



اَبُو سَيْفِ بْنِ كَلْبٍ
UNIVERSITI
TEKNOLOGI
MARA



Software Design Document for Performance and Evaluation System for Real Max Consult Sdn. Bhd. With Data Visualization

SEMESTER MARCH 2022 – JULY 2022

Submit To:

**Miss Fadilah Ezlina Shahbudin
Madam Faiqah Hafidzah Binti Halim**

Prepared By:

Nor Huwaida Binti Mat Isa, 2020985135

Version: 1.0

Due Date: June 2017

Table of Contents

1. Introduction.....	1
1.1 Purpose	1
1.2 Scope	1
1.3 References	1
1.4 Document Structure.....	2
1.5 Definitions, Acronyms and Abbreviations	2
1.6 System Overview	2
2. System Architecture.....	5
2.1 Architectural Description	5
2.2 Design Rational	6
2.3 Domain Class Diagram	6
2.4 Design Class Diagram	7
3. Data Design.....	8
3.1 Database Description	8
3.2 Data Dictionary	9
4. Package Diagram	12
4.1 Package Diagram	12
4.1.2 Package Functions	13
4.1.3 Package Content	13
5. Class Description.....	15
5.1 SDD Class 201 - User.....	15
5.1.1 Class Attributes & Methods.....	15
5.1.2 Method Description	16
5.2 SDD Class 202 - Task.....	16
5.2.1 Class Attributes & Methods.....	16
5.2.2 Method Description	17
5.3 SDD Class 203 - Sales.....	23
5.4 SDD Class 204 - Property	23
5.4.1 Class Attributes & Methods.....	24
5.4.2 Method Description	24
5.5 SDD Class 205 - Evaluation.....	25
6. Human Interface Design (Screens)	26
6.1 Overview of the User Interface	26
6.1.1 Screen Images.....	26
6.1.2 Report	32
7. Traceability Requirements Matrix.....	33
7.1 View Layer	33
7.2 Domain Layer	34
7.3 Data Access Layer	35
8. Appendices.....	36
8.1 Detailed Class Diagram.....	36
8.2 Create Account	37
8.2.1 Multilayer Sequence Diagram	37
8.3 View Employee Account.....	37
8.3.1 Multilayer Sequence Diagram	37
8.4 Delete Employee Account.....	37
8.4.1 Multilayer Sequence Diagram	37
8.5 Create Tasks.....	38
8.5.1 Multilayer Sequence Diagram	38
8.6 View Tasks	38
8.6.1 Multilayer Sequence Diagram	38
8.7 Update Tasks	39
8.7.1 Multilayer Sequence Diagram	39

8.8	Delete Tasks.....	39
8.8.1	Multilayer Sequence Diagram	39
8.9	Update Task Progress.....	40
8.9.1	Multilayer Sequence Diagram	40
8.10	Create Properties.....	40
8.10.1	Multilayer Sequence Diagram	40
8.11	View Properties	41
8.11.1	Multilayer Sequence Diagram	41
8.12	Create Properties Sales	41
8.12.1	Multilayer Sequence Diagram	41
8.13	View Properties Sales	41
8.13.1	Multilayer Sequence Diagram	41
8.14	Delete Properties Sales.....	41
8.14.1	Multilayer Sequence Diagram	41
8.15	Create Employee Evaluation	41
8.15.1	Multilayer Sequence Diagram	41
8.16	View Evaluation.....	42
8.16.1	Multilayer Sequence Diagram	42

1. Introduction

This document is a Software Design Document (SDD) which is also known as a software design specification. This document shows how the system will be developed to meet the requirements from the Software Requirements Specification document. Hence, this document is significant during the development as it must include the details that the software programmer needs to refer during the development.

1.1 Purpose

The purpose of this document is to develop a web-based performance and evaluation system for Real Max Consult Sdn. Bhd. with Data Visualization in order to assign the tasks and update tasks. Moreover, it also able to track the tasks or sales that have been made. The required software is to build a platform for the company to ensure the performance and evaluation system is properly managed and each use cases are successful. Hence, the results of developed system will be able to assign and update tasks and sales. Thus, data visualization will visualize the data of tracking the performance of the employee and the system allows the director to evaluate the employees. This software also makes it easier for the company to manage the task, performance and evaluation system.

1.2 Scope

A performance and evaluation System for Real Max Consult Sdn. Bhd. with Data Visualization is a system that assist three categories of end users, which known as directors, administrators, and sales employee of Real Max Consult Sdn. Bhd. Hence, this system helps to facilitate the company to properly manage the tasks, sales, performance and evaluation. Moreover, this system eases the company to keep the track of the sales and tasks and evaluation that are developed. Furthermore, this system includes the data visualization that eases the company to track the performance of the employee.

1.3 References

System Requirements Specification for Performance and Evaluation System for Real
Max Consult Sdn. Bhd. with Data Visualization

1.4 Document Structure

This document is written based on the standards for Software Design Documentation explained in "IEEE Recommended Practice for Software Design Documentation". This document also completely describes on the system progression from the last progress report, which was stated in Software Requirements Specification. As every progression that was made is shown in diagrams and screen images are included in this document.

1.5 Definitions, Acronyms and Abbreviations

Acronyms	Definition
SRS	Software Requirement Specification
SDD	Software Design Document
UCD	Use Case Diagram
DCD	Domain Class Diagram
DCD	Design Class Diagram
UML	Unified Modeling Language
MLSD	Multilayer Sequence Diagram

1.6 System Overview

In the system, there will be three end users which are directors, administrators and sales employees. Every user has its role and function, which include:

Director

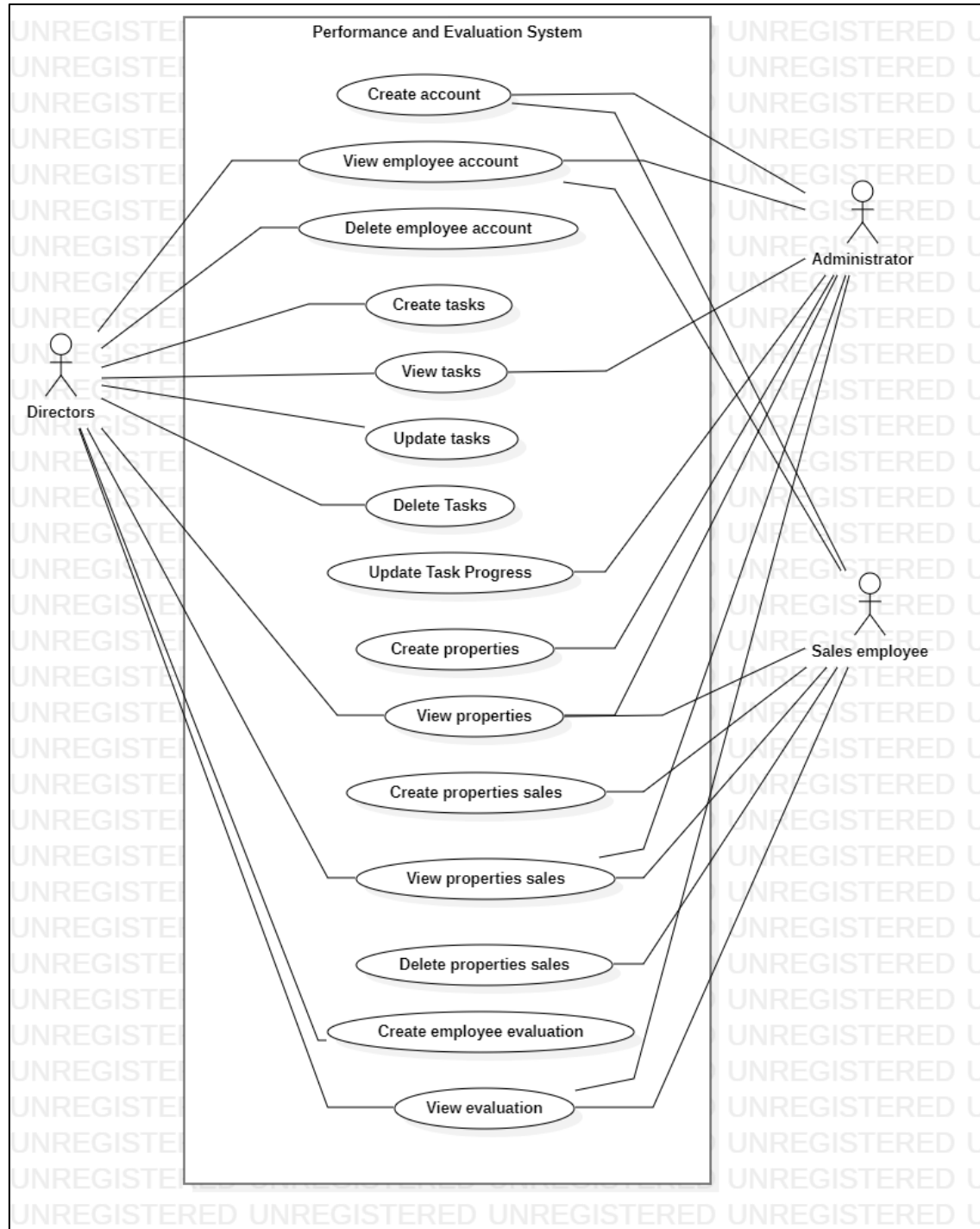
- View employee account
- Delete employee account
- Create tasks
- View tasks
- Update tasks
- Delete tasks
- View properties
- View properties sales
- Create employee evaluation
- View Evaluation

Administrators

- Create account
- View employee account
- View tasks
- Update task progress
- Create properties
- View properties
- View properties sales
- View evaluation

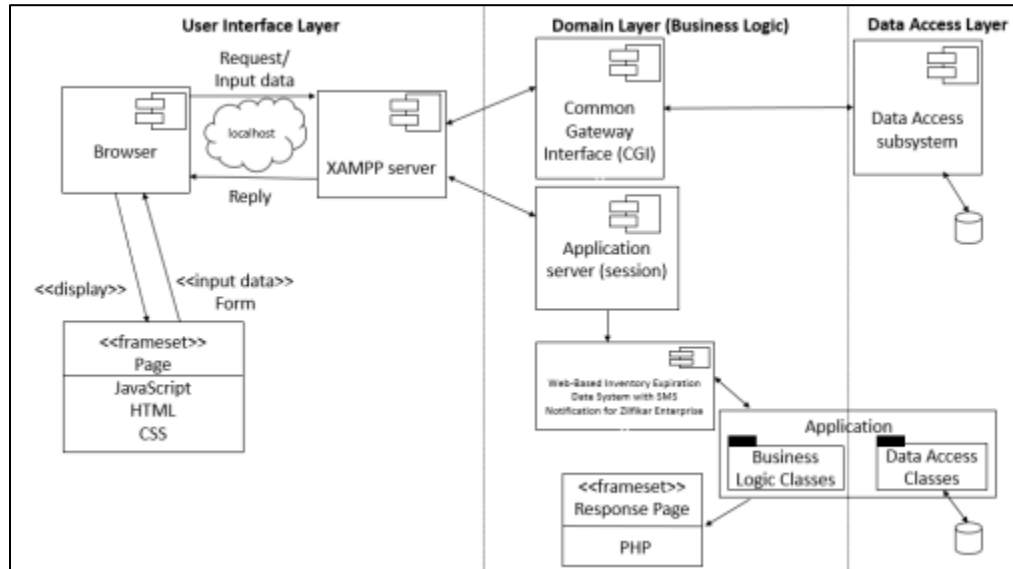
Sales employee

- Create account
- View employee account
- View properties
- Create properties sales
- Delete properties sales
- View properties sales
- View evaluation



2. System Architecture

2.1 Architectural Description



This diagram shows how the system will be developed using the three-layer application during the development of the software system. Therefore, the three-layer application consists of three division as it is a client and server architecture.

