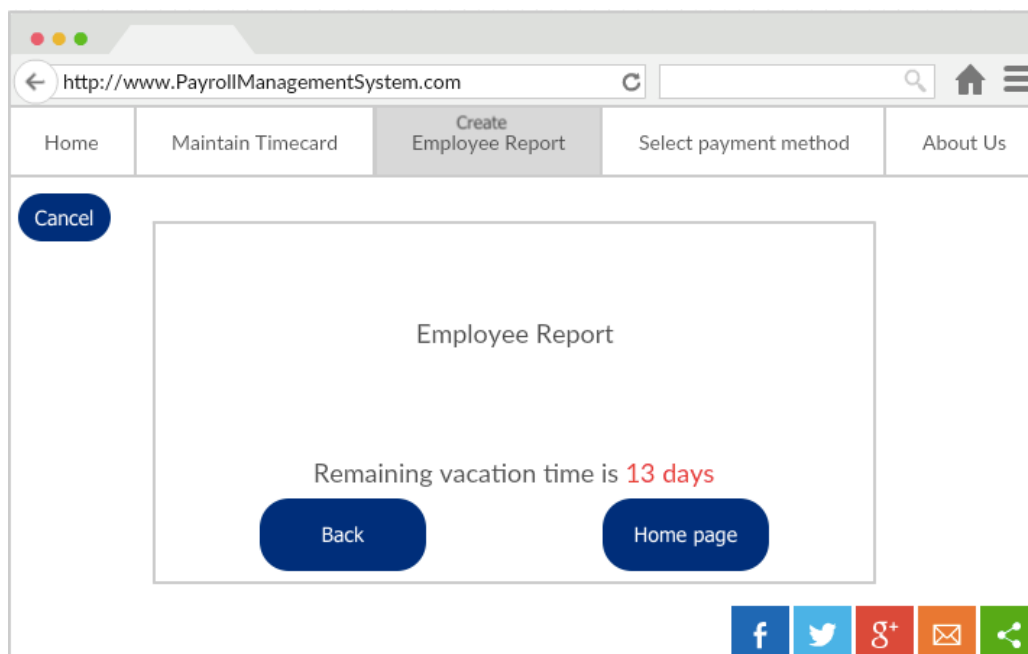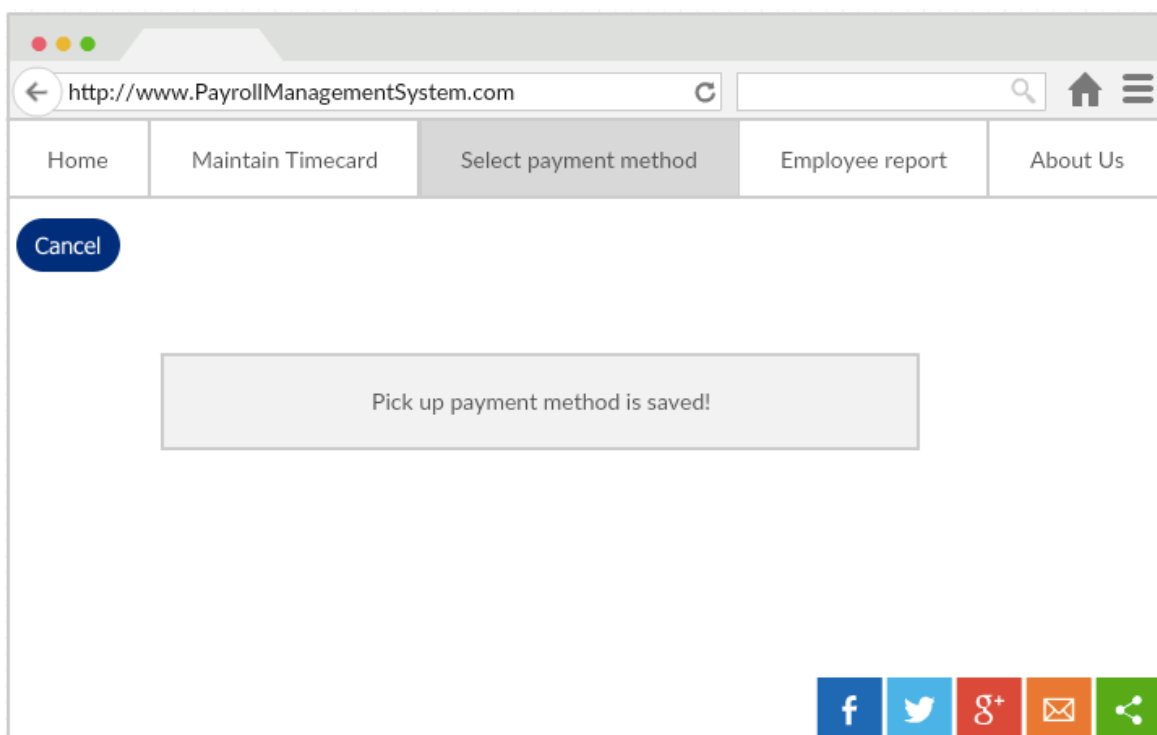
(9:Screen mock-up corresponding to Create Employee Report use case)

2-Select Payment Method :

(10:Screen mock-up corresponding to Select Payment Method use case)

(11:Screen mock-up corresponding to Select Payment Method use case)

3-Maintain Timecard :

(12:Screen mock-up corresponding to Maintain Timecard use case)

(13:Screen mock-up corresponding to Maintain Timecard use case)

Sequence Diagram :

