



SDD Document SEM I 20212022

PSM Management System (PSMMS)

Dingo Gang

- | | |
|--|-----------|
| 1. Muhammad Afif bin Ab Rahman | [CB19007] |
| 2. Muhammad Ilman Nashran bin Mohd Nasruddin | [CB19014] |
| 3. Siti Nurasma' binti Hashim | [CB19123] |
| 4. Nureen Nasuha binti Mohd Rosli | [CB19124] |
| 5. Nabilah Ayuni binti Mohd Nasir | [CB19125] |
| 6. Nur Alia Shahira binti Mohd Zaidi | [CB19141] |

TABLE OF CONTENTS

1	INTRODUCTION	4
1.1	Purpose	4
1.2	System Identification	4
1.3	Module Overview	5
1.3.1	Manage Lecturer Expertise	5
1.3.2	Manage Profile	5
1.3.3	Manage Supervisor Hunting	5
1.3.4	Manage Title from Student and Lecturer	5
1.3.5	Manage Approval and Proposal Submission	5
1.3.6	Manage Logbook	5
2	REFERENCE DOCUMENT	7
3	Data design	8
3.1	General Architecture	8
3.1.1	3-Tier Architecture Layer	9
3.1.1.1	Manage Lecturer Expertise [SDD-REQ-100]	9
3.1.1.2	Manage Profile [SDD-REQ-200]	10
3.1.1.3	Manage Supervisor Hunting [SDD-REQ-300]	11
3.1.1.4	Manage Title [SDD-REQ-400]	12
3.1.1.5	Manage Approval and Proposal Submission [SDD-REQ-500]	13
3.1.1.6	Manage Logbook [SDD-REQ-600]	14
3.1.2	Framework	15
3.2	Package Module	17
3.3	Data Dictionary	18
4	DETAIL DESIGN	21
4.1	Manage Lecturer Expertise [SDD-REQ-100]	21
4.2	Manage Profile [SDD-REQ-200]	29
4.3	Manage Supervisor Hunting [SDD-REQ-300]	37
4.4	Manage Title [SDD-REQ-400]	39
4.5	Manage Approval and Proposal Submission [SDD-REQ-500]	48
4.6	Manage Logbook [SDD-REQ-600]	57
5.	REQUIREMENT TRACEABILITY	69

5.1 Manage Lecturer Expertise [SRS-REQ-100]	69
5.2 Manage Profile [SRS-REQ-200]	69
5.3 Manage Supervisor Hunting [SRS-REQ-300]	70
5.4 Manage Title from Student and Lecturer [SRS-REQ-400]	70
5.5 Manage Approval and Proposal Submission [SRS-REQ-500]	71
5.6 Manage Logbook [SRS-REQ-600]	71

1 INTRODUCTION

1.1 Purpose

The goal of this System Design Document (SDD) is to provide the essential guidelines for designing and implementing software that meets all the criteria.

The purpose of this document in the instance of the PSM Management System (PSMMS) is to provide an online system to manage both supervisor and student for PSM subjects where students need to find a supervisor to guide them through their project. It will describe the system's goal and features, as well as the system's interfaces, what the system will perform, the constraints that it must function under, and how the system will react to external stimuli.

This paper will be design-agnostic, focusing solely on the end criteria that must be met by the final software. This document can also be utilized by clients and stakeholders as a representation of their software requirements. Fulfilling all the SDD's requirements is the same as fulfilling the client's contract.

1.2 System Identification

System Title: PSM Management System

System Abbreviation: PSMMS

Year: 2021

System Identification: SRS-PSMMS-2021-V1

1.3 Module Overview

1.3.1 Manage Lecturer Expertise

Manage Lecturer Expertise module allows the lecturers to manage their expertise by adding or editing their expertise in the system. The students can browse a list of lecturers and their expertise by picking one from the list.

1.3.2 Manage Profile

Manage Profile module allows the users (Student and Lecturer) to view their profile. The student can view the lecturer profile to gain any information about the lecturer while the lecturer also can view the student profile. The users can manage their profiles by editing, updating, or adding new information to the profile.

1.3.3 Manage Supervisor Hunting

Manage Supervisor Hunting module allows the student to search for a potential supervisor or PSM title available to request. Students can book the supervisor chosen and fill in a project proposal form if he/she wants to propose their own title. The student also can choose the supervisor title available and even filter the search based on the lecturer's name or the lecturer's field of expertise.

1.3.4 Manage Title from Student and Lecturer

Manage Title module allows the lecturer to add and modify their title, as well as view it in the system. After the lecturer enters their title into the system, the student can view the title and can book the title that they want. The student also can search for their preferred title in the system if they have the title that they want to search.

1.3.5 Manage Approval and Proposal Submission

Manage Approval and Proposal Submission module allows the student to submit their proposals to the lecturers by filling out a proposal form and viewing the results of their proposal application. The lecturers can manage the approval process for each proposal submitted in this PSMMS system.

1.3.6 Manage Logbook

Manage Logbook module allows the students to update their PSM progress in the logbook. If there is any discussion or meeting with their supervisor, they need to report the progress in the logbook with some information such as Week, Date, Time, Current Progress and Upcoming Progress. The student can

also view and edit their logbook. The student's logbook can be generated and printed before the evaluation presentation. The supervisor can monitor their student's logbook progress and add some comments on the progress. The supervisor needs to approve their student's logbook.

2 REFERENCE DOCUMENT

This document is prepared in reference to the following documents:

Armeli, J. (2014). *Web application development with Laravel PHP Framework version 4*. [Bachelor of Engineering, Helsinki Metropolia University of Applied Sciences].

<https://www.theseus.fi/bitstream/handle/10024/74052/Author.pdf>

Rajput, S. (2020). What are the reasons to choose Laravel MVC for web development. *MSC IT Research Methods and Professional Issues*. Published. <https://doi.org/10.31219/osf.io/fgq3z>

Tutorials Point (2016). *Laravel Tutorial*. Tutorials Point (I) Pvt. Ltd.

https://www.tutorialspoint.com/laravel/laravel_tutorial.pdf

Laravel. (n.d.). *Request Lifecycle - Laravel - The PHP Framework For Web Artisans*. Laravel – The PHP Framework For Web Artisans. Retrieved December 1, 2021, from

<https://laravel.com/docs/8.x/lifecycle>

Patel, J. (2021, November 18). *Why Laravel PHP Framework is Best Considered to Use in 2021?*

Monocubed. Retrieved December 2, 2021, from <https://www.monocubed.com/why-laravel-php-framework/>

Ighodaro, N. (2018, August 22). *How Laravel implements MVC and how to use it effectively*. Blog -

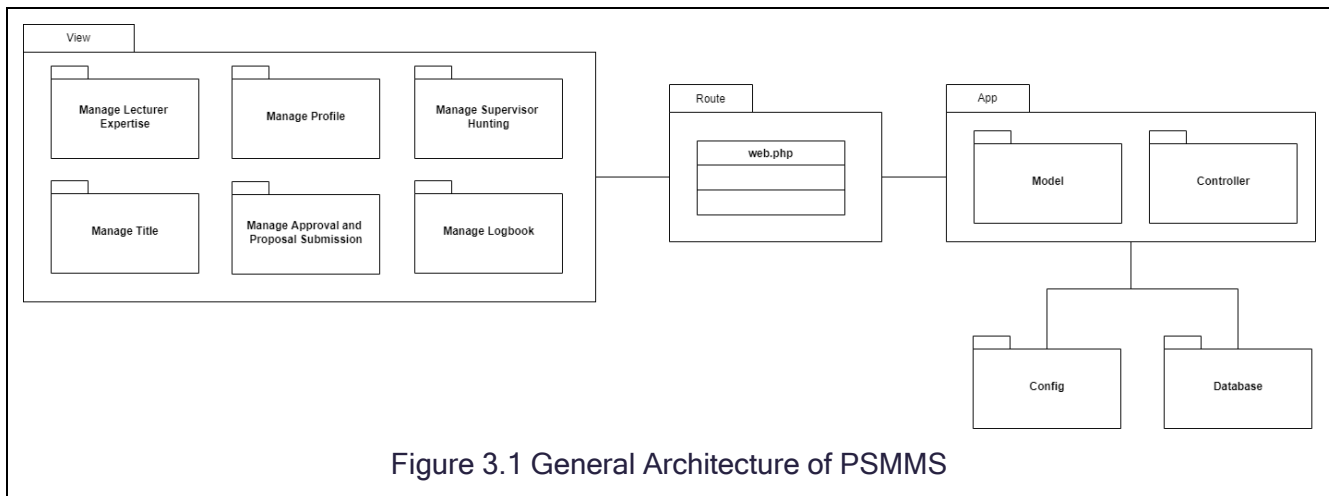
Pusher. Retrieved December 2, 2021, from <https://blog.pusher.com/laravel-mvc-use/>

Vincy. (2021, July 27). *PHP Laravel Project Example for Beginners*. Phppot. Retrieved December 2,

2021, from <https://phppot.com/php/php-laravel-project-example/>

3 DATA DESIGN

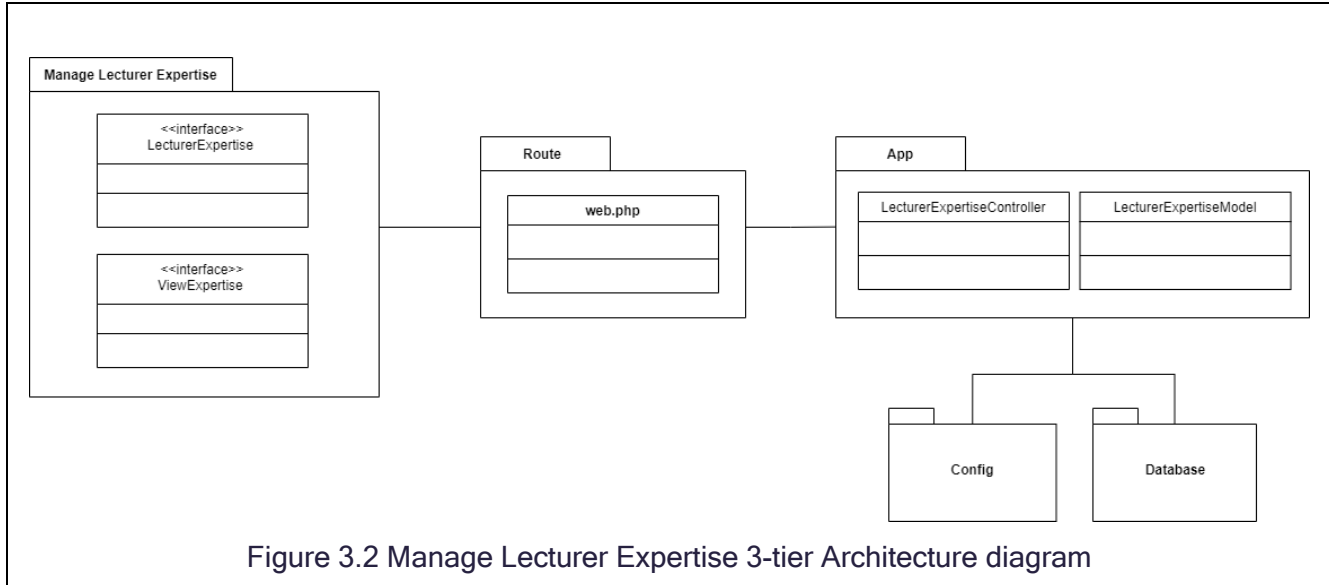
3.1 General Architecture



3.1.1 3-Tier Architecture Layer

3.1.1.1 Manage Lecturer Expertise [SDD-REQ-100]

Siti Nurasma' binti Hashim [CB19123]