User Interface Layer

The top-most level of the system which is the first layer of the system is the user interface layer, also known as view layer. It is responsible for converting the coding into something that users can understand by looking at it. Therefore, during this layer, the web browser interacts with the application as it developed by using HTML, CSS and JavaScript which the languages are used for front-end application.

Domain Layer

The domain layer known as the business logic consists the programs that implements the business rules and processes as it communicates the data with the other layers.

Data Access Layer

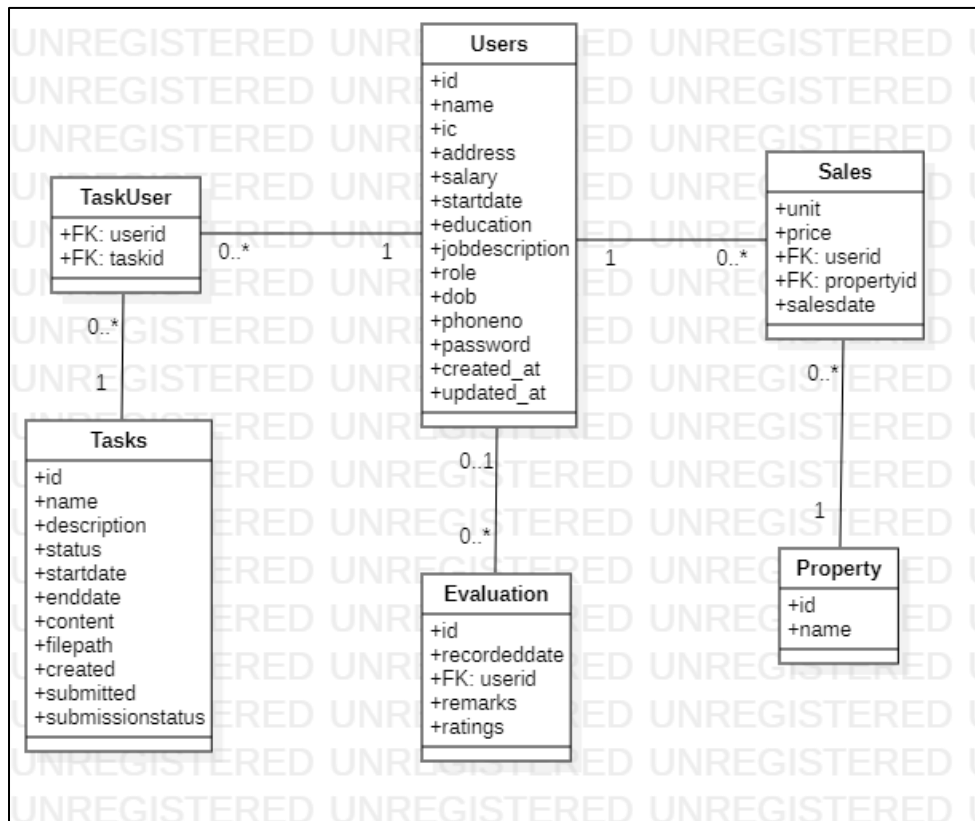
The data access layer is where interaction of data occurs as it performs any operation such as CRUD in the database such as create, update, read and delete.

2.2 Design Rational

The three-layer architecture was chosen because it meets the system requirements. This architecture helps to organize the system in three tiers that are divided into user interface layer, domain layer, and data access layer. As, the user interface layer is the top or highest level of architecture. The domain layer is the middle layer of architecture as it helps to interact the data between the other layers. Therefore, the data layer is used for the data interaction from the domain layer to the database system. There are some benefits by applying the three-layer application to the system as it eases the programmer for the future use as it can be expanded. Therefore, it is easy to maintain and modify the system for further improvement. Hence, it improves the data integrity as the updated information must go through to the second layer. Therefore, by applying the three-layer application may not affect the function or layer.

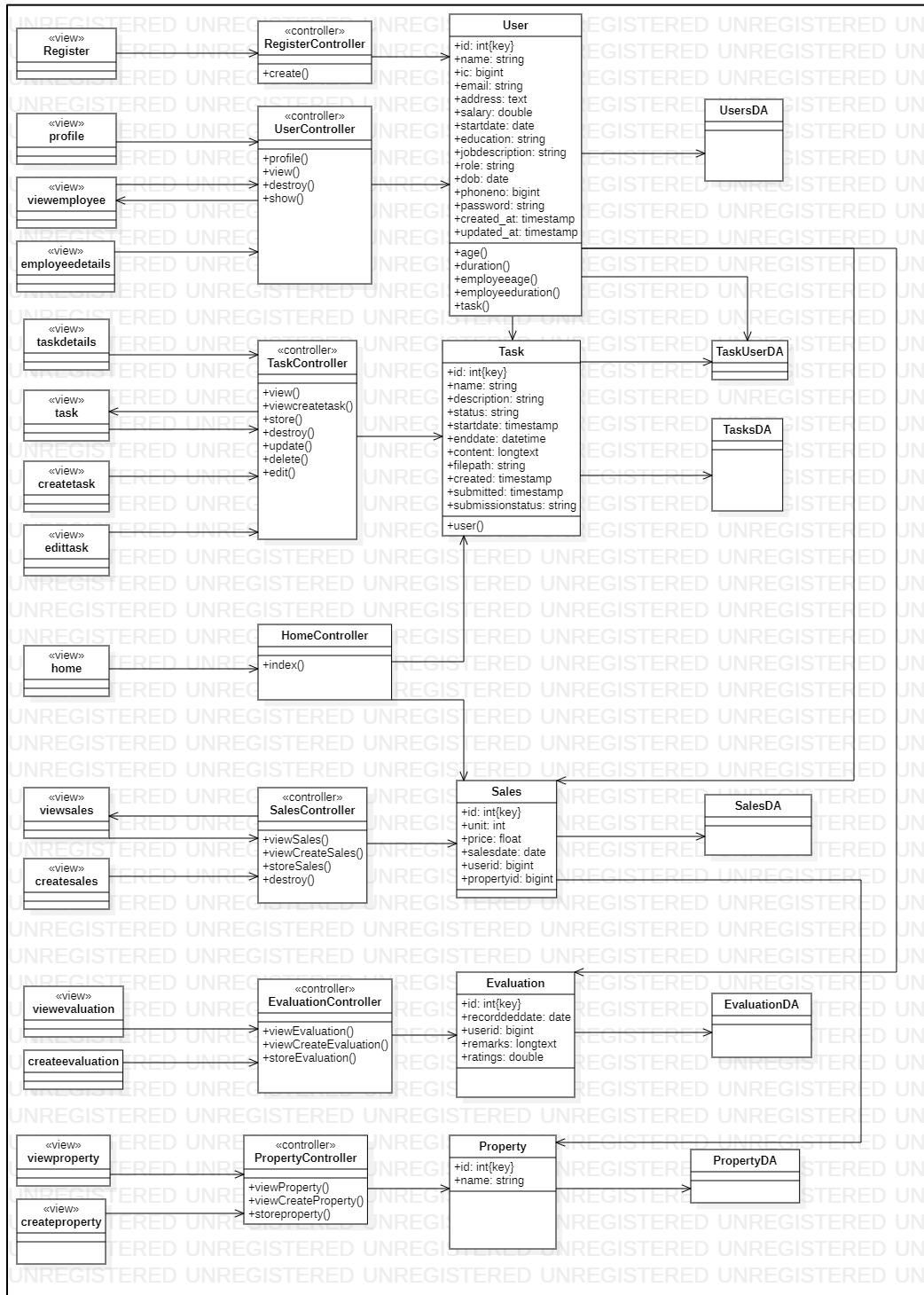
2.3 Domain Class Diagram

Domain class diagram consists of six classes which are tasks, taskuser, users, evaluation, sales and property.