1-Maintain Timecard:
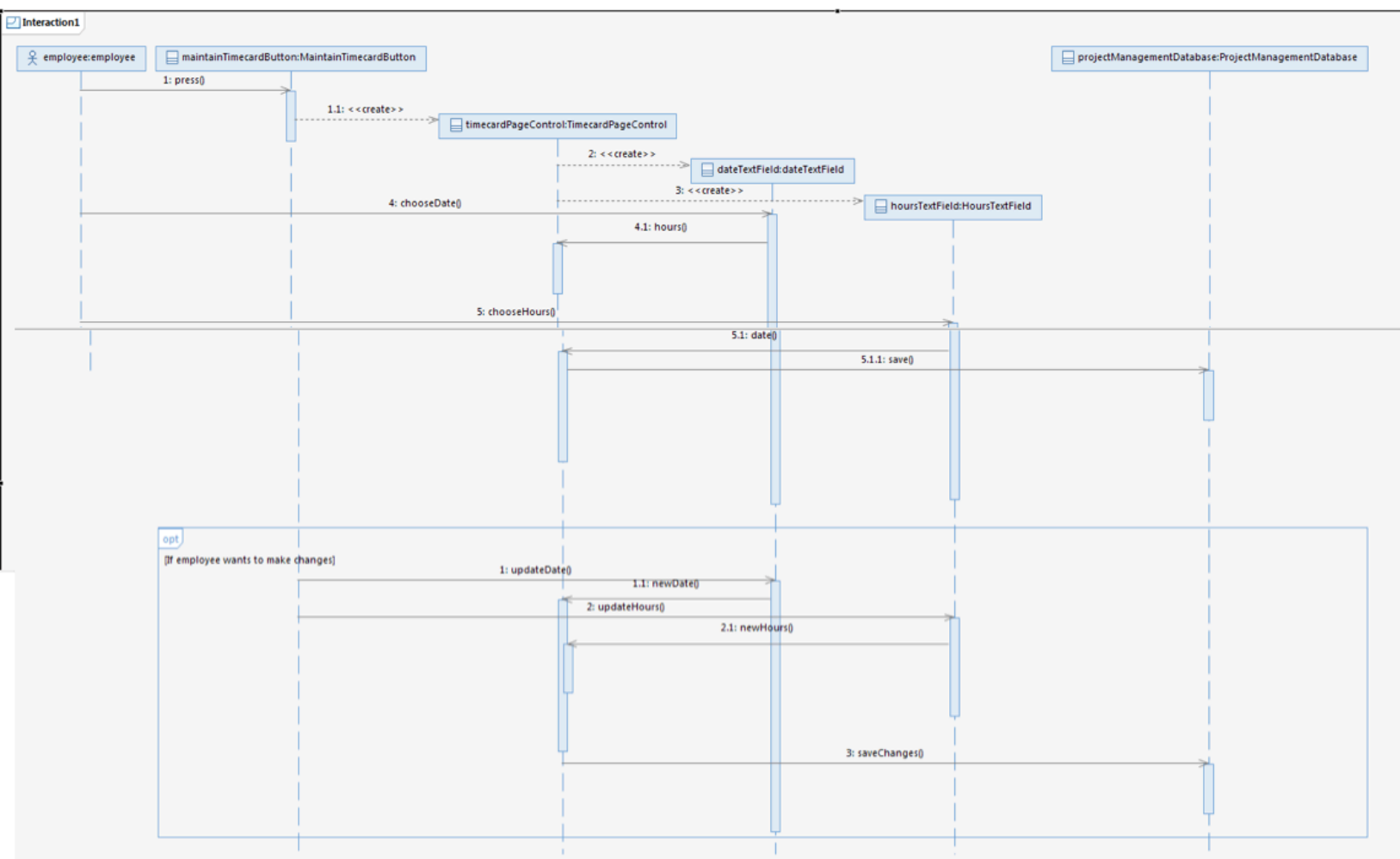
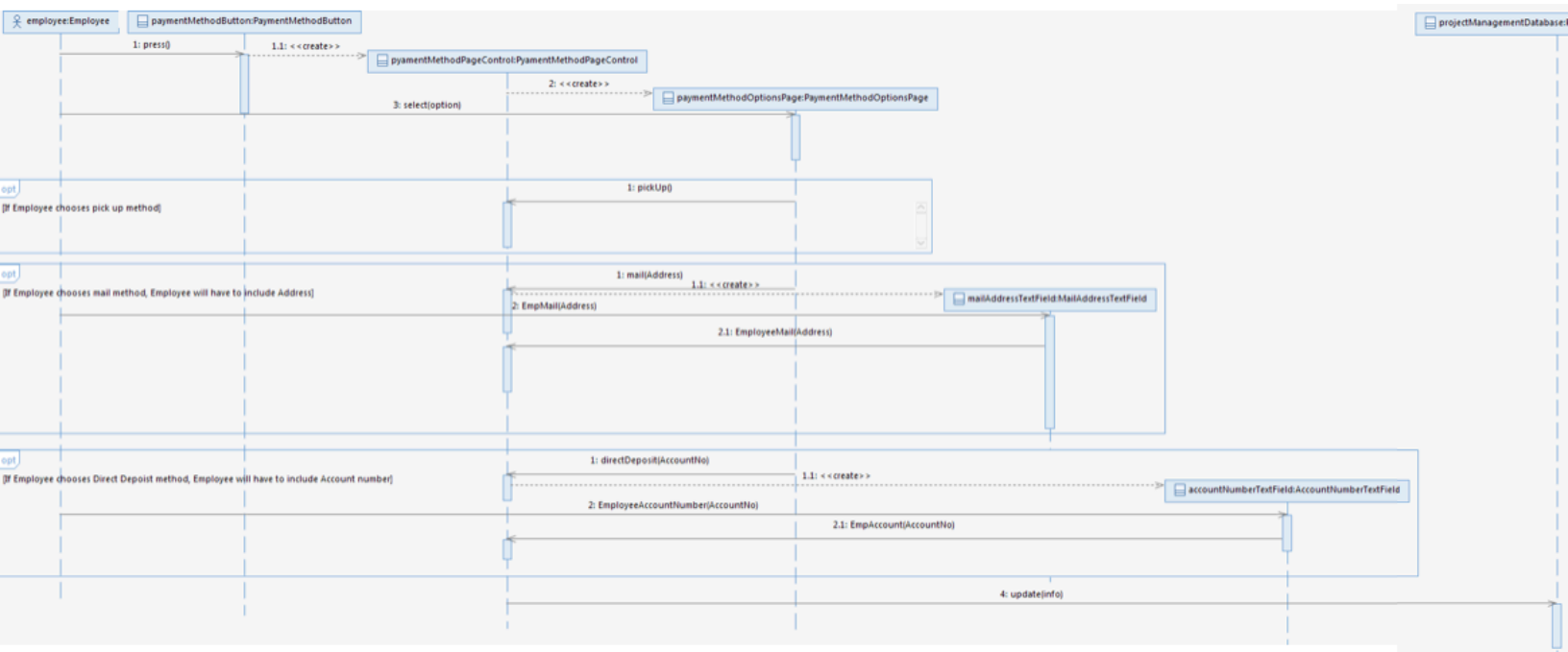(14:Sequence diagram for Maintain Timecard use case)

