

# Object Oriented Analysis and Design

## Automated Bill Payment System

### Project Deliverable 2



#### Submitted to:

Ma'am Amina mirza

#### Group Members:

Bsef19m013 - Osama Sultan

Bsef19m032 - Ameena Abdullah

Bsef19m029 - Iqra Hanif

# **Automated Bill payments Application**

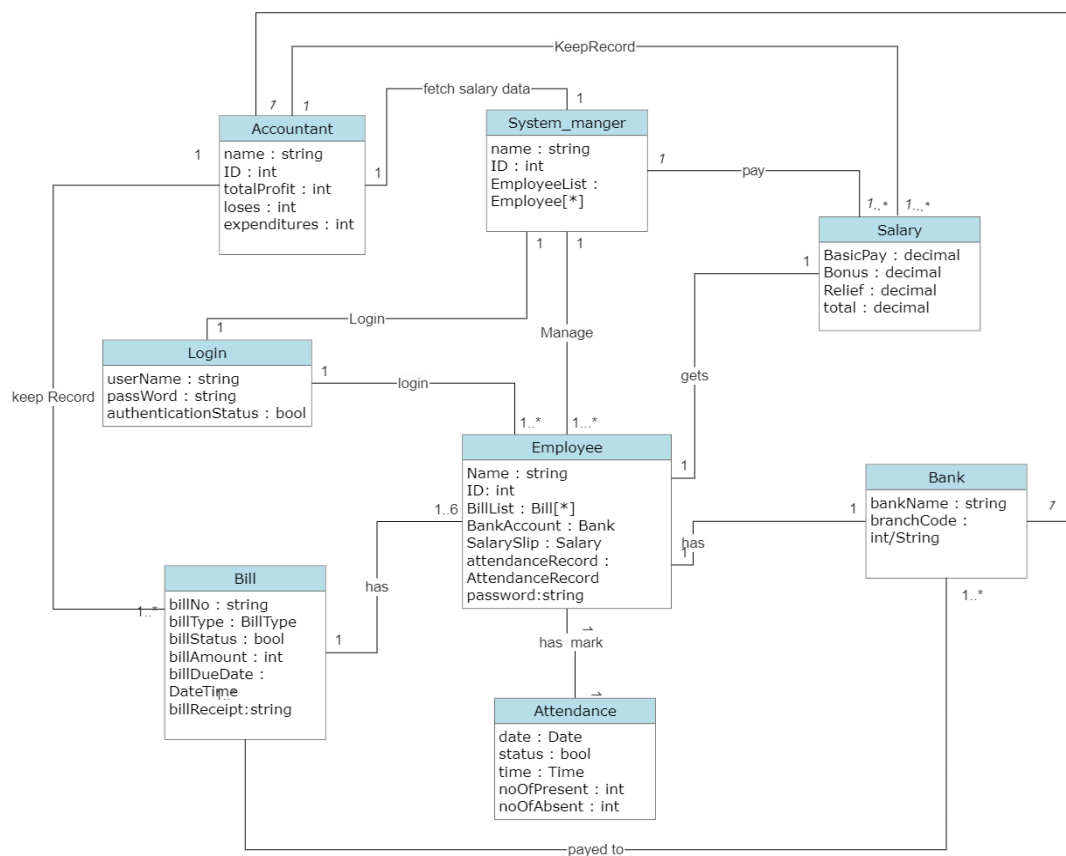
## **Table of Contents**

1. Domain Model
2. Design Class Diagram
3. Sequence Diagram
4. System Sequence Diagram
  - 4.1- Risk Management Use case
  - 4.2- Business Level
  - 4.3- Architecture Level (MVC)
5. Collaboration Diagram
6. Prototypes
  - 5.1- Admin End UI
  - 5.2- Employee End UI
7. Use Case Contracts
8. References

\*All the diagrams in this document are created on [smartdraw.com](https://www.smartdraw.com)