3.1.1.1.1 Manage Lecturer Expertise

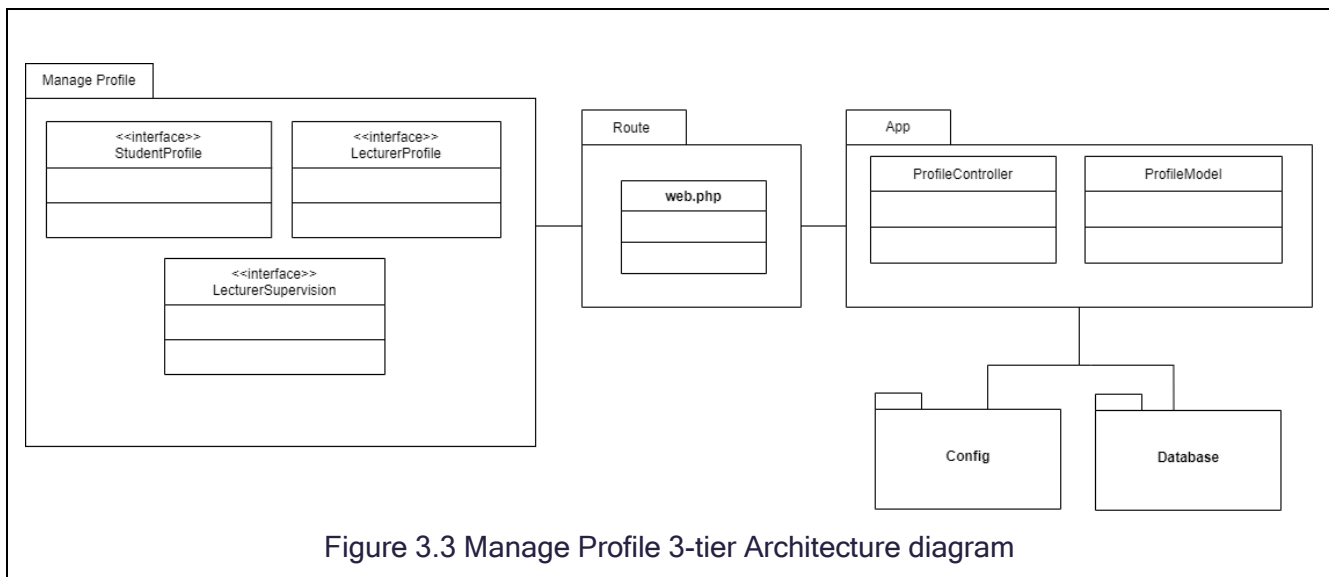
Class Name	Description
LecturerExpertise	This class allows the lecturer to view the expertise details, and perform an action which are, add new expertise and edit existing expertise details.
ViewExpertise	This class allows the student to view the lecturer's expertise.

3.1.1.1.2 App

Class Name	Description
LecturerExpertiseController	This controller is used to communicate Manage Lecturer Expertise Interface with the entity class Model.
LecturerExpertiseModel	This model is used to fetch data from the database to display in the LecturerExpertise and viewExpertise.

3.1.1.2 Manage Profile [SDD-REQ-200]

Nur Alia Shahira binti Mohd Zaidi [CB19141]



3.1.1.2.1 Manage Profile

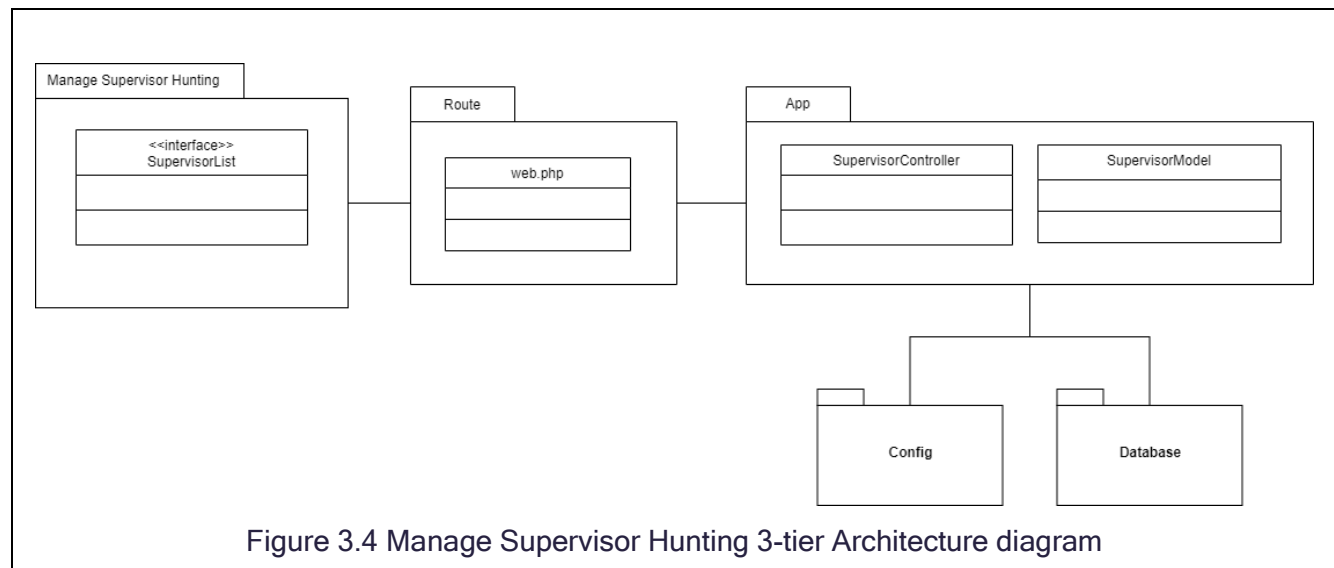
Class Name	Description
LecturerProfile	This class allows the student and lecturer to view lecturer's profile.
StudentProfile	This class allows the student and lecturer to view student's profile.
LecturerSupervision	This class allows the lecturer to view the student under his/her supervision

3.1.1.2.2 App

Class Name	Description
ProfileController	The controller that communicates with Manage Profile Interface with entity class Model (ProfileModel)
ProfileModel	This model is used to query and retrieve data from database that is being displayed in StudentProfile interface and LecturerProfile interface

3.1.1.3 Manage Supervisor Hunting [SDD-REQ-300]

Muhammad Afif bin Ab Rahman [CB19007]



3.1.1.3.1 Manage Supervisor Hunting

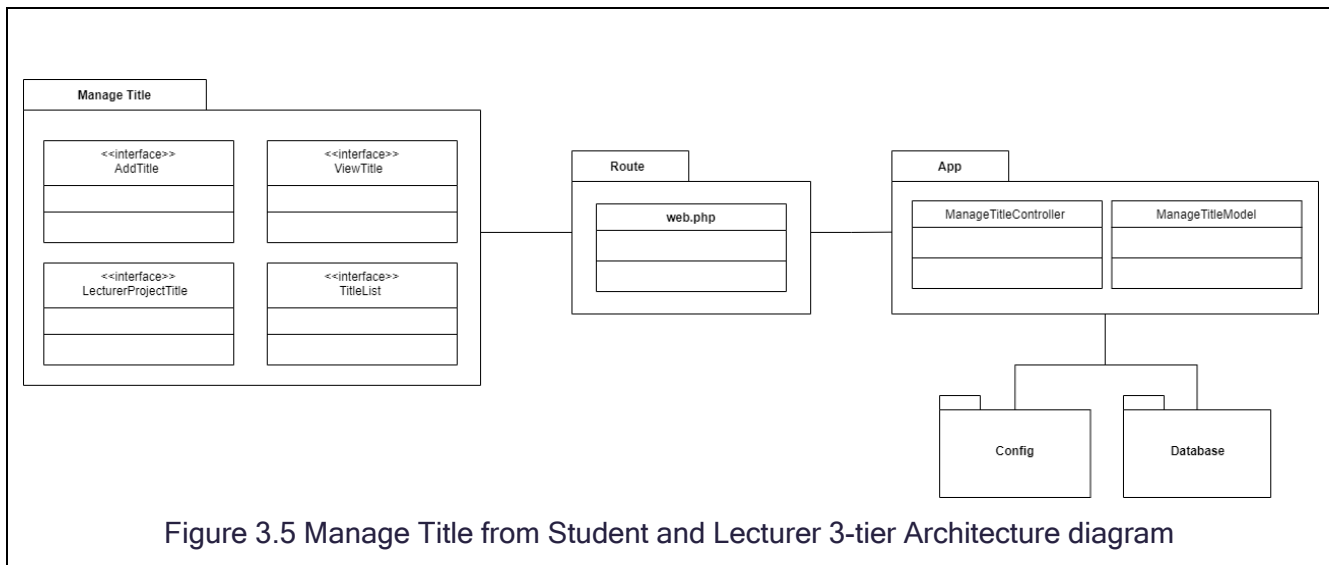
Class Name	Description
SupervisorList	This class allows the students to browse lecturer list with their basic information and book any supervisor that they want

3.1.1.3.2 App

Class Name	Description
SupervisorController	This controller is used to communicate with the Manage Supervisor Hunting Interface with the entity class Model (Supervisor)
SupervisorModel	This model is used to fetch data from the database to display in the SupervisorList interface

3.1.1.4 Manage Title [SDD-REQ-400]

Nabilah Ayuni binti Mohd Nasir [CB19125]



3.1.1.4.1 Manage Title

Class Name	Description
AddTitle	This class allows the lecturer to add their title into the system.
ViewTitle	This class allows the lecturer to view and edit lecturer's title details.
LecturerProjectTitle	This class allows the lecturer to view title list, edit and delete their title information.
TitleList	This class allows the student to view the project title list, project title status and book their preferred title.

3.1.1.4.2 App

Class Name	Description
ManageTitleController	This controller is being used to communicate with ManageTitleView and the entity class Model.
ManageTitleModel	This model is used to store and fetch data from the database for the Manage Title from Student and Lecturer module.