2.4 Design Class Diagram

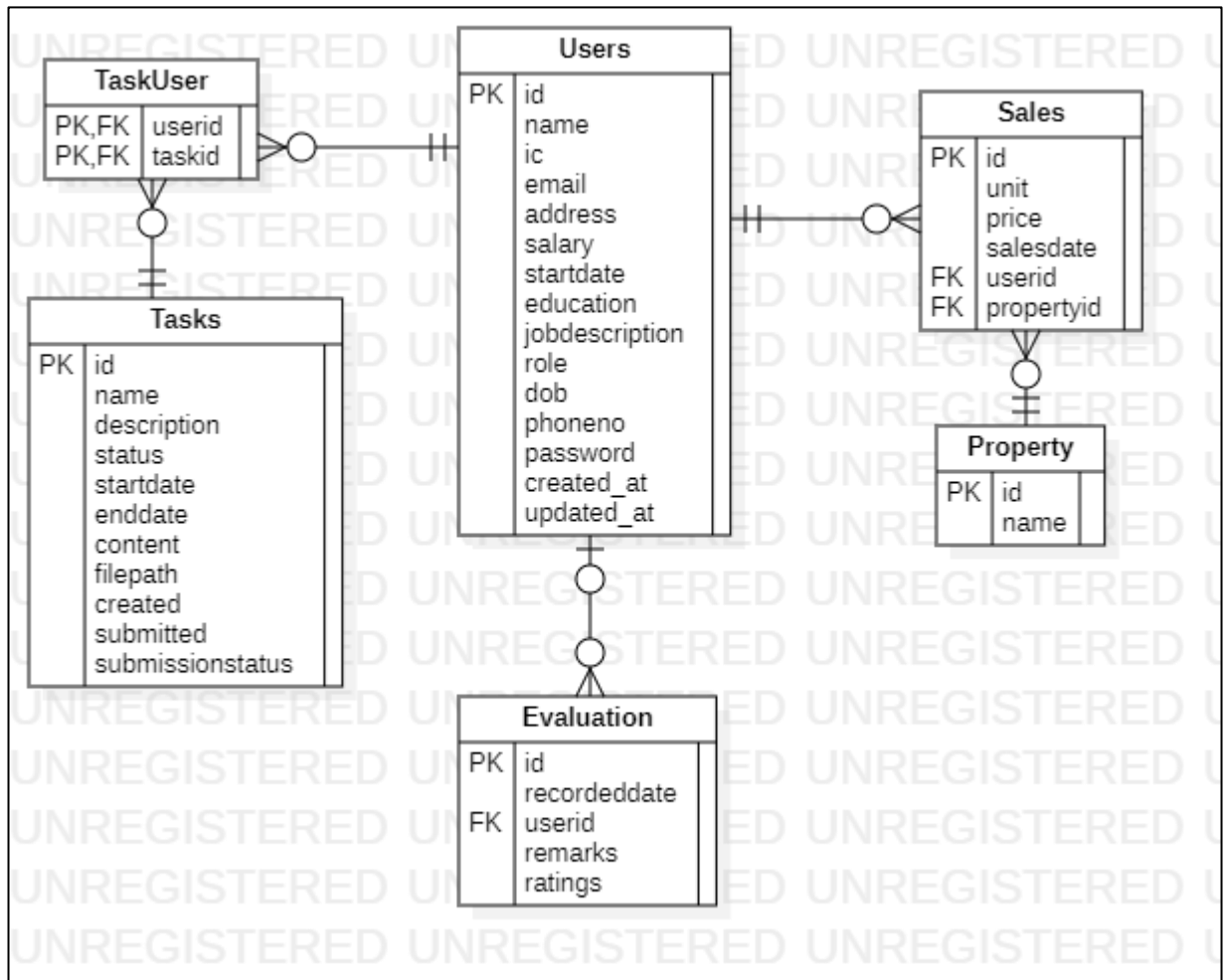
This section shows about the design class diagram on how the flow of the system will be as it consists of seven controllers, fifteen views, five models and six databases.



3. Data Design

3.1 Database Description

This section describes about the entity relationship diagram which consists of six entities that will be used for the development process to store the data in the database.



3.2 Data Dictionary

Users						
Attribute Name	Description	Type	Additional Type Information	Default Value	M	U
id	Id of each user	Integer	Max = 10	-	Y	Y
name	Name of user	Varchar	Max = 255	-	Y	N
ic	Users' identity card	Big Integer	Max = 20	-	Y	Y
email	Email of user	Varchar	Max = 255	-	Y	Y
address	Address of user	Text	-	-	Y	N
salary	Salary of user	Double	-	-	Y	N
startdate	Start day of work	Date	-	-	Y	N
education	Highest level of education	Varchar	Max = 255	-	Y	N
jobdescription	Job description of user	Varchar	Max = 255	-	Y	N
role	Role of user	Varchar	Max = 255	-	Y	N
dob	Date of birth	Date	-	-	Y	N
phoneno	Phone number of user	Big Integer	Max = 20	-	Y	N
password	Password of user	Varchar	Max = 255	-	Y	N
created_at	Created account date	Timestamp	-	-	Y	N
updated_at	Updated account date	Timestamp	-	-	Y	N

M=Mandatory?, U=Unique?, Y= Yes, N = No

TaskUser						
Attribute Name	Description	Type	Additional Type Information	Default Value	M	U
userid	Id of each user	Integer	Max = 10	-	Y	N
taskid	Id of each task	Integer	Max = 10	-	Y	N

M=Mandatory?, U=Unique?, Y= Yes, N = No

Tasks						
Attribute Name	Description	Type	Additional Type Information	Default Value	M	U
id	Id of each task	Integer	Max = 10	-	Y	Y
name	Name of task	Varchar	Max = 255	-	Y	N
description	Description of task	Varchar	Max = 255	-	Y	N
status	Status of task	Varchar	Max = 255	-	Y	N
startdate	Start date of task	Timestamp	-	-	Y	N
enddate	End date of task	Datetime	-	-	Y	N
content	Content of task	LongText	-	-	N	N
filepath	File path of task	LongText	-	-	N	N
created	Time task created	Timestamp	-	-	N	N
submitted	Time task submitted	Timestamp	-	-	N	N
submissionstatus	Status of submitted task	Varchar	Max = 50	-	N	N

M=Mandatory?, U=Unique?, Y= Yes, N = No

Sales						
Attribute Name	Description	Type	Additional Type Information	Default Value	M	U
id	Id of each sales	Integer	Max= 10	-	Y	Y
unit	Unit of each sales	integer	Max = 10	-	Y	N
price	Price of sales	Double	Max = 8.2	-	Y	N
salesdate	Date when sold	Date	-	-	Y	N
userid	Id of each user	Big Integer	Max = 20	-	Y	N
propertyid	Id of each property	Big Integer	Max = 20	-	Y	N

M=Mandatory?, U=Unique?, Y= Yes, N = No

Property						
Attribute Name	Description	Type	Additional Type Information	Default Value	M	U
id	Id of each property	Integer	Max = 10	-	Y	Y
name	Name of property	Varchar	Max = 255	-	Y	N

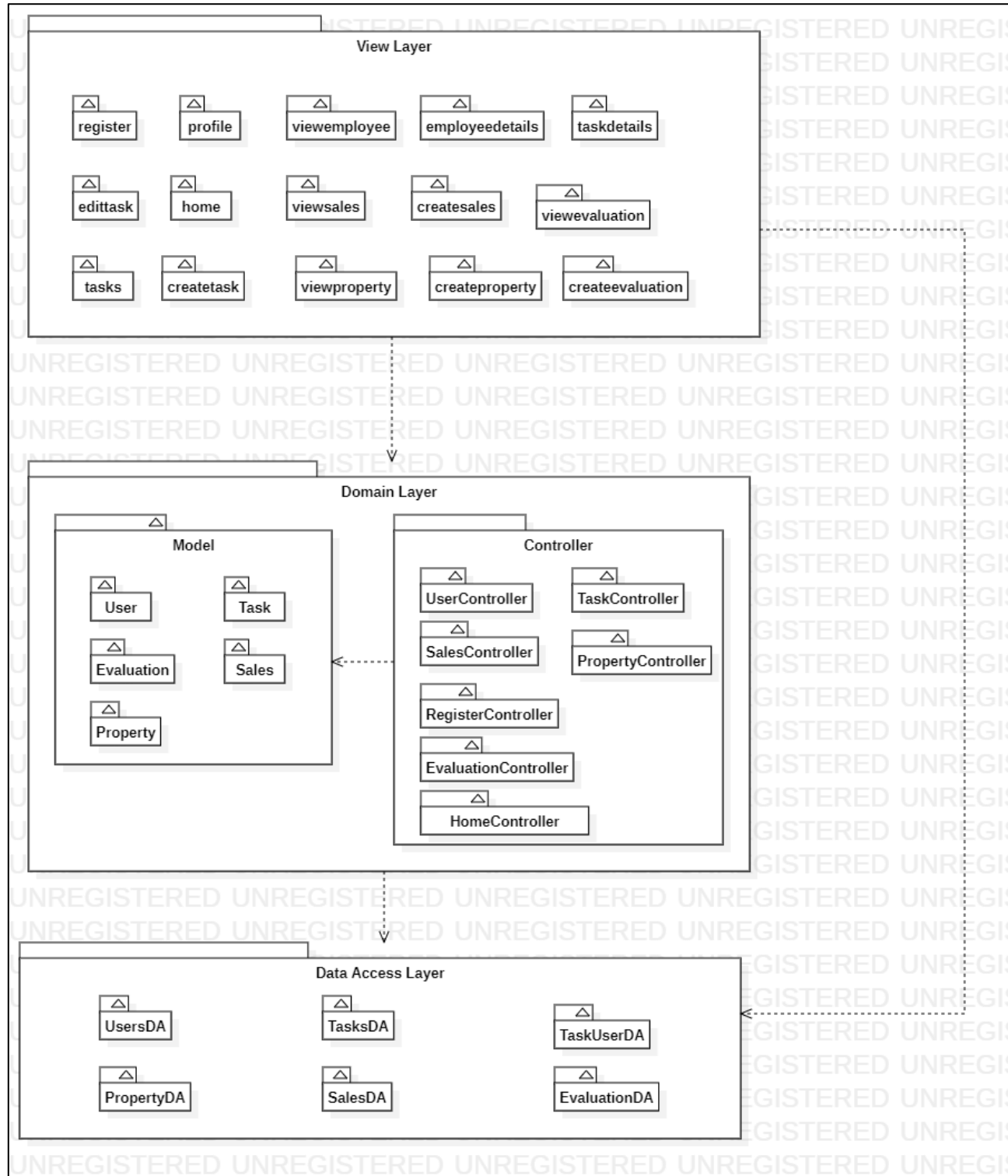
M=Mandatory?, U=Unique?, Y= Yes, N = No

Evaluation						
Attribute Name	Description	Type	Additional Type Information	Default Value	M	U
id	Id of each evaluation	Integer	Max = 10	-	Y	Y
recordeddate	Recorded date of evaluation	Date	-	-	Y	N
userid	Id of user	Big Integer	Max = 20	-	Y	N
remarks	Remarks of evaluation	Text	-	-	Y	N
ratings	Ratings of evaluation	Double	Max = 8.2	-	Y	N

M=Mandatory?, U=Unique?, Y= Yes, N = No

4. Package Diagram

4.1 Package Diagram



4.1.2 Package Functions

Table 4.1 Package functions

Package	Package Functions
View Layer (SDD_PKG_100)	<ul style="list-style-type: none"> • To display required form as needs • To display data and accepts input data • To edit and validate input data • To be able to view the data
Domain Layer (SDD_PKG_200)	<ul style="list-style-type: none"> • To carry out the process of business rules
Data Access Layer (SDD_PKG_300)	<ul style="list-style-type: none"> • To establish and maintain connection of database • To retrieve data from database • To store the data in the database • To delete the data in the database • To update the data in the database

4.1.3 Package Content

Table 4.2 Package Content

Package	Class ID	Class Name
SDD_PKG_100	SDD_CLASS_101	Register
	SDD_CLASS_102	Profile
	SDD_CLASS_103	Viewemployee
	SDD_CLASS_104	Employeeedetails
	SDD_CLASS_105	Taskdetails
	SDD_CLASS_106	Tasks
	SDD_CLASS_107	Createtask
	SDD_CLASS_108	Edittask
	SDD_CLASS_109	Home
	SDD_CLASS_110	Viewsales
	SDD_CLASS_111	Createsales

	SDD_CLASS_112	Viewproperty
	SDD_CLASS_113	Createproperty
	SDD_CLASS_114	Viewevaluation
	SDD_CLASS_115	Createevaluation
SDD_PKG_200	SDD_CLASS_201	User
	SDD_CLASS_202	Task
	SDD_CLASS_203	Sales
	SDD_CLASS_204	Property
	SDD_CLASS_205	Evaluation
	SDD_CLASS_206	UserController
	SDD_CLASS_207	TaskController
	SDD_CLASS_208	SalesController
	SDD_CLASS_209	PropertyController
	SDD_CLASS_210	PerformanceController
	SDD_CLASS_211	EvaluationController
	SDD_CLASS_212	RegisterController
SDD_PKG_300	SDD_CLASS_301	Users
	SDD_CLASS_302	TaskUser
	SDD_CLASS_303	Tasks
	SDD_CLASS_304	Sales
	SDD_CLASS_305	Property
	SDD_CLASS_306	Evaluation

5. Class Description

5.1 SDD Class 201 - User

To manage all information of users.