2-Submit Timecard:
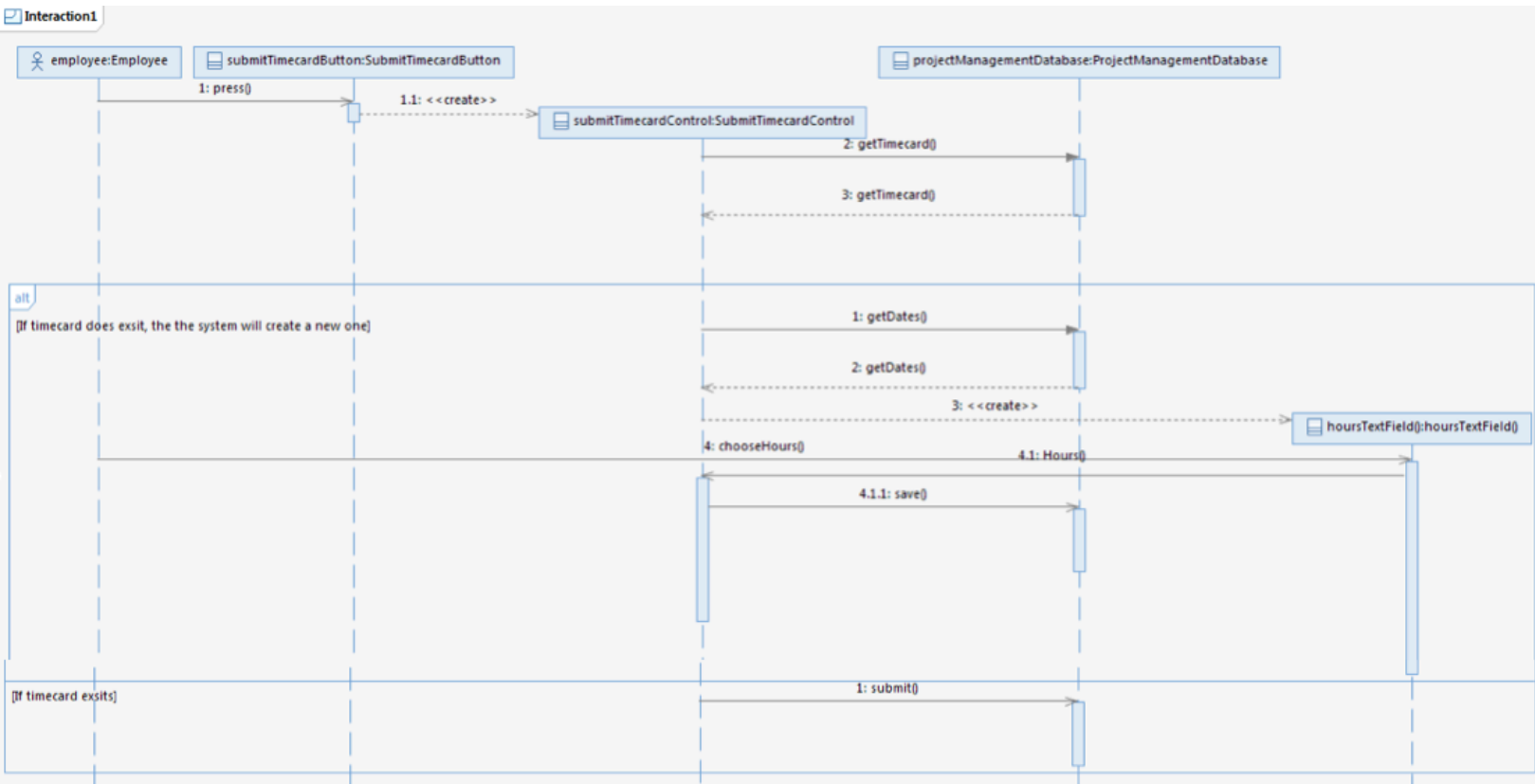
(15:Sequence diagram for Submit Timecard use case)