3.1.1.5 Manage Approval and Proposal Submission [SDD-REQ-500]

Muhammad Iman Nashran bin Mohd Nasruddin [CB19014]

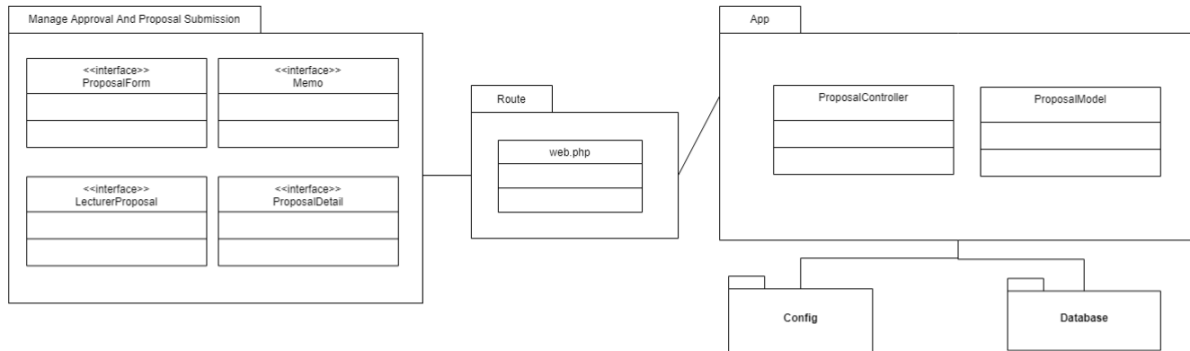


Figure 3.6 Manage Approval and Proposal Submission 3-tier Architecture diagram

3.1.1.5.1 Manage Approval and Proposal Submission

Class Name	Description
ProposalForm	This class allows the students to submit their proposal by inserting the proposal from.
Memo	This class allows the students to view their proposal status.
LecturerProposal	This class allows the lecturer to view, accept and reject the proposal that is being submitted by students.
ProposalDetail	This class allows the lecture to view details of the proposal.

3.1.1.5.2 App

Class Name	Description
ProposalController	This controller is used to communicate within the Proposal Model and Proposal View.
ProposalModel	This model is used to store and fetch data from the database for the Manage Approval and proposal submission.

3.1.1.6 Manage Logbook [SDD-REQ-600]

Nureen Nasuha binti Mohd Rosli [CB19124]

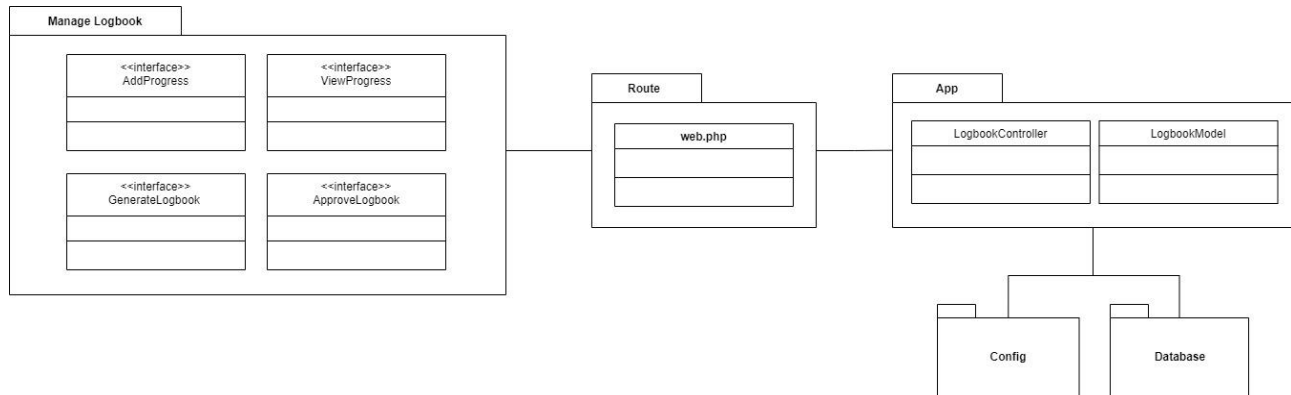


Figure 3.7 Manage Logbook 3-tier Architecture diagram

3.1.1.6.1 Manage Logbook

Class Name	Description
AddProgress	This class allows the student to add, view and edit their logbook progress into the system
GenerateLogbook	This class allows the system to generate the logbook in the system
ViewProgress	This class allows the lecturer to view and add comment into their student under supervision logbook progress in the system
ApproveLogbook	This class allows the lecturer to approve their student under supervision logbook progress in the system

3.1.1.6.2 App

Class Name	Description
LogbookController	This controller is used to communicate with Manage Logbook Interface with the entity class Model.
LogbookModel	This model is used to store, query, and retrieve data from the database and display in ViewProgress

3.1.2 Framework

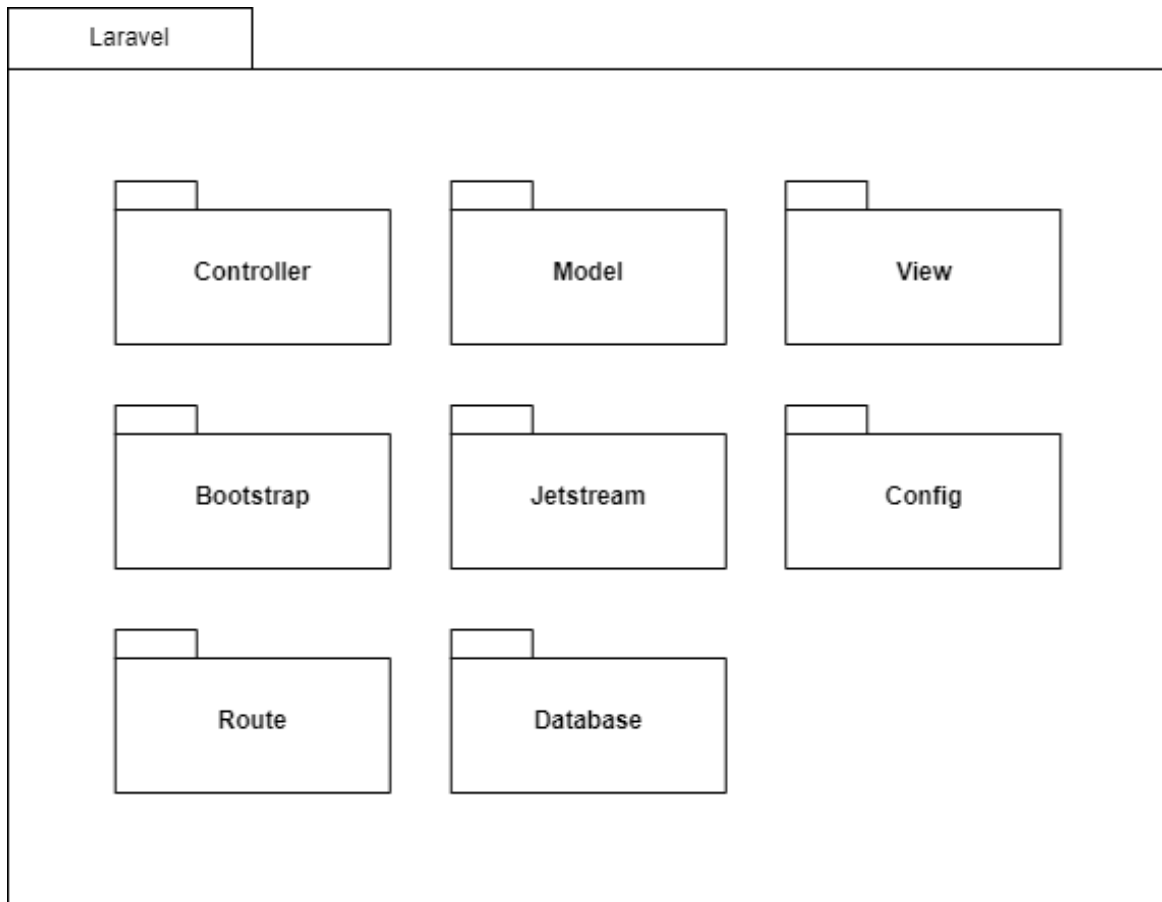


Figure 3.8 Framework of PSMMS

Package Name	Description
Bootstrap	Bootstrap is a popular CSS framework which has simple, yet functional design. Bootstrap has a lot of templates available online due to its popularity, which will potentially ease development
Controllers	Controllers act as an interface between Model and View components to process all the business logic and incoming requests, manipulate data using the Model component and interact with the Views to render the final output.
Models	The Model component corresponds to all the data-related logic that the user works with.

Database	A database is a data structure that stores organized information.
Views	View component is used for all the UI logic of the application.
Routes	Route is a way of creating a request URL of your application.
Config	Database links the information, mail server information, and other core configuration values such as application time zone, and encryption key can all be configured using these configuration files.
Jetstream	Jetstream provides the implementation for your application's login, registration, email verification, two-factor authentication, session management, API via Laravel Sanctum, and optional team management features.

3.2 Package Module



Figure 3.9 Package Module of PSMMS

3.3 Data Dictionary

3.3.1 Manage Lecturer Expertise table

Field Name	Description	Data Type	Constraint
Expertise_ID	Lecturer expertise	Varchar(10)	PK
Lecturer_ID	Lecturer ID number	Varchar(10)	FK
Expertise_Name	Expertise name	Varchar(255)	-
Expertise_Level	Expertise level	Varchar(255)	-

3.3.2 Manage Profile

Lecturer table

Field Name	Description	Data Type	Constraint
Lecturer_ID	Lecturer ID number	Varchar(10)	PK
Lecturer_Name	Lecturer's Name	Varchar(255)	-
Lecturer_Email	Lecturer's Email Address	Varchar(255)	-
Lecturer_OfficeNo	Lecturer's Office Number	Varchar(10)	-
Lecturer_Biography	Lecturer's Biography details	Varchar(255)	-
Lecturer_Image	Lecturer profile image	Varchar(10)	-

Lecturer_Education table

Field Name	Description	Data Type	Constraint
Education_ID	Education ID number	Varchar(10)	PK
Lecturer_ID	Lecturer ID number	Varchar(10)	FK
Institution_Name	Institution Name	Varchar(255)	
Level_Education	Level of Education	Varchar(255)	
Major_Education	Major of Education	Varchar(255)	
Year_Education	Year of Education	Int	

Student table

Field Name	Description	Data Type	Constraint
Student_ID	Student ID number	Varchar(10)	PK
Lecturer_ID	Lecturer ID number	Varchar(10)	FK
Student_Name	Student's Name	Varchar(255)	
Student_Email	Student's Email Address	Varchar(255)	
Student_PhoneNo	Student's Phone Number	Varchar(15)	
Student_Major	Student's Major Course	Varchar(255)	
Student_Year	Student's Year	Int	
PSM_Level	PSM Level	Int	
Student_Image	Student Profile Image	Varchar(10)	

3.3.3 Manage Supervisor Hunting table

Data obtained from lecturer table in 2.2.2

Field Name	Description	Data Type	Constraint
Lecturer_ID	Lecturer ID number	Varchar(10)	PK
Lecturer_Name	Lecturer's Name	Varchar(255)	-
Lecturer_Email	Lecturer's Email Address	Varchar(255)	-
Lecturer_Image	Lecturer profile image	Varchar(10)	-