5.1.1 Class Attributes & Methods

Attribute Name	Attribute Type
id	INT (PK)
name	STRING
ic	BIGINT
email	STRING
address	TEXT
salary	DOUBLE
startdate	DATE
education	STRING
jobdescription	STRING
role	STRING
dob	DATE
phoneno	BIGINT
password	STRING
created_at	INT
updated_At	STRING

Method Name	Return Type	Parameter
profile()	profile	
view()	director.viewemployee	
destroy()	view()	id
show()	Director.employeeedetails	

5.1.2 Method Description

- destroy()

5.1.2.1 Method 1

START

```
public function destroy(User $id) {  
    $id->delete();  
  
    return redirect()->route('employee')->with('Success, employee successfully deleted.');
```

END

5.2 SDD Class 202 - Task

To manage all information of tasks

5.2.1 Class Attributes & Methods

Attribute Name	Attribute Type
id	INT (PK)
name	STRING
description	STRING
status	STRING
startdate	TIMESTAMP
enddate	DATETIME
content	LONGTEXT
filepath	STRING
created	TIMESTAMP
submitted	TIMESTAMP
submissionstatus	STRING

Method Name	Return Type	Parameter
view()	assign, completed, adminassign, admincompleted	
viewcreatetask()	users	
store()		request
destroy()		assign
show()	db, task	
edit()	task, db, users	
update()		id
delete()		id

5.2.2 Method Description

- store()
- destroy()
- update()
- delete()

5.2.2.1 Method 1

START

```
public function store(Request $request)
{
    $now = Carbon::now();

    $request->validate([
        'name' => 'required',
        'enddate' => 'required',
    ]);

    if($request->hasFile('file')){
        $file = $request->file('file') ;
    }
}
```

```
$fileName = $file->getClientOriginalName() ;
$destinationPath = public_path().'/files' ;
$file->move($destinationPath,$fileName);

$data = DB::table('tasks')->insertGetId([
    'name' => $request->name,
    'description' => $request->description,
    'startdate' => $now,
    'enddate' => $request->enddate,
    'content' => $request->content,
    'status' => 'Assigned',
    'filepath' => $fileName,
    'created' => $now,
]);
}

$data = DB::table('tasks')->insertGetId([
    'name' => $request->name,
    'description' => $request->description,
    'startdate' => $now,
    'enddate' => $request->enddate,
    'content' => $request->content,
    'status' => 'Assigned',
    'created' => $now,
]);

foreach ($request->userid as $value => $task)
{
    DB::table('taskuser')->insert([
        'taskid' => $data,
        'userid' => $request->userid[$value],
    ]);
}

/*if (count($request->userid) > 0){
```

```
// foreach ($request->userid as $task => $t{
    $data = new TaskUser;
    $data->taskid => $task->id,
    $data->userid => $request->userid[$task],
    $data->save();
}
}
*/
// User::create($request->all());

return redirect()->route('task')
    ->with('success','Student created successfully.');
```

END

5.2.2.2 Method 2

START

```
public function destroy($assign) {

    DB::table('tasks')->where('id', $assign)->delete();
    DB::table('taskuser')->where('taskid', $assign)->delete();

    return redirect()->route('task')->with('Success, employee successfully deleted.');
```

END

5.2.2.3 Method 3

START

```
public function update(Request $request)
{
    $now = Carbon::now();

    $enddate = Carbon::parse($request->enddate);

    if (Auth::user()->role == "administrator")
    {

        if($request->hasFile('file')){
            $file = $request->file('file') ;
            $fileName = $file->getClientOriginalName() ;
            $destinationPath = public_path().'/files' ;
            $file->move($destinationPath,$fileName);

            if($now->lessThan($enddate))
                $submissionstatus = "On-Time";
            else
                $submissionstatus = "Late";

            DB::table('tasks')->where('id',$request->id)->update([
                'filepath' => $fileName,
                'status' => "Completed",
                'submitted' => $now,
                'submissionstatus' => $submissionstatus,
            ]);
        }
        else if($request->content){

            if($now->lessThan($enddate))
```

```
$submissionstatus = "On-Time";
else
$submissionstatus = "Late";

DB::table('tasks')->where('id',$request->id)->update([
    'content' => $request->content,
    'status' => "Completed",
    'submitted' => $now,
    'submissionstatus' => $submissionstatus,

]);
}
else{
    $file = $request->file('file') ;
    $fileName = $file->getClientOriginalName() ;
    $destinationPath = public_path().'/files' ;
    $file->move($destinationPath,$fileName);

    if($now->lessThan($enddate))
    $submissionstatus = "On-Time";
    else
    $submissionstatus = "Late";

    DB::table('tasks')->where('id',$request->id)->update([
        'filepath' => $fileName,
        'status' => "Completed",
        'content' => $request->content,
        'submitted' => $now,
        'submissionstatus' => $submissionstatus,

    ]);

}
}
else if (Auth::user()->role == "director")
```



```
{
  if($request->hasFile('file')){
    $file = $request->file('file') ;
    $fileName = $file->getClientOriginalName() ;
    $destinationPath = public_path().'files' ;
    $file->move($destinationPath,$fileName);

    DB::table('tasks')->where('id',$request->id)->update([
      'name' => $request->name,
      'description' => $request->description,
      'startdate' => $now,
      'enddate' => $request->enddate,
      'content' => $request->content,
      'filepath' => $fileName,
      'status' => 'Assigned',
    ]);
  }
  else{
    DB::table('tasks')->where('id',$request->id)->update([
      'name' => $request->name,
      'description' => $request->description,
      'startdate' => $now,
      'enddate' => $request->enddate,
      'content' => $request->content,
      'status' => 'Assigned',
    ]);
  }

  DB::table('taskuser')->where('taskid', $request->id)->delete();

  foreach ($request->userid as $value => $task)
  {
    DB::table('taskuser')->insert([
      'taskid' => $request->id,
```

```
        'userid' => $request->userid[$value],
    ]);
    }
}
// $student->update($request->all());

return redirect()->route('task')
    ->with('success','Student updated successfully');
}

END
```

5.2.2.4 Method 4

START

```
public function delete($id)
{
    $task = DB::table('tasks')->where('id',$id)->update([
        'filepath' => null,
    ]);

    return back();    }
```

END

5.3 SDD Class 203 - Sales

5.4 SDD Class 204 - Property

To manage all information of property.

5.4.1 Class Attributes & Methods

Attribute Name	Attribute Type
id	INT (PK)
name	STRING

Method Name	Return Type	Parameter
viewProperty()	property	
viewCreateProperty()	property.createproperty	
storeproperty()		request

5.4.2 Method Description

- storeproperty()

5.4.2.1 Method 1

START

```
public function storeproperty(Request $request)
{
    $request->validate([
        'name' => 'required| unique:property',
    ]);

    DB::table('property')->insert([
        'name' => $request->name,]);

    return $this->viewProperty();
}
```

END

5.5 SDD Class 205 - Evaluation

6. Human Interface Design (Screens)

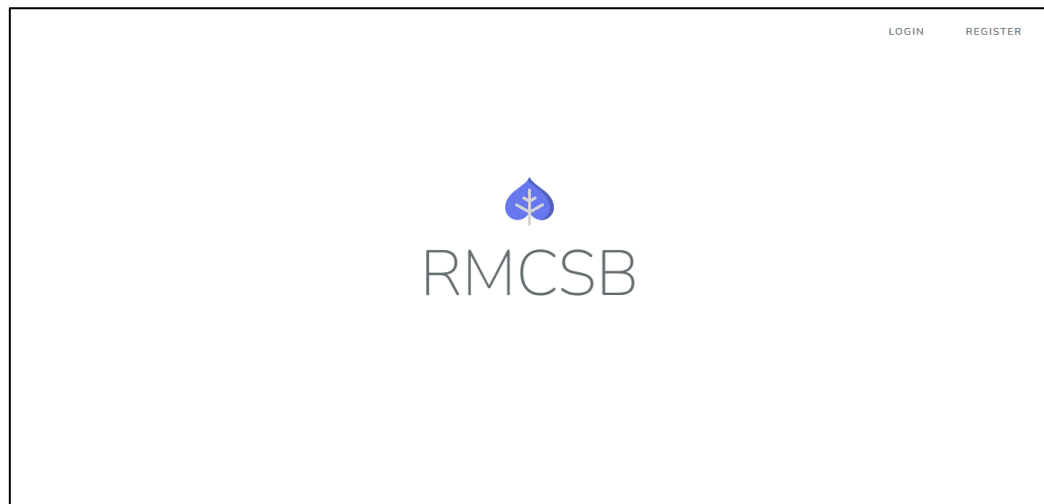
6.1 Overview of the User Interface

The main function of the system is to track the performance of the employees by tracking the tasks and sales that has been made by the employees. The users may ensure that the tasks are assigned and updated and recorded. Moreover, the system allows the user to track the performance with data visualization to ease the decision making. Moreover, the system also has evaluation system which the user may evaluate the user.

