JavaTech Group

Payroll Management System
Software Architecture Document

Version 1.0

Revision History

Date	Version	Description	Author
20/12/2019	1.0	Final	Quản Trọng Tú

Table of Contents

Software Architecture Document

1. Introduction

1.1 Purpose

This document provides a comprehensive architectural overview of the system, using a number of different architectural views to depict different aspects of the system. It is intended to capture and convey the significant architectural decisions which have been made on the system.

1.2 Scope

This document applies to the Payroll Management System which will be developed by JavaTech Group.

1.3 Definitions, Acronyms, and Abbreviations

- Employee – a person who work for the IT company.

- Accountant – an employee who manage salary.

- Administrator – a person who manage system.

1.4 References

None.

1.5 Overview

In the following section, architectural design of the Payroll Management System is provided in detail. First, the primary software architecture of the system will be defined. Then, there are further discussion about the goals and constraints that will be imposed upon the quality of the final product, which including but not limited to security, distribution and reuse. In the precedence sections, the key views of the system are demonstrated to depict different aspects of the system. Lastly, criteria concerning with size, performance and quality of the system will be proposed.

2. Architectural Representation

This document presents the architectural as a series of mandatory views: Use-Case View, Logical View and Data View. These views are presented as Visual Paradigm Community Edition Models and use the Unified Modeling Language (UML).

Use-Case View

- **Audience:** all the stakeholders of the system, including the end-users.

- **Area:** describes the set of scenarios and/or use cases that represent significant, central functionality to the system.

- **Related artifacts:** Use-Case Model, Analysis Model, Use-Case-Realization documents.

Logical View

- **Audience:** designers, programmers.

- **Area:** functional requirements: describes the design's object model.

- **Related artifacts:** Design Model.

Data View

- **Audience:** data specialists, database administrators.

- **Area:** persistence: describes the architecturally significant persistent elements in the data model.

- **Related artifacts**: Data Model.

3.   Architectural Goals and Constraints

There are some key requirements and system constraints that have a significant bearing on the architecture. They are:

- The Payroll Management System must be designed to fulfills all system requirements specified in requirements definition.

- The Payroll Management design must be structured to be robust, easy to change if and when functional requirements change.

- The Payroll Management System must be designed to allow the re-use of business logic across applications; therefore, the design separate the three components: model, view and controller.

- The separation of the three components: model, view and controller are also necessary to provide a convenient cooperation between different development teams.

- The Payroll Management System will run on a dedicated platform with access to a database.

- The Payroll Management website provides most of the content display. An interface to this system must be capable of handling large traffic volumes.

4.   Use-Case View

A description of the Use-Case View of the system architecture. The Use Case View is important input to the selection of the set of scenarios and/or use cases that are the focus of an iteration. It describes the set of scenarios and/or use cases that represent some significant, central functionality. It also describes the set of scenarios and/or use cases that have a substantial architectural coverage (that exercise many architectural elements) or that stress or illustrate a specific, delicate point of the architecture.

The significant use cases in this system are listed below:

- Login/Logout
- Change Password
- Manage Report
- View Report
- Manage Department
- Manage Role
- View Profile
- Export File
- Manage User
- Approve-Reject Leave
- Apply Feedback
- Approve-Reject Feedback
- Manage Salary
- Apply Leave

- Manage Project
- Manage Working Time
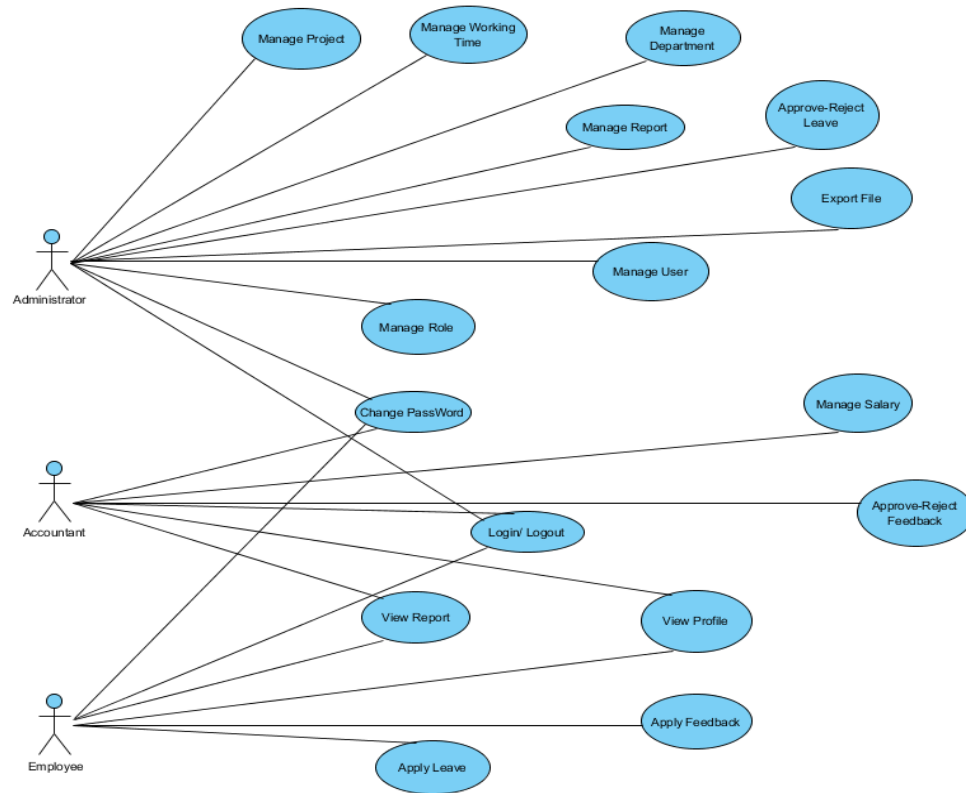
4.1 Use-Case Realizations

- **Login:**
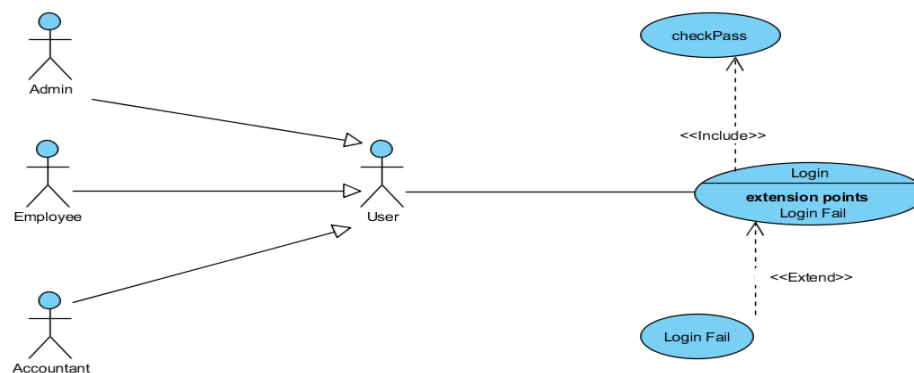
- **Brief Description:** A user login into System.

- **Specification:** See Use-Case-Realization Specification: Login.
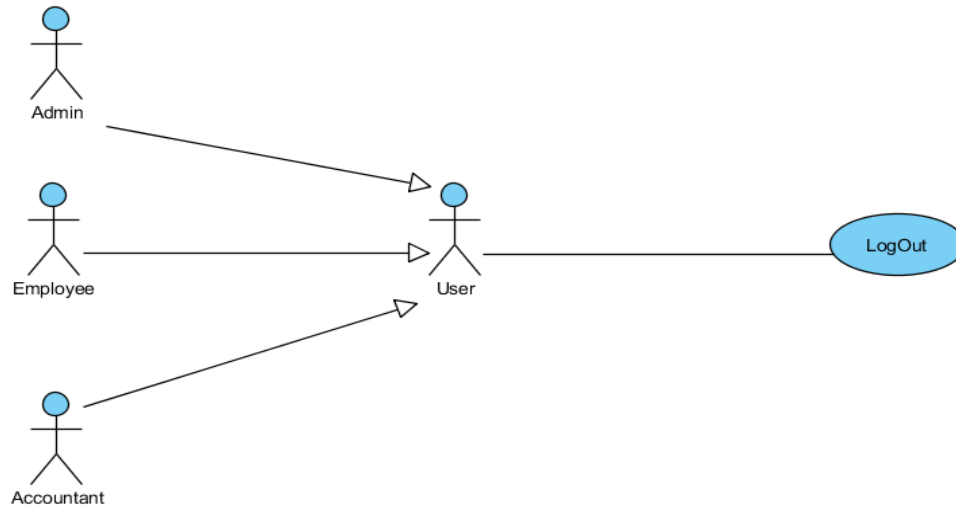
- **Log out:**

- • **Brief Description:** A user logout to the system.