3.3.4 Manage Title from Student and Lecturer table

Field Name	Description	Data Type	Constraint
Title_ID	Title ID number	Varchar(10)	PK
Lecturer_ID	Lecturer ID number	Varchar(10)	FK
Student_ID	Student ID number	Varchar(10)	FK
Title_Name	Title Name	Varchar(225)	-
Title_Type	Title Type	Varchar(225)	-
Title_Description	Title Description	Varchar(225)	-
Required_Skill	Knowledge/Skill Required	Varchar(225)	-
Title_Status	Title Status	Varchar(225)	-
Lecturer_Email	Lecturer's Email Address	Varchar(255)	-

3.3.5 Manage Approval and Proposal Submission table

Field Name	Description	Data Type	Constraint
Proposal_ID	Proposal ID number	Varchar(10)	PK
Student_ID	Student ID number	Varchar(10)	FK
Lecturer_ID	Lecturer ID number	Varchar(10)	FK
Proposal_Title	Student's proposal title	Varchar(225)	-
Proposal_Type	Student's proposal type	Varchar(225)	-
SV_Name	Student's supervisor name	Varchar(225)	-
Proposal_Status	Student's proposal status	Varchar(10)	-
Proposal_Doc	Student's proposal document	Varchar(225)	-

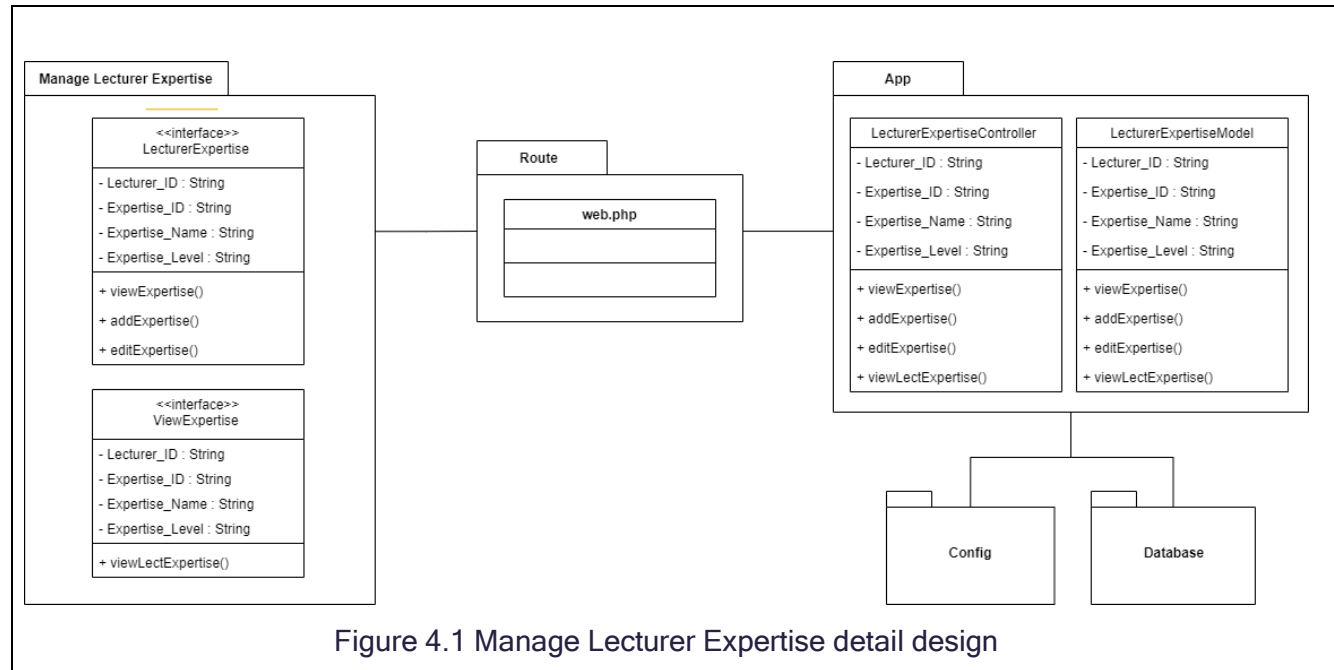
3.3.6 Manage Logbook table

Field Name	Description	Data Type	Constraint
Logbook_ID	Logbook ID	Varchar(10)	PK
Student_ID	Student matric ID	Varchar(10)	FK
Student_Name	Student name	Varchar(255)	-
Student_Title	Student PSM title	Varchar(255)	-
Week	Logbook week	Varchar(10)	-
Date	Logbook date	Date	-
Time	Logbook time	Time	-
Current_Progress	Add student current progress	Varchar(255)	-
Upcoming_Progress	Add student upcoming progress	Varchar(255)	-
Add_Comment	Add lecturer comment	Varchar(255)	-
Status_Approval	Status approval	Varchar(255)	-

4 DETAIL DESIGN

4.1 Manage Lecturer Expertise [SDD-REQ-100]

Siti Nurasma' binti Hashim [CB19123]



4.1.1 LecturerExpertise [SDD-REQ-101]

Class Type	Boundary Class	
Responsibility	An interface to display lecturer expertise and perform an action to add new expertise and edit existing expertise.	
Attributes	Attributes Name	Attributes Type
	Lecturer_ID	String
	Expertise_ID	String
	Expertise_Name	String
	Expertise_Level	String
Methods	Method Name	Description
	viewExpertise()	Method to display lecturer expertise for the lecturer.
	addExpertise()	Method for the lecturer to add new expertise into database.
	editExpertise()	Method for the lecturer to edit their expertise details.
Algorithm	<pre> viewExpertise() BEGIN GET Expertise_ID FROM expertise table; System query from database; DISPLAY Expertise_Name, Expertise_Name ACCORDING Expertise_ID. END addExpertise() BEGIN INSERT Expertise_Name, Expertise_Name INTO expertise table; System query from database; DISPLAY LecturerExpertise page. END editExpertise() BEGIN </pre>	

	<p>GET Expertise_ID FROM expertise table;</p> <p>SEND variable Expertise_ID TO LecturerExpertiseController</p> <p> FOR editExpertise() class;</p> <p> System query from database;</p> <p>DISPLAY LecturerExpertise page.</p> <p>END</p>
--	--

4.1.2 ViewExpertise [SDD-REQ-102]

Class Type	Boundary Class	
Responsibility	An interface to display lecturer's expertise.	
Attributes	Attributes Name	Attributes Type
	Lecturer_ID	String
	Expertise_ID	String
	Expertise_Name	String
	Expertise_Level	String
Methods	Method Name	Description
	viewLectExpertise()	Method to display lecturer expertise for the student.
Algorithm	<pre> viewLectExpertise() BEGIN GET Expertise_ID FROM expertise table; System query from database; DISPLAY Expertise_Name, Expertise_Name ACCORDING Expertise_ID. END </pre>	

4.1.3 LecturerExpertiseController [SDD-REQ-103]

Class Type	Controller Class	
Responsibility	Control the flow of input between ManageLecturerExpertiseInterface and ManageLecturerExpertiseModel.	
Attributes	Attributes Name	Attributes Type
	Lecturer_ID	String
	Expertise_ID	String
	Expertise_Name	String
	Expertise_Level	String
Methods	Method Name	Description
	viewExpertise()	Method to view lecturer expertise for the lecturer.
	addExpertise()	Method to add new expertise into database.
	editExpertise()	Method to edit lecturer expertise details.
	viewLectExpertise()	Method to view lecturer expertise for the student.
Algorithm	<pre> viewExpertise() BEGIN IF viewExpertise() is called; THEN create view lecturer expertise details to connect to LecturerExpertiseModel; RETURN to LecturerExpertise page. END IF END addExpertise() BEGIN IF addExpertise() is called; THEN create object add expertise form to connect to LecturerExpertiseModel; RETURN to addExpertise form LecturerExpertise page. END IF END editExpertise() </pre>	

BEGIN

IF editExpertise() is called;

THEN create object edit expertise form to connect to
LecturerExpertiseModel;

RETURN to editExpertise form LecturerExpertise page.

END IF

END

viewLectExpertise()

BEGIN

IF viewLectExpertise() is called;

THEN create view lecturer expertise details to connect to
LecturerExpertiseModel;

RETURN to ViewExpertise page.

END IF

END

4.1.4 LecturerExpertiseModel [SDD-REQ-104]

Class Type	Entity Class	
Responsibility	To save data to the database and retrieve request data from the database.	
Attributes	Attributes Name	Attributes Type
	Lecturer_ID	String
	Expertise_ID	String
	Expertise_Name	String
	Expertise_Level	String
Methods	Method Name	Description
	viewExpertise()	Method to retrieve and display lecturer expertise for the lecturer.
	addExpertise()	Method to save and display new expertise into database.
	editExpertise()	Method to save the update and display edited lecturer expertise details.
	viewLectExpertise()	Method to retrieve and display lecturer expertise for the student.
Algorithm	<pre> viewExpertise() BEGIN RETRIEVE Expertise_Name, Expertise_Level FROM Expertise table WHERE Expertise_ID = Expertise_ID AND Lecturer_ID = Lecturer_ID; SEND Expertise_Name, Expertise_Level TO LecturerExpertiseController FOR viewExpertise() class. END addExpertise() BEGIN GET Expertise_Name, Expertise_Level FROM addExpertise() class; SEND Expertise_Name, Expertise_Level TO expertise table. END </pre>	

editExpertise()

BEGIN

UPDATE Expertise_Name, Expertise_Level **FROM** Expertise table

WHERE Expertise_ID = Expertise_ID **AND** Lecturer_ID =
Lecturer_ID;

SEND Expertise_Name, Expertise_Level **TO**

LecturerExpertiseController **FOR** editExpertise() class.

END

viewLectExpertise()

BEGIN

RETRIEVE Expertise_Name, Expertise_Level **FROM** Expertise table

WHERE Expertise_ID = Expertise_ID **AND** Lecturer_ID =
Lecturer_ID;

SEND Expertise_Name, Expertise_Level **TO**

LecturerExpertiseController **FOR** viewLectExpertise() class.