6.1.1 Screen Images

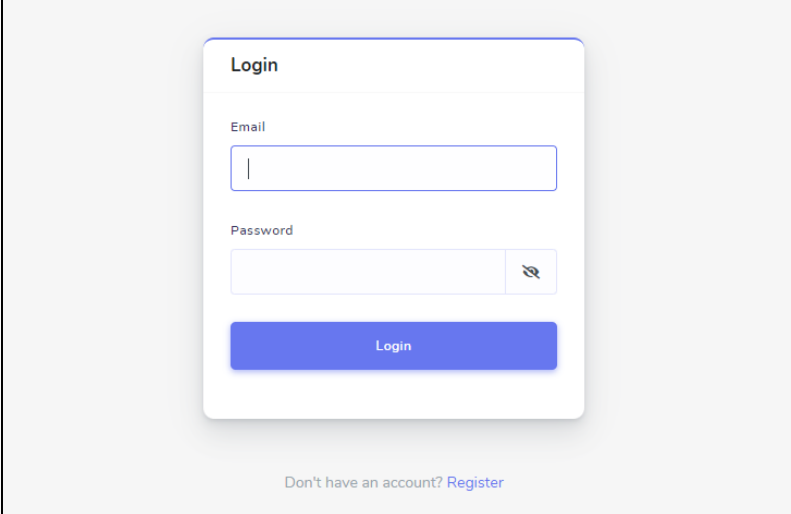
6.1.1.1 Main Page

The figure shows the main page of the system. The shortform of the company, logo and login and register will be appeared at the right top page to help the users to login.



6.1.1.2 Login Page

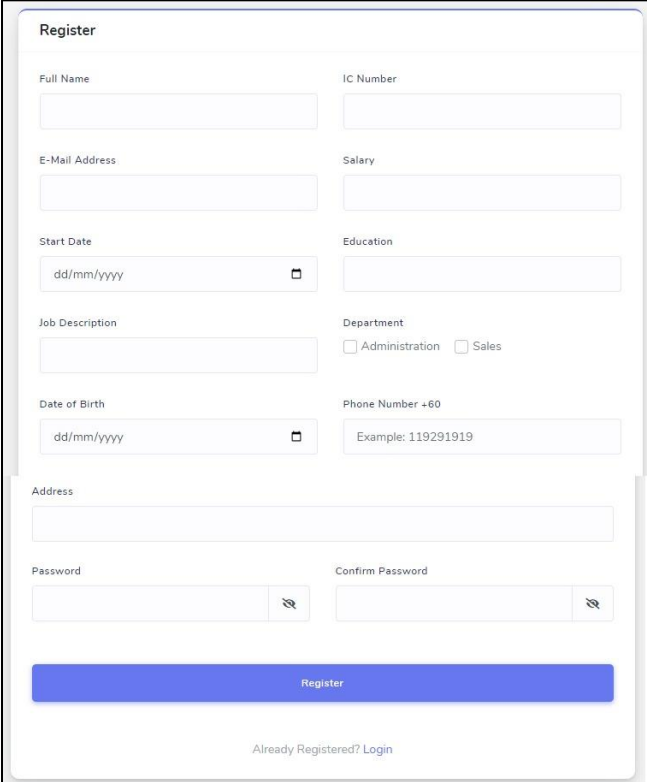
The figure shows the login page of the system. To access the system, the employees must enter the email and password that they have registered.



The login form is a white card with rounded corners on a light gray background. It has a title 'Login' at the top. Below the title are two input fields: 'Email' and 'Password'. The 'Email' field is a simple text box. The 'Password' field is a text box with a toggle icon (an eye) to its right. Below these fields is a blue button labeled 'Login'. At the bottom of the card, there is a link that says 'Don't have an account? [Register](#)'.

6.1.1.3 Register Page

The figure shows the register page of the system. Therefore, this is the page where the new employees must register in the required form.



The register form is a white card with rounded corners on a light gray background. It has a title 'Register' at the top. Below the title are several input fields and checkboxes. The fields are: 'Full Name', 'IC Number', 'E-Mail Address', 'Salary', 'Start Date' (with a date picker icon), 'Education', 'Job Description', 'Date of Birth' (with a date picker icon), 'Phone Number +60' (with an example '119291919'), and 'Address'. There are also two checkboxes for 'Department': 'Administration' and 'Sales'. Below these fields is a blue button labeled 'Register'. At the bottom of the card, there is a link that says 'Already Registered? [Login](#)'.

6.1.1.4 Home Page

6.1.1.5 Account Page

The figure shows the account page of the user of the system. Therefore, this is the page where the user's data will be displayed.

6.1.1.6 List of Employee Page

The figure shows the list of employees in the system. Therefore, the directors can access the page.

No	Name	Date of Birth	Job Description	Salary	Phone Number	Action
1	Mira Filzah	2022-05-07	234234	RM 23423	+60-234234	Detail Delete
2	Nor Huwaida Binti Mat Isa	2022-05-05	Account Executive	RM 342424	+60-56567567	Detail Delete
3	Ebit Lew	2022-05-12	123	RM 4324	+60-9	Detail Delete

6.1.1.7 List of Task Page

The figure shows a page of view the tasks that has been assigned to administrator employees and completed tasks.

No	Task	Status	Created	End Date	Action
1	aasdsad	Assigned	3 days ago	2022-05-10 19:16:00 3 days before	Edit Detail Delete
2	CMV for MKH	Assigned	1 second ago	2022-05-14 13:25:00 3 minutes after	Edit Detail Delete

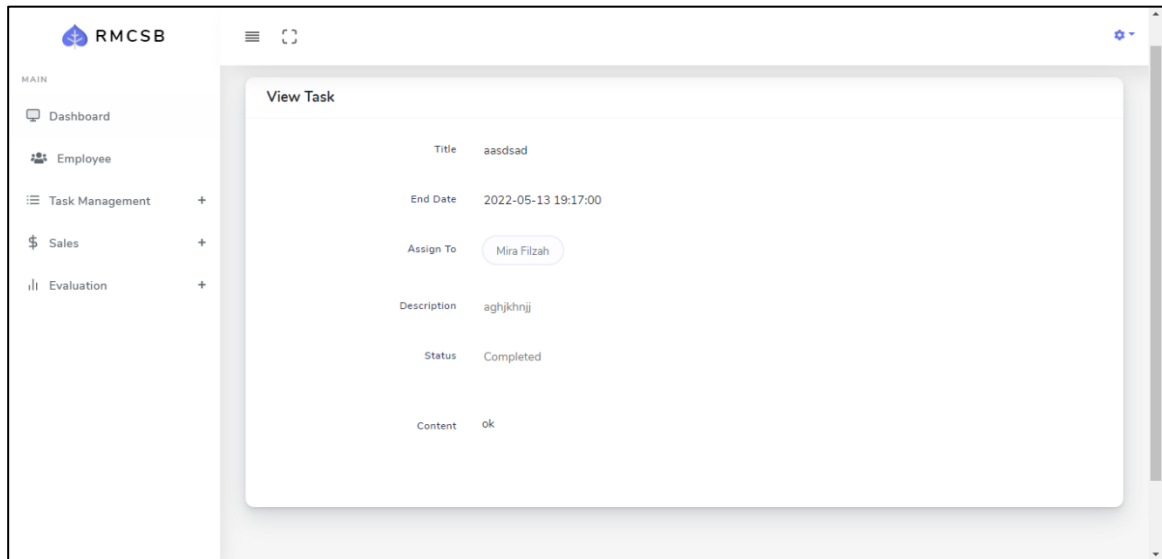
Showing 1 to 2 of 2 entries

Previous 1 Next

No	Task	Status	Submitted	Start Date	End Date	Action
----	------	--------	-----------	------------	----------	--------

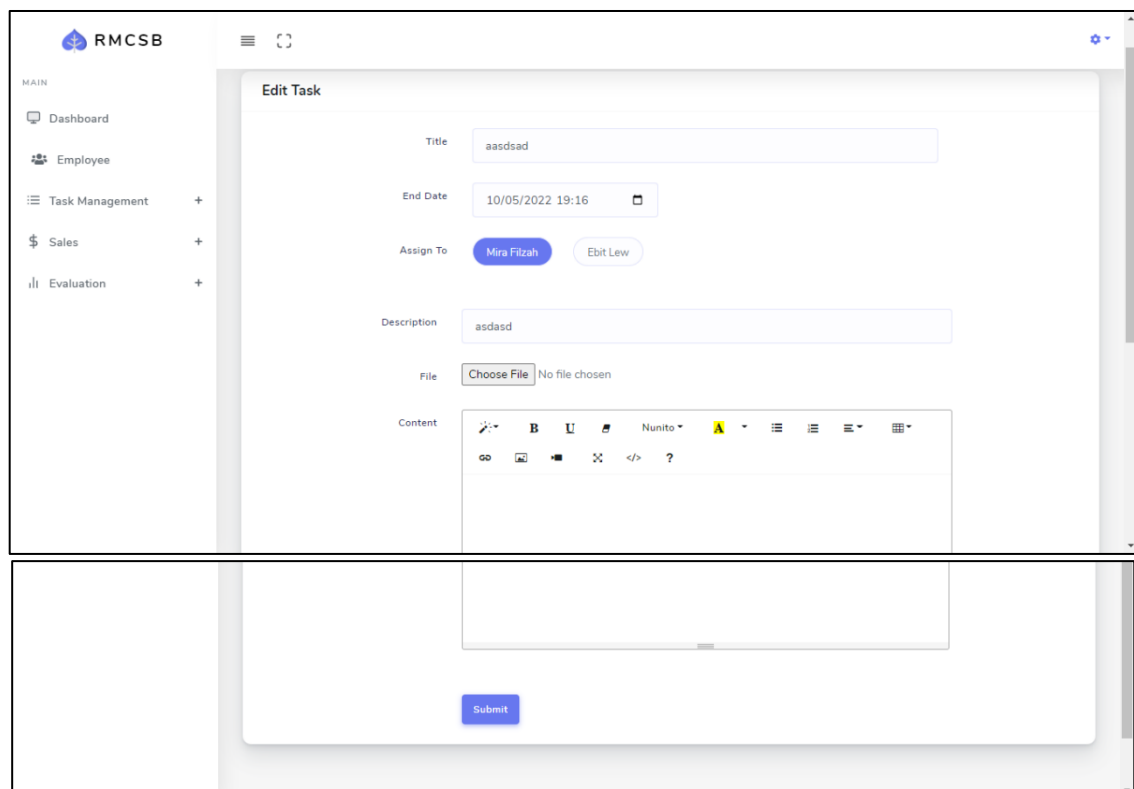
6.1.1.9 View Task Page

The figure shows a page of viewing the tasks that that has been assigned or completed to administrator employees.