4-Create Employee Report:
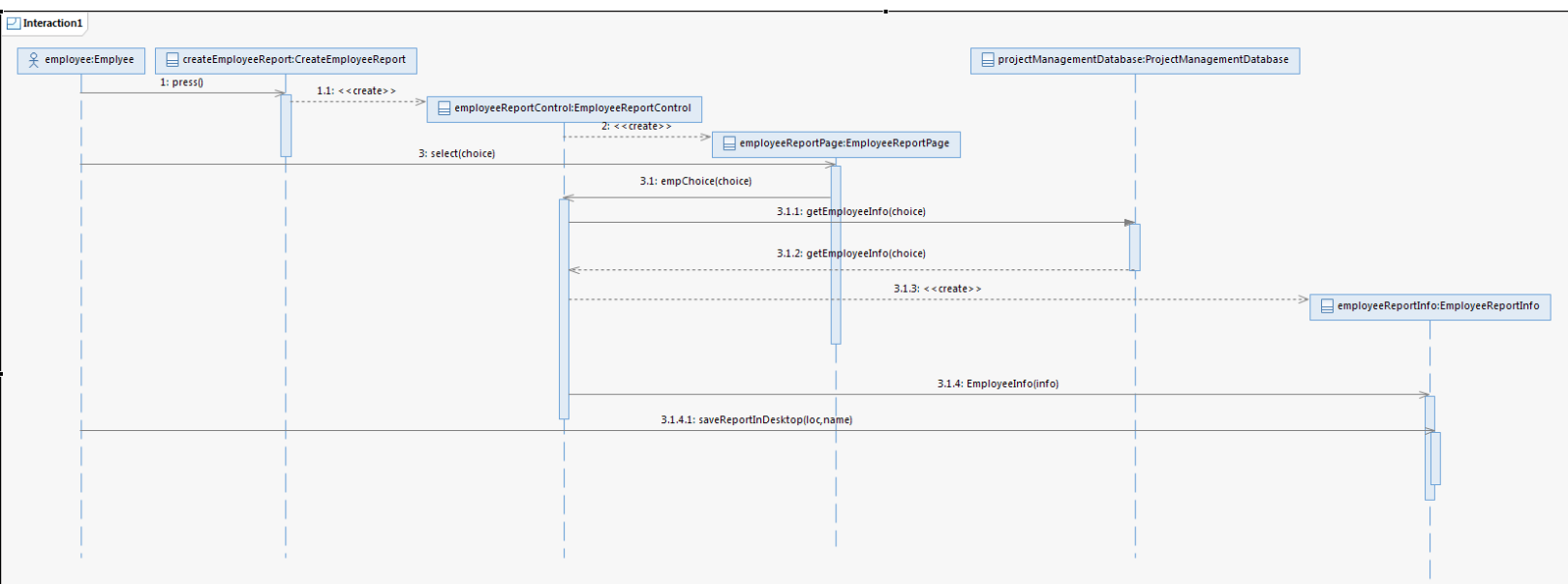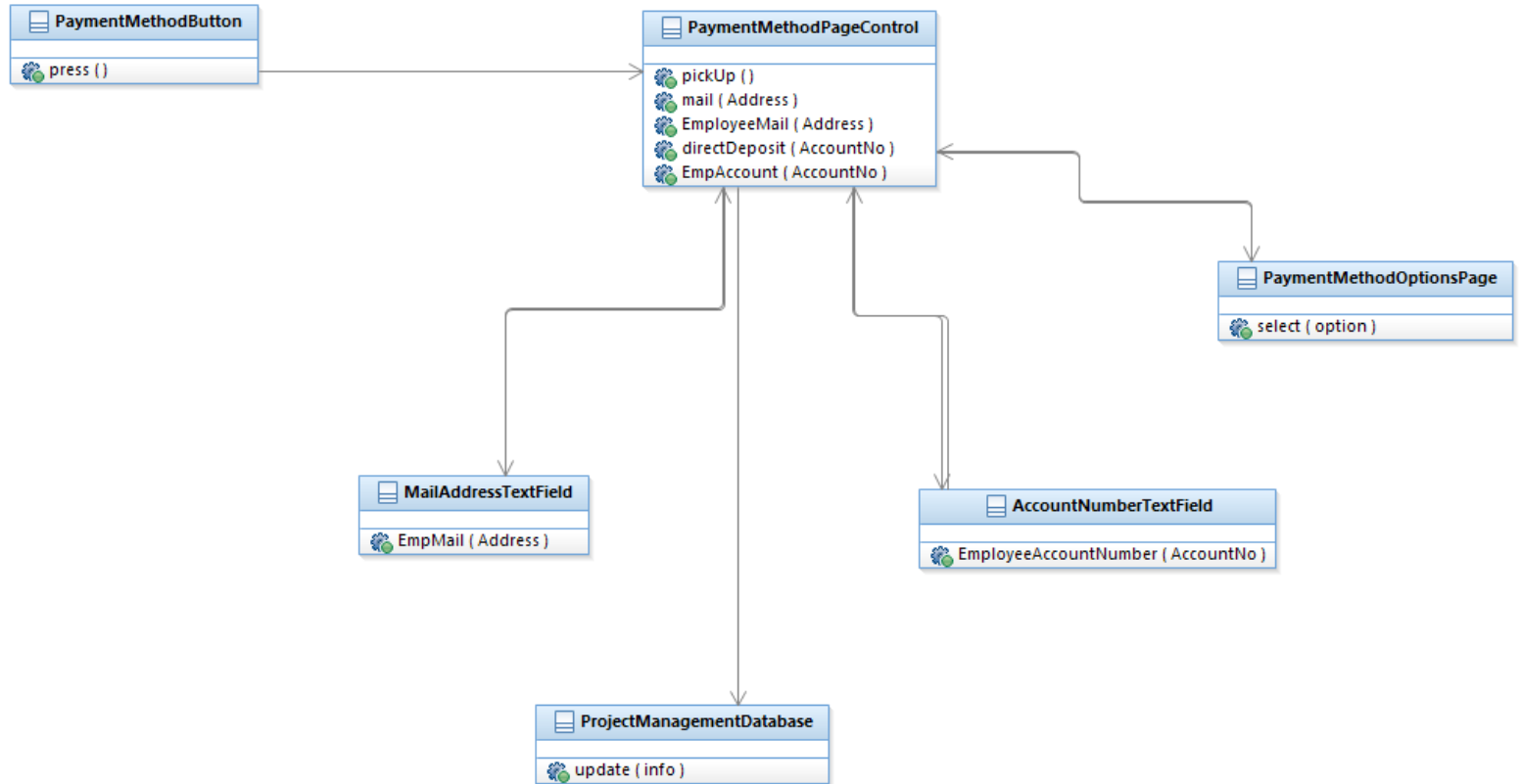
18:Sequence diagram for Create Employee Report use case)

5-Select payment method sequence:

(19:Sequence diagram for Select Payment Method use case)

VOPC:

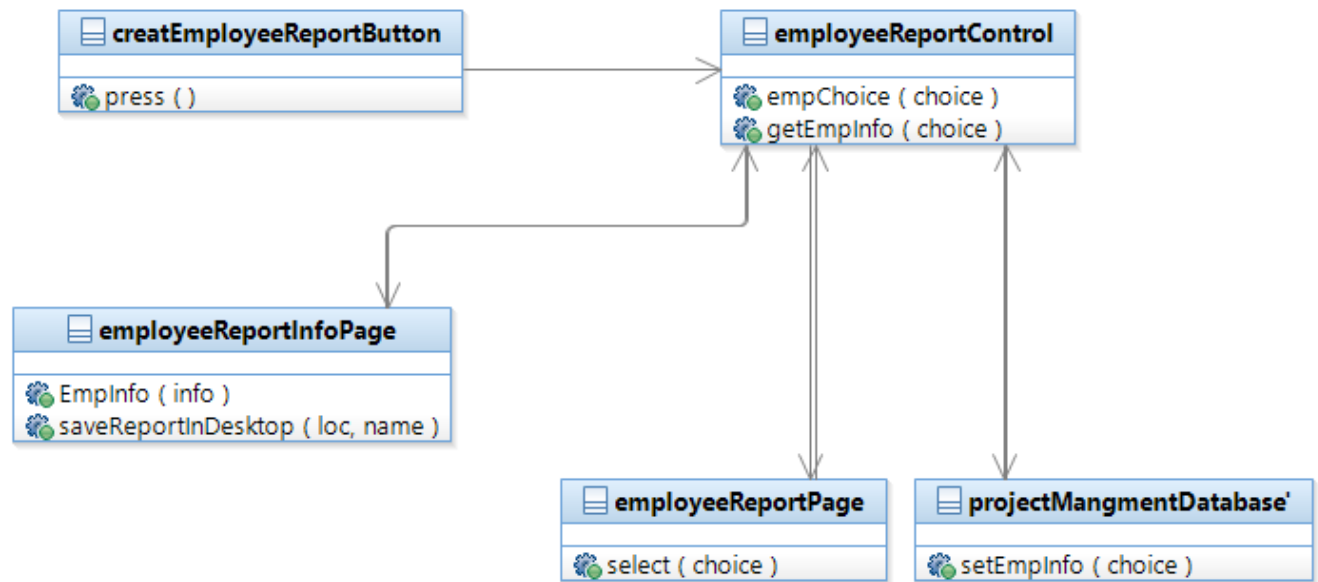1- Select Payment Method :
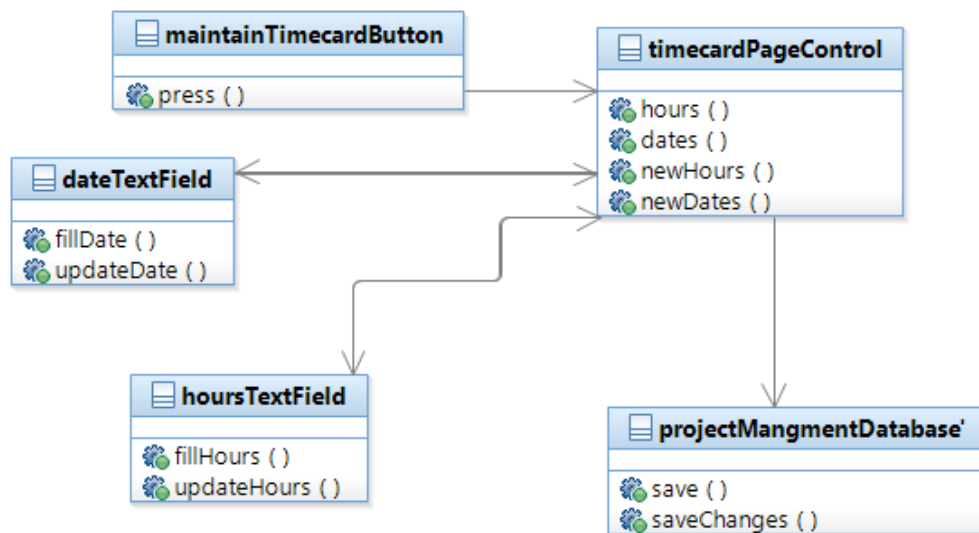