END

4.2 Manage Profile [SDD-REQ-200]

Nur Alia Shahira binti Mohd Zaidi [CB19141]

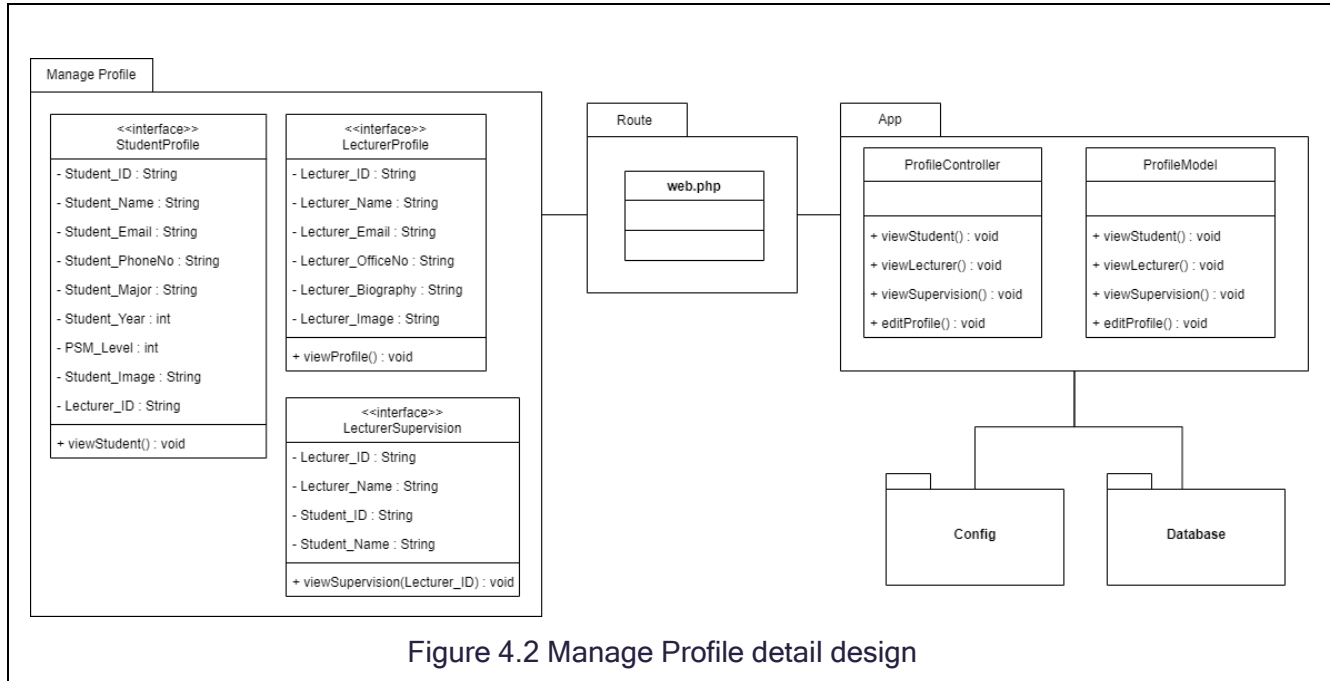


Figure 4.2 Manage Profile detail design

4.2.1 StudentProfile [SDD-REQ-201]

Class Type	Boundary Class	
Responsibility	An interface to display the student's profile.	
Attributes	Attributes Name	Attributes Type
	Student_ID	String
	Student_Name	String
	Student_Email	String
	Student_PhoneNo	String
	Student_Major	String
	Student_Year	Int
	PSM_Level	Int
Methods	Method Name	Description
	viewStudent()	To display the student profile
Algorithm	<pre> viewStudent() BEGIN GET Student_ID DISPLAY Student_Name, Student_Email, Student_PhoneNo, Student_Major, Student_Year, PSM_Level ACCORDING Student_ID END </pre>	

4.2.2 LecturerProfile [SDD-REQ-202]

Class Type	Boundary Class	
Responsibility	An interface to display the lecturer's profile.	
Attributes	Attributes Name	Attributes Type
	Lecturer_ID	String
	Lecturer_Name	String
	Lecturer_Email	String
	Lecturer_OfficeNo	String
	Lecturer_Biography	String
	Institution_Name	String
	Level_Education	String
	Major_Education	String
	Year_Education	Int
Methods	Method Name	Description
	viewLecturer()	To display the lecturer profile
Algorithm	viewLecturer() BEGIN GET Lecturer_ID DISPLAY Lecturer_Name, Lecturer_Email, Lecturer_OfficeNo, Lecturer_Biography, Institution_Name, Level_Education, Major_Education, Year_Education ACCORDING Lecturer_ID END	

4.2.3 LecturerSupervision [SDD-REQ-203]

Class Type	Boundary Class	
Responsibility	An interface to display the student list under lecturer's supervision.	
Attributes	Attributes Name	Attributes Type
	Student_ID Student_Name	String String
Methods	Method Name	Description
	viewSupervision(LecturerID)	To display the lecturer supervision list
Algorithm	viewSupervision(LecturerID) BEGIN GET Student_ID, Student_Name ACCORDING Lecturer_ID END	

4.2.4 ProfileController [SDD-REQ-204]

Class Type	Controller Class	
Responsibility	Control the flow of input and output between ManageProfileInterface and ManageProfileModel	
Attributes	Attributes Name	Attributes Type
	-	-
Methods	Method Name	Description
	viewStudent()	To display the student profile details
	viewLecturer()	To display the lecturer profile details
	viewSupervision()	To display the lecturer's supervision list
	editProfile()	To edit and update profile details
Algorithm	viewStudent() BEGIN IF viewStudent() is called THEN select Student_ID, Student_Name, Student_Email, Student_PhoneNo, Student_Major, Student_Year, PSM_Level, Student_Image, Lecturer_ID from table Student RETURN data to StudentProfile page ENDIF END	
	viewLecturer() BEGIN IF viewLecturer() is called THEN select Lecturer_ID, Lecturer_Name, Lecturer_Email, Lecturer_OfficeNo, Lecturer_Biography, Lecturer_Image from table lecturer in psmms database RETURN data to LecturerProfile page ENDIF END	
	viewSupervision(LecturerID) BEGIN	

```
IF viewSupervision(LecturerID) is called
    THEN select all Student_ID, Student_Name WHERE
        Lecturer_ID equal to LecturerID from table Student in
        psmms database.
    RETURN array data to LecturerSupervision page
ENDIF
END
```

```
editProfile()
```

```
BEGIN
    IF editProfile() is called
        THEN Update and store Lecturer_Name, Lecturer_Email,
            Lecturer_OfficeNo, Lecturer_Biography, Lecturer_Image
            for Lecturer_ID in table Lecturer in psmms database
    ENDIF
END
```

4.2.5 ProfileModel [SDD-REQ-205]

Class Type	Entity Class	
Responsibility	Represent data of lecturer and student	
Attributes	Attributes Name	Attributes Type
	-	-
Methods	Method Name	Description
	viewStudent() viewLecturer(LecturerID) viewSupervision() editProfile()	To retrieve the student profile details To retrieve the lecturer profile details To retrieve the lecturer's supervision list To update profile details
Algorithm	<pre> viewStudent() BEGIN RETRIEVE Student_Name, Student_Email, Student_PhoneNo, Student_Major, Student_Year, PSM_Level FROM table Student WHERE Student_ID = Student_ID RETURN array Student_Name, Student_Email, Student_PhoneNo, Student_major, Student_Year, PSM_Level END viewLecturer() BEGIN RETRIEVE Lecturer_Name, Lecturer_Email, Lecturer_OfficeNo, Lecturer_Biography FROM table Lecturer WHERE Lecturer_ID = Lecturer_ID RETURN array Lecturer_Name, Lecturer_Email, Lecturer_OfficeNo, Lecturer_Biography END viewSupervision(Lecturer_ID) BEGIN RETRIEVE all Student_ID, Student_Name FROM table Student WHERE Lecturer_ID = Lecturer_ID RETURN array Student_ID, Student_Name END </pre>	

editProfile()

BEGIN

UPDATE Lecturer_Name, Lecturer_Email, Lecturer_OfficeNo,
Lecturer_Biography **FROM** table Lecturer

WHERE Lecturer_ID = Lecturer_ID

UPDATE Institution_Name, Level_Education,
Major_Education,

Year_Education **FROM** table Lecturer_Education

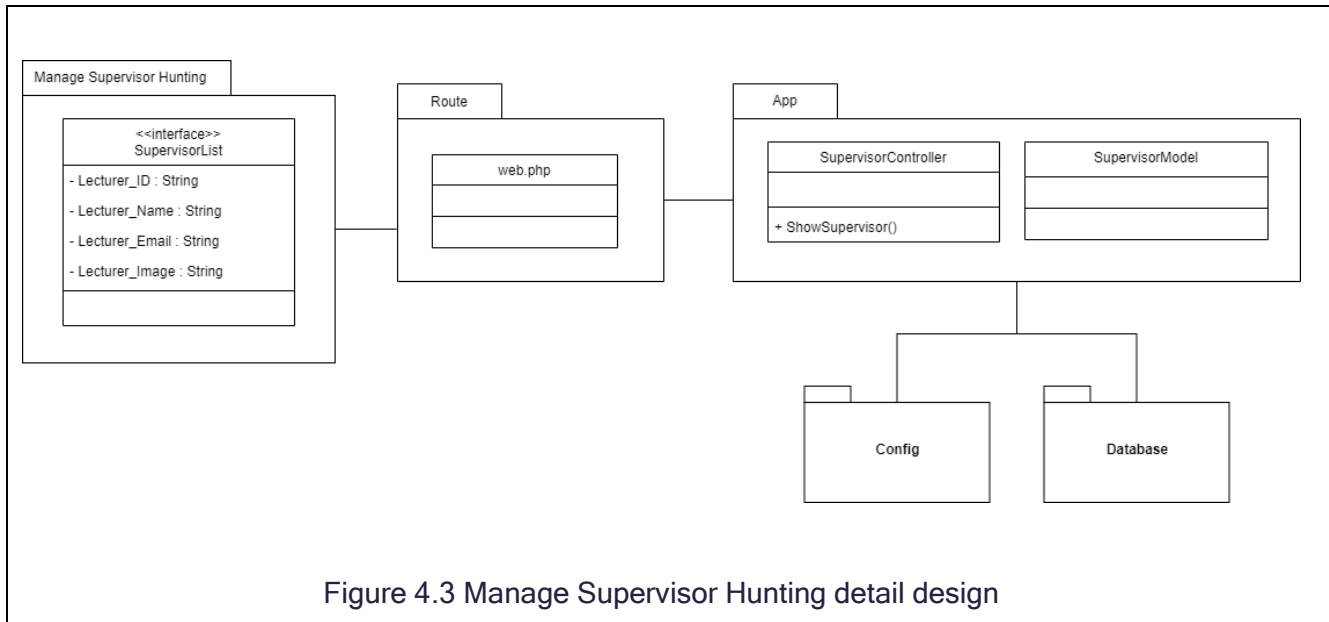
WHERE Lecturer_ID = Lecturer_ID **AND**

Education_ID = Education_ID

END

4.3 Manage Supervisor Hunting [SDD-REQ-300]

Muhammad Afif bin Ab Rahman [CB19007]



4.3.1 SupervisorList [SDD-REQ-301]

Class Type	Boundary Class	
Responsibility	An interface to display the list of supervisors for the student	
Attributes	Attributes Name	Attributes Type
	Lecturer_ID	String
	Lecturer_Name	String
	Lecturer_Email	String
	Lecturer_Image	String
Methods	Method Name	Description
	Not applicable	Not applicable
Algorithm	Not applicable	

4.3.2 SupervisorController [SDD-REQ-302]

Class Type	Controller Class	
Responsibility	Control the flow of input and output between ManageSupervisorHuntingInterface and ManageSupervisorHuntingModel	
Attributes	Attributes Name	Attributes Type
	-	-
Methods	Method Name	Description
	ShowSupervisor()	Method to display all supervisor in the SupervisorList.blade.php
Algorithm	ShowSupervisor() BEGIN GET Lecturer_ID FROM table Lecturer DISPLAY Lecturer_Name, Lecturer_Email, Lecturer_Image ACCORDING Lecturer_ID END	

4.3.3 SupervisorModel [SDD-REQ-303]

Class Type	Entity Class	
Responsibility	Represent data of supervisor	
Attributes	Attributes Name	Attributes Type
	-	-
Methods	Method Name	Description
	Not applicable	Not applicable
Algorithm	Not applicable	

4.4 Manage Title [SDD-REQ-400]

Nabilah Ayuni binti Mohd Nasir [CB19125]

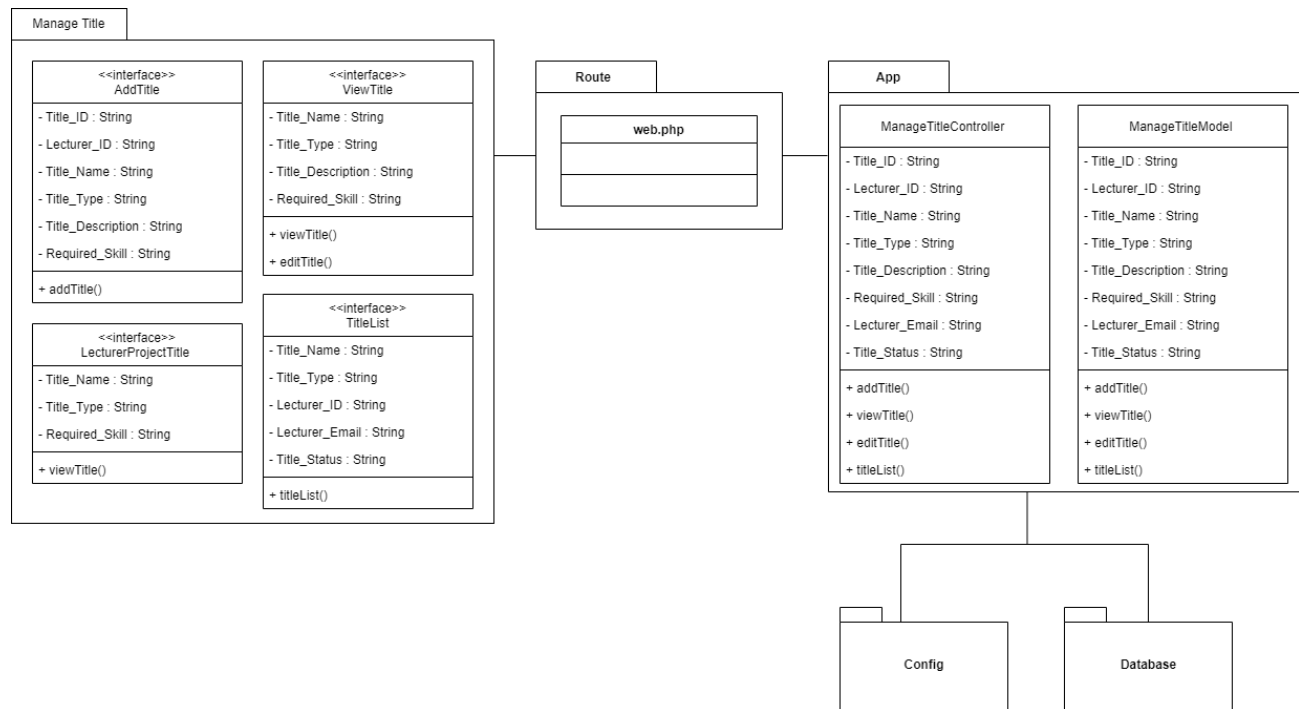


Figure 4.4 Manage Title detail design

4.4.1 AddTitle [SDD-REQ-401]

Class Type	Boundary Class	
Responsibility	An interface for the lecturer to add their new title.	
Attributes	Attributes Name	Attributes Type
	Title_ID	String
	Lecturer_ID	String
	Title_Name	String
	Title_Type	String
	Title_Description	String
	Required_Skill	String
Methods	Method Name	Description
	addTitle()	To save the title information entered by the lecturer in the title database.
Algorithm	addTitle() BEGIN INSERT Title_Name, Title_Type, Title_Description, Required_Skill INTO title table; System query from database. DISPLAY ViewTitle page. END	

4.4.2 ViewTitle [SDD-REQ-402]

Class Type	Boundary Class	
Responsibility	An interface for the lecturer to view title information and edit their title information.	
Attributes	Attributes Name	Attributes Type
	Title_Name	String
	Title_Type	String
	Title_Description	String
	Required_Skill	String
Methods	Method Name	Description
	viewTitle()	To retrieve the lecturers' title information and display it to them.
	editTitle()	To edit and update the lecturer's title information.
Algorithm	<pre> viewTitle() BEGIN GET Title_ID field FROM title table DISPLAY Title_Name, Title_Type, Title_Description, Required_Skill ACCORDING Title_ID END editTitle() BEGIN READ Title_ID FROM title table SEND title information details TO ManageTitleModel FOR editTitle() END </pre>	

4.4.3 LecturerProjectTitle [SDD-REQ-403]

Class Type	Boundary Class	
Responsibility	An interface for the lecturer to view title list and perform an action to edit their existing title and delete their title.	
Attributes	Attributes Name	Attributes Type
	Title_Name	String
	Title_Type	String
	Required_Skill	String
Methods	Method Name	Description
	viewTitle()	To retrieve the lecturers' title information and display it to them.
	editTitle()	To edit and update the lecturer's title information.
Algorithm	<pre> viewTitle() BEGIN GET Title_ID field FROM title table DISPLAY Title_Name, Title_Type, Required_Skill ACCORDING Title_ID END editTitle() BEGIN READ Title_ID FROM title table SEND title information details TO ManageTitleController FOR editTitle() END </pre>	

4.4.4 TitleList [SDD-REQ-404]

Class Type	Boundary Class	
Responsibility	An interface for the student to view the project title list, project title status and book their preferred title.	
Attributes	Attributes Name	Attributes Type
	Title_Name	String
	Title_Type	String
	Lecturer_ID	String
	Lecturer_Email	String
	Title_Status	String
Methods	Method Name	Description
	titleList()	To retrieve the lecturers' title information list and display it to them.
Algorithm	<pre> titleList() BEGIN GET Title_ID field FROM title table DISPLAY Title_Name, Title_Type, Lecturer_ID, Lecturer_Email, Title_Status ACCORDING Title_ID END </pre>	

4.4.5 ManageTitleController [SDD-REQ-405]

Class Type	Controller Class	
Responsibility	Control the flow of input and output between ManageTitleView and ManageTitleModel	
Attributes	Attributes Name	Attributes Type
	Title_ID	String
	Lecturer_ID	String
	Title_Name	String
	Title_Type	String
	Title_Description	String
	Required_Skill	String
	Lecturer_Email	String
	Title_Status	String
Methods	Method Name	Description
	addTitle()	To save the title information entered by the lecturer in the title database.
	viewTitle()	To retrieve the lecturers' title information and display it to them.
	editTitle()	To edit and update the lecturer's title information.
	titleList()	To retrieve the lecturers' title information list and display it to them.
Algorithm	addTitle() BEGIN IF addTitle() is called THEN create object lecturer add title form to connect to ManageTitleModel RETURN to addTitle form AddTitle page END IF END	

```
viewTitle()
BEGIN
    IF viewTitle() is called
        THEN create view lecturer title details information to connect
            to ManageTitleModel
        RETURN to view title details ViewTitle page.
    END IF
END

editTitle()
BEGIN
    IF editTitle() is called
        THEN create object edit title information form to connect to
            ManageTitleModel
        RETURN to editTitle form ViewTitle page.
    END IF
END

titleList()
BEGIN
    IF titleList() is called
        THEN create view lecturer title list to connect to
            ManageTitleModel.
        RETURN to TitleList page
    END IF
END
```

4.4.6 ManageTitleModel [SDD-REQ-406]

Class Type	Entity Class	
Responsibility	To save data to the database and retrieve request data from the database.	
Attributes	Attributes Name	Attributes Type
	Title_ID	String
	Lecturer_ID	String
	Title_Name	String
	Title_Type	String
	Title_Description	String
	Required_Skill	String
	Lecturer_Email	String
	Title_Status	String
Methods	Method Name	Description
	addTitle()	To save the title information entered by the lecturer in the title database.
	viewTitle()	To retrieve the lecturers' title information and display it to them.
	editTitle()	To edit and update the lecturer's title information.
	titleList()	To retrieve the lecturers' title information list and display it to them.
Algorithm	addTitle() BEGIN GET Title_Name, Title_Type, Title_Description, Required_Skill FROM addTitle() Class SEND Title_Name, Title_Type, Title_Description, Required_Skill TO title table END	

```
viewTitle()
BEGIN
    RETRIEVE Title_Name, Title_Type, Title_Description, Required_Skill
    FROM title table
        WHERE Title_ID = Title_ID AND Lecturer_ID = Lecturer_ID;
    SEND Title_Name, Title_Type, Title_Description, Required_Skill
        TO ManageTitleController FOR viewTitle() class.
END
```

```
editTitle()
BEGIN
    UPDATE Title_Name, Title_Type, Title_Description, Required_Skill
    FROM title table
        WHERE Title_ID = Title_ID AND Lecturer_ID = Lecturer_ID;
    SEND Title_Name, Title_Type, Title_Description, Required_Skill
        TO ManageTitleController FOR editTitle() class.
END
```

```
titleList()
BEGIN
    RETRIEVE Title_Name, Title_Type, Lecturer_ID, Lecturer_Email
    FROM title table
        WHERE Title_ID = Title_ID AND Lecturer_ID = Lecturer_ID;
    SEND Title_Name, Title_Type, Lecturer_ID, Lecturer_Email
        TO ManageTitleController FOR titleList() class.
END
```

4.5 Manage Approval and Proposal Submission [SDD-REQ-500]

Muhammad Ilman Nashran bin Mohd Nasruddin [CB19014]

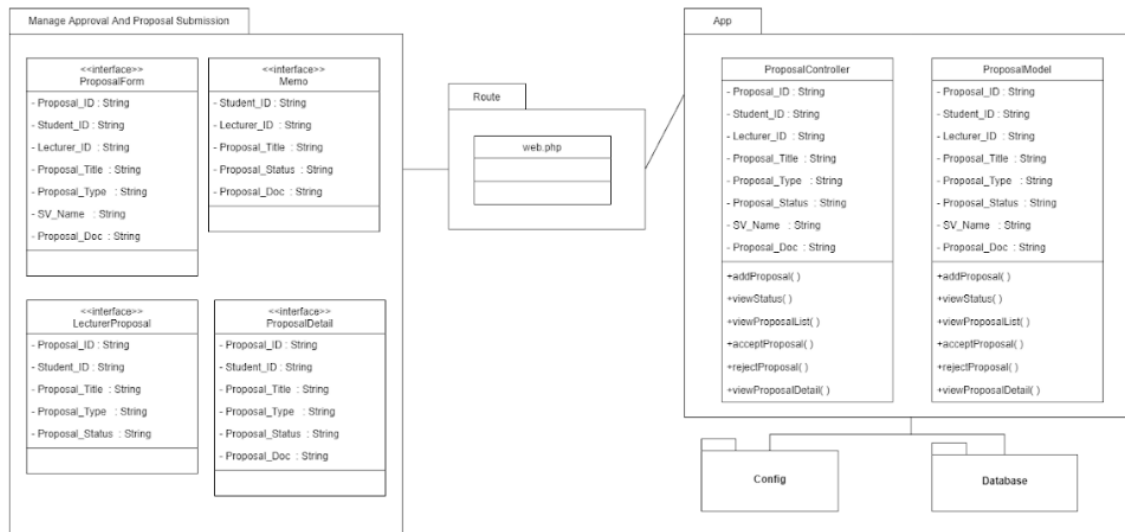


Figure 4.5 Manage Approval and Proposal Submission detail design

4.5.1 ProposalForm [SDD-REQ-501]

Class Type	Boundary Class	
Responsibility	An interface for the students to add their new proposal.	
Attributes	Attributes Name	Attributes Type
	Proposal_ID	String
	Student_ID	String
	Lecturer_ID	String
	Proposal_Title	String
	Proposal_Type	String
	Proposal_Doc	String
	SV_Name	String
Methods	Method Name	Description
	Not applicable	-
Algorithm	Not applicable	

4.5.2 Memo [SDD-REQ-502]

Class Type	Boundary Class	
Responsibility	An interface for the students to view their proposal status.	
Attributes	Attributes Name	Attributes Type
	Proposal_ID	String
	Student_ID	String
	Proposal_Title	String
	Proposal_Status	String
	Proposal_Doc	String
Methods	Method Name	Description
	Not applicable	-
Algorithm	Not applicable	

4.5.3 LecturerProposal [SDD-REQ-503]

Class Type	Boundary Class	
Responsibility	An interface for the lecturer to view the list of submitted proposals from their student.	
Attributes	Attributes Name	Attributes Type
	Proposal_ID	String
	Student_ID	String
	Proposal_Title	String
	Proposal_Type	String
	Proposal_Status	String
Methods	Method Name	Description
	Not applicable	-
Algorithm	Not applicable	

4.5.4 ProposalDetail [SDD-REQ-504]

Class Type	Boundary Class	
Responsibility	An interface for the lecturer to view the details of submitted proposals.	
Attributes	Attributes Name	Attributes Type
	Proposal_ID	String
	Student_ID	String
	Proposal_Title	String
	Proposal_Type	String
	Proposal_Status	String
	Proposal_Doc	String
Methods	Method Name	Description
	Not applicable	-
Algorithm	Not applicable	

4.5.5 ProposalController [SDD-REQ-505]

Class Type	Controller Class	
Responsibility	This class controls the flow of data between Manage Approval and Proposal Submission Interface and Proposal Model	
Attributes	Attributes Name	Attributes Type
	Proposal_ID	String
	Student_ID	String
	Lecturer_ID	String
	Proposal_Title	String
	Proposal_Type	String
	Proposal_Status	String
	Proposal_Doc	String
	SV_Name	String
Methods	Method Name	Description
	addProposal()	Method for student to add and save the proposal submission details into database
	viewStatus()	Method for student to view the proposal submission status.
	viewProposalList()	Method for lecturer to view the list proposal submission from students.
	acceptProposal()	Method to allow lecturer to accept the proposal.
	rejectProposal()	Method to allow lecturer to reject the proposal.
	viewProposalDetail()	Method for lecturer to view the details of the proposal submission from students.
Algorithm	addProposal() BEGIN IF addProposal() is called THEN create object Student add proposal form to connect to ProposalModel RETURN to ProposalForm page END IF END	

```
viewStatus( )  
BEGIN  
    IF viewStatus() is called  
        THEN create a view student proposal status to connect  
             to ProposalModel  
        RETURN to view proposal Memo page  
    END IF  
END
```

```
viewProposalList( )  
BEGIN  
    IF viewProposalList() is called  
        THEN create a view Lecturer list proposal to connect  
             to ProposalModel  
        RETURN to view proposal list LecturerProposal page.  
    END IF  
END
```

```
acceptProposal( )  
BEGIN  
    IF acceptProposal() is called  
        THEN create an object to connect to ProposalModel  
        RETURN to view proposal list LecturerProposal page.  
    END IF  
END
```

```
rejectProposal( )  
BEGIN  
    IF rejectProposal() is called  
        THEN create an object to connect to ProposalModel  
        RETURN to view proposal list LecturerProposal page.  
    END IF  
END
```

```
viewProposalDetail( )
```

```
BEGIN
```

```
  IF viewProposalDetail() is called
```

```
    THEN create a lecturer view proposal detail to connect
```

```
      to ProposalModel
```

```
    RETURN to view ProposalDetail page
```

```
  END IF
```

```
END
```

4.5.6 ProposalModel [SDD-REQ-506]

Class Type	Entity Class	
Responsibility	To save data to the database and retrieve request data from the database.	
Attributes	Attributes Name	Attributes Type
	Proposal_ID	String
	Student_ID	String
	Lecturer_ID	String
	Proposal_Title	String
	Proposal_Type	String
	Proposal_Status	String
	Proposal_Doc	String
	SV_Name	String
Methods	Method Name	Description
	addProposal()	Method to add and save the proposal submission details into database.
	viewStatus()	Method to view the proposal submission status
	viewProposalList()	Method to view the list proposal submission from students.
	acceptProposal()	Method to allow lecturer to accept the proposal.
	rejectProposal()	Method to allow lecturer to reject the proposal.
	viewProposalDetail()	Method to view the details of the proposal submission from students.
Algorithm	addProposal() BEGIN GET Proposal_Title, Proposal_Type, Proposal_Doc, SV_Name, Lecturer_ID FROM addProposal() Class SEND Proposal_Title, Proposal_Type, Proposal_Doc, SV_Name, Lecturer_ID TO ProposalController Class FOR addProposal() class. END	

```
viewStatus( )
BEGIN
    RETRIEVE Proposal_Title, Proposal_Type, Lecturer_Name,
    Proposal_Status
    FROM proposal table
    SEND Proposal_Title, Proposal_Type, Lecturer_Name,
    Proposal_Status
    TO ProposalController FOR viewStatus ( ) class.
END

viewProposalList( )
BEGIN
    RETRIEVE Proposal_Title, Proposal_Type, Student_Name,
    Proposal_Status
    FROM proposal table
    SEND Proposal_Title, Proposal_Type, Student_Name,
    Proposal_Status
    TO ProposalController FOR viewProposalList( ) class.
END

acceptProposal( )
BEGIN
    UPDATE Proposal SET Proposal_Status =Proposal_Status
    WHERE Student_ID=Student_ID
END

rejectProposal( )
BEGIN
    UPDATE Proposal SET Proposal_Status =Proposal_Status
    WHERE Student_ID=Student_ID
END
```

```
viewProposalDetail( )
```

```
BEGIN
```

```
RETRIEVE Student_ID, Student_Name, PSM_Level, Proposal_Title,  
Proposal_Type,
```

```
FROM proposal table
```

```
WHERE Proposal_ID = Proposal_ID AND Student_ID = Student_ID;
```

```
SEND Student_ID, Student_Name, PSM_Level, Proposal_Title,  
Proposal_Type,
```

```
TO ProposalController FOR viewProposalDetail() class.
```

```
END
```


4.6 Manage Logbook [SDD-REQ-600]

Nureen Nasuha binti Mohd Rosli [CB19124]

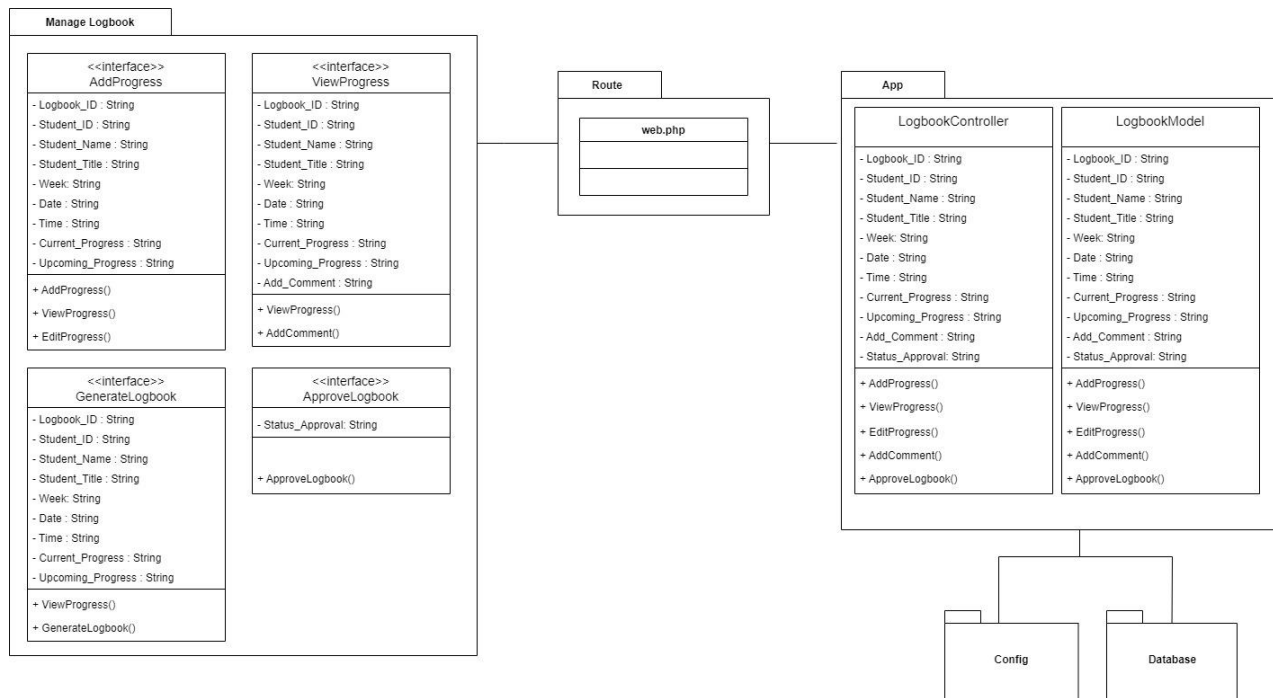


Figure 4.6 Manage Logbook detail design

4.6.1 AddProgress [SDD-REQ-601]

Class Type	Boundary Class	
Responsibility	An interface for the student to add, view and edit their logbook progress	
Attributes	Attributes Name	Attributes Type
	Logbook_ID	String
	Student_ID	String
	Student_Name	String
	Student_Title	String
	Week	String
	Date	String
	Time	String
	Current_Progress	String
	Upcoming_Progress	String
Methods	Method Name	Description
	AddProgress()	Method to add and save the logbook progress details into database
	ViewProgress()	Method to view the logbook progress details
	EditProgress()	Method to edit and save the logbook progress details into database
Algorithm	<pre> AddProgress() BEGIN INSERT Student_Name, Student_Title, Week, Date, Time, Current_Progress, Upcoming_Progress INTO logbook table; System query from database; DISPLAY ViewProgress page END </pre>	

ViewProgress()

BEGIN

GET Logbook_ID **FROM** logbook table;

System query from database;

DISPLAY Student_Name, Student_Title, Week, Date, Time,
Current_Progress, Upcoming_Progress **ACCORDING**
Logbook_ID.