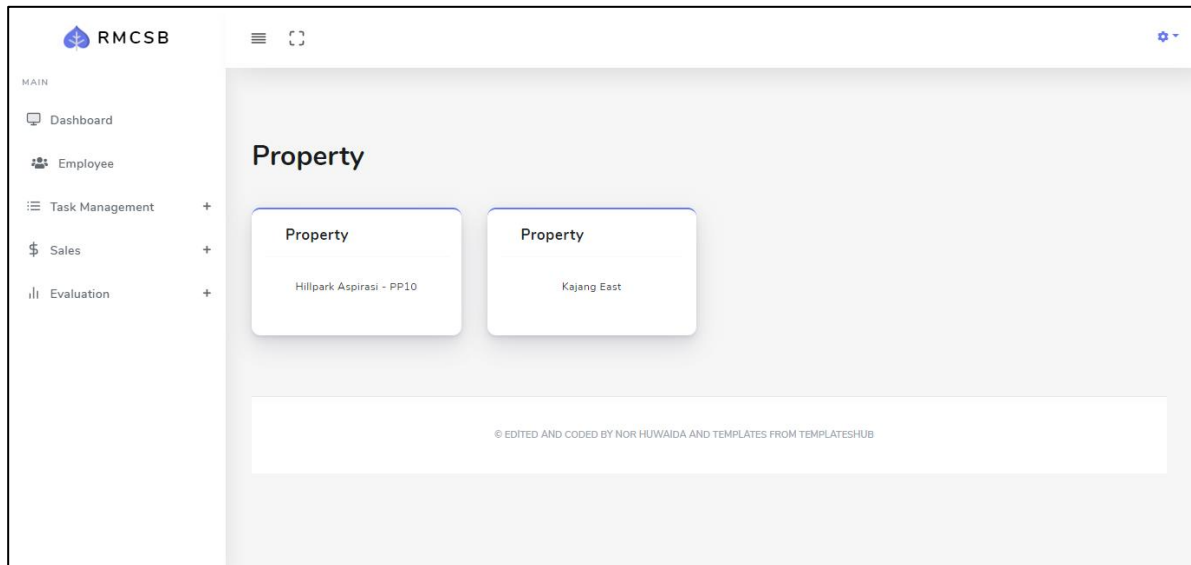
6.1.1.10 Edit Task Page

The figure shows a page of editing the tasks that that has been assigned to administrator employees.



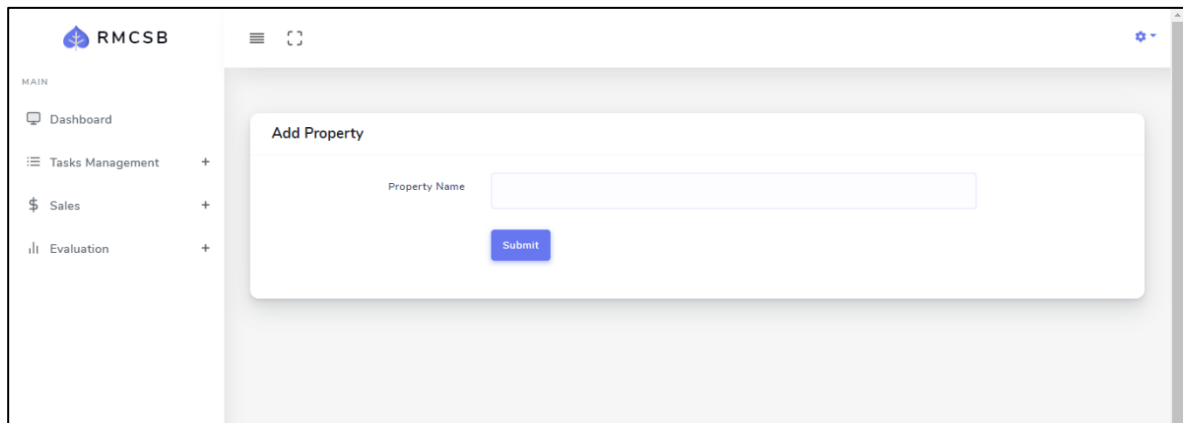
6.1.1.11 List of Property Page

The figure shows a page of list of the properties that the company associated with under the sales.



6.1.1.12 Create Property Page

The figure shows a page of creating of the properties that the company associated with under the sales.



6.1.1.13 List of Sales Property Page

6.1.1.14 Create Property Sales Page

6.1.1.15 Create Evaluation Page

6.1.1.16 View Evaluation Page

6.1.2 Report

<A description of major reports provided by the system (e.g., type of report, etc) if applicable>

7.1 View Layer

[illegible]

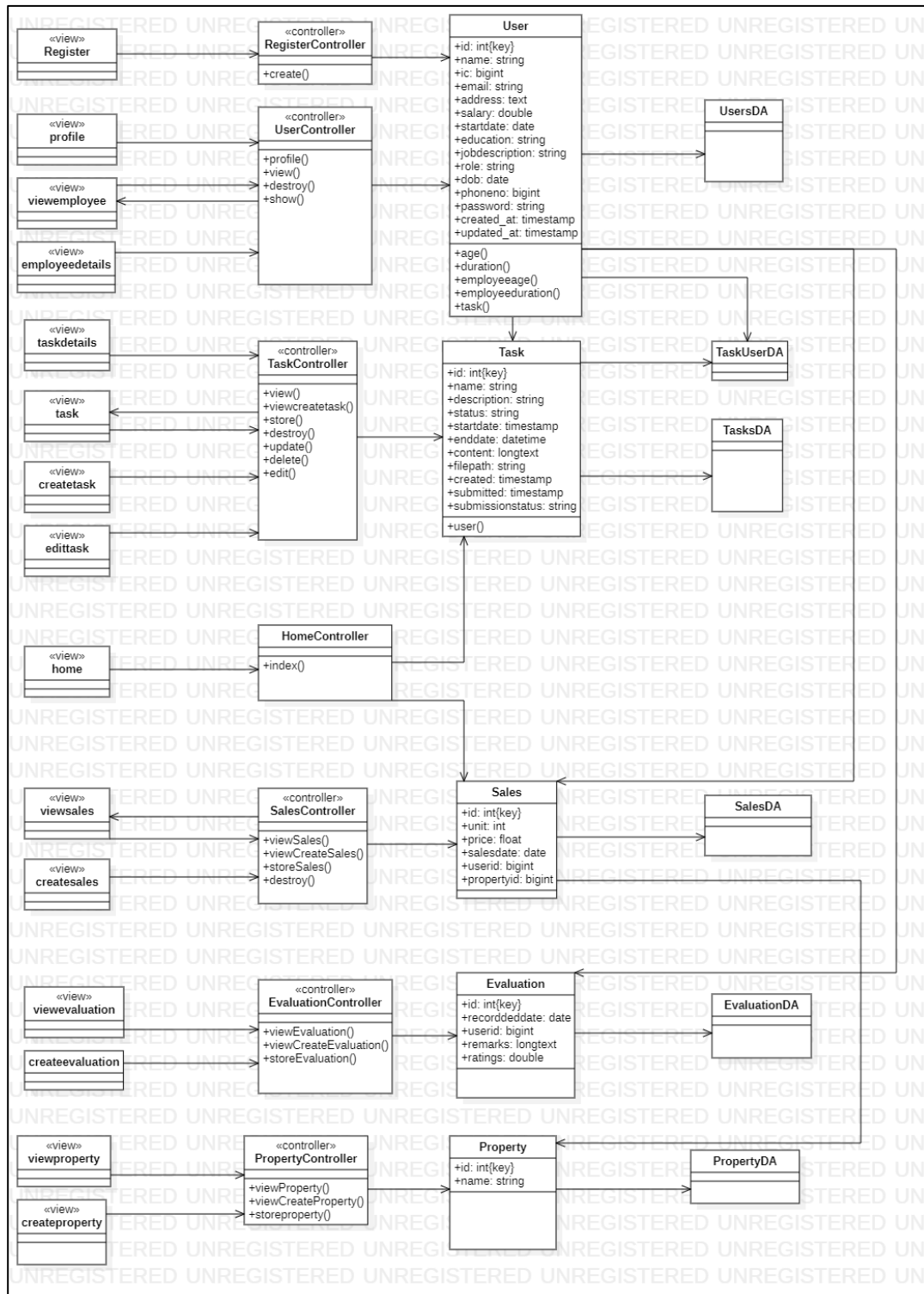
7.2 Domain Layer

7.3 Data Access Layer

CLASS	SDD_CLASS_301	SDD_CLASS_302	SDD_CLASS_303	SDD_CLASS_304	SDD_CLASS_305	SDD_CLASS_306
CREATE ACCOUNT	✓					
VIEW EMPLOYEE ACCOUNT	✓					
DELETE EMPLOYEE ACCOUNT	✓					
CREATE TASKS	✓	✓	✓			
VIEW TASKS	✓	✓	✓			
UPDATE TASKS	✓	✓	✓			
DELETE TASKS		✓	✓			
UPDATE TASKS PROGRESS			✓			
CREATE PROPERTIES					✓	
VIEW PROPERTIES					✓	
CREATE PROPERTIES SALES				✓		
VIEW PROPERTIES SALES				✓		
DELETE PROPERTIES SALES				✓		
CREATE EMPLOYEE EVALUATION						✓
VIEW EVALUATION						✓