- • **Specification:** See Use-Case-Realization Specification: Log out

- **Change Password:**
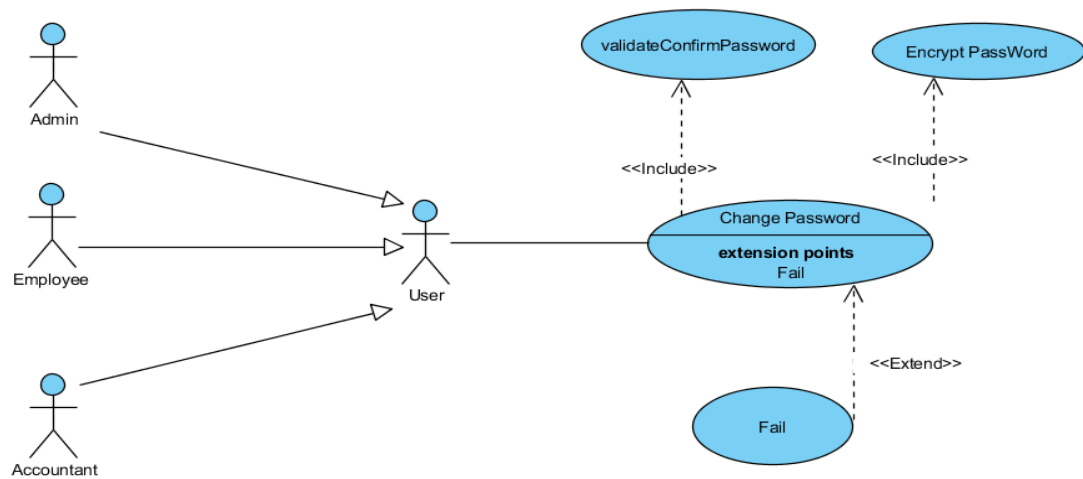
- • **Brief Description:** A user change Password

- • **Specification:** See Use-Case-Realization Specification: Change Password
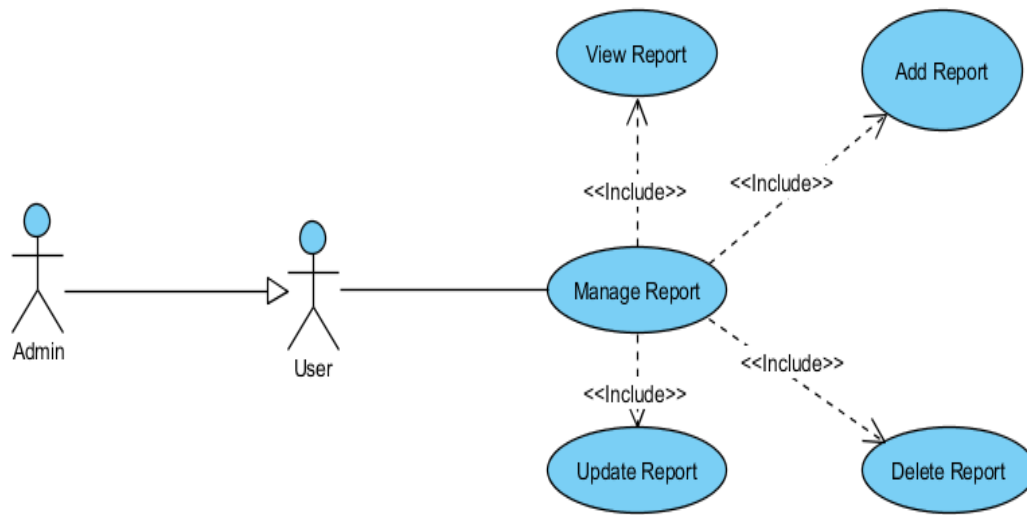
- **Manage Report**

- **Brief Description:** Admin manages report Salary as Notification
- **Specification:** See Use-Case-Realization Specification: Manage Report

- **View Report:**
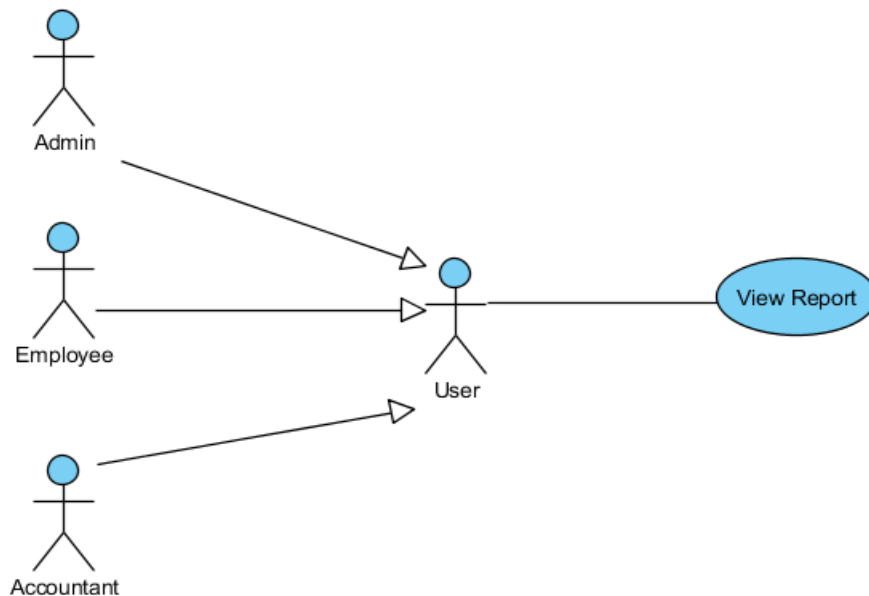
- **Brief Description:** A user can view Report to check Salary

- **Specification:** See Use-Case-Realization Specification: View Report
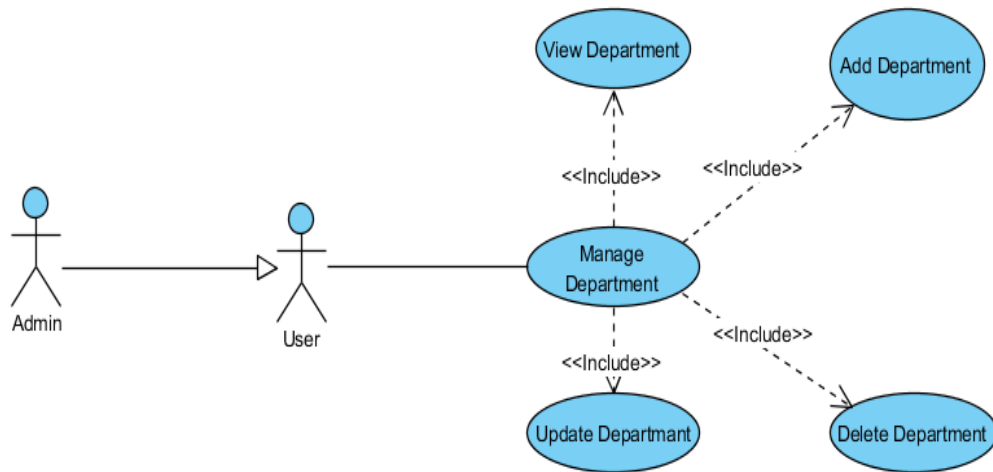
- **Manage Department:**

- • **Brief Description:** Admin can manage department of company.
- • **Specification:** See Use-Case-Realization Specification: Manage Department
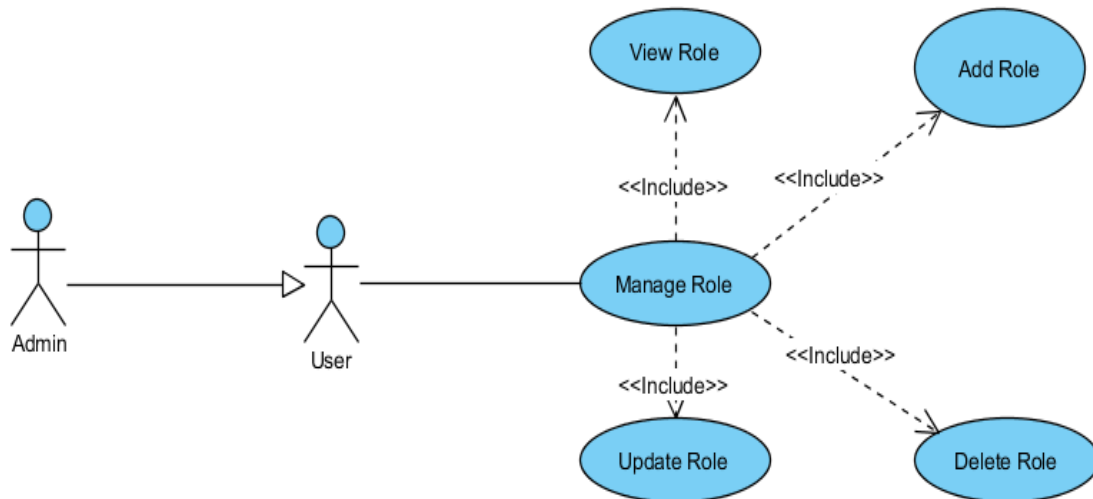
- **Manage Role**