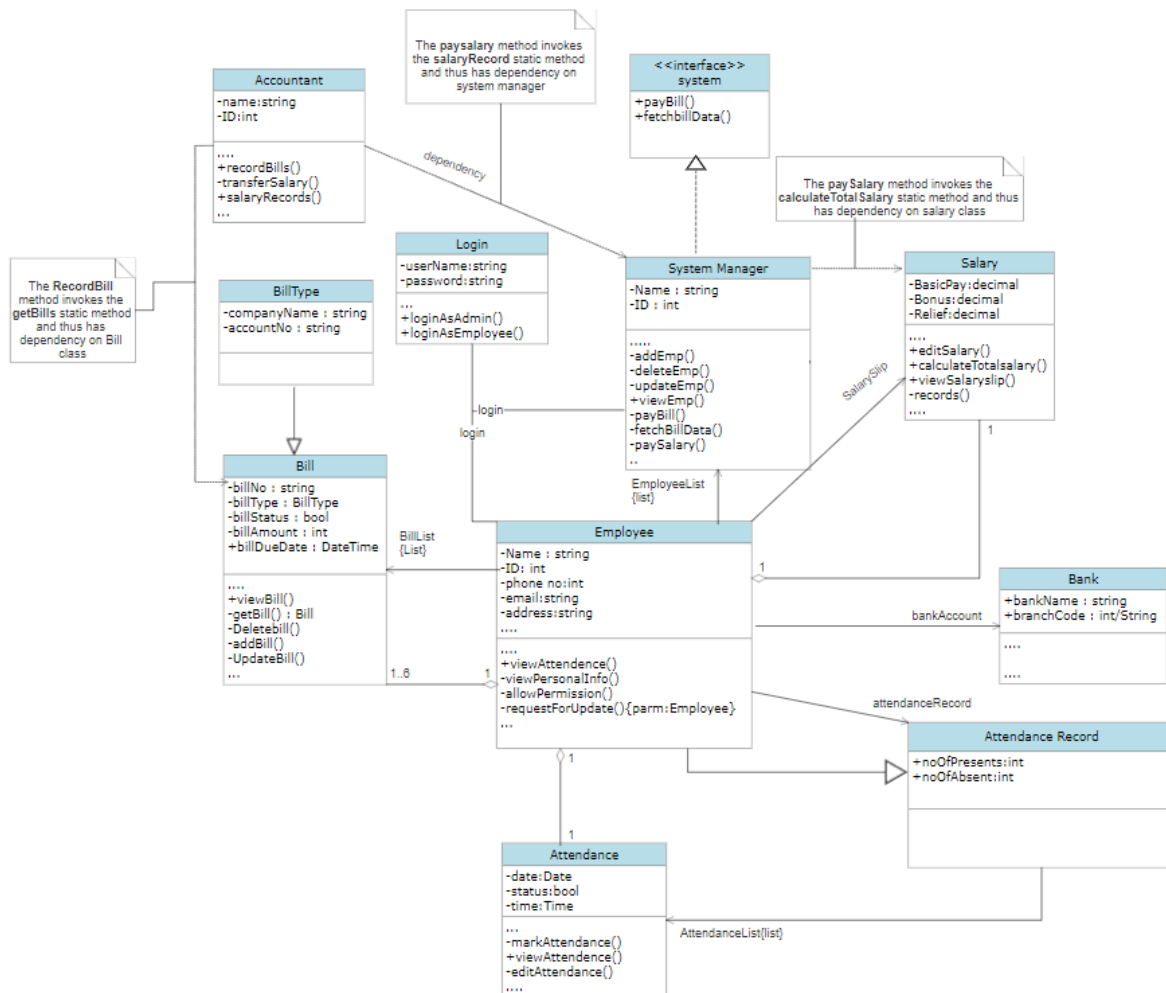
# Domain Model

(Visual representation of conceptual classes of Automated Bill payment System)



# Design Class Diagram

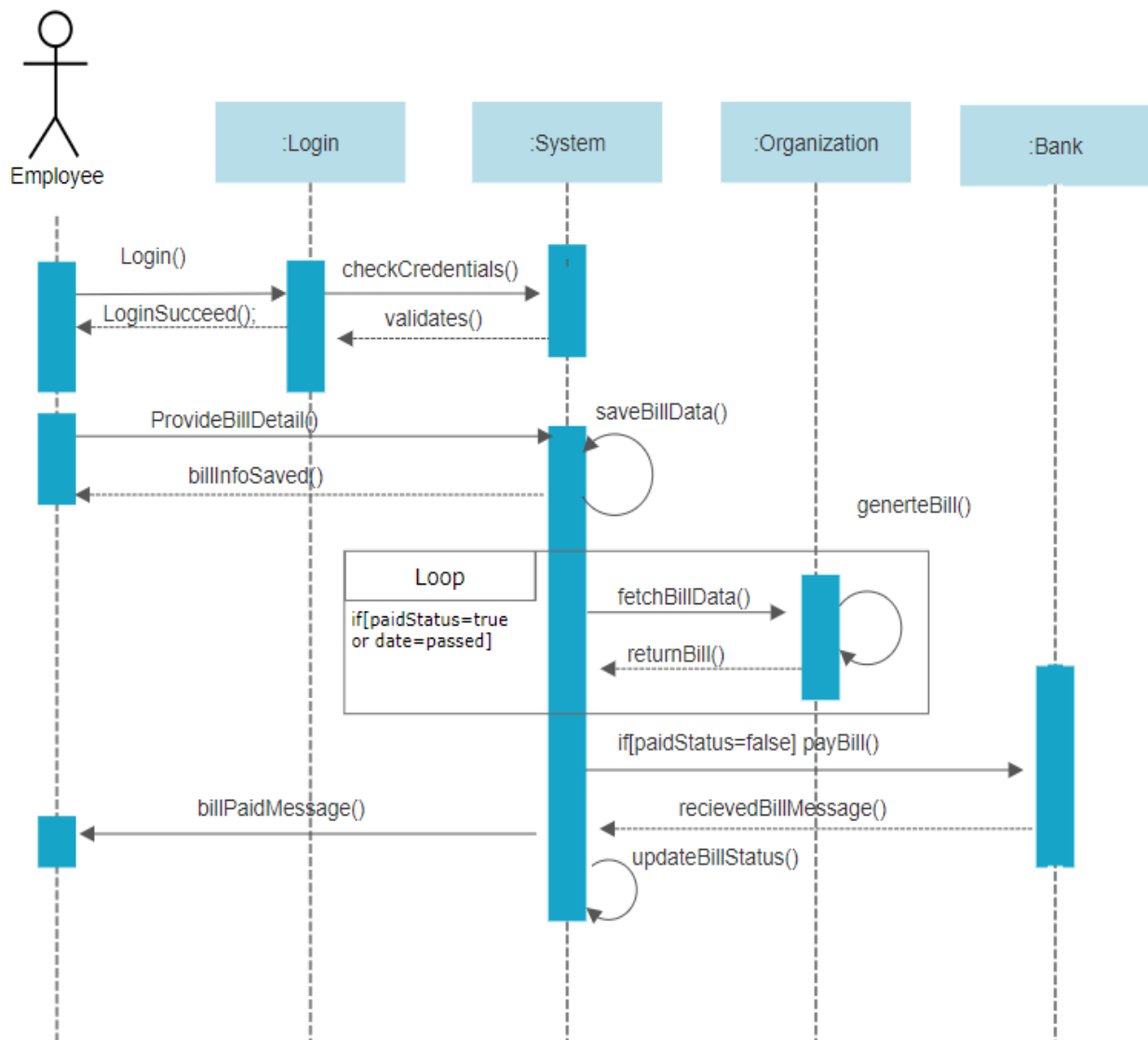
(Detailed Visual Representation of system class and their relationships)