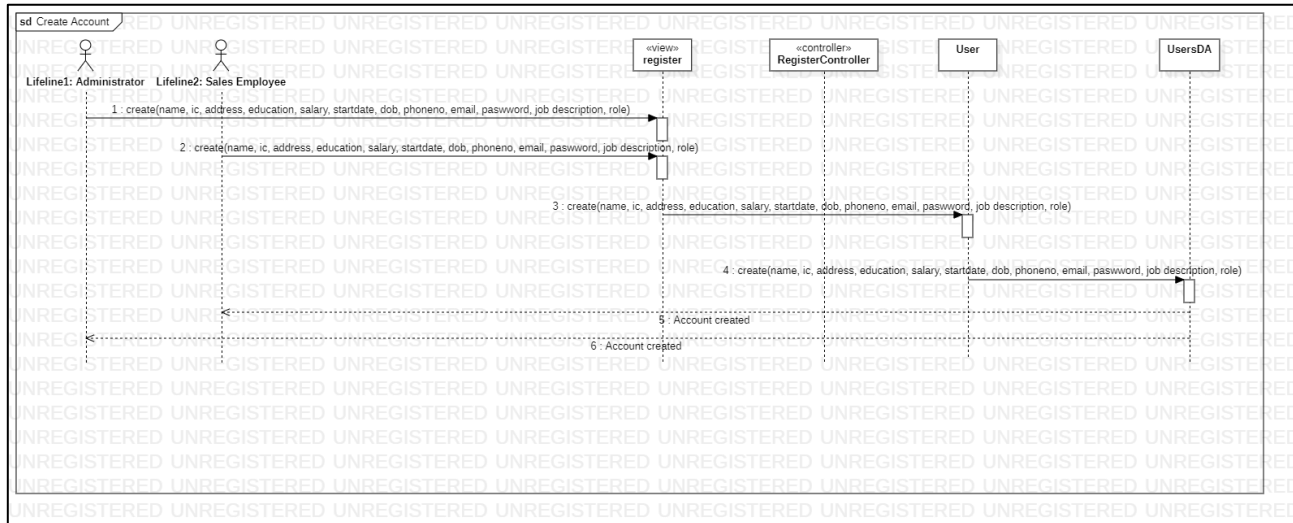
8. Appendices

8.1 Detailed Class Diagram



8.2 Create Account

8.2.1 Multilayer Sequence Diagram

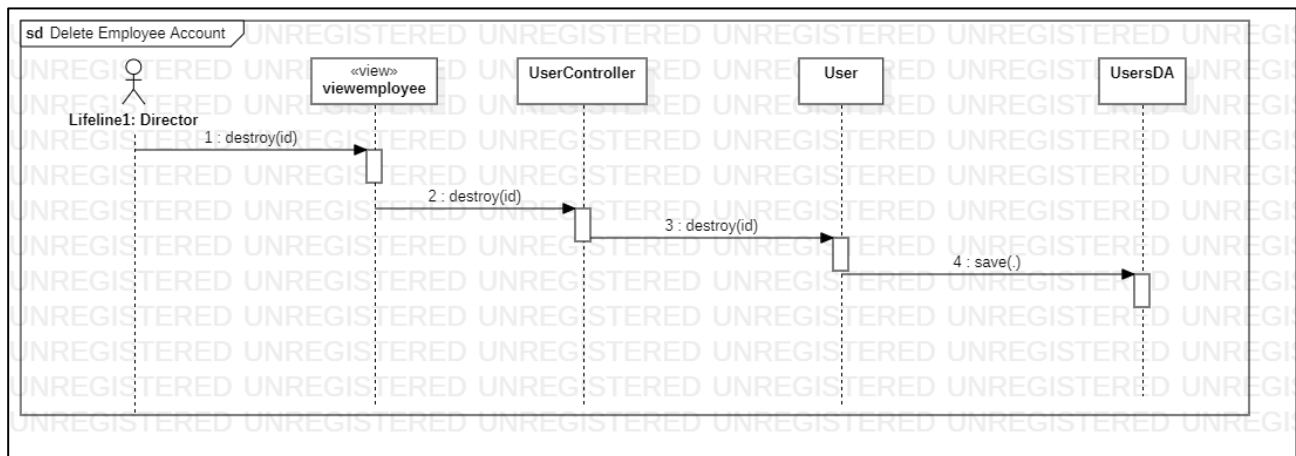


8.3 View Employee Account

8.3.1 Multilayer Sequence Diagram

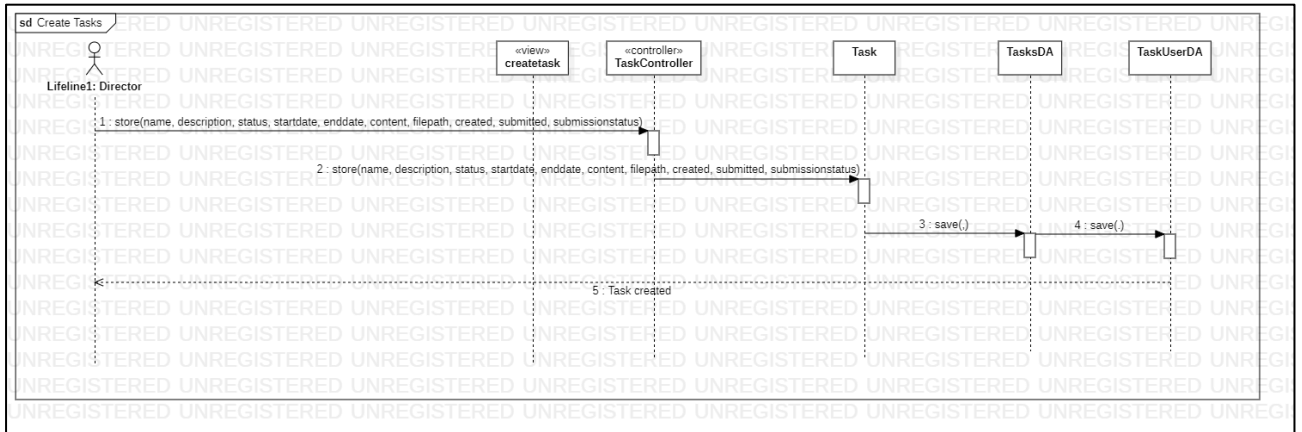
8.4 Delete Employee Account

8.4.1 Multilayer Sequence Diagram



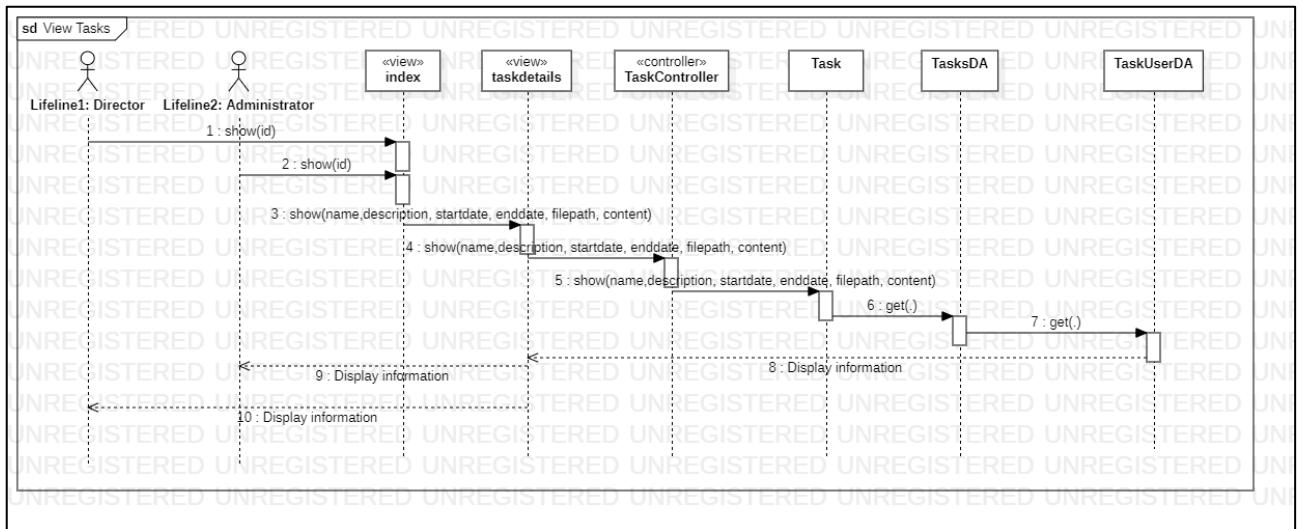
8.5 Create Tasks

8.5.1 Multilayer Sequence Diagram



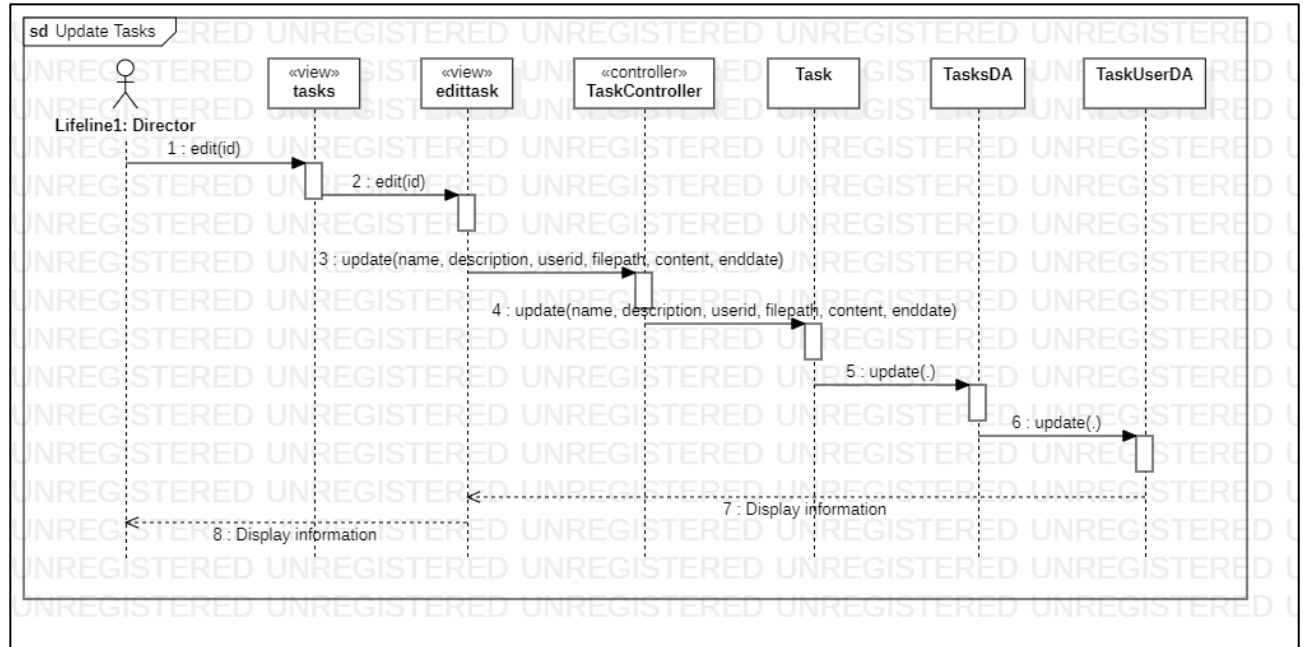
8.6 View Tasks

8.6.1 Multilayer Sequence Diagram



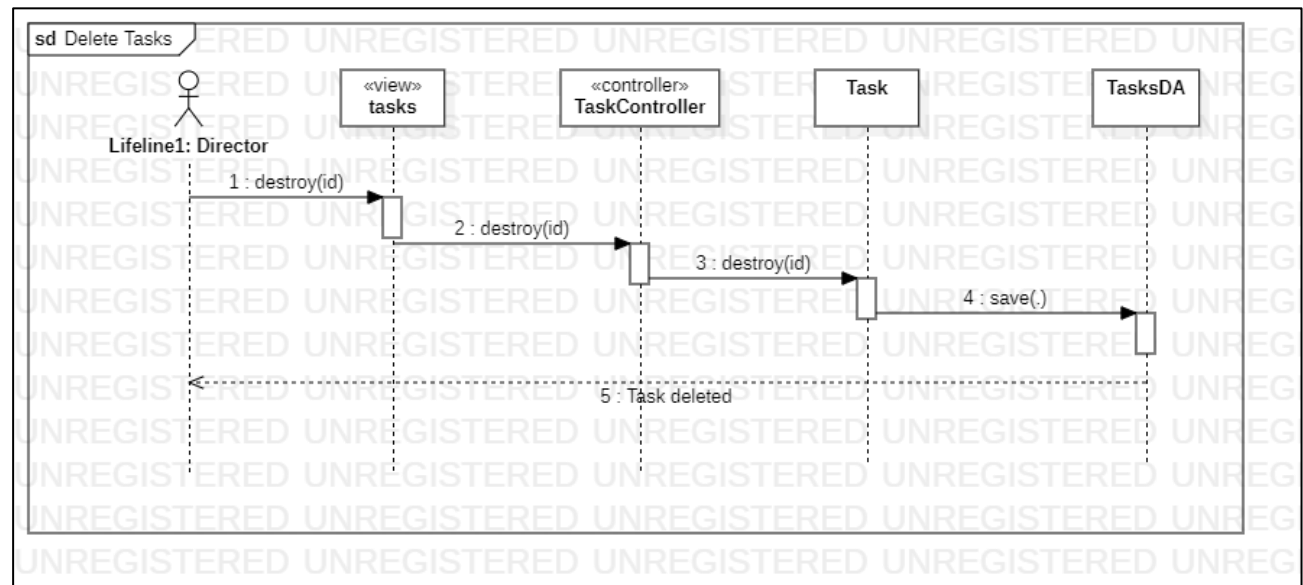
8.7 Update Tasks

8.7.1 Multilayer Sequence Diagram



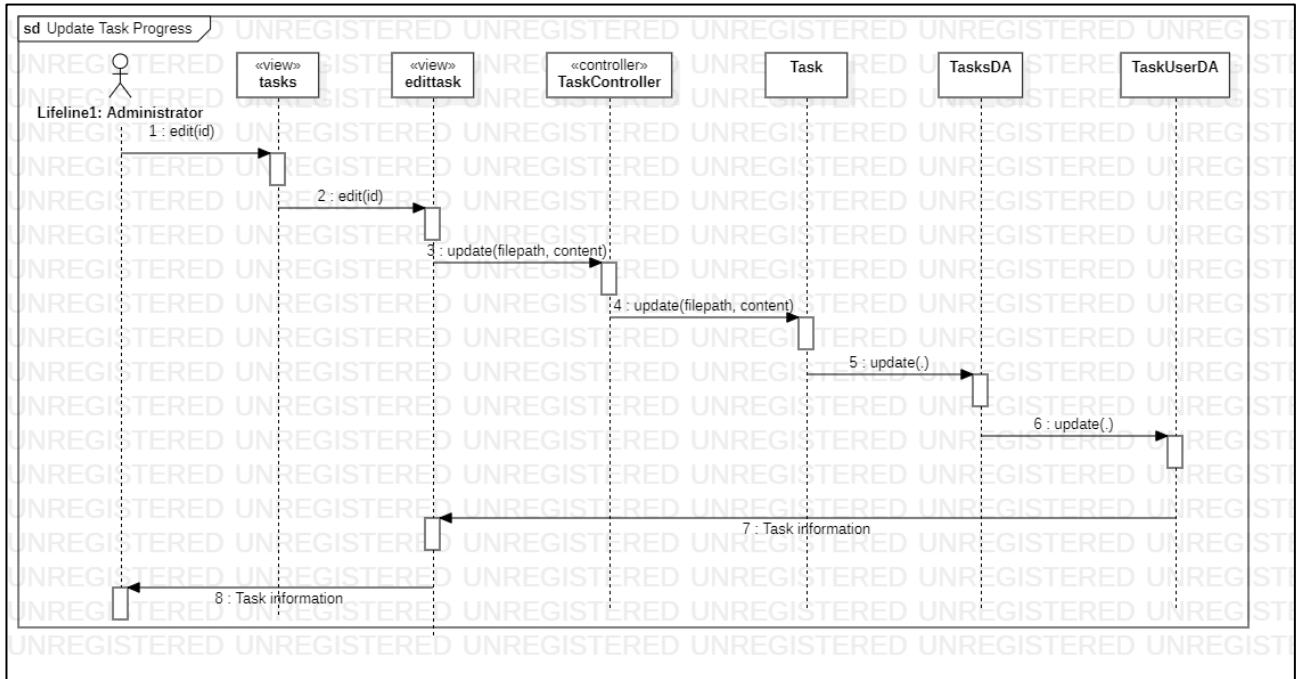