- • **Brief Description:** Admin can manage Role of company.
- • **Specification:** See Use-Case-Realization Specification: Manage Role
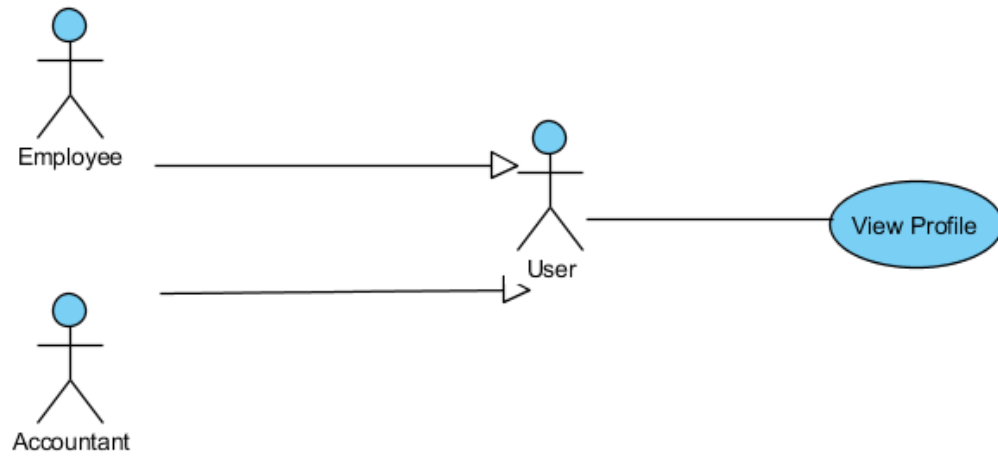
- **View Profile:**

- **Brief Description:** Employee and Accountant can view profile, Admin just view profile through manage user.

- **Specification:** See Use-Case-Realization Specification: View Profile

- **Export File:**
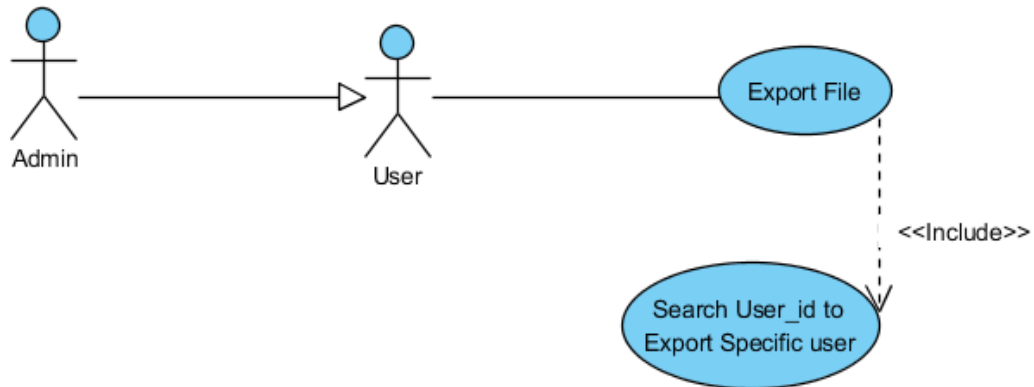
- **Brief Description:** Admin export user's profile if it exists.

- **Specification:** See Use-Case-Realization Specification: Export profile
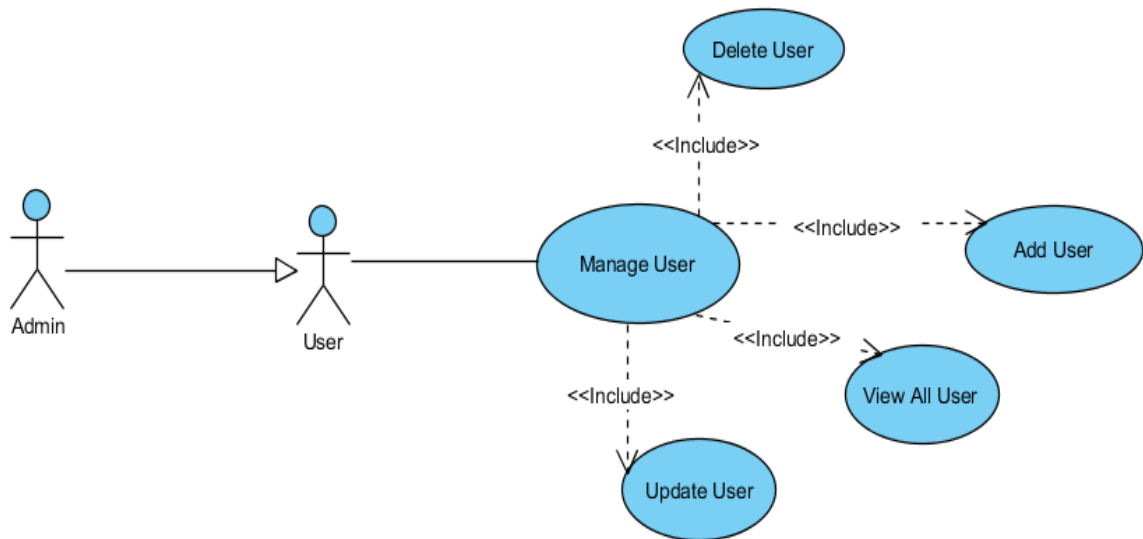
- **Manage User:**

- • **Brief Description:** Admin can manage User by sub-function.
- • **Specification:** See Use-Case-Realization Specification: Manage User.

- **Manage Salary:**
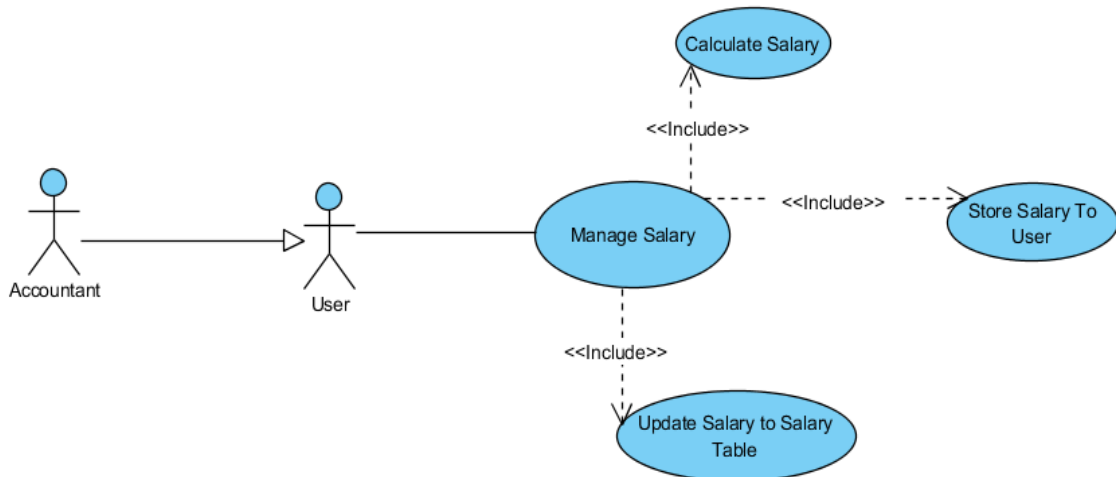
- • **Brief Description:** Accountant can manage Salary by sub-function.
- • **Specification:** See Use-Case-Realization Specification: Manage Salary
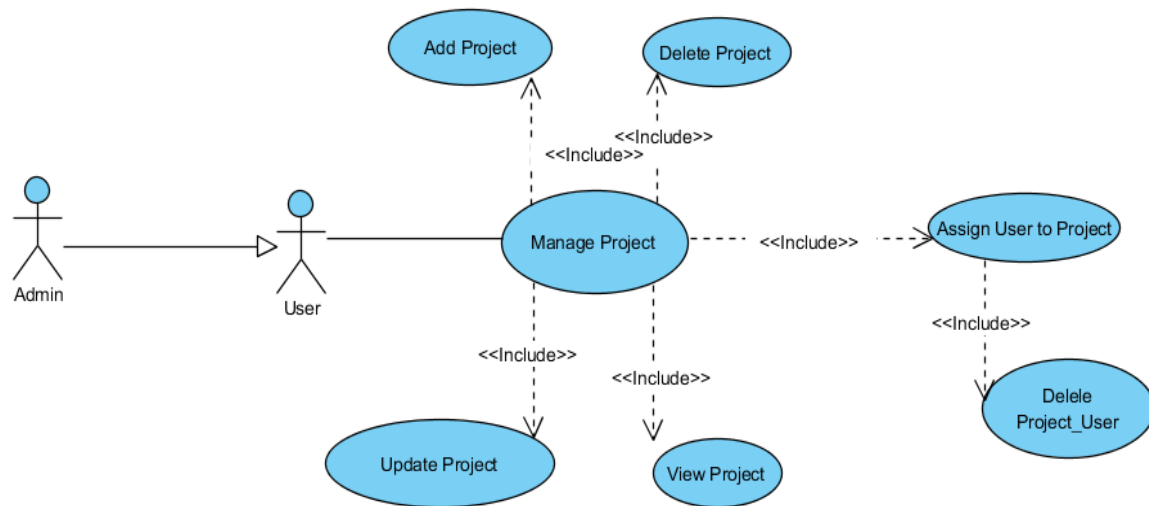
- **Manage Project:**

- **Brief Description:** Admin can manage Project by sub-function.