# Sequence Diagram

## (Sequence flow of bill payment among different states)

- Employee with add bill information once and system will then automatically fetch data from organization site, check status and performs accordingly.

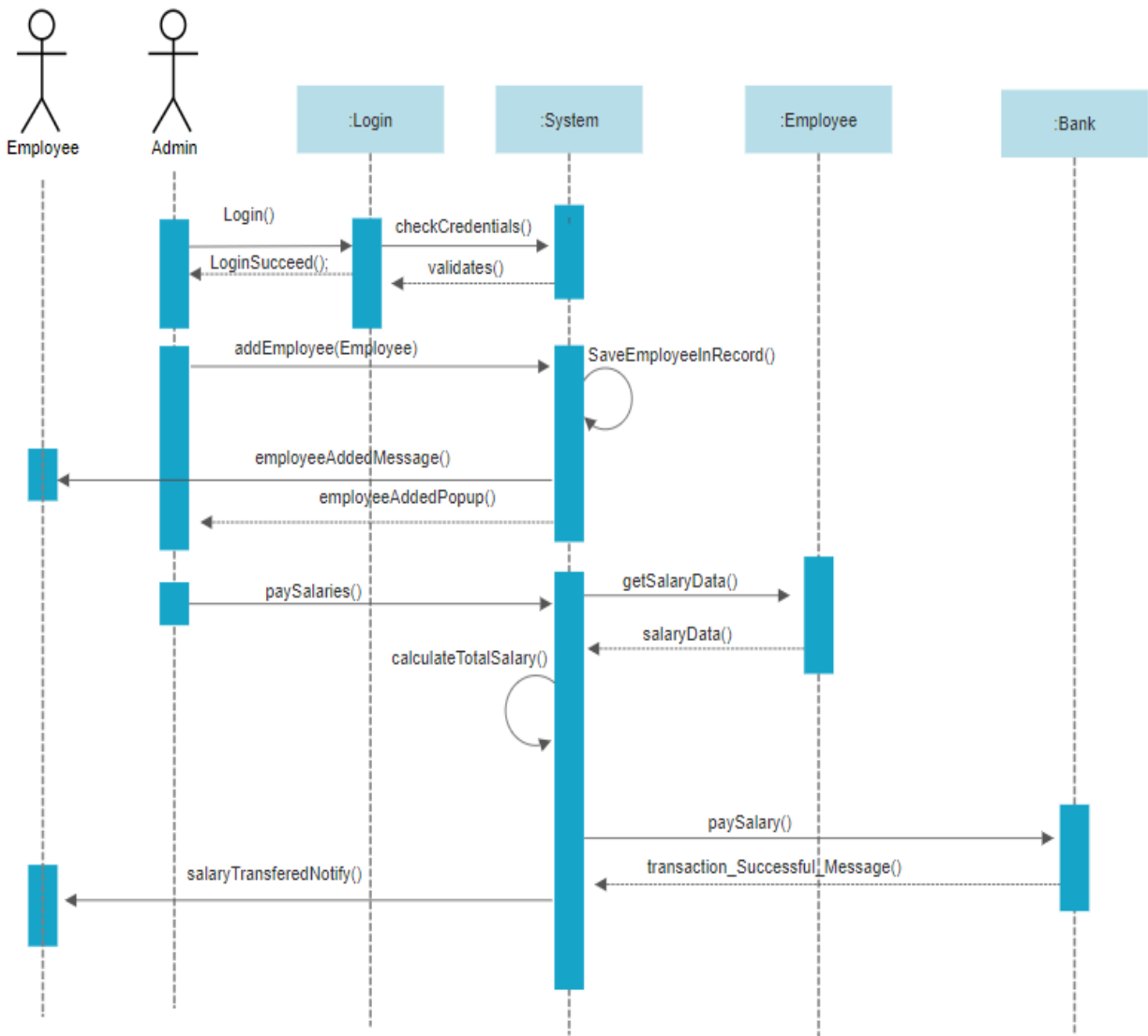


\*We can have multiple actors in a Sequence Diagram

Source: <https://www.geeksforgeeks.org/unified-modeling-language-uml-sequence-diagrams/>