8.8 Delete Tasks

8.8.1 Multilayer Sequence Diagram



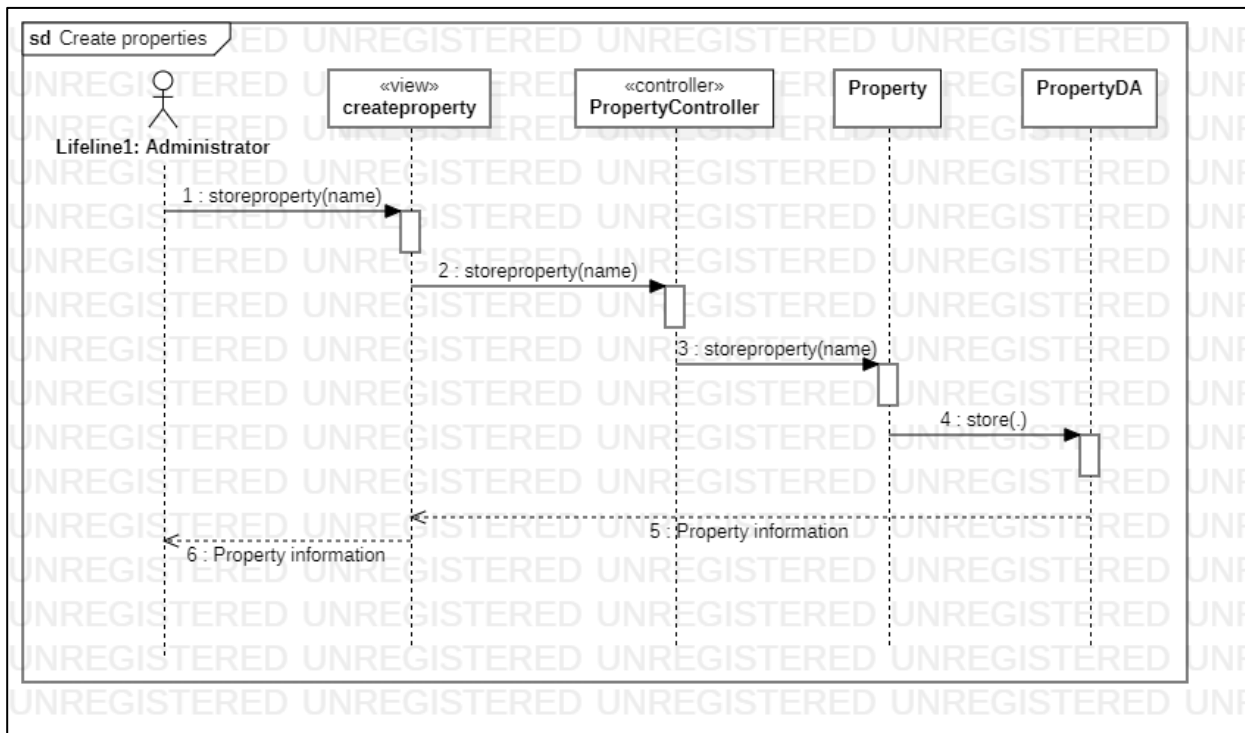
8.9 Update Task Progress

8.9.1 Multilayer Sequence Diagram



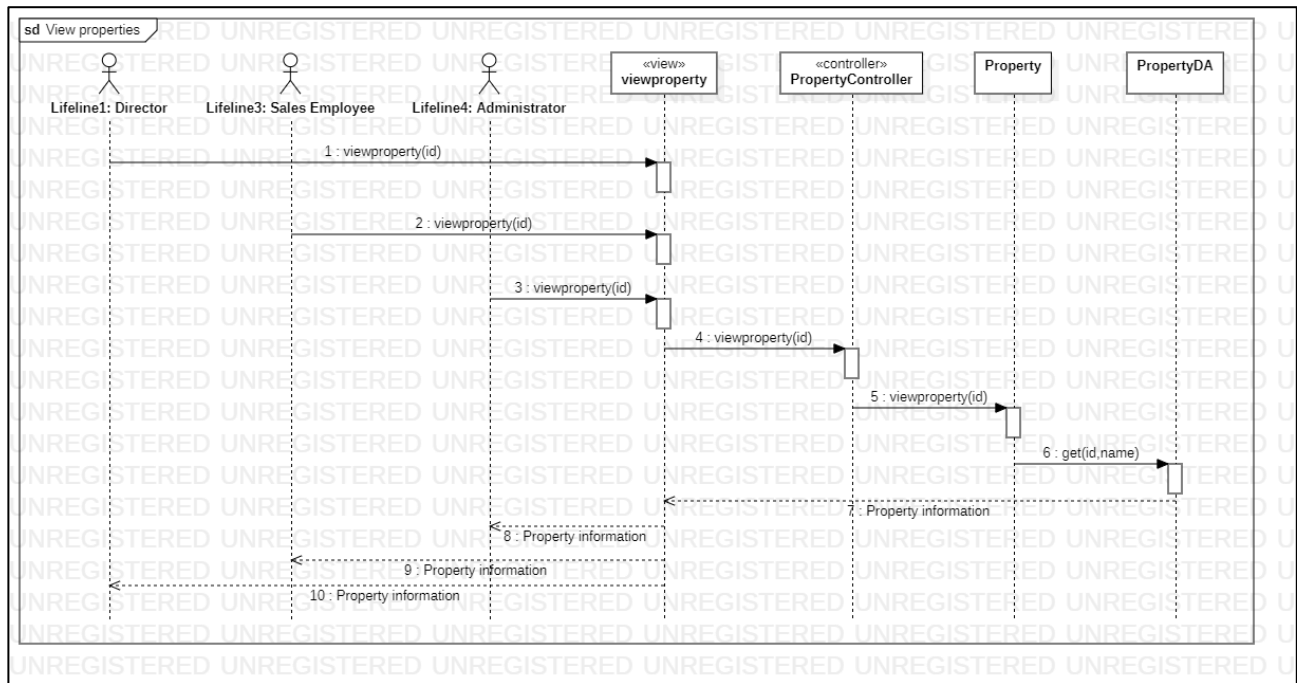
8.10 Create Properties

8.10.1 Multilayer Sequence Diagram



8.11 View Properties

8.11.1 Multilayer Sequence Diagram



8.12 Create Properties Sales

8.12.1 Multilayer Sequence Diagram

8.13 View Properties Sales

8.13.1 Multilayer Sequence Diagram

8.14 Delete Properties Sales

8.14.1 Multilayer Sequence Diagram

8.15 Create Employee Evaluation

8.15.1 Multilayer Sequence Diagram

8.16 View Evaluation

8.16.1 Multilayer Sequence Diagram