- **Specification:** See Use-Case-Realization Specification: Manage Project.

- **Manage Working Time:**
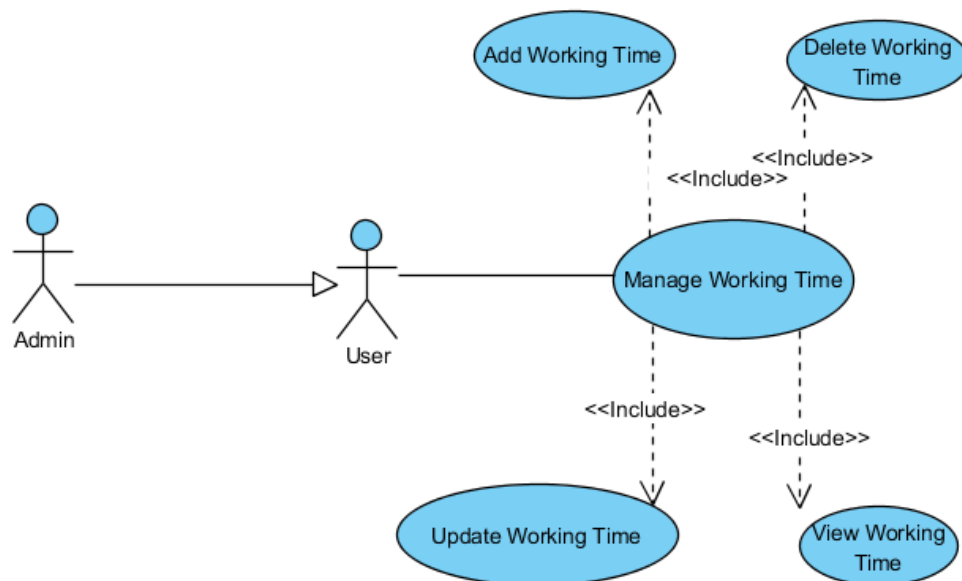
- **Brief Description:** Accountant can manage Working Time by sub-function.

- **Specification:** See Use-Case-Realization Specification: Manage Working Time.

- **Approve-Reject FeedBack:**
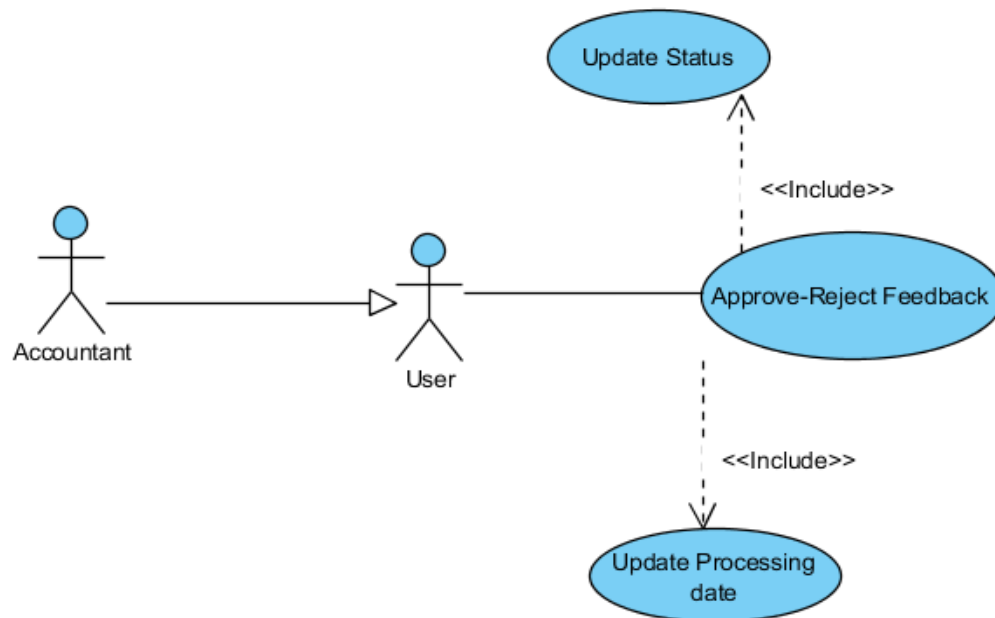
- • **Brief Description:** Accountant can Approve-Reject Leave which are belong to user by sub-function.
- • **Specification:** See Use-Case-Realization Specification: Approve-Reject Feedback.

- **Apply FeedBack**
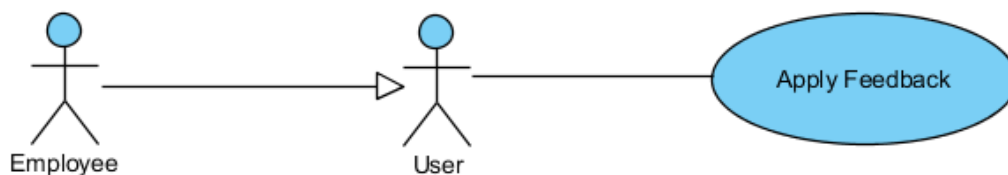
- • **Brief Description:** Employee can Apply Feedback.
- • **Specification:** See Use-Case-Realization Specification: Apply Feedback.

- **Approve – Reject Leave:**
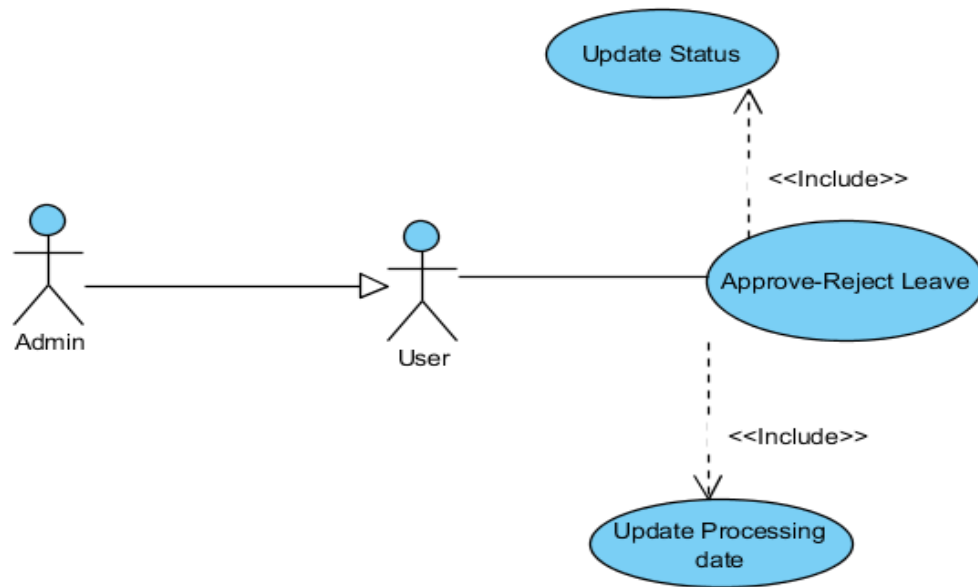
- • **Brief Description:** Admin can Approve-Reject Leave which are belong to user by sub-function.
- • **Specification:** See Use-Case-Realization Specification: Approve-Reject Leave.

- **Apply Leave:**
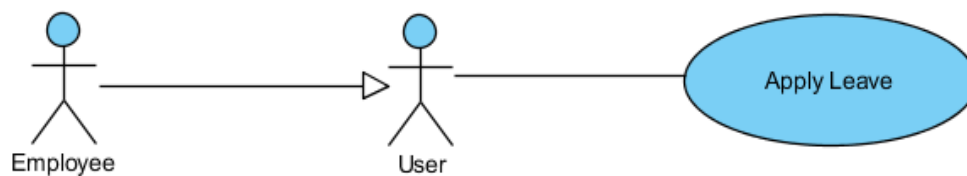
- • **Brief Description:** Employee can Apply Leave.
- • **Specification:** See Use-Case-Realization Specification: Apply Leave.

5. Logical View

5.1 Overview

A description of the logical view of the architecture. Describes the overall decomposition of the design model in terms of package hierarchy and layers.

The logical view of the Payroll Management System is comprised of 3 significant packages:

- **model:** contains classes that directly manages the data, logic and rules of the Payroll Management System and displayed in the view.

- **view:** contains classes that generates output representation of information to the user based on changes in the model. View return the text, button,….