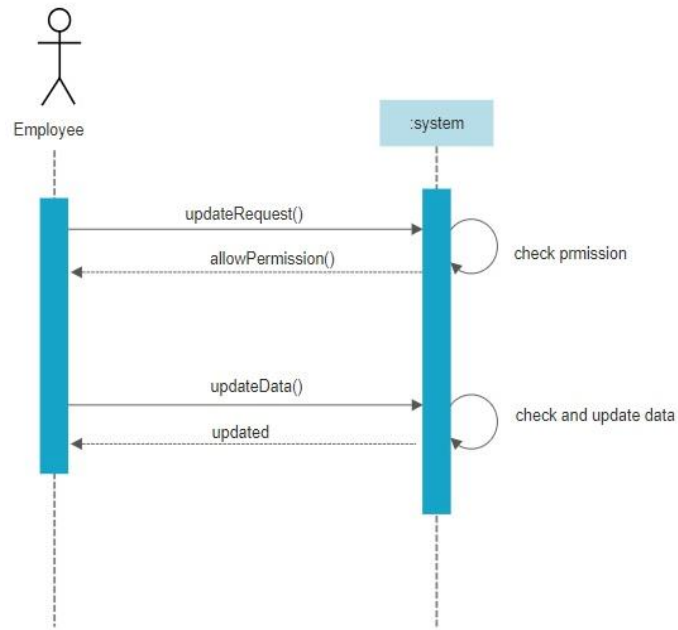
**(Sequence flow of login to add employee and then paying salaries to employees among different states of admin)**

- Admin will login and add employee to employee list then admin will pay salaries to all existing employee. System will calculate salary of each employee after checking employee salary data. Bills will be deducted from salary