(20: VOPC for Select Payment Method use case)

2- Create Employee Report
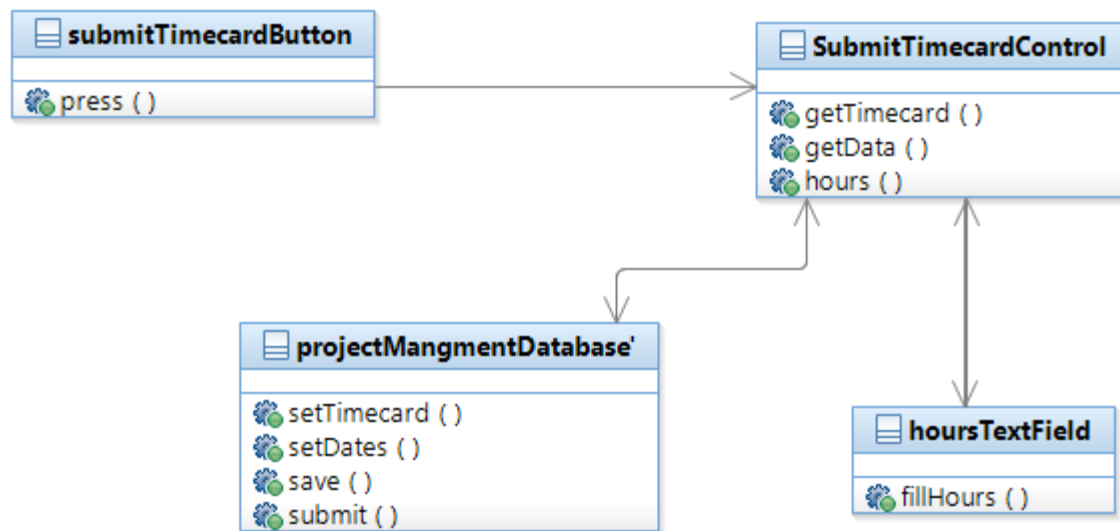

(21:VOPC for Create Employee Report use case)

3- Maintain Timecard

(22:VOPC for Maintain Timecard use case)

4- Submit Timecard

submitTimecardButton
press ( )

SubmitTimecardControl
getTimecard ( )
getData ( )
hours ( )

projectMangmentDatabase'
setTimecard ( )
setDates ( )
save ( )
submit ( )

hoursTextField
fillHours ( )

(23:VOPC for Submit Timecard use case)

Q9:
Security: The system has to authenticate any employee before

submitting the timecard. Technique: https and encrypt commission rate so the informations will stay up secure .

Realibility : the quality and the performing of the system must be reliable

The Technique: when the system become down all the information will be updated before when system stop , no information will be lost , the srart date and end date of the timecard must be correct.

18

Q10:

we chose Client-Sever architecture style because most functions in the system are un form of request and response between user and server . the system is decomposed into two subsystem : one : client (request to the server) two : server (which receives the request) it can be implemented by designing two application layers one for the client and another for server , client and server will run on different processes and on different machines .so if client will be down the system will have a backup client so the system will continue working same to the server side .
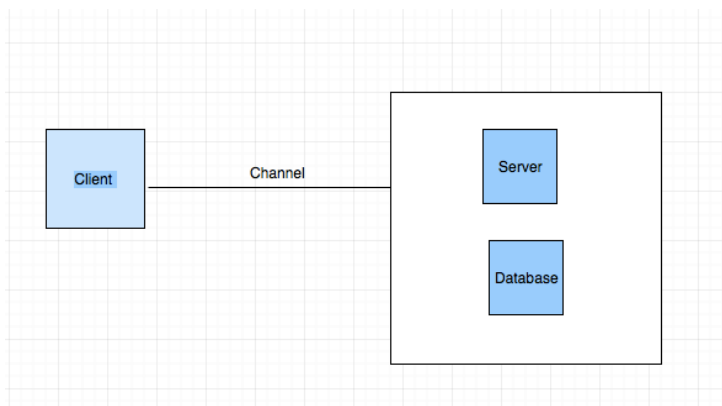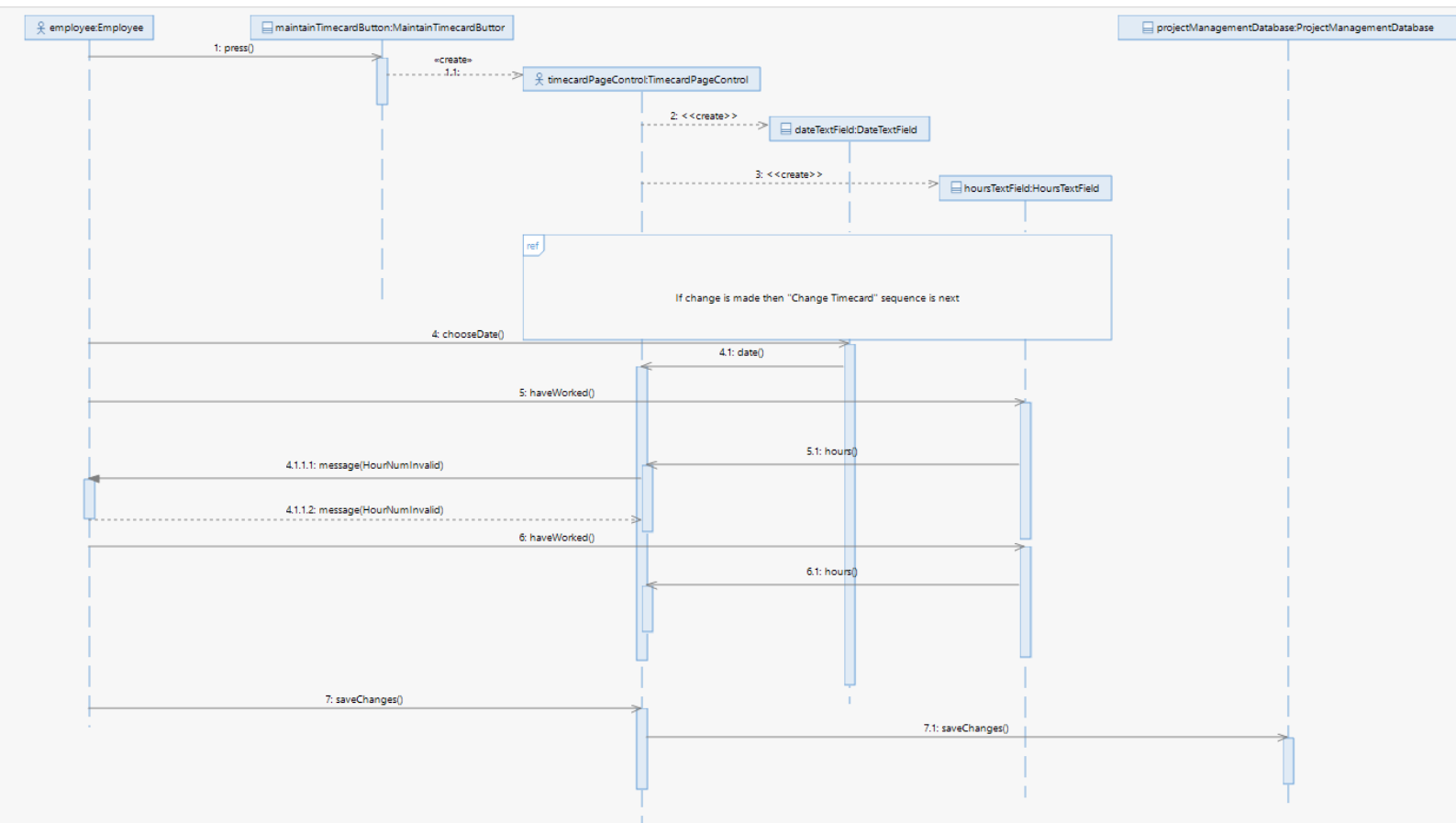

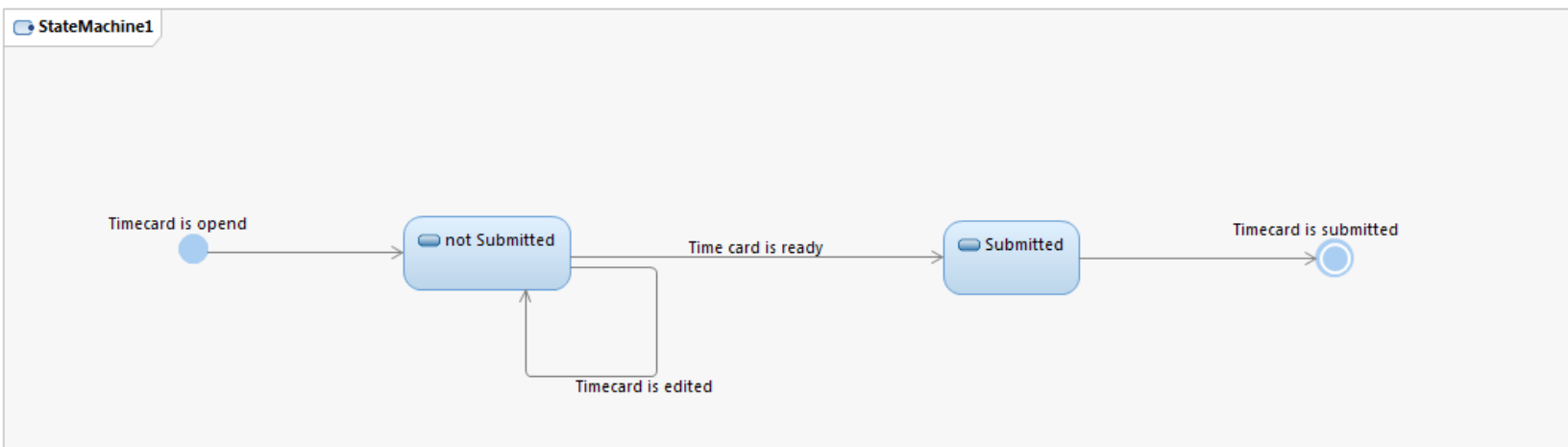(25: Block diagram)