END

EditProgress()

BEGIN

GET Logbook_ID **FROM** logbook table;

SEND Student_Name, Student_Title, Week, Date, Time,

Current_Progress, Upcoming_Progress **TO** LogbookController

FOR EditProgress() class.

System query from database

END

4.6.2 GenerateLogbook [SDD-REQ-602]

Class Type	Boundary Class	
Responsibility	An interface for the student to view and generate the logbook	
Attributes	Attributes Name	Attributes Type
	Logbook_ID	String
	Student_ID	String
	Student_Name	String
	Student_Title	String
	Week	String
	Date	String
	Time	String
	Current_Progress	String
	Upcoming_Progress	String
Methods	Method Name	Description
	ViewProgress()	Method to view the logbook progress details
	GenerateLogbook()	Method to generate the logbook
Algorithm	<pre> ViewProgress() BEGIN GET Logbook_ID FROM logbook table; System query from database; DISPLAY Student_Name, Student_Title, Week, Date, Time, Current_Progress, Upcoming_Progress ACCORDING Logbook_ID. END GenerateLogbook() BEGIN System query all the logbook progress details from database IF "Print" System generate and print the logbook ELSE "Cancel" System display ViewProgress END </pre>	

4.6.3 ViewProgress [SDD-REQ-603]

Class Type	Boundary Class	
Responsibility	An interface for the lecturer to view and add comment into the student logbook progress	
Attributes	Attributes Name	Attributes Type
	Logbook_ID	String
	Student_ID	String
	Student_Name	String
	Student_Title	String
	Week	String
	Date	String
	Time	String
	Current_Progress	String
	Upcoming_Progress	String
	Add_Comment	String
Methods	Method Name	Description
	ViewProgress()	Method to view the logbook progress details
	AddComment()	Method to add comment into the logbook progress
Algorithm	<pre> ViewProgress() BEGIN GET Logbook_ID FROM logbook table; System query from database; DISPLAY Student_Name, Student_Title, Week, Date, Time, Current_Progress, Upcoming_Progress ACCORDING Logbook_ID. END AddComent() BEGIN INSERT Add_Comment INTO logbook table; System query from database DISPLAY ViewProgress page END </pre>	

4.6.4 ApproveLogbook [SDD-REQ-604]

Class Type	Boundary Class	
Responsibility	An interface for the lecturer to approve the student logbook progress	
Attributes	Attributes Name	Attributes Type
	Status_Approval	String
Methods	Method Name	Description
	ApproveLogbook()	Method to approve the student logbook progress
Algorithm	ApproveLogbook() BEGIN INSERT Status_Approval INTO logbook table; System query from database; DISPLAY ViewProgress page END	

4.6.5 LogbookController [SDD-REQ-605]

Class Type	Controller Class	
Responsibility	This class control the flow of data between ManageLogbookInterface and ManageLogbookModel	
Attributes	Attributes Name	Attributes Type
	Logbook_ID	String
	Student_ID	String
	Student_Name	String
	Student_Title	String
	Week	String
	Date	String
	Time	String
	Current_Progress	String
	Upcoming_Progress	String
	Add_Comment	String
	Status_Approval	String
Methods	Method Name	Description
	AddProgress()	Method to add and save the logbook progress details into database
	ViewProgress()	Method to view the logbook progress details
	EditProgress()	Method to edit and save the logbook progress details into database
	AddComment()	Method to add comment into the logbook progress
	ApproveLogbook()	Method to approve the logbook progress
Algorithm		

AddProgress()

BEGIN

IF AddProgress() is called

THEN create object student add progress to connect to
 ManageLogbookModel

RETURN to AddProgress page

END IF

END

ViewProgress()

BEGIN

IF ViewProgress() is called

THEN create view student progress details to connect to
 ManageLogbookModel

RETURN to ViewProgress page

END IF

END

EditProgress()

BEGIN

IF EditProgress() is called

THEN create object edit progress to connect to
 ManageLogbookModel

RETURN to EditProgress page

END IF

END

AddComment()

BEGIN

IF AddComment is called

THEN create object lecturer add comment to connect to
ManageLogbookModel

RETURN to ViewProgress page

END IF

END

ApproveLogbook()

BEGIN

IF ApproveLogbook is called

THEN create object lecturer approve logbook to connect to
ManageLogbookModel

RETURN to ViewProgress page

END IF

END

4.6.6 LogbookModel [SDD-REQ-606]

Class Type	Entity Class	
Responsibility	This model class is to store and retrieve the data from the database	
Attributes	Attributes Name	Attributes Type
	Logbook_ID	String
	Student_ID	String
	Student_Name	String
	Student_Title	String
	Week	String
	Dat	String
	Time	String
	Current_Progress	String
	Upcoming_Progress	String
	Add_Comment	String
	Status_Approval	String
Methods	Method Name	Description
	AddProgress()	Method to add and save the logbook progress details into database
	ViewProgress()	Method to view the logbook progress details
	EditProgress()	Method to edit and save the logbook progress details into database
	AddComment()	Method to add comment into the logbook progress
	ApproveLogbook()	Method to approve the logbook progress
Algorithm		

AddProgress()

BEGIN

GET Student_Name, Student_Title, Week, Date, Time,
 Current_Progress, Upcoming_Progress
 FROM AddProgress() Class
 SEND Student_Name, Student_Title, Week, Date, Time,
 Current_Progress, Upcoming_Progress
 TO logbook table

END

ViewProgress()

BEGIN

RETRIEVE Student_Name, Student_Title, Week, Date, Time,
 Current_Progress, Upcoming_Progress
 FROM logbook table
 WHERE Logbook_ID = Logbook_ID
 SEND Student_Name, Student_Title, Week, Date, Time,
 Current_Progress, Upcoming_Progress **TO** LogbookController **FOR**
 ViewProgress() class

END

EditProgress()

BEGIN

UPDATE Student_Name, Student_Title, Week, Date, Time,
 Current_Progress, Upcoming_Progress
 FROM logbook table
 WHERE Logbook_ID = Logbook_ID
 SEND Student_Name, Student_Title, Week, Date, Time,
 Current_Progress, Upcoming_Progress **TO** LogbookController **FOR**
 EditProgress() class

END

```
AddComment()
BEGIN
    GET Add_Comment
    FROM AddComment() Class
    SEND Add_Comment
    TO logbook table
END

ApproveLogbook()
BEGIN
    GET Status_Approval
    FROM ApproveLogbook() class
    SEND Status_Approval
    TO logbook table
END
```

5. REQUIREMENT TRACEABILITY

5.1 Manage Lecturer Expertise [SRS-REQ-100]

Siti Nurasma' binti Hashim [CB19123]

Requirement ID	Description	Design ID
SRS-REQ-101	The system shall be able to allow the lecturer to add new lecturer expertise.	SDD-REQ-101
SRS-REQ-102	The system shall be able to allow the lecturer to view their expertise details.	SDD-REQ-101
SRS-REQ-103	The system shall be able to allow the lecturer to edit the existing expertise details.	SDD-REQ-101
SRS-REQ-104	The system shall be able to allow the student to view the lecturer expertise details.	SDD-REQ-102

5.2 Manage Profile [SRS-REQ-200]

Nur Ali Shahira binti Mohd Zaidi [CB19141]

Requirement ID	Description	Design ID
SRS-REQ-201	The system shall be able to display the lecturer profile	SDD-REQ-202
SRS-REQ-202	The system shall be able to display the student profile	SDD-REQ-201
SRS-REQ-203	The lecturer should be able to edit and add information to his/her profile.	SDD-REQ-202
SRS-REQ-204	The system must be able to display the list of students under lecturer supervision.	SDD-REQ-203

5.3 Manage Supervisor Hunting [SRS-REQ-300]

Muhammad Afif bin Ab Rahman [CB19007]

Requirement ID	Description	Design ID
SRS-REQ-301	The system shall allow the student to view available for supervisor	SDD-REQ-301
SRS-REQ-302	The system shall allow the student to view supervisor detailed information	SDD-REQ-302
SRS-REQ-303	The system shall allow the student to filter supervisor	SDD-REQ-301
SRS-REQ-304	The system shall allow the student to book the supervisor	SDD-REQ-301

5.4 Manage Title from Student and Lecturer [SRS-REQ-400]

Nabilah Ayuni binti Mohd Nasir [CB19125]

Requirement ID	Description	Design ID
SRS-REQ-401	The system shall allow the lecturer to add their title	SDD-REQ- 401
SRS-REQ-402	The system shall allow the lecturer to view, edit and delete their title information	SDD-REQ- 403
SRS-REQ-403	The system shall allow the lecturer to edit lecturer's title details.	SDD-REQ- 402
SRS-REQ-404	The system shall allow the student to view the title list.	SDD-REQ- 404

5.5 Manage Approval and Proposal Submission [SRS-REQ-500]

Muhammad Ilman Nashran bin Mohd Nasruddin [CB19014]

Requirement ID	Description	Design ID
SRS-REQ-501	The system shall allow the student to add and submit proposal submission.	SDD-REQ-501
SRS-REQ-502	The system shall allow the lecturer to view submitted proposals from students.	SRS-REQ-502
SRS-REQ-503	The system shall allow the lecturer to reject or accept submitted proposals from students.	SRS-REQ-502

5.6 Manage Logbook [SRS-REQ-600]

Nureen Nasuha binti Mohd Rosli [CB19124]

Requirement ID	Description	Design ID
SRS-REQ-601	The system shall allow the student to add new progress, view and edit the progress	SDD-REQ-601
SRS-REQ-602	The system shall allow the student to generate the logbook	SDD-REQ-602
SRS-REQ-603	The system shall allow the lecturer to view and add comments into the student under supervision progress	SDD-REQ-603
SRS-REQ-604	The system shall allow the lecturer to approve the student under supervision logbook progress	SDD-REQ-604