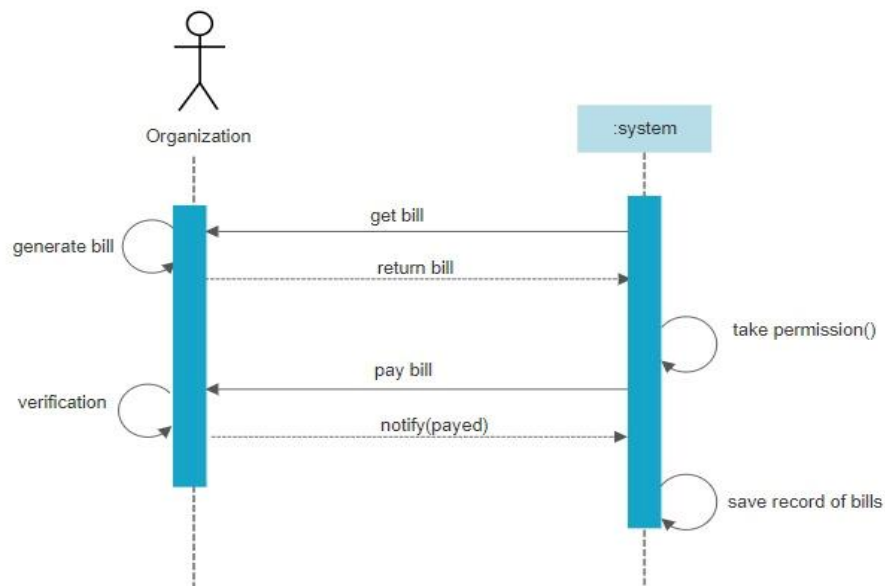


# System Sequence Diagrams

## (System Sequence Flow of Risk Management Use Case)



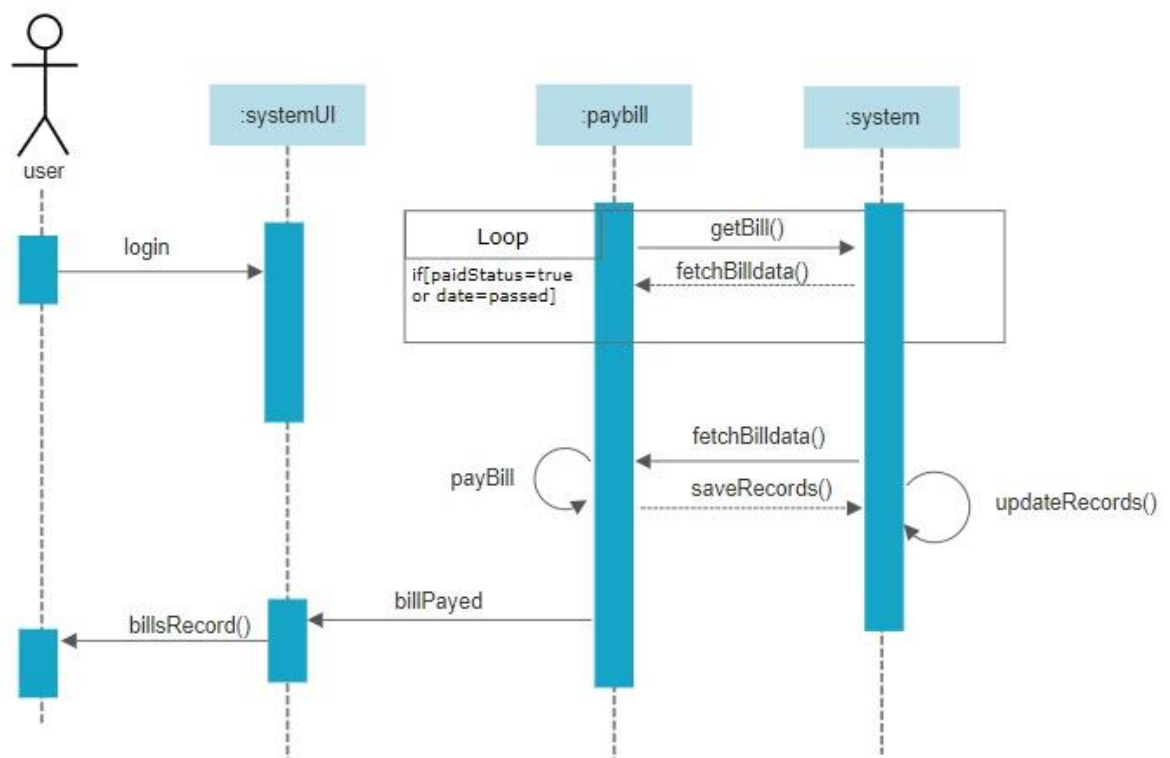
## (System Sequence Flow of Business Management Use Case)