- **controller:** contains classes that can send commands to the model to update the model's state (e.g., add a new computer); it can also send commands to its associated view to change the view's presentation of the model (e.g., scrolling through computer's reviews). Specially, Payroll Management System was built in Java Swing following ORM and OPP design. Therefore, Controller execute event of view.
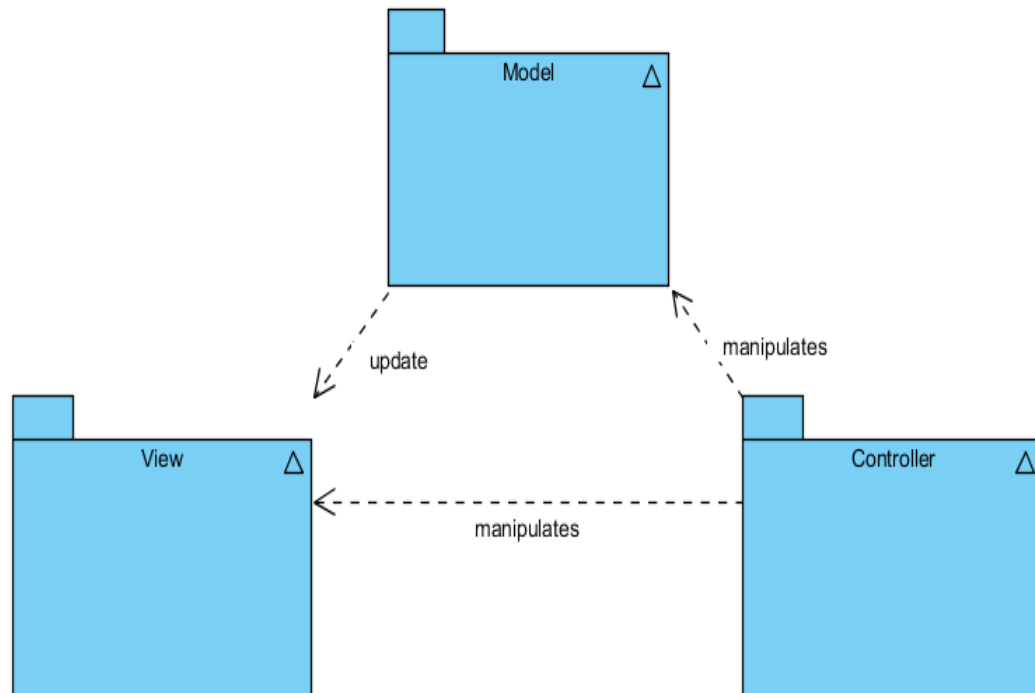
5.2 Architecturally Significant Design Packages

5.2.1 Package model:

- Entity Class

Department
- -department_name : String
- -department_basic_salary : long
- +getDepartment_name() : String
- +setDepartment_name(department_name : String) : void
- +getDepartment_basic_salary() : long
- +setDepartment_basic_salary(department_basic_salary : long) : void

Project
- -projectId : int
- -projectTitle : String
- -projectDescription : String
- -projectContent : String
- -projectTotalEarning : long
- -project_startDate : String
- -project_endDate : String
- +getProjectId() : int
- +setProjectId(projectId : int) : void
- +getProjectTitle() : String
- +setProjectTitle(projectTitle : String) : void
- +getProjectDescription() : String
- +setProjectDescription(projectDescription : String) : void
- +getProjectContent() : String
- +setProjectContent(projectContent : String) : void
- +getProjectTotalEarning() : long
- +setProjectTotalEarning(projectTotalEarning : long) : void
- +getProject_startDate() : String
- +setProject_startDate(project_startDate : String) : void
- +getProject_endDate() : String
- +setProject_endDate(project_endDate : String) : void

WorkingTime
- -workingTimt_id : int
- -user_id : int
- -absence : int
- -over : int
- +getWorkingTimt_id() : int
- +setWorkingTimt_id(workingTimt_id : int) : void
- +getUser_id() : int
- +setUser_id(user_id : int) : void
- +getAbsence() : int
- +setAbsence(absence : int) : void
- +getOver() : int
- +setOver(over : int) : void

Role
- -role_name : String
- -role_basic_salary : long
- +getRole_name() : String
- +setRole_name(role_name : String) : void
- +getRole_basic_salary() : long
- +setRole_basic_salary(role_basic_salary : long) : void

Project_User
- -project_id
- -user_id : int
- -salary : double
- -projectPercentSalary : int
- +getProject_id()
- +setProject_id(project_id) : void
- +getUser_id() : int
- +setUser_id(user_id : int) : void
- +getSalary() : long
- +setSalary(salary : long) : void
- +getProjectPercentSalary() : int
- +setProjectPercentSalary(projectPercentSalary : int) : void

Report
- -report_id : int
- -user_id : int
- -report_title : String
- -report_description : String
- -report_content : String
- -report_post_date : String
- +getReport_id() : int
- +setReport_id(report_id : int) : void
- +getUser_id() : int
- +setUser_id(user_id : int) : void
- +getReport_title() : String
- +setReport_title(report_title : String) : void
- +getReport_description() : String
- +setReport_description(report_description : String) : void
- +getReport_content() : String
- +setReport_content(report_content : String) : void
- +getReport_post_date() : String
- +setReport_post_date(report_post_date : String) : void

User
- -user_id : int
- -user_name : String
- -user_password : String
- -user_type : String
- -user_gender : String
- -user_email : String
- -user_address : String
- -user_dob : String
- -user_phone : String
- -user_role : String
- -user_department : String
- -user_salary : long
- +getUser_id() : int
- +setUser_id(user_id : int) : void
- +getUser_name() : String
- +setUser_name(user_name : String) : void
- +getUser_password() : String
- +setUser_password(user_password : String) : void
- +getUser_type() : String
- +setUser_type(user_type : String) : void
- +getUser_gender() : String
- +setUser_gender(user_gender : String) : void
- +getUser_email() : String
- +setUser_email(user_email : String) : void
- +getUser_address() : String
- +setUser_address(user_address : String) : void
- +getUser_dob() : String
- +setUser_dob(user_dob : String) : void
- +getUser_phone() : String
- +setUser_phone(user_phone : String) : void
- +getUser_role() : String
- +setUser_role(user_role : String) : void
- +getUser_department() : String
- +setUser_department(user_department : String) : void
- +getUser_salary() : double
- +setUser_salary(user_salary : double) : void

Documents
- -document_id : int
- -user_id : int
- -document_title : String
- -document_description : String
- -document_content : String
- -document_status : String
- -document_post_date : String
- -document_process_date : String
- +getDocument_id() : int
- +setDocument_id(document_id : int) : void
- +getUser_id() : int
- +setUser_id(user_id : int) : void
- +getDocument_title() : String
- +setDocument_title(document_title : String) : void
- +getDocument_description() : String
- +setDocument_description(document_description : St...
- +getDocument_content() : String
- +setDocument_content(document_content : String) : ...
- +getDocument_status() : String
- +setDocument_status(document_status : String) : void
- +getDocument_post_date() : String
- +setDocument_post_date(document_post_date : Stri...
- +getDocument_process_date() : String
- +setDocument_process_date(document_process_dat...

Salary
- -user_id : int
- -user_name : String
- -role_salary : long
- -department_salary : long
- -projectSalary : long
- -absenceSalary : long
- -overTimeSalary : long
- -bonusSalary : long
- -minusSalary : long
- -totalSalary : long
- -salary_month : String
- -salary_year : String
- +getUser_id() : int
- +setUser_id(user_id : int) : void
- +getUser_name() : String
- +setUser_name(user_name : String) : void
- +getRole_salary() : long
- +setRole_salary(role_salary : long) : void
- +getDepartment_salary() : long
- +setDepartment_salary(department_salary : long) : void
- +getProjectSalary() : long
- +setProjectSalary(projectSalary : long) : void
- +getAbsenceSalary() : long
- +setAbsenceSalary(absenceSalary : long) : void
- +getOverTimeSalary() : long
- +setOverTimeSalary(overTimeSalary : long) : void
- +getBonusSalary() : long
- +setBonusSalary(bonusSalary : long) : void
- +getMinusSalary() : long
- +setMinusSalary(minusSalary : long) : void
- +getTotalSalary() : long
- +setTotalSalary(totalSalary : long) : void
- +getSalary_month() : String
- +setSalary_month(salary_month : String) : void
- +getSalary_year() : String
- +setSalary_year(salary_year : String) : void

LeaveOff

FeedBack

- Access Database class:

UserDB
- -cn : Connection
- +getAllUserInfor() : ArrayList<User>
- +getUserInfor(user_id : int) : User
- +insertUserInfor(user : User) : boolean
- +deleteUserInfor(user_id : int) : boolean
- +updateUserInfor(user : User) : boolean
- +changePassUser(user : User) : boolean
- +updateSalary(user_id : int, user_salary : user_salary) : boolean
- +getUserByDepartment(user_department : String) : ArrayList<User>
- +getUserByRole(user_role : String) : ArrayList<User>

Project_UserDB
- -cn : Connection
- +insertPersonInvolve(user_id : int, project_id : int, project_salary : long, project_persentSalary : int) : boolean
- +getByProjectID(project_id : int) : ArrayList<Project_User>
- +getProject_User(project_id : int, user_id : int) : Project_User
- +getAllProjects() : ArrayList<Project_User>
- +updateProjectUserSalary(pu_ : Project_User) : boolean
- +getByUserID(user_id : int) : ArrayList<Project_User>
- +deleteProUser(project_id : int, user_id : int) : boolean