Unified class diagram:
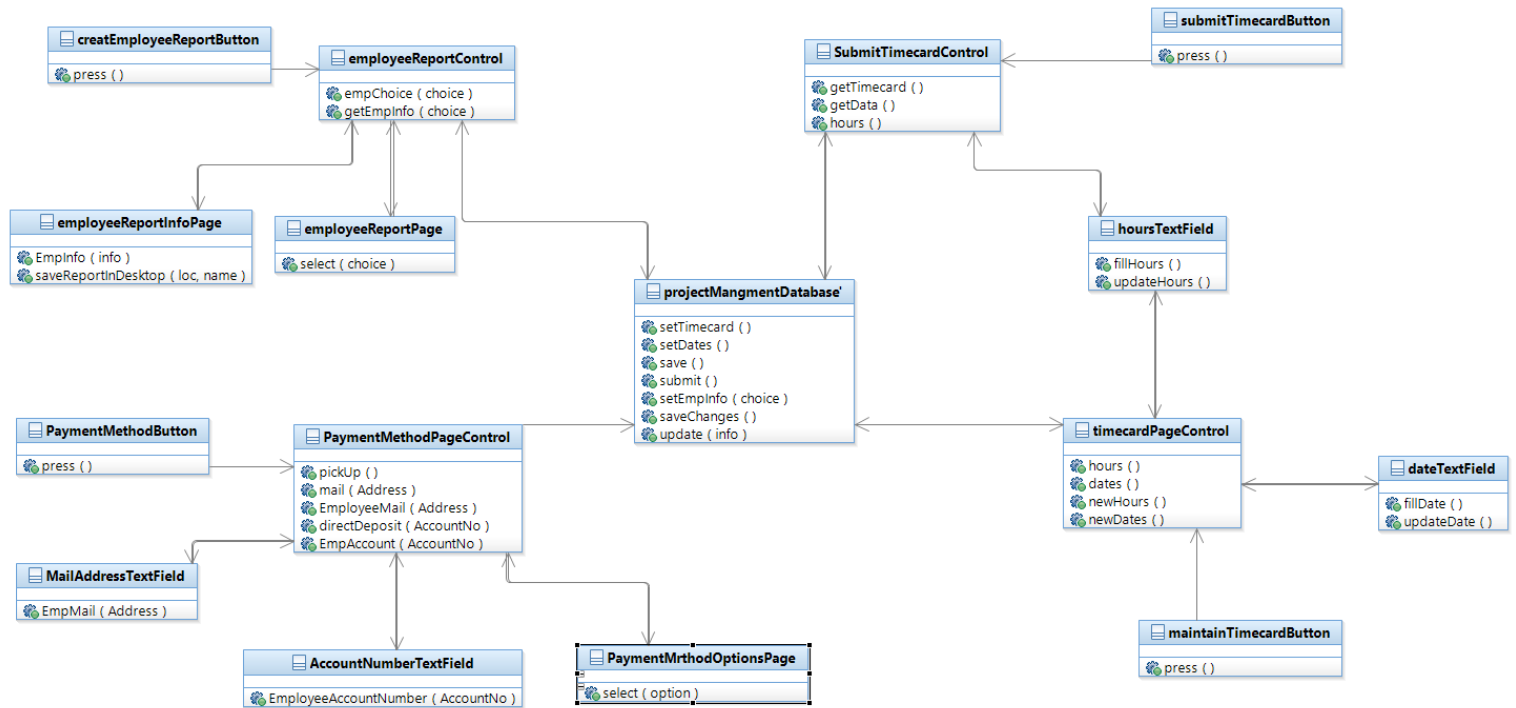
(26: Unified class diagram representing the part of the design model)

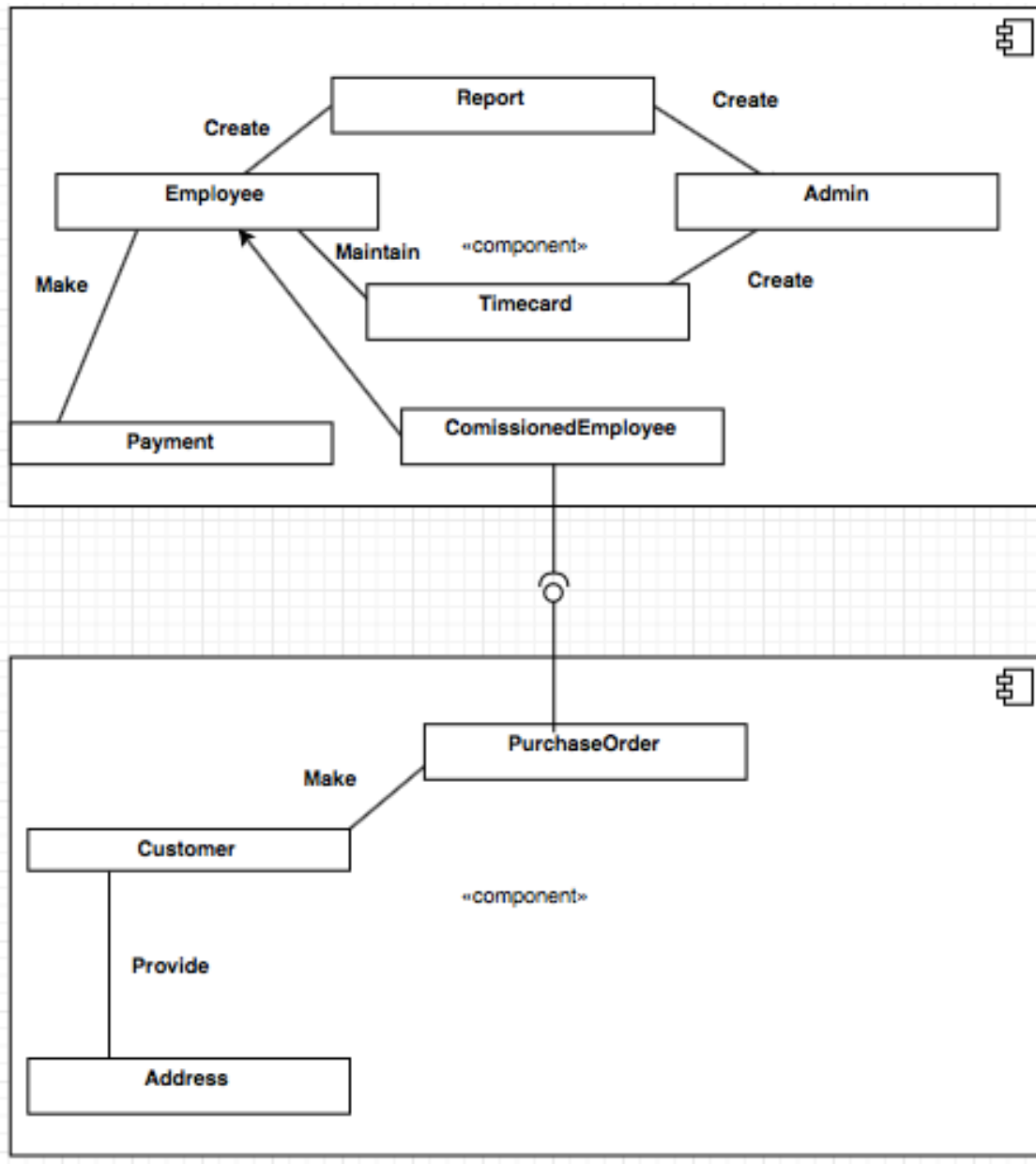# (Change Timecard) use case alternative Sequence diagram:

(27: Sequence diagram of the alternative flow of Change Timecard use case)
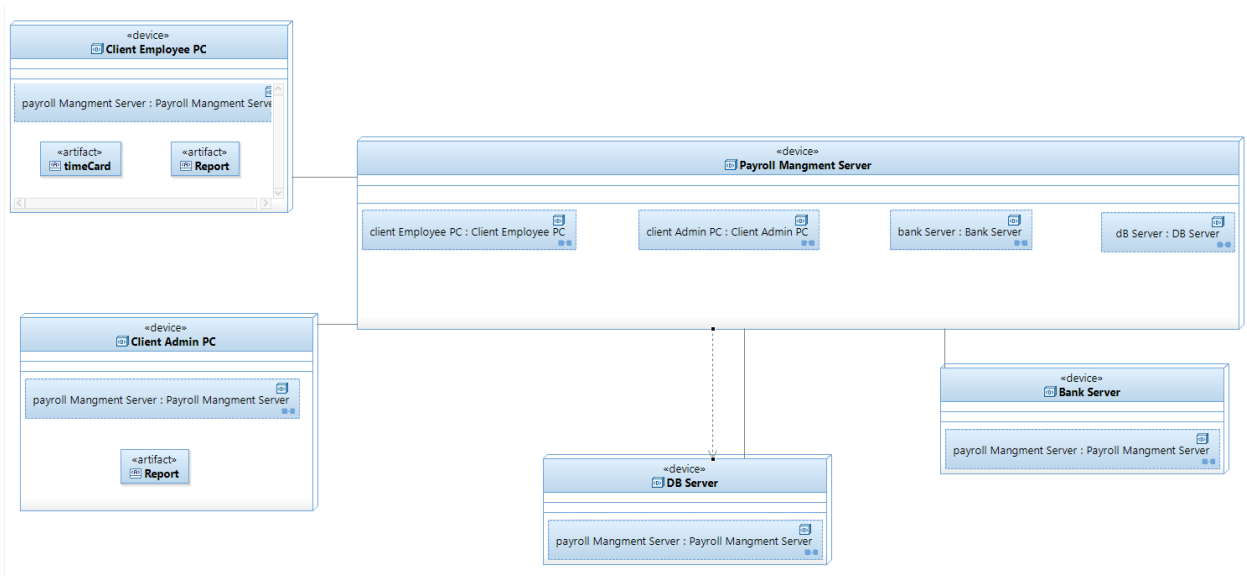
State Machine Diagram of (Maintain timecard):

(26:State machine diagram for Maintain Timecard use case)

Component Diagram:

(27:Component Diagram)

Deployment diagram:

(2 : Deployment diagram for the whole system)

Design Pattern :

Design pattern: Proxy
Why we chose it:
The proxy pattern improve the performance and the security of the system by checking access before loading an object into memory . And use memory only when needed , which we found it suitable with our system .
How to apply it:
An Access association class contains a set of operations that an employee can use to create a report.
The ReportProxy class contains a set of operations that employee use to create a report. The ReportProxy class contains operation to check first with it Accessible() the invoking employee has legitimate access.
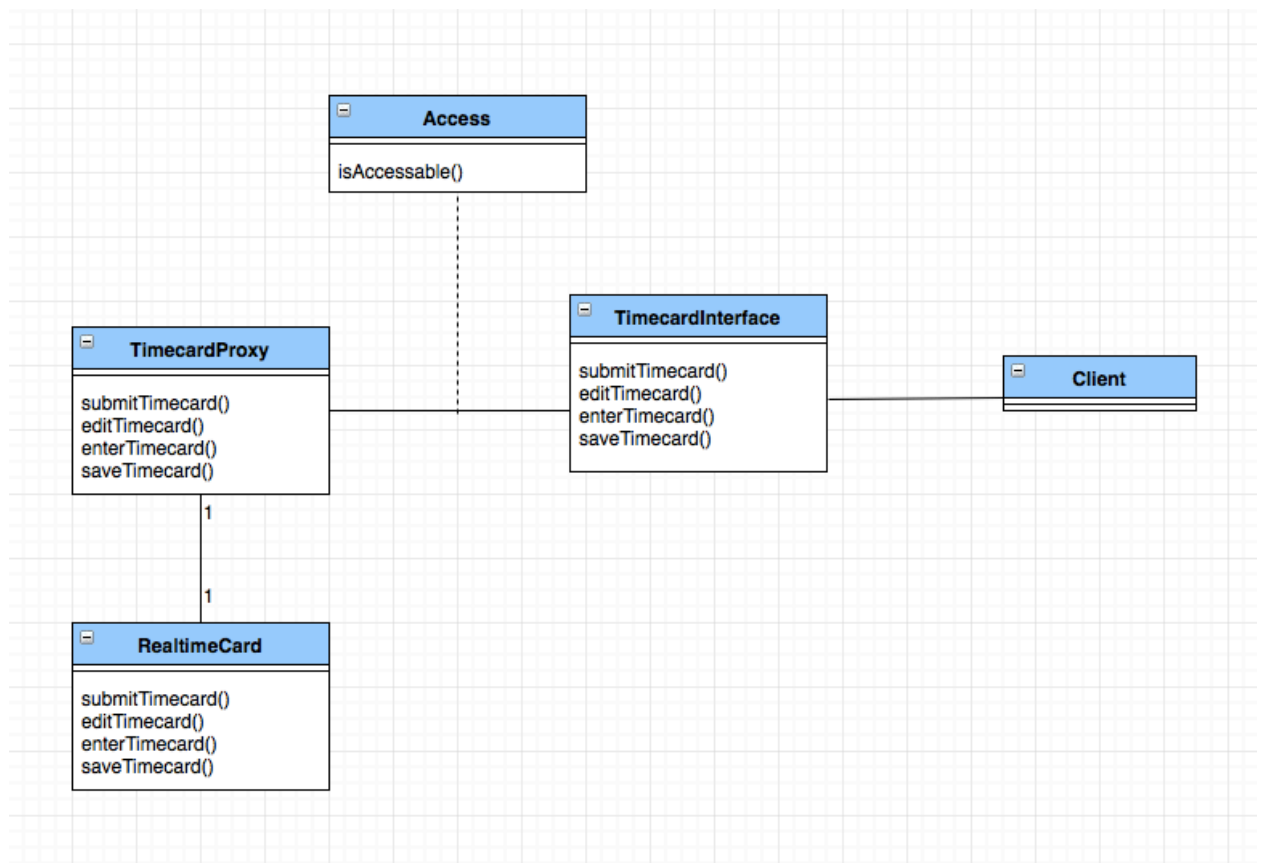Once it access ReportProxy delegates the operation to the actual Report object. If it doesn't access , the actual Report object is not loaded into memory.

Proxy server is a server (a computer system or an application) that acts as an intermediary for requests from clients seeking resources from other servers.A client connects to the proxy server, requesting some service, such as a file, connection, web page, or other resource available from a different server and the proxy server evaluates the request as a way to simplify and control its complexity
A proxy server may reside on the user's local computer, or at various points between the user's computer and destination servers on the Internet.

The Features that proxy provide :
-Monitoring and filtering
-improving performance
-Translation
-security

(29 : Proxy class diagram)