## (System Sequence Diagram of Architecture)

Here system is divided into three parts

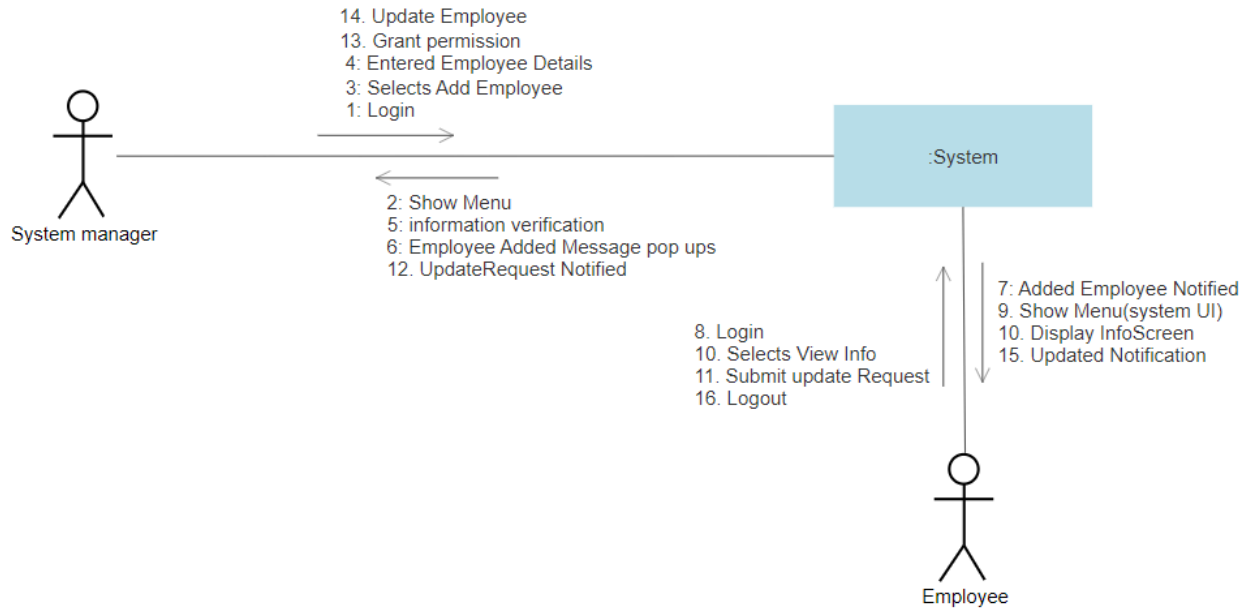
- Model
- View
- Controller



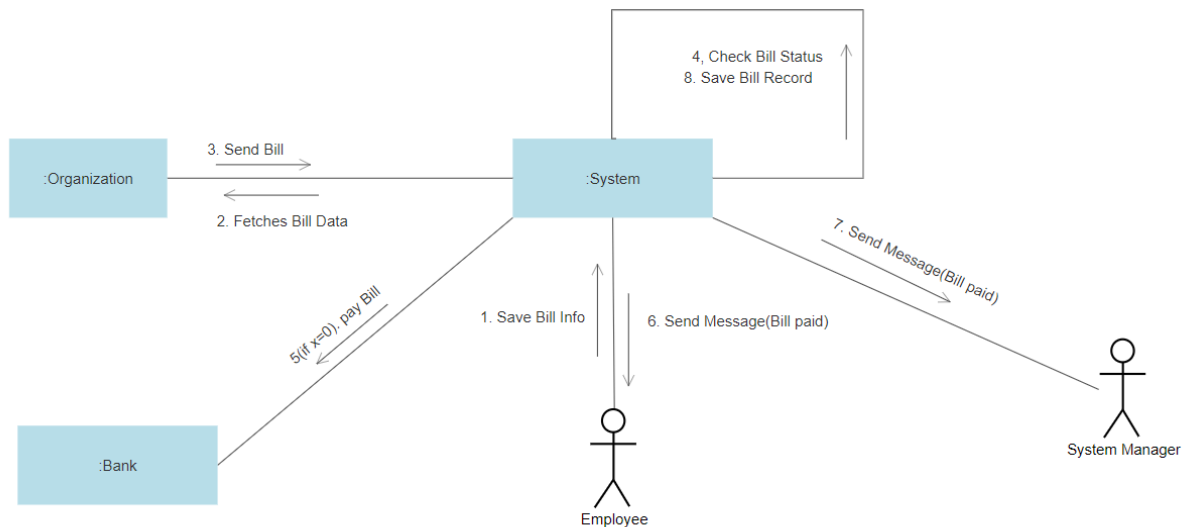


# Collaboration Diagram

## (Risk Use Case- Update Permission and Employee update Request)



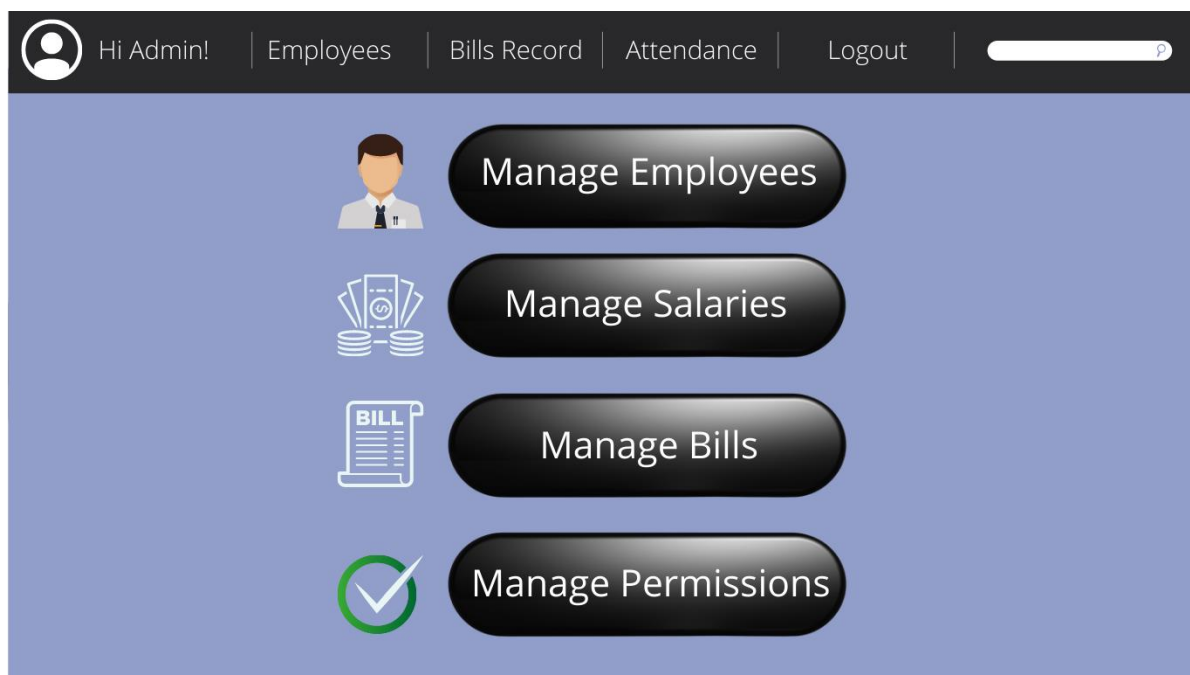
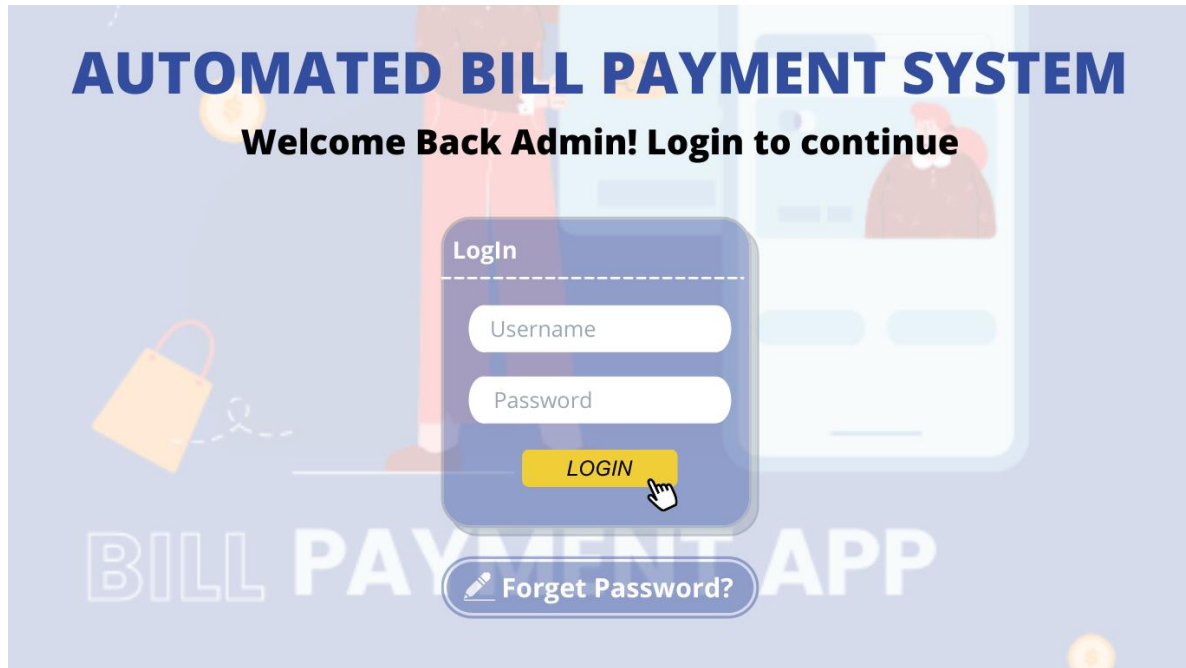
## (Pay Bills and Save Record)