ReportDB
- -cn : Connection
- +getAllReport() : ArrayList<Report>
- +getReport(report_id : int) : Report
- +updateReport(report : Report) : boolean
- +addReport(report : Report) : boolean
- +deleteReport(report_id : int) : boolean
- +getReportByUserId(user_id : int) : ArrayList<Report>

FeedBackDB
- -cn : Connection
- +getAllFeedBack() : ArrayList<FeedBack>
- +getFeedBack(feedback_id : int) : FeedBack
- +getByUserId(user_id : int) : ArrayList<FeedBack>
- +updateStatus(fb : FeedBack) : boolean
- +deleteFeedBack(feedback_id : int) : boolean
- +addFeedBack(feedBack : FeedBack) : boolean

DBconnection
- -db_string : String
- -db_user : String
- -db_pass : String
- -connect : Connection
- -statement : Statement
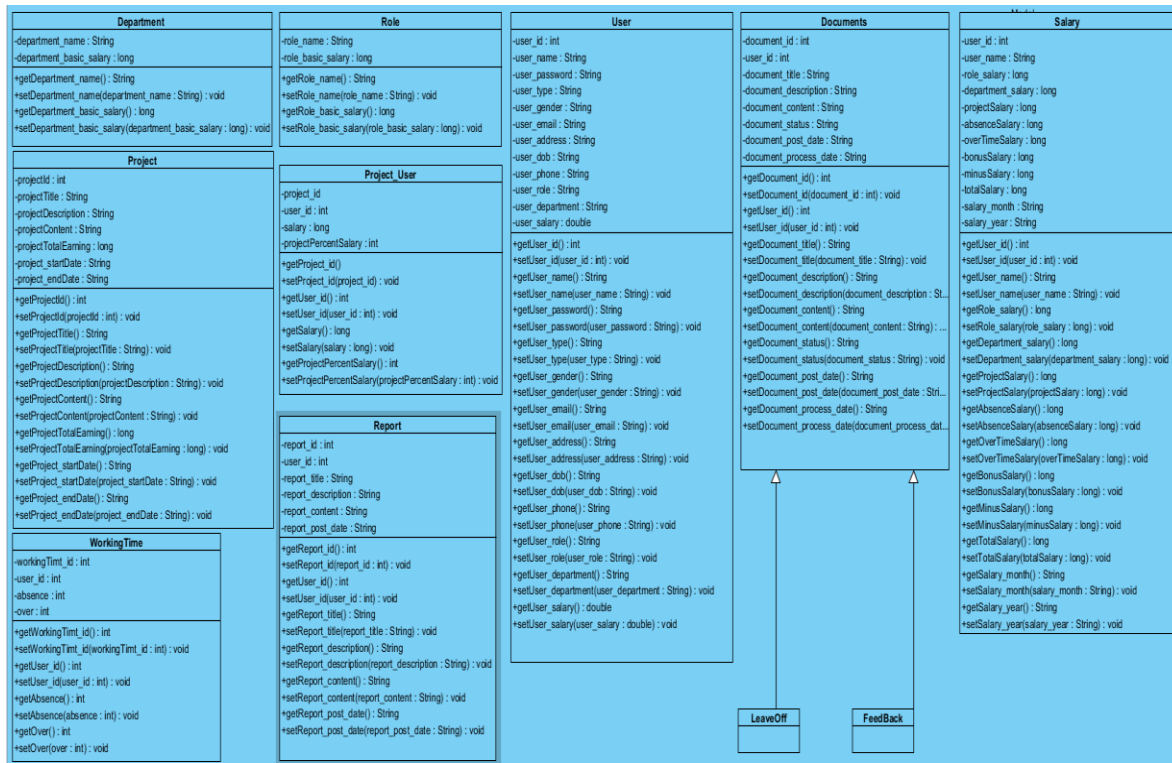- +getConnection() : Connection

Model.ModelDB: Model
- -cn : Connection
- +getAllRole() : ArrayList<Role>
- +getRole(role_name : String) : Role
- +updateRole(role : Role, previous_name : String) : boolean
- +addRole(role : Role) : boolean
- +deleteRole(role_name : String) : boolean

DepartmentDB
- -cn : Connection
- +getAllDepartments() : ArrayList<Department>
- +getDepartment(de_name : String) : Department
- +updateDepartment(department : Department, previous_name : String) : boolean
- +addDeparment(department Department) : boolean
- +deleteDepartment(de_name : String) : boolean

ProjectDB
- -cn : Connection
- +insertProjectInfor(project : Project) : boolean
- +getAllProjectInfor() : ArrayList<Project>
- +getProjectInfor(project_id : int) : Project
- +updateProjectInfor(project : Project) : boolean
- +deleteProInfor(project_id : int) : boolean

LeaveOffDB
- -cn : Connection
- +getAllLeave() : ArrayList<LeaveOff>
- +getLeave(leave_id : int) : LeaveOff
- +updateLeave(leaveoff : LeaveOff) : boolean
- +getByUser(user_id : int) : ArrayList<LeaveOff>
- +addLeave(leave : LeaveOff) : boolean
- +deleteLeave(leave_id : int) : boolean

SalaryDB
- -cn : Connection
- +getAllSalary() : ArrayList<Salary>
- +getSalaryByUser(user_id : int) : ArrayList<Salary>
- +getSalaryByMonth(month : String) : ArrayList<Salary>
- +getSalaryByYear(year : String) : ArrayList<Salary>
- +addSalary(salary : Salary) : boolean
- +updateSalary(salary : Salary) : boolean

WorkingTimeDB
- -cn : Connection
- +getAllWorkTime() : ArrayList<WorkingTime>
- +getWorkingTimeInf(user_id : int) : WorkingTime
- +insertWorkingTime(workingTime : WorkingTime) : boolean
- +updateWorkingTime(workingTime : WorkingTime) : boolean

Name	Model Entities
Brief Description	Contains classes that directly manages the data, logic and rules of the Payroll Management System and displayed in the view.
Classes	Department, Role, User, FeedBack, LeaveOff, Documents, Project, Project_User, Report, Salary, WorkingTime.

- Class: User

Name	User						
Brief Description	Data model for User table in database.						
Attributes							
Name	**Type**	**Access**	**Mutable**	**Optional**	**Length**	**Min**	**Max**
user_id	int	Private	True	False	N/A	1	N/A
user_name	String	Private	True	False	N/A	N/A	N/A
user_password	String	Private	True	False	N/A	N/A	N/A
user_type	String	Private	True	False	N/A	N/A	N/A
user_gender	String	Private	True	False	N/A	N/A	N/A
user_email	String	Private	True	False	N/A	N/A	N/A
user_address	String	Private	True	False	N/A	N/A	N/A
user_dob	String	Private	True	False	N/A	N/A	N/A
user_phone	String	Private	True	False	N/A	N/A	N/A
user_role	String	Private	True	False	N/A	N/A	N/A
user_department	String	Private	True	False	N/A	N/A	N/A
user_salary	double	Private	True	False	N/A	N/A	N/A

- Class: Documents

Name	Documents						
Brief Description	This is class for LeaveOff and Feedback extends						
Attributes							
Name	**Type**	**Access**	**Mutable**	**Optional**	**Length**	**Min**	**Max**
document_id	int	Private	True	False	N/A	1	N/A
user_id	int	Private	True	False	N/A	1	N/A
document_title	String	Private	True	False	N/A	N/A	N/A
document_description	String	Private	True	False	N/A	N/A	N/A
document_content	String	Private	True	False	N/A	N/A	N/A
document_status	String	Private	True	False	N/A	N/A	N/A
document_post_date	String	Private	True	False	N/A	N/A	N/A
document_process_date	String	Private	True	False	N/A	N/A	N/A

- Class: Salary

Name	Salary						
Brief Description	Data model for department table in database.						
Attributes							
Name	**Type**	**Access**	**Mutable**	**Optional**	**Length**	**Min**	**Max**
user_id	int	Private	True	False	N/A	N/A	N/A
user_name	String	Private	True	False	N/A	N/A	N/A
role_salary	long	Private	True	False			
department_salary	long	Private	True	False			
projectSalary	long	Private	True	False			
absenceSalary	long	Private	True	False			
overTimeSalary	long	Private	True	False			
bonusSalary	long	Private	True	False			
minusSalary	long	Private	True	False			
totalSalary	long	Private	True	False			
salary_month	String	Private	True	False			
salary_year	String	Private	True	False			

Name	Model Access Database
Brief Description	Contains classes that directly manages the data, logic and rules of the Payroll Management System and displayed in the view.
Classes	DBConnection, DepartmentDB, FeedBackDB, LeaveOffDB, ProjectDB, Project_UserDB, ReportDB, RoleDB, SalaryDB, UserDB, WorkingTimeDB

- Class DBConnection:

Name	DBConnection						
Brief Description	To Connect Database						
Attributes							
Name	**Type**	**Access**	**Mutable**	**Optional**	**Length**	**Min**	**Max**
db_string	String	Private	False	False			
db_user	String	Private	False	False			
db_pass	String	Private	False	False			
connect	Connection	Private	False	False			
statement	Statement	Private	False	False			
Operations							
Header	**Return Type**	**Access**	**Scope**	**Specification**			
getConnection()	Connection	Public	Instance	Connect to database			

- Class UserDB:

Name	UserDB			
Brief Description	Get Database and store it to User entity			
Operations				
Header	**Return Type**	**Access**	**Scope**	**Specification**
getAllUserInfor()	ArrayList<User>	Public	Instance	Return all user in database
getUserInfor()	User	Public	Instance	Return specific user
insertUserInfor()	boolean	Public	Instance	add user to database
deleteUserInfor()	boolean	Public	Instance	delete specific user from database
updateUserInfor()	boolean	Public	Instance	update user information in database
changePassUser()	boolean	Public	Instance	change user password
updateSalary()	boolean	Public	Instance	update User salary
getUserByRoles()	ArrayList<User>	Public	Instance	Return Users by role
getUserByDepartment()	ArrayList<User>	Public	Instance	Return Users by department

5.2.2 Package View:

Name	view
Brief Description	Contains classes that generates output representation of information to the user based on changes in the model.
Classes	- ViewAdminManageAddReport - ViewAdminManageUpdateUser - ViewAdminManageAddUser - ViewAdminManageUpdateProject - ViewAdminManageAddProject - ViewAdminManageUpdateWorkingTime

	- ViewAdminManageAssignProject
	- ViewAdminManageUpdateReport
	- ViewAccountantUpdateSalary
	- ViewAdmin
	- ViewLogin
	- ViewEmployee
	- ViewAccountant
	- ViewEmployeeAddLeave
	- ViewEmployeeAddFeedback

- Class: ViewAdminManageAddUser

Name	ViewAdminManageAddUser					
Brief Description	Represents the visualization of the data that model contains.					
Attributes						
Name	**Type**	**Access**	**Mutable**	**Optional**	**Length**	
button_add	**javax.swing.JButton**	**private**	False	True		
get_userAddress	**javax.swing.JTextField**	**private**	False	True		
get_userDepartment	**javax.swing.JComboBox**	**private**	False	True		
get_userDob	**com.toedter.calendar.JDateChooser**	**private**	False	True		
get_userEmail	**javax.swing.JTextField**	**private**	False	True		
get_userGender	**javax.swing.JTextField**	**private**	False	True		
get_userName	**javax.swing.JTextField**	**private**	False	True		
get_userPass	**javax.swing.JPasswordField**	**private**	False	True		
get_userPhone	**javax.swing.JTextField**	**private**	False	True		
get_userRole	**javax.swing.JComboBox**	**private**	False	True		
get_userSalary	**javax.swing.JTextField**	**private**	False	True		
get_userType	**javax.swing.JComboBox**	**private**	False	True		
jLabel10	**javax.swing.JLabel**	**private**	False	True		
jLabel11	**javax.swing.JLabel**	**private**	False	True		
jLabel12	**javax.swing.JLabel**	**private**	False	True		
jLabel13	**javax.swing.JLabel**	**private**	False	True		
jLabel3	**javax.swing.JLabel**	**private**	False	True		
jLabel4	**javax.swing.JLabel**	**private**	False	True		
jLabel5	**javax.swing.JLabel**	**private**	False	True		
jLabel6	**javax.swing.JLabel**	**private**	False	True		
jLabel7	**javax.swing.JLabel**	**private**	False	True		

jLabel8	**javax.swing.JLabel**	**private**		False	True	
jLabel9	**javax.swing.JLabel**	**private**		False	True	
jPanel1	**javax.swing.JPanel**	**private**		False	True	
jPanel2	**javax.swing.JPanel**	**private**		False	True	
Operations						
Header	**Return Type**	**Access**	**Scope**	**Specification**		
getGet_userDob()	JDateChooser	Public	Instance	Return Date to choose		
formatDate()	void	Public	Instance	to format JDateChooser		
buttonAdd()	JButton	Public	Instance	return button in view		
getUserName()	JTextField	Public	Instance	return Text field		
getUserPass()	JPasswordField	Public	Instance	return password field		
getUserType()	JComboBox	Public	Instance	return box to choose		
getUserGender()	JTextField	Public	Instance	return Text field		
getUserEmail()	JTextField	Public	Instance	return Text field		
getUserAddress()	JTextField	Public	Instance	return Text field		
getUserPhone()	JTextField	Public	Instance	return Text field		
getUserRole()	JComboBox	Public	Instance	return box to choose		
getUserDepartment()	JComboBox	Public	Instance	return box to choose		
getUserSalary()	JTextField	Public	Instance	return Text field		

5.2.3   Package Controller
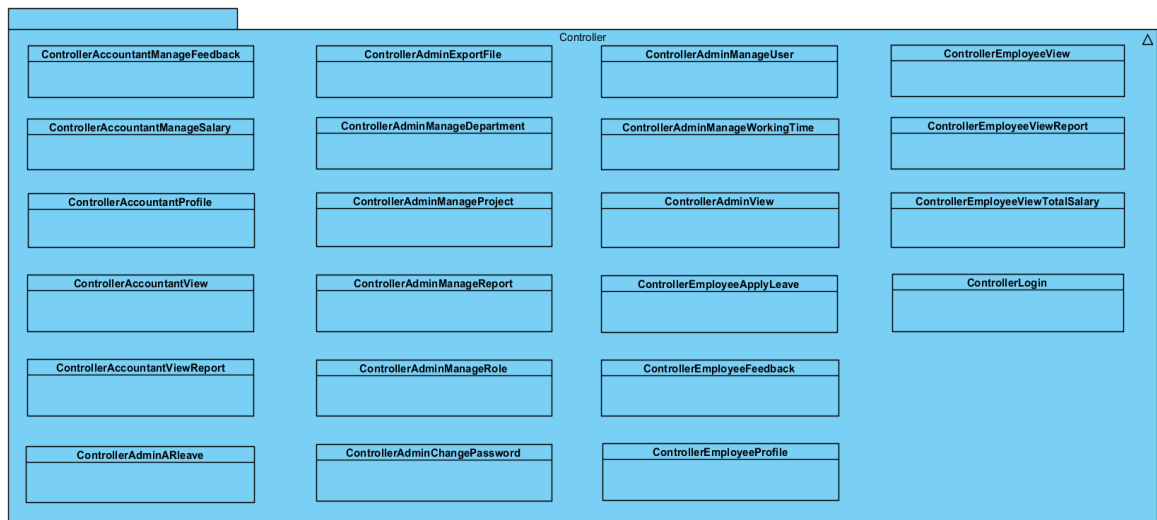
Name	Controller
Brief Description	Contains classes that directly manages the data, logic and rules of the Payroll Management System and displayed in the view.

Classes	- ControllerAccountantManageFeedback - ControllerAccountantManageSalary - ControllerAccountantProfile - ControllerAccountantView - ControllerAccountantViewReport - ControllerAdminARleave - ControllerAdminChangePassword - ControllerAdminExportFile - ControllerAdminManageProject - ControllerAdminManageReport - ControllerAdminManageRole - ControllerAdminManageUser - ControllerAdminManageWorkingTime - ControllerAdminView - ControllerEmployeeApplyLeave - ControllerEmployeeFeedback - ControllerEmployeeProfile - ControllerEmployeeView - ControllerEmployeeViewReport - ControllerEmployeeViewTotalSalary - ControllerLogin

- Class: ControllerAdminView

Brief Description	Controller to Manage all of admin's function						
Attributes							
Name	**Type**	**Access**	**Mutable**	**Optional**	**Length**	**Min**	**Max**
userDB	UserDB	Private	False	False	N/A	N/A	N/A
viewLogin	ViewLogin	Private	False	False	N/A	N/A	N/A
viewAdmin	ViewAdmin	Private	False	False			
projectDB	ProjectDB	Private	False	False			
project_UserDB	Project_UserDB	Private	False	False			
reportDB	ReportDB	Private	False	False			
leaveOffDB	LeaveOffDB	Private	False	False			
wdb	WorkingTimeDB	Private	False	False			
roleDB	RoleDB	Private	False	False			
departmentDB	DepartmentDB	Private	False	False			
Operations							
Header	**Return Type**	**Access**	**Scope**	**Specification**			
logOut()	void	Public	Instance	Handling log out request.			
changeView()	void	Public	Instance	Handling change view request.			

6. Implementation View

The implementation of the system is strictly driven from the design; therefore, the implementation view will not be considered in this document.

