Software Design Document

for

PETAKOM Mart Management System

Version 1.0 approved

Prepared by Tee Jia Mei (CB20068)

Tan Teng Xing (CB20121)

Loo Chang Herng (CB20114)

Tan Kha Hui (CB20064)

Immortal Tech Sdn Bhd

19 APRIL 2023

TABLE OF CONTENTS

1. INTRODUCTION

1.1 Purpose

Software Design Document (SDD) explains how a software product or feature will be built to meet a set of technical requirements. The purpose of doing this SDD is to track the necessary information required to effectively define architecture and system design in order to provide the development team guidance on the architecture of the PETAKOM Mart Management System to be developed. In this document, it provides an overview of the functions, data design, system architecture and detail design of each module. It can provide the client and developer a clear view on the development of PETAKOM Mart Management System.

1.2 System Identification

This Software Design Document (SDD) belongs to the "PETAKOM Mart Management System" (PMMS).

System title	: PETAKOM Mart Management System
System abbreviation	: PMMS
System identification number	: PMMS-2023-V1
System version number	: 1.0
System release number	: 1.0
Document identification number	: PMMS-SDD-2023-V1
Package identification number	: PMMS-SDD-REQ-X00

PMMS-2023-V1 is used to record the system identification number while the document identification number uses the format of PMMS-SDD-2023-V1. The PMMS is the abbreviation name of the system, while 2023 refers to the published year of the system and V1 stands for version 1 of the software documented in the document. The system version number is written as 1.0 whereas the number 1 refers to the major version while number 0 refers to the minor version. The system release number is the same as the system version number since the system release the system version of 1.0. The SDD refers to the Software Design Document.

For the package identification number, it follows the format of ‘PMMS-SDD-REQ-X00' in which ‘PMMS’ is the system abbreviation, ‘SDD’ stands for Software Design Document, and ‘REQ’ represents requirement. ‘X’ is the numerical variable referring to the sequence of packages. The number starts with 1 which is for the first package, then the number will be increased by 1 for every next package. For example, ‘PMMS-SDD-REQ-100' refers to the first package, ‘PMMS-SDD-REQ-200' refers to the second package and it continues for the following package.

For the design ID in a package, it follows the format of ‘PMMS-SDD-REQ-YXX’ in which Y will represent the package ID and the XX will represent the design ID in that package. The XX will start from 01, 02, and it continues for each design in that package. For example, for the first design in the first package, the design ID will be ‘PMMS-SDD-REQ-101’.

1.3 System Overview

This system was created for managing inventory of PETAKOM's mart in computing faculty. As students in computing faculty will buy accessories or foods frequently in PETAKOM's mart, it needs a management system for convenient management. This system will have the main function of inventory management, where the inventory in the mart is recorded in the system, and perform auto-increment or auto-decrement when the staff restock the items or sell the items. The system will also notify admins or person-in-charge when the stocks of some items are low.

Besides, the system shall allow admin to manage user registration for the PETAKOM Committee and Secretariat and the PETAKOM Coordinator. Furthermore, another requirement is the system should be able to allow admin to create and update staff duty roster. The PETAKOM Committee and Secretariat will be able to choose the available time slot to take the responsibility as cashier in PETAKOM mart. Other than that, payment made will be recorded into the system's database for record purposes when a payment is made by the customer. The system is also able to generate weekly, monthly, and yearly sales/inventory reports. Lastly, the system shall allow admin to create and update the announcement for the PETAKOM Committee and Secretariat and the PETAKOM Coordinator to view the announcement.

1.4 Module Overview

1. Inventory Management Module:
    - This module will allow the admin to manage inventory in the PETAKOM's mart. The inventory data will be recorded in the system and perform auto-increment or auto-decrement when the staff restock or sell the items. The system will also notify admins or person-in-charge when the stocks of some items are low.
2. User Management Module:
    - This module will allow the admin to manage user registration for the PETAKOM Committee and Secretariat and the PETAKOM Coordinator. The admin can create, update and delete user accounts as per the requirement.
3. Staff Duty Roster Module:
    - This module will allow the admin to create and update staff duty roster. The PETAKOM Committee and Secretariat will be able to choose the available time slot to take the responsibility as a cashier in the PETAKOM mart. The PETAKOM Coordinator is allowed to view the duty roster.
4. Payment Management Module:
    - This module will allow the staff to record payment data into the system's database for record purposes when a payment is made by the customer.
5. Sales/Inventory Reporting Module:
    - This module will allow the system to generate weekly, monthly, and yearly sales/inventory reports to help the admin to monitor the mart's performance.
6. Announcement Management Module:
    - This module will allow the admin to create and update the announcement for the PETAKOM Committee and Secretariat and the PETAKOM Coordinator to view the announcement.

2. REFERENCES

Bates, A. (2022, July 26). The essential guide to requirements management and traceability. Retrieved from https://www.jamasoftware.com/requirements-management-guide.

How to write a good software design Doc. (2018, June 8). Retrieved from https://www.freecodecamp.org/news/how-to-write-a-good-software-design-document-66fcf019569c/.

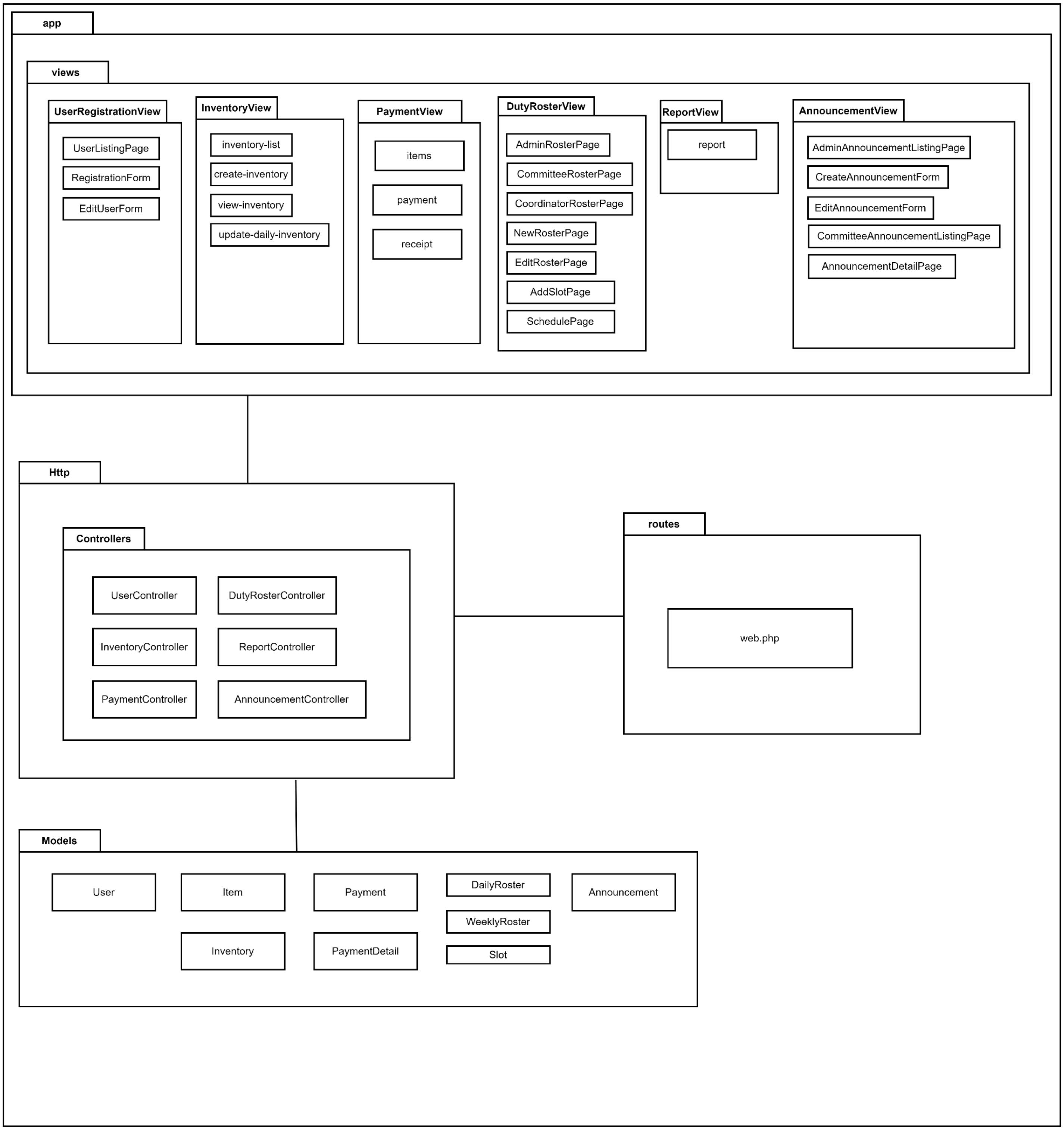
Inventory management system ER diagram. (n.d.). Retrieved from https://www.freeprojectz.com/entity-relationship/inventory-management-system-er-diagram.

Larman, C. (2005). *Applying UML and patterns: An introduction to object-oriented analysis and design and iterative development*. Pearson.

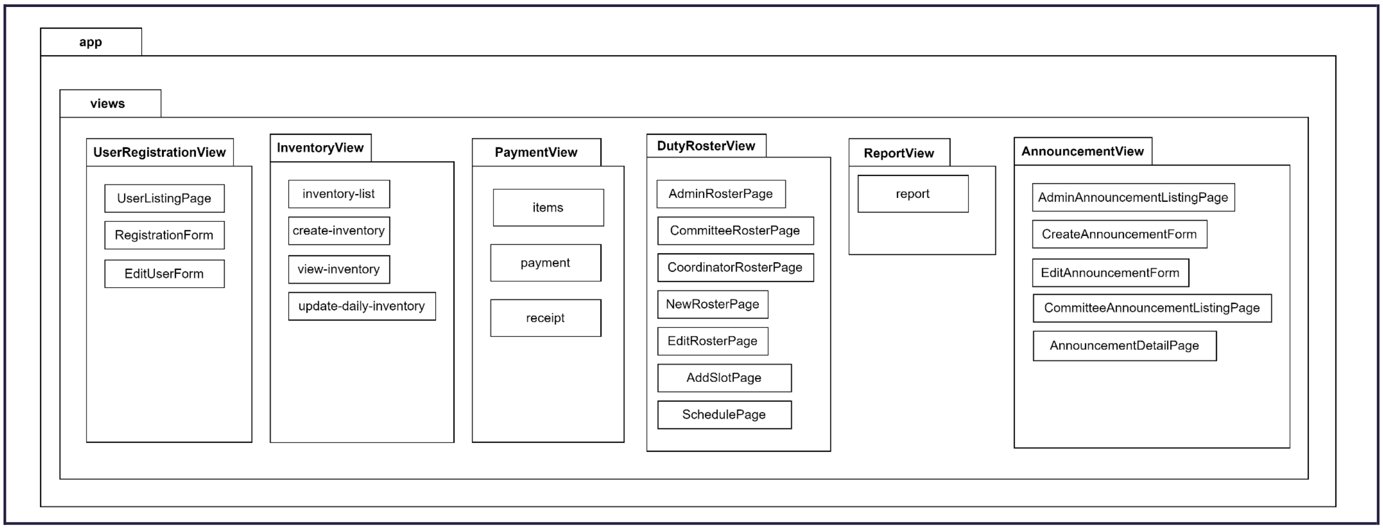
Santosa, I., Kurniawan, F., Rahmanita, E., & Rachmad, A. (2018). Application of model view controller architecture in hospital inventory logistic management. *Proceedings of the International Conference on Science and Technology (ICST 2018)*. doi:10.2991/icst-18.2018.72.

3. ARCHITECTURE DESCRIPTION

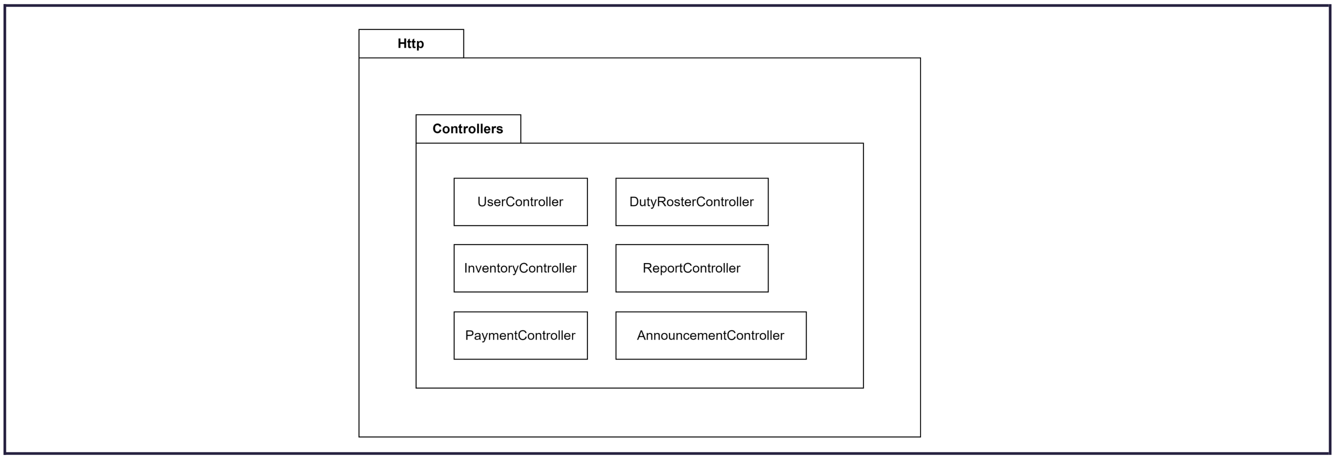
3.1 General Architecture



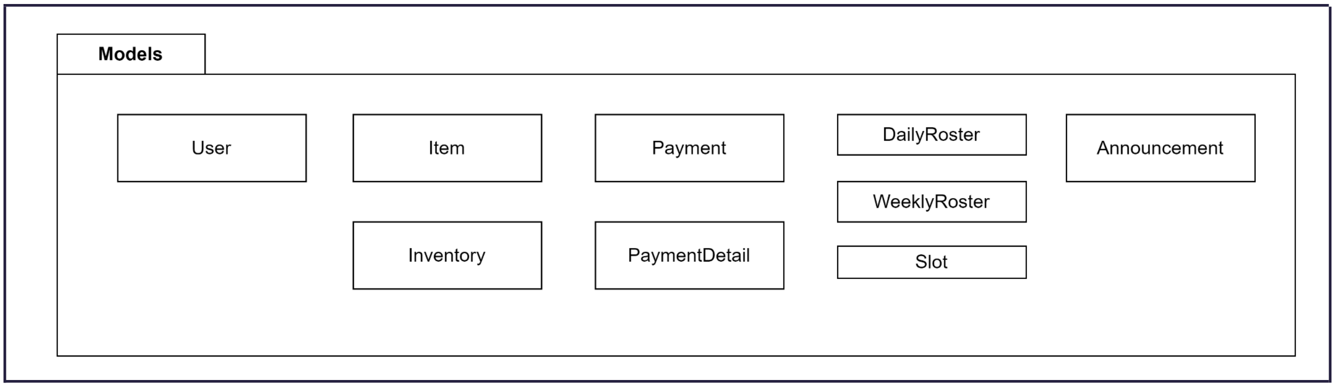
3.1.1 View Layer



3.1.2 Controller Layer



3.1.3 Model Layer



3.2 Package Module

3.2.1 Manage User Registration (CB20068 - TEE JIA MEI)

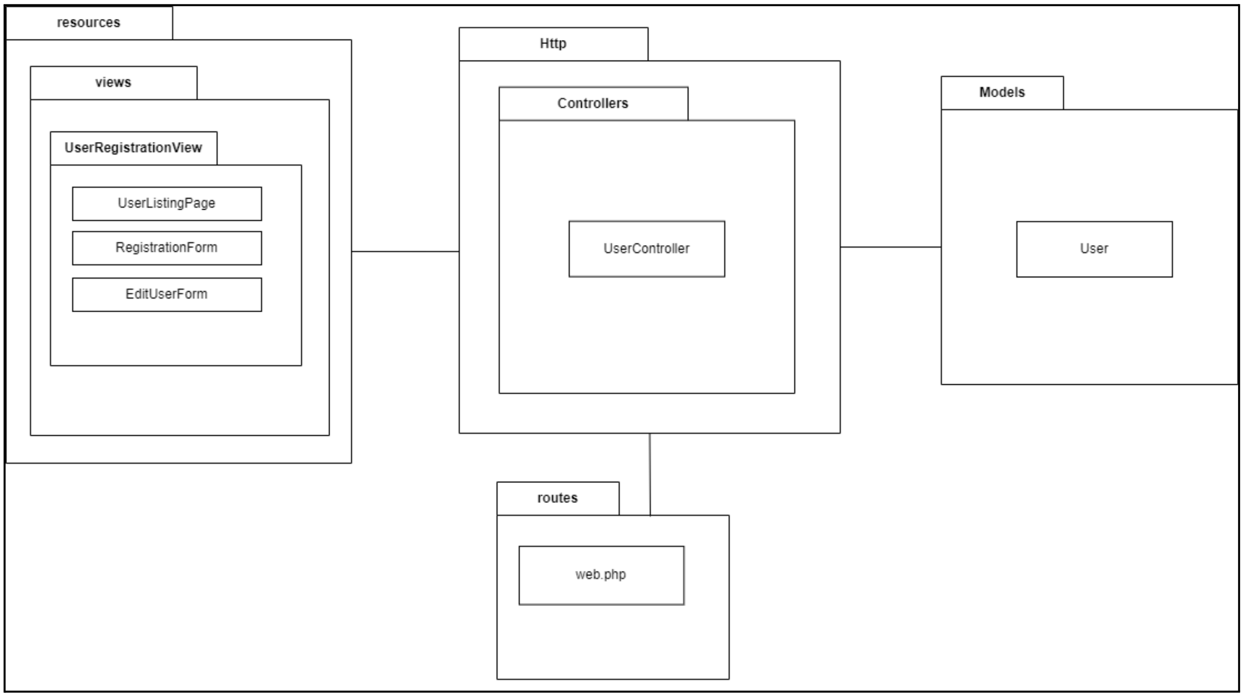


Figure 3.2.1 Package Diagram for Manage User Registration

3.2.1.1 UserRegistrationView [PMMS-SDD-REQ-100]

Class Name	Description
UserListingPage	Interface that displays all the registered user information.
RegistrationForm	Interface that allows admin to create a new user account for PETAKOM Committee and Secretariat or PETAKOM Coordinator.
EditUserForm	Interface that allows admin to update user account information for PETAKOM Committee and Secretariat or PETAKOM Coordinator.

3.2.1.2 Controllers [PMMS-SDD-REQ-200]

Class Name	Description
UserController	Controller class that is responsible to manage the flow of user registration module, including receiving input from RegistrationForm or EditUserForm user interface, validating user input, interacting with User Model to create or update user accounts, and returning appropriate responses to the user interface based on the success or failure of the registration process.

3.2.1.3 Models [PMMS-SDD-REQ-300]

Class Name	Description
User	Model class that is responsible to store and retrieve user data in the database.

3.2.2 Manage Inventory (CB20121 - TAN TENG XING)

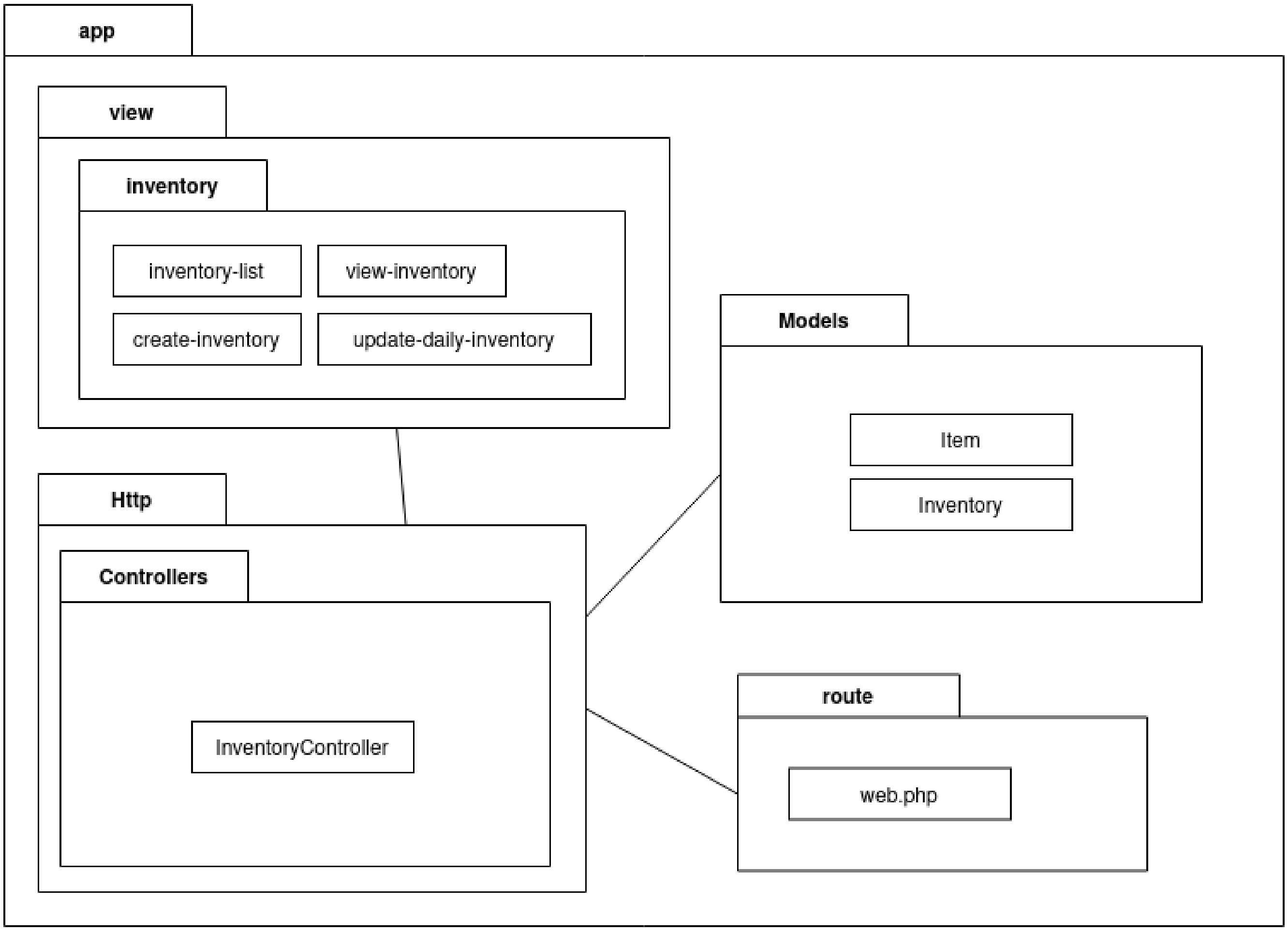


Figure 3.2.2 Package Diagram for Manage Inventory

3.2.2.1 InventoryView [PMMS-SDD-REQ-400]

Class Name	Description
inventory-list (index)	System displays inventory list in a table.
create-inventory	Allows users to create new inventory items.
view-inventory	System displays a particular inventory details in one page.
update-daily-inventory	Allows cashiers to update daily opening and closing inventory.

3.2.2.2 Controllers [PMMS-SDD-REQ-200]

Class Name	Description
InventoryController	Controller class that controls the flow of inventory module, including navigation between inventory boundaries, validates user input, and stores inventory details into database.

3.2.2.3 Models [PMMS-SDD-REQ-300]

Class Name	Description
Item	Model class that declares data type of items according to data dictionary. Declare functions that manage “Item” table, including createItem, viewItem, editItem, and deleteItem.
Inventory	Model class that declares data type of inventories according to data dictionary. Declare functions that manage “Inventory” table, including openingInventory and closingInventory.

3.2.3 Manage Payment (CB20114 - LOO CHANG HERNG)

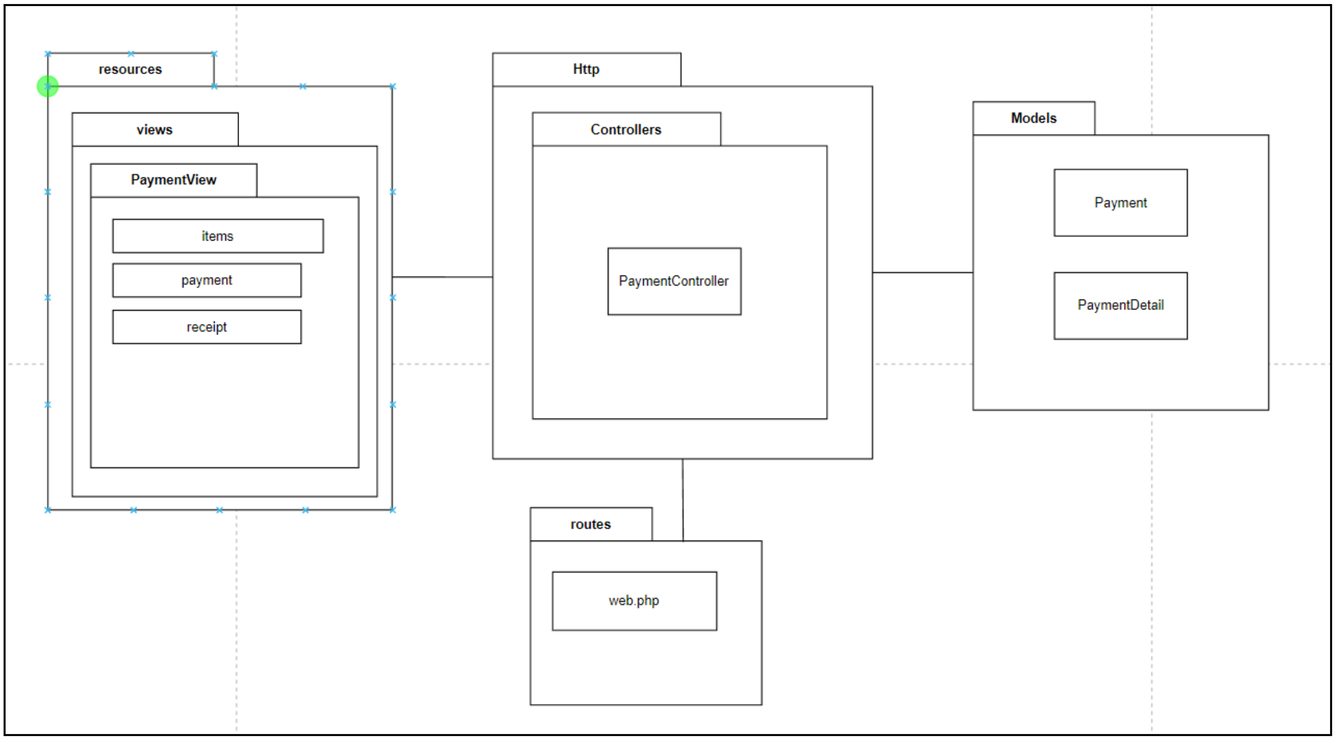


Figure 3.2.3 Package Diagram for Manage Payment

3.2.3.1 PaymentView [PMMS-SDD-REQ-500]

Class Name	Description
items	System displays every item available
payment	System displays every item selected and its quantity, allows cashier to update the quantity of items.
receipt	System displays a receipt that contains information of that payment including, payment date, item purchase, and its' quantity.

3.2.3.2 Controllers [PMMS-SDD-REQ-200]

Class Name	Description
PaymentController	Controller class that controls the flow of payment module, including navigation between payment interfaces, calculation of prices, and update of item's quantity

3.2.3.3 Models [PMMS-SDD-REQ-300]

Class Name	Description
Payment	Model class that declares data type of payment according to data dictionary.
PaymentDetail	Model class that declares data type of payment details according to data dictionary.

3.2.4 Manage Duty Roster (CB20064 - TAN KHA HUI)

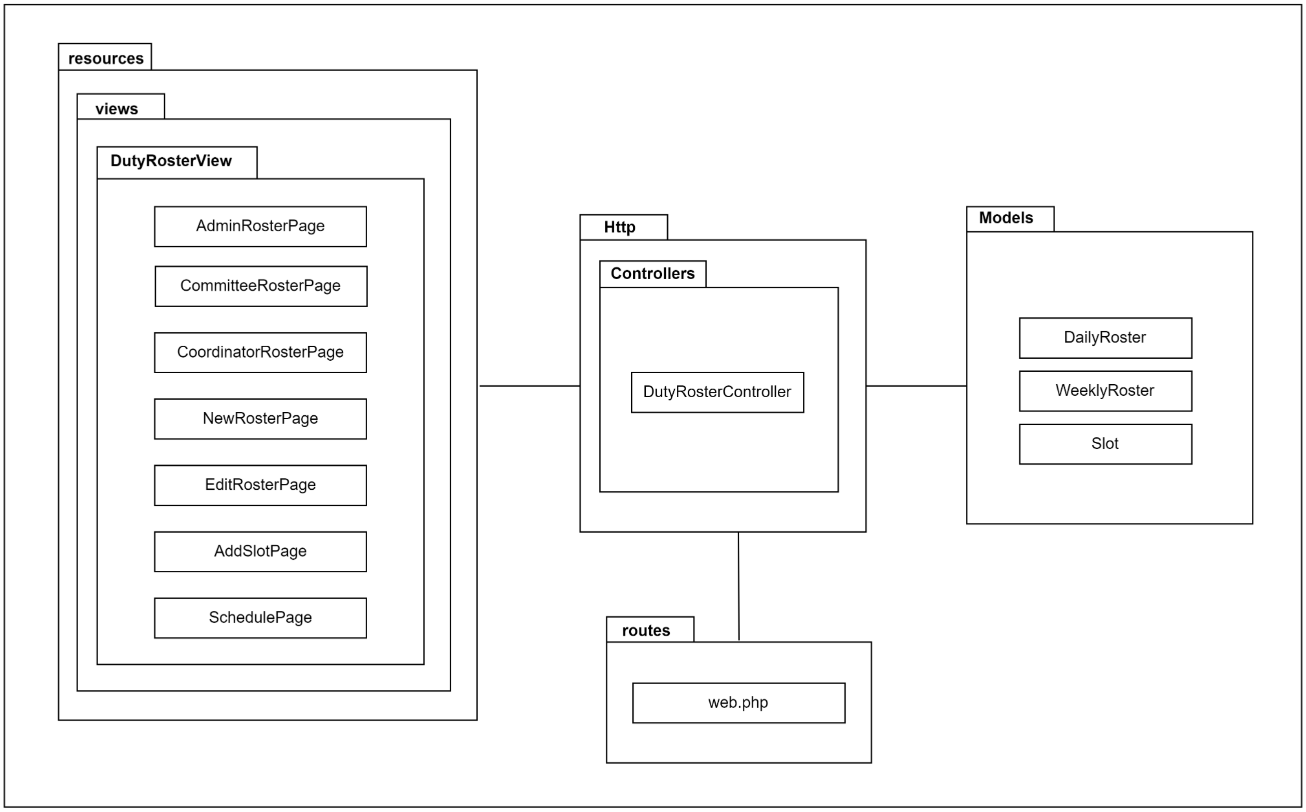


Figure 3.2.4 Package Diagram for Manage Duty Roster

3.2.4.1 DutyRosterView [PMMS-SDD-REQ-600]

Class Name	Description
AdminRosterPage	Interface that displays the latest duty roster retrieved from the database for admin with provided <<new>>, <<edit>> and <<delete>> button.
CommitteeRosterPage	Interface that displays the latest duty roster retrieved from the database for the PETAKOM Committee and Secretariat.
CoordinatorRosterPage	Interface that displays the latest duty roster retrieved from the database for the PETAKOM Coordinator.
NewRosterPage	Interface that allows admin to create a new weekly roster and save the data in the database.
EditRosterPage	Interface that allows admin to edit the weekly roster and update in the database.
AddSlotPage	Interface that allows the PETAKOM Committee to add slots to his/her duty schedule and save in the database.
SchedulePage	Interface that allows the PETAKOM Committee to view or delete his/her added slot.

3.2.4.2 Controllers [PMMS-SDD-REQ-200]

Class Name	Description
DutyRosterController	Controller class that controls the flow of duty roster module, including navigation between duty roster interfaces, creating and adding slots.

3.2.4.3 Models [PMMS-SDD-REQ-300]

Class Name	Description
DailyRoster	Model class that declares data type of daily roster details according to data dictionary.
WeeklyRoster	Model class that declares data type of weekly roster according to data dictionary.
Slot	Model class that declares data type of slots according to data dictionary.

3.2.5 Manage Report (CB20114 - LOO CHANG HERNG)

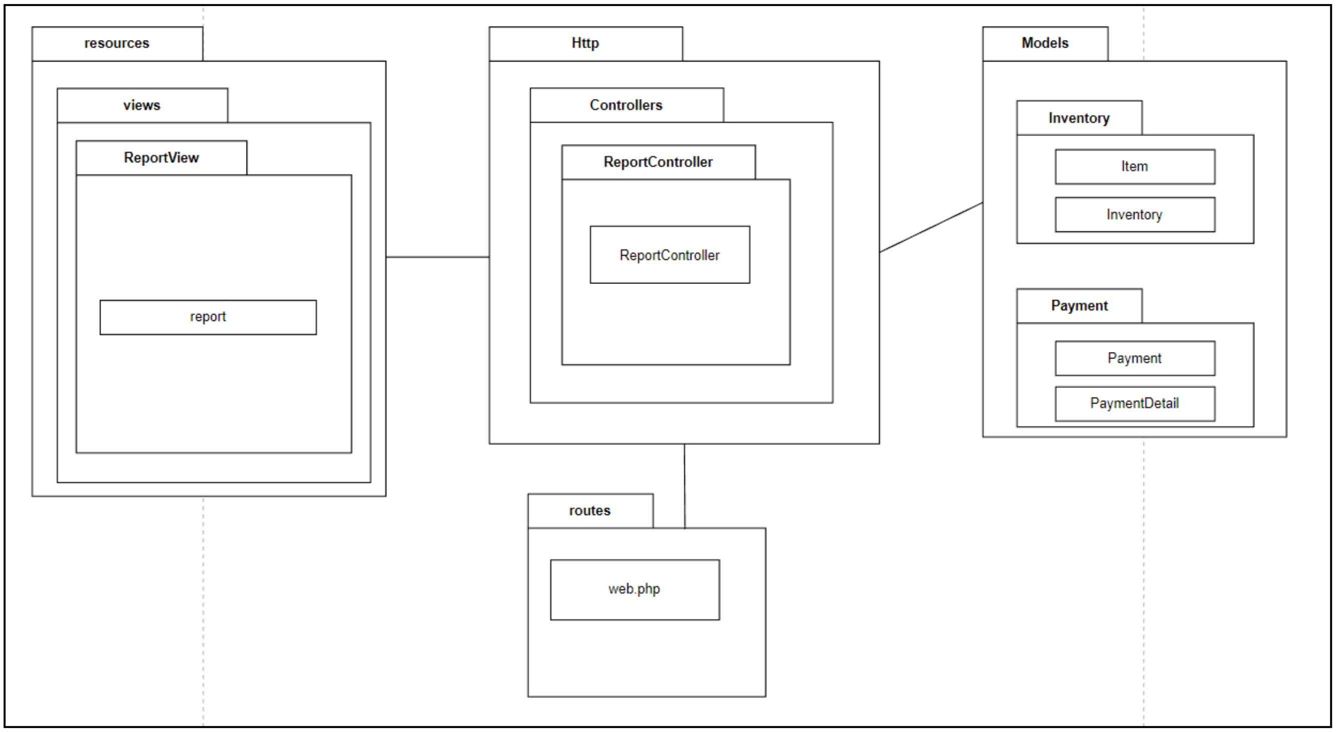


Figure 3.2.5 Package Diagram for Manage Report

3.2.5.1 ReportView [PMMS-SDD-REQ-700]

Class Name	Description
report	System displays the statistic of sales and inventory.

3.2.5.2 Controllers [PMMS-SDD-REQ-200]

Class Name	Description
ReportController	Controller class that controls the flow of report module, including navigation between report interfaces, sorting of data, and displaying it in a graph view.

3.2.6 Manage Announcement (CB20068 - TEE JIA MEI)

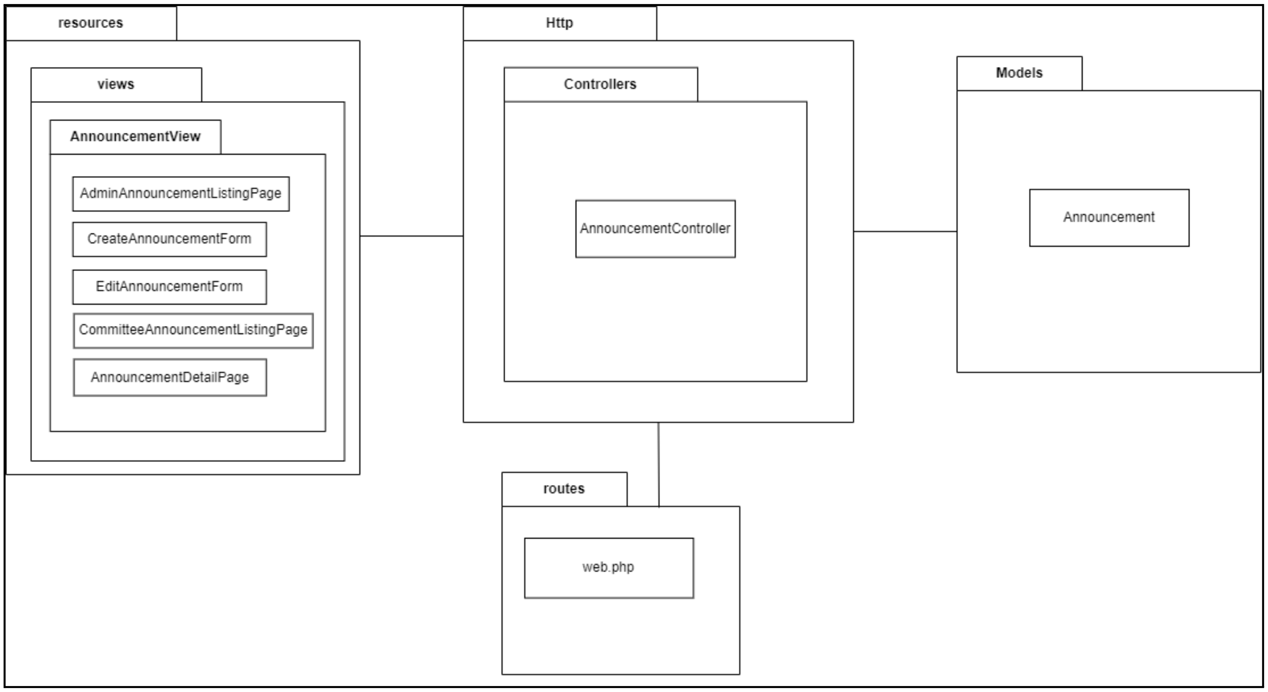


Figure 3.2.6 Package Diagram for Manage Announcement

3.2.6.1 AnnouncementView [PMMS-SDD-REQ-800]

Class Name	Description
AdminAnnouncementListingPage	Interface that displays the list of announcements for admin.
CreateAnnouncementForm	Interface that allows admin to create a new announcement.
EditAnnouncementForm	Interface that allows admin to update an existing announcement details.
CommitteeAnnouncementListingPage	Interface that displays the list of announcements for the PETAKOM Committee, Secretariat and the PETAKOM Coordinator.
AnnouncementDetailPage	Interface that allows PETAKOM Committee and Secretariat, PETAKOM Coordinator to view the selected announcement details.

3.2.6.2 Controllers [PMMS-SDD-REQ-200]

Class Name	Description
AnnouncementController	Controller class that is responsible to manage the flow of manage announcement module, including receiving input from CreateAnnouncementForm or EditAnnouncementForm user interface, validating user input, interacting with Announcement Model to create or update announcement, and returning appropriate responses to the user interface based on the success or failure of the creating and editing announcement process.

3.2.6.3 Models [PMMS-SDD-REQ-300]

Class Name	Description
Announcement	Model class that is responsible to store and retrieve announcement data in the database.

3.3 Data Dictionary

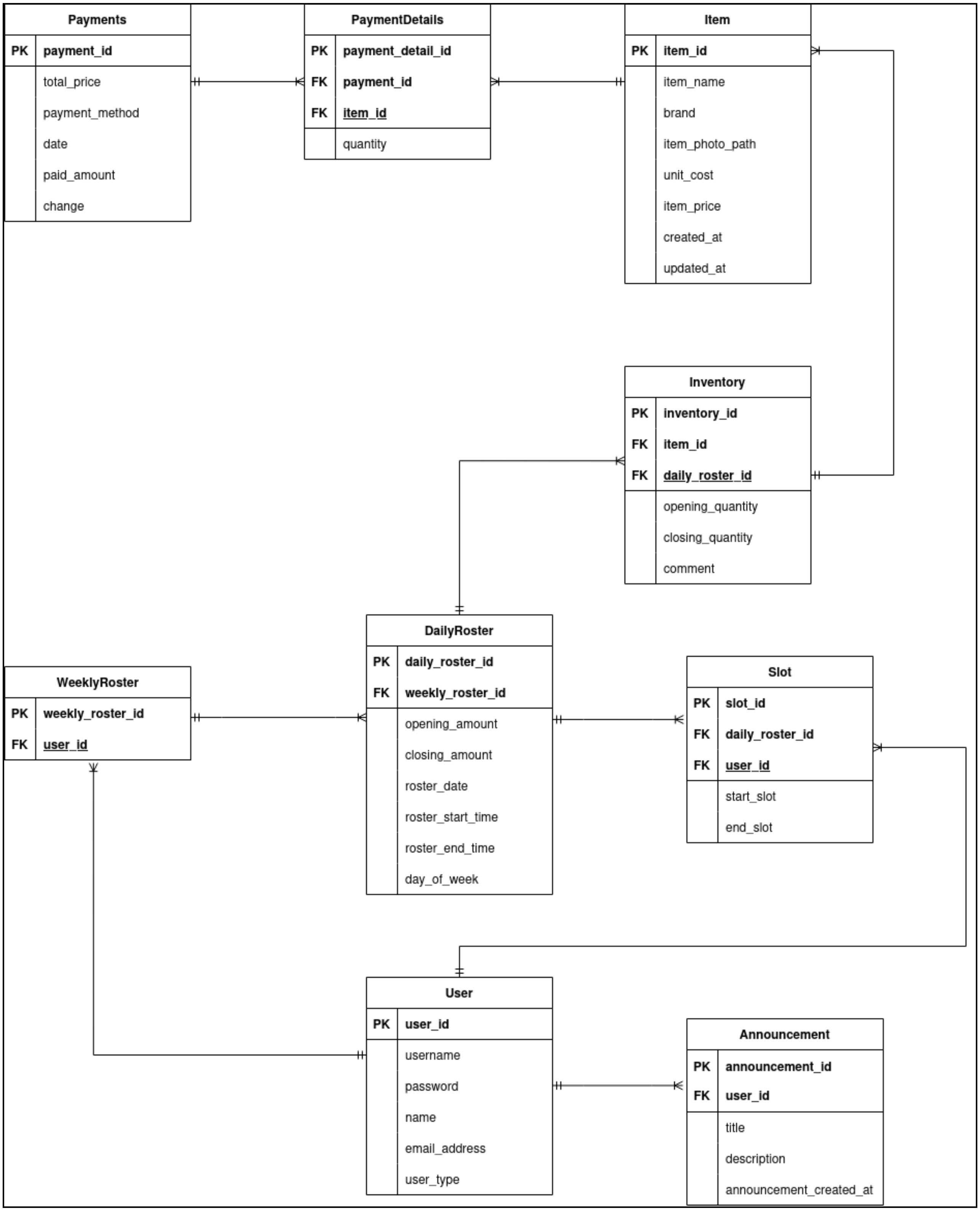


Figure 3.3: Entity Relationship Diagram for PETAKOM Mart Management System

3.3.1 Manage User Registration (CB20068 - TEE JIA MEI)

User

Field Name	Description	Data Type	Constraint
user_id	User ID	INTEGER	PK, NOT NULL
username	Matric ID of the PETAKOM Committee, Secretariat or Coordinator	VARCHAR(10)	NOT NULL
password	Password of the user account	VARCHAR(20)	NOT NULL
name	Full name of the PETAKOM Committee, Secretariat or Coordinator	VARCHAR(50)	-
email_address	Email address of the PETAKOM Committee, Secretariat or Coordinator	VARCHAR(30)	-
user_type	User type either admin, PETAKOM Committee and Secretariat or PETAKOM Coordinator	VARCHAR(50)	-

3.3.2 Manage Inventory (CB20121 - TAN TENG XING)

Item

Field Name	Description	Data Type	Constraint
item_id	Item ID	INTEGER	PK
item_name	Item name	VARCHAR2(50)	NOT NULL
brand	Item brand	VARCHAR2(50)	-
item_photo_path	Path of the item photo	VARCHAR2(255)	-
unit_cost	Cost of item per unit	FLOAT	NOT NULL
item_price	Price of item	FLOAT	NOT NULL
created_at	Date & time where item created	TIMESTAMP	NOT NULL
updated_at	Date & time where item updated	TIMESTAMP	NOT NULL

Inventory

Field Name	Description	Data Type	Constraint
inventory_id	Inventory ID	INTEGER	PK
item_id	Item ID	INTEGER	FK
daily_roster_id	Daily roster ID	INTEGER	FK
current_quantity	Current quantity	INTEGER	-
opening _quantity	Opening quantity	INTEGER	-
closing_quantity	Closing quantity	INTEGER	-
comment	Comment or reason for missing inventory	TEXT	-
created_at	Date & time where item created	TIMESTAMP	NOT NULL
updated_at	Date & time where item updated	TIMESTAMP	NOT NULL

3.3.3 Manage Payment (CB20114 - LOO CHANG HERNG)

Payment

Field Name	Description	Data Type	Constraint
payment_id	payment ID	INTEGER	PK
total_price	total price of this payment	FLOAT	NOT NULL
payment_method	payment method of this payment	VARCHAR2(20)	NOT NULL
date	date of this payment	TIMESTAMP	NOT NULL
paid_amount	payment from customer	FLOAT	NOT NULL
change	change to customer	FLOAT	NOT NULL

PaymentDetail

Field Name	Description	Data Type	Constraint
payment_detail_id	payment detail ID	INTEGER	PK
payment_id	payment ID	INTEGER	FK
item_id	item ID	INTEGER	FK
quantity	quantity bought of this item	INTEGER	NOT NULL

3.3.4 Manage Duty Roster (CB20064 - TAN KHA HUI)

WeeklyRoster

Field Name	Description	Data Type	Constraint
weekly_roster_id	Weekly Roster ID	INTEGER	PK
user_id	User ID	INTEGER	FK

DailyRoster

Field Name	Description	Data Type	Constraint
daily_roster_id	Daily Roster ID	INTEGER	PK
weekly_roster_id	Weekly Roster ID	INTEGER	FK
opening_amount	Opening amount of the day	FLOAT	NOT NULL
closing_amount	Closing amount of the day	FLOAT	NOT NULL
roster_date	Date of the daily roster	TIMESTAMP	NOT NULL
roster_start_time	Starting time of the roster	TIMESTAMP	NOT NULL
roster_end_time	Closing time of the roster	TIMESTAMP	NOT NULL
day_of_week	Day of the week	STRING	NOT NULL

Slot

Field Name	Description	Data Type	Constraint
slot_id	Slot ID	INTEGER	PK
daily_roster_id	Daily Roster ID	INTEGER	FK
user_id	User ID	INTEGER	FK
start_slot	Starting time of slot	TIMESTAMP	NOT NULL
end_slot	End time of slot	TIMESTAMP	NOT NULL

3.3.5 Manage Announcement (CB20068 - TEE JIA MEI)

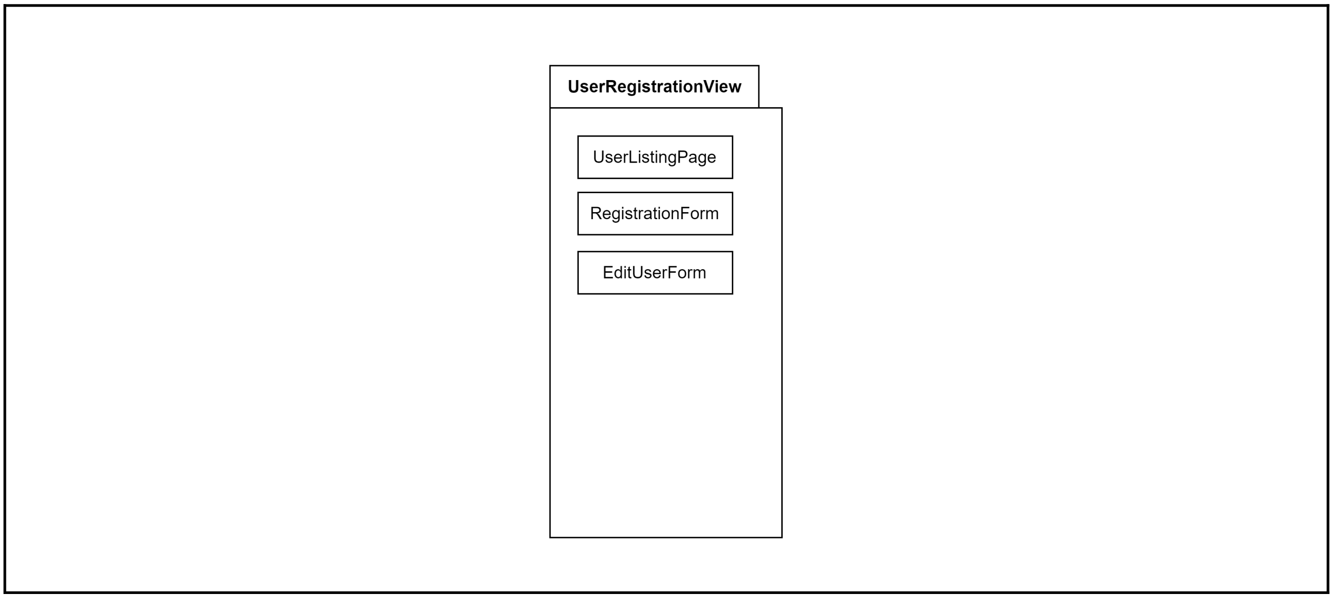
Announcement

Field Name	Description	Data Type	Constraint
announcement_id	Announcement ID	INTEGER	PK, NOT NULL
user_id	User ID	INTEGER	FK
title	Title of the announcement	VARCHAR(100)	-
description	Description of the announcement	VARCHAR(300)	-
announcement_created_at	Date of creating the announcement	DATE	-

4. DETAIL DESIGN

4.1 Manage User Registration (CB20068 - TEE JIA MEI)

4.1.1 UserRegistrationView [PMMS-SDD-REQ-100]



4.1.1.1 UserListingPage [PMMS-SDD-REQ-101]

Class Type	Boundary class	
Responsibility	An interface to display all the registered user information.	
Attributes	Attributes Name	Attributes Type
	-	-
Methods	Method Name	Description
	-	-
Algorithm	1. Start 2. Add <<Create New User>> button and once click the button, send a GET request to the “create-user” route to display the RegistrationForm. 3. Create a table to display the list of users, including columns for the username, name, email address and user type. 4. Use @foreach to iterate over the list of users and display each one in the appropriate row and column of the table. 5. For each user row, add <<Edit>> button and once click the button, send a GET request to the “edit-user/{id}” route to display the EditUserForm. 6. For each user row, add <<Delete>> button and once click the button, send a GET request to the “delete-user/{id}” route to delete the corresponding user record from the database . 7. End	

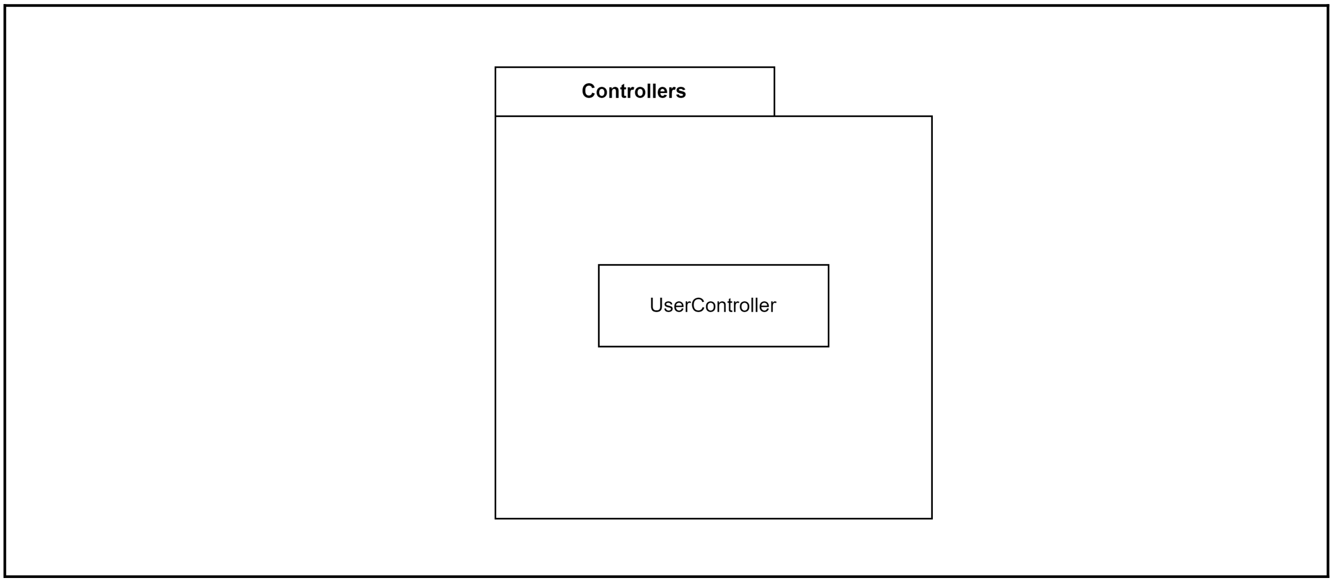
4.1.1.2 RegistrationForm [PMMS-SDD-REQ-102]

Class Type	Boundary class	
Responsibility	An interface that allows admin to create a new user account for PETAKOM Committee and Secretariat or PETAKOM Coordinator.	
Attributes	Attributes Name	Attributes Type
	-	-
Methods	Method Name	Description
	-	-
Algorithm	1. Start 2. Create a form that uses the HTTP method ‘POST” and has an action attribute that points to a route that maps to the “store” method of the UserController. 3. Include input fields such as username, name, email address, password and select field which is user type inside the form. 4. Add a CSRF token field to the form, using the “@csrf” directive. 5. Add <<Save>> button to trigger form submission. 6. End	

4.1.1.3 EditUserForm [PMMS-SDD-REQ-103]

Class Type	Boundary class	
Responsibility	An interface that allows admin to update user account information for the PETAKOM Committee and Secretariat or PETAKOM Coordinator.	
Attributes	Attributes Name	Attributes Type
	-	-
Methods	Method Name	Description
	-	-
Algorithm	1. Start 2. Create a form that uses the HTTP method ‘POST” and has an action attribute that points to a route that maps to the “update” method of the UserController. 3. Include input fields such as username, name, email address, password and select field which is user type inside the form. 4. Add a CSRF token field to the form, using the “@csrf” directive. 5. Add <<Save>> button to trigger form submission. 6. End	

4.1.2 Controllers [PMMS-SDD-REQ-200]

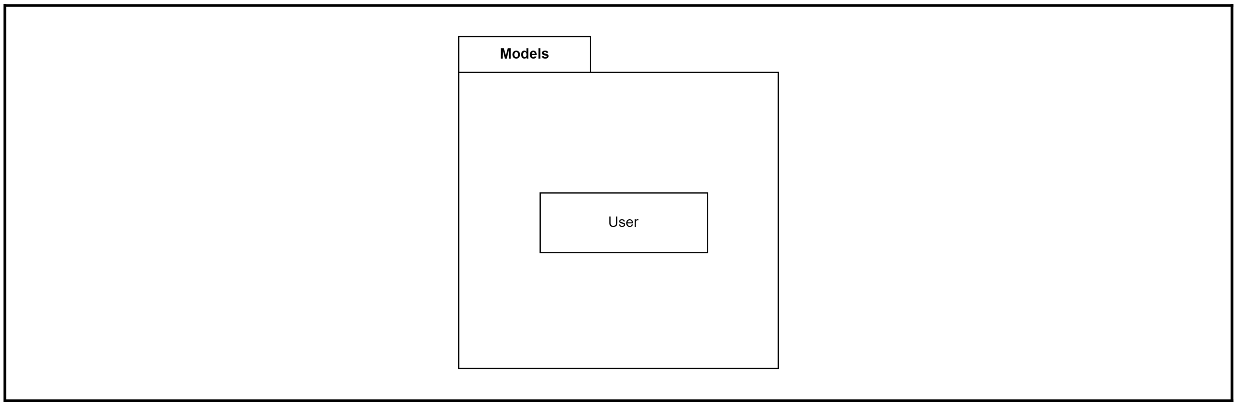


4.1.2.1 UserController [PMMS-SDD-REQ-201]

Class Type	Controller Class	
Responsibility	Controller class that is responsible to manage the flow of user registration module, including receiving input from RegistrationForm or EditUserForm user interface, validating user input, interacting with User Model to create or update user accounts, and returning appropriate responses to the user interface based on the success or failure of the registration process.	
Attributes	Attributes Name	Attributes Type
	-	-
Methods	Method Name	Description
	index()	To retrieve all announcements from the database and return them to the “UserListingPage” view for display.
	create()	To return a blank “RegistrationForm” view for the user to fill out and submit.
	store()	To receive the form data from the “RegistrationForm” view and save a new user to the database.
	edit()	To retrieve an user by ID from the database and return it to the “EditUserForm” view for the user to edit and submit.

	update()	To receive the edited form data from the "EditUserForm" view, validate the data, and update the corresponding user in the database.
	destroy()	To receive a request containing the ID of a user to delete, delete the corresponding user from the database, and redirect the user back to the "UserListingPage" view.
Algorithm	1. Start 2. Create a "index()" method. 3. Create a "create()" method. 4. Create a "store()" method. 5. Create a "edit()" method. 6. Create a "update()" method. 7. Create a "destroy()" method. 8. End	

4.1.3 Models [PMMS-SDD-REQ-300]