- In message 5 : x means Bill Status and Date and 0 means false

# Prototype's

- Admin End UI:



 Hi Admin! | Employees | Bills Record | Salaries | Logout |


## Manage Employees

 [Add Employees](#)

 [Delete Employee](#)


 [Update Employee](#)


 [View Employees](#)

 Hi Admin! | Employees | Bills Record | Salaries | Logout |

## Manage Salaries

 [Pay Salaries](#)

 [View Salary History](#)

 [Update Employee Salary](#)



Hi Admin!

Employees

Bills Record

Salaries

Logout



## AddEmployee

### Personal Information

Name	<input type="text" value="Bilal"/>	Father Name	<input type="text" value="Adam"/>
Date of birth	<input type="text" value="21/09/1991"/>	NIC No.	<input type="text" value="3740512345678"/>
Mobile No.	<input type="text" value="923331234567"/>	Landline No.	<input type="text" value="0511234567"/>
Nationality	<input type="text" value="Pakistani"/>	Gender	<input checked="" type="radio"/> Male <input type="radio"/> Female
Religion	<input type="text" value="Islam"/>	Active	<input checked="" type="radio"/> Active <input type="radio"/> Inactive
Appointment Date	<input type="text" value="08/09/2016"/>	Joining Date	<input type="text" value="12/09/2016"/>
Passport No.	<input type="text"/>		
Department	<input type="text" value="Admin"/>	Designation	<input type="text" value="Teacher"/>
City	<input type="text" value="Islamabad"/>	Street Address	<input type="text"/>

Save Employee



Hi Admin!

Employees

Bills Record

Salaries

Logout



## Manage Bills



View Bills



Manage Bill Permissions



View Stats



Hi Admin!

Employees



Bills Record


Salaries



Logout















## Employee List


**Filters**  

 +7 [Add to List](#)

125 results   Sort By **Last Name**    Color by **None** 

 Maria Albert New York Vice President, Human Resources 	 Tomas Alvares France General Office Clerk 	 Janette Andrews Atlanta Programmer / Analyst (IT) 
 Marissa Andrews Chicago Director 	 Lloyd Arthur Los Angeles Retail Salesperson 	 Doug Barden San Jose Telemarketer 

- Employee End UI:




**Welcome Back**  
Sign in to continue

Email

Password ☐

☐ Remember me [Forgot Password](#)

**LOGIN**

Employee 

**Menus**




[View Personal Info](#)

[View Salary Slip](#)


[Manage Bills](#)


[View Attendance](#)

[UPDATE INFO REQUEST](#)

Employee 


**Employee Profile**







Name	Ameena Abdullah
Contact	03XXXXXXXXXX
Email	bsef19m032@pucit.edu.pk
Role	Senior Developer

[UPDATE INFO REQUEST](#)

Employee 

**Salary Slip**





Name	Ameena Abdullah
Basic Pay	Rs. XXXXX
Bonus	Rs. XXXX
Deduction	Rs. XXX
Total	Rs. XXXXXXXX

[VIEW DETAILED SLIP](#)