7. Data View (optional)
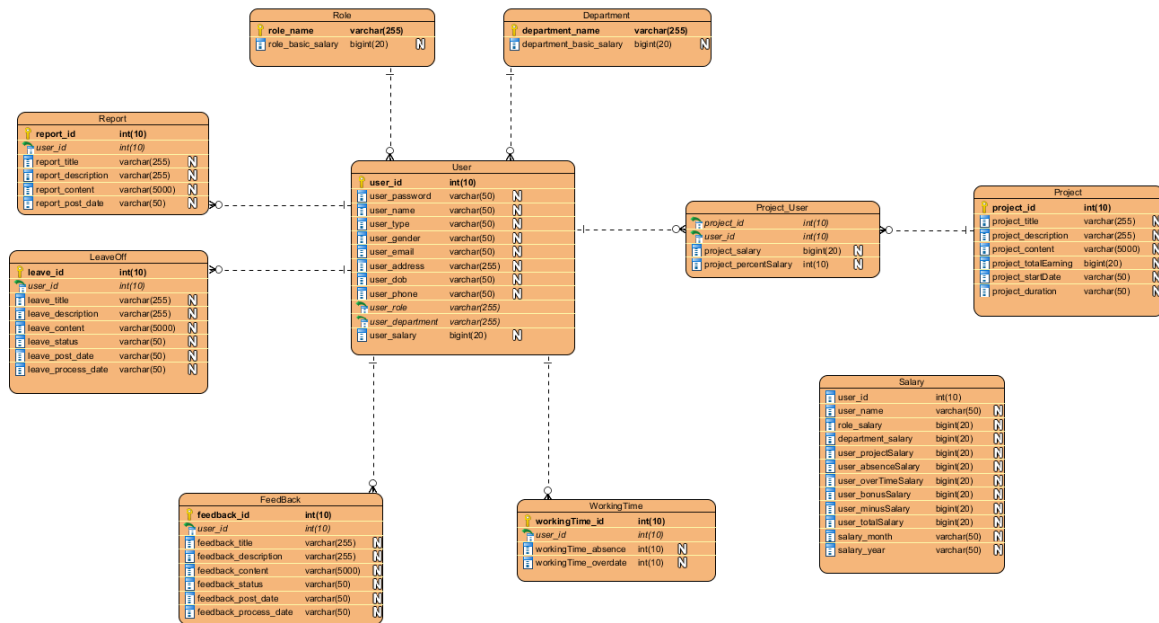
Data Dictionary:
- User:

```
+------------------+---------------+------+-----+---------+----------------+
| Field            | Type          | Null | Key | Default | Extra          |
+------------------+---------------+------+-----+---------+----------------+
| user_id          | int(10)       | NO   | PRI | NULL    | auto_increment |
| user_password    | varchar(250)  | YES  |     | NULL    |                |
| user_name        | varchar(50)   | YES  |     | NULL    |                |
| user_type        | varchar(50)   | YES  |     | NULL    |                |
| user_gender      | varchar(50)   | YES  |     | NULL    |                |
| user_email       | varchar(50)   | YES  |     | NULL    |                |
| user_address     | varchar(255)  | YES  |     | NULL    |                |
| user_dob         | varchar(50)   | YES  |     | NULL    |                |
| user_phone       | varchar(50)   | YES  |     | NULL    |                |
| user_role        | varchar(255)  | YES  | MUL | NULL    |                |
| user_department  | varchar(255)  | YES  | MUL | NULL    |                |
| user_salary      | bigint(20)    | YES  |     | NULL    |                |
+------------------+---------------+------+-----+---------+----------------+
```

- Project:

```
+-------------------+---------------+------+-----+---------+----------------+
| Field             | Type          | Null | Key | Default | Extra          |
+-------------------+---------------+------+-----+---------+----------------+
| project_id        | int(10)       | NO   | PRI | NULL    | auto_increment |
| project_title     | varchar(255)  | YES  |     | NULL    |                |
| project_description | varchar(255) | YES  |     | NULL    |                |
| project_content   | varchar(5000) | YES  |     | NULL    |                |
| project_totalEarning | bigint(20) | YES  |     | NULL    |                |
| project_startDate | varchar(50)   | YES  |     | NULL    |                |
| project_endDate   | varchar(50)   | YES  |     | NULL    |                |
```

- Project_User:

```
+---------------------+------------+------+-----+---------+-------+
| Field               | Type       | Null | Key | Default | Extra |
+---------------------+------------+------+-----+---------+-------+
| project_id          | int(10)    | NO   | MUL | NULL    |       |
| user_id             | int(10)    | NO   | MUL | NULL    |       |
| project_salary      | bigint(20) | YES  |     | NULL    |       |
| project_percentSalary | int(10)  | YES  |     | NULL    |       |
```

- Salary:

```
+---------------------+-------------+------+-----+---------+-------+
| Field               | Type        | Null | Key | Default | Extra |
+---------------------+-------------+------+-----+---------+-------+
| user_id             | int(10)     | YES  |     | NULL    |       |
| user_name           | varchar(50) | YES  |     | NULL    |       |
| role_salary         | bigint(20)  | YES  |     | NULL    |       |
| department_salary   | bigint(20)  | YES  |     | NULL    |       |
| user_projectSalary  | bigint(20)  | YES  |     | NULL    |       |
| user_absenceSalary  | bigint(20)  | YES  |     | NULL    |       |
| user_overTimeSalary | bigint(20)  | YES  |     | NULL    |       |
| user_bonusSalary    | bigint(20)  | YES  |     | NULL    |       |
| user_minusSalary    | bigint(20)  | YES  |     | NULL    |       |
| user_totalSalary    | bigint(20)  | YES  |     | NULL    |       |
| salary_month        | varchar(50) | YES  |     | NULL    |       |
| salary_year         | varchar(50) | YES  |     | NULL    |       |
```

- Working Time:

```
+---------------------+---------+------+-----+---------+----------------+
| Field               | Type    | Null | Key | Default | Extra          |
+---------------------+---------+------+-----+---------+----------------+
| workingTime_id      | int(10) | NO   | PRI | NULL    | auto_increment |
| user_id             | int(10) | NO   | MUL | NULL    |                |
| workingTime_absence | int(10) | YES  |     | NULL    |                |
| workingTime_overdate | int(10)| YES  |     | NULL    |                |
```

- Feedback:

```
| Field                | Type          | Null | Key | Default | Extra          |
+----------------------+---------------+------+-----+---------+----------------+
| feedback_id          | int(10)       | NO   | PRI | NULL    | auto_increment |
| user_id              | int(10)       | NO   | MUL | NULL    |                |
| feedback_title       | varchar(255)  | YES  |     | NULL    |                |
| feedback_description | varchar(255)  | YES  |     | NULL    |                |
| feedback_content     | varchar(5000) | YES  |     | NULL    |                |
| feedback_status      | varchar(50)   | YES  |     | NULL    |                |
| feedback_post_date   | varchar(50)   | YES  |     | NULL    |                |
| feedback_process_date| varchar(50)   | YES  |     | NULL    |                |
```

- LeaveOff:

```
| Field             | Type          | Null | Key | Default | Extra          |
+-------------------+---------------+------+-----+---------+----------------+
| leave_id          | int(10)       | NO   | PRI | NULL    | auto_increment |
| user_id           | int(10)       | NO   | MUL | NULL    |                |
| leave_title       | varchar(255)  | YES  |     | NULL    |                |
| leave_description | varchar(255)  | YES  |     | NULL    |                |
| leave_content     | varchar(5000) | YES  |     | NULL    |                |
| leave_status      | varchar(50)   | YES  |     | NULL    |                |
| leave_post_date   | varchar(50)   | YES  |     | NULL    |                |
| leave_process_date| varchar(50)   | YES  |     | NULL    |                |
```

- Report:

```
| Field              | Type          | Null | Key | Default | Extra          |
+--------------------+---------------+------+-----+---------+----------------+
| report_id          | int(10)       | NO   | PRI | NULL    | auto_increment |
| user_id            | int(10)       | NO   | MUL | NULL    |                |
| report_title       | varchar(255)  | YES  |     | NULL    |                |
| report_description | varchar(255)  | YES  |     | NULL    |                |
| report_content     | varchar(5000) | YES  |     | NULL    |                |
| report_post_date   | varchar(50)   | YES  |     | NULL    |                |
```

- Role:

```
| Field             | Type         | Null | Key | Default | Extra  |
+-------------------+--------------+------+-----+---------+--------+
| role_name         | varchar(255) | NO   | PRI | NULL    |        |
| role_basic_salary | bigint(20)   | YES  |     | NULL    |        |
```

- Department:

```
| Field                   | Type         | Null | Key | Default | Extra  |
+-------------------------+--------------+------+-----+---------+--------+
| department_name         | varchar(255) | NO   | PRI | NULL    |        |
| department_basic_salary | bigint(20)   | YES  |     | NULL    |        |
```

8. Size and Performance

The major dimensioning characteristics of the software that impact the architecture and performance constraints:

The system shall support up to 1000 concurrent users against the primary database at any given time, and up to 500 concurrent users against the local servers at any one time.

The system must perform all functions with minimal time delays. The system must also accurately save all information transactions

9. Quality

The system architecture supports the quality requirements:

- In order to maintain the highest degree of system integrity, the system is capable of ensuring that all information transitions are saved.

- Databases will be backed up on a daily basis in concern with safety implications.

- This system can install in some Operating System: MacOS, Window, ….