Employee



## Manage Bills



Check The Bills you want to pay  
through system



Electricity Bill



Gas Bill



Internet Bill



Water Bill

ADD NEW BILL

# USE CASE CONTRACTS

## 1. Contract #1:

Operation:	PayBill(Bill_Id: Bill_Id, Amount: integer)
Cross References:	Use Case- Bill Payment
Pre-Condition:	The data is fetched from the billing service
Post Condition:	<ul style="list-style-type: none"><li>• PayBill instance p was created.</li><li>• p is associated with BillRecord.</li><li>• p.Bill_Id becomes Bill_Id.</li><li>• Based on the Bill_Id, the bill is paid.</li></ul>

## 2. Contract #2:

Operation:	AllowPermission(GrantPermission:GrantPermission)
Cross References:	Use Case- Manage Employee Permission
Pre-Condition:	Employee has logged in and, on the panel, to either check mark or not.
Post Condition:	<ul style="list-style-type: none"><li>• An instance of Employee e is created.</li><li>• Instance e is associated with System Manager.</li><li>• If an employee marks yes which displays the information on UI.</li></ul>

## 3. Contract #3:

Operation:	LoginEmployee(Username:string, Password:string)
Cross References:	Use Case- Login
Pre-Condition:	Employee has to visit the app, and enter the login credentials.
Post Condition:	<ul style="list-style-type: none"><li>• An Employee e instance is created.</li></ul>



	<ul style="list-style-type: none"> <li>• If the credentials match with the record, the user is logged in.</li> <li>• If credentials fail to match, an instance f is called that says forget password notification.</li> </ul>
--	---

#### 4. Contract #4

Operation:	ViewEmpInfo(DisplayInfo:DisplayInfo)
Cross References:	Use Case- View Employee Information
Pre-Condition:	User is logged as employee e into the app and able to click on the view information button.
Post Condition:	<ul style="list-style-type: none"> <li>• e.DisplayInfo is called.</li> <li>• UI is displayed to the user with all his information.</li> </ul>

#### 5. Contract #5

Operation:	SendUpdateRequest(EmpInfo:Empinfo)
Cross References:	Use Case- Update Request
Pre-Condition:	Employee is logged in to the app and is on UI where he can view information.
Post Condition:	<ul style="list-style-type: none"> <li>• An instance manager m is created.</li> <li>• Employee is associated with the system manager.</li> <li>• .EmployeeInfo is called.</li> <li>• If an employee's information needs updating, a request will be sent to the admin panel.</li> </ul>

#### 6. Contract #6

Operation:	UpdateInfo()
Cross References:	Update Employee Information

Pre-Condition:	Employee's information update requests have been received to the admin panel (one who has logged in as employee.
Post Condition:	<ul style="list-style-type: none"> <li>• An instance employee e is created</li> <li>• Instance e.employeeInfo is associated with the manager.</li> <li>• If an update request is received, the admin updates the employees information.</li> </ul>

### 7. Contract #7

Operation:	PaySalary(EmployeeSalary:EmployeeSalary, Amount: Integer)
Cross References:	Use Case- Salary Details
Pre-Condition:	Admin is logged in to the system to pay salary to the employees.
Post Condition:	<ul style="list-style-type: none"> <li>• Instance salary p is already created.</li> <li>• A function that calculates the net salary is called.</li> <li>• p.EmployeeSalary becomes EmployeeSalary</li> <li>• Net salary is transferred to the employee's bank.</li> </ul>

### 8. Contract #8

Operation:	ViewAttendance(Attendances:Attendances, Absents:Integer, NoOfDays: Integer)
Cross References:	Use Case- View Attendance
Pre-Condition:	Employee is logged in to the system and clicks on View Attendance button to see his attendance record.
Post Condition:	<ul style="list-style-type: none"> <li>• An instance employee e is created.</li> <li>• e.Attendance is associate with Attendances</li> </ul>

	<ul style="list-style-type: none"> <li>• Attendance record is displayed.</li> </ul>
--	---

## 9. Contract #9

Operation:	MarkLeave(Leave:Leave, No_ofAttendances:Integer)
Cross References:	Use Case- Attendance System
Pre-Condition:	Employee is on panel/page to mark leave.
Post Condition:	<ul style="list-style-type: none"> <li>• An instance employee e is already created.</li> <li>• It is associated with Leave.</li> <li>• A request is sent to the admin to either mark leave or decline.</li> </ul>

## 10. Contract #9

Operation:	BillRecords()
Cross References:	Use Case- update Bill Records
Pre-Condition:	Manager is logged in to the system and clicks on the update bill Records button to update records.
Post Condition:	<ul style="list-style-type: none"> <li>• An instance employee e is created.</li> <li>• e.updatebill is associated with billRecord .</li> <li>• Updated record stored in system.</li> </ul>

## References

- <https://www.geeksforgeeks.org/unified-modeling-language-uml-sequence-diagrams/>
- <https://creately.com/blog/diagrams/sequence-diagram-tutorial/>
- <https://searchsoftwarequality.techtarget.com/definition/collaboration-diagram#:~:text=A%20collaboration%20diagram%2C%20also%20known,the%20role%20of%20each%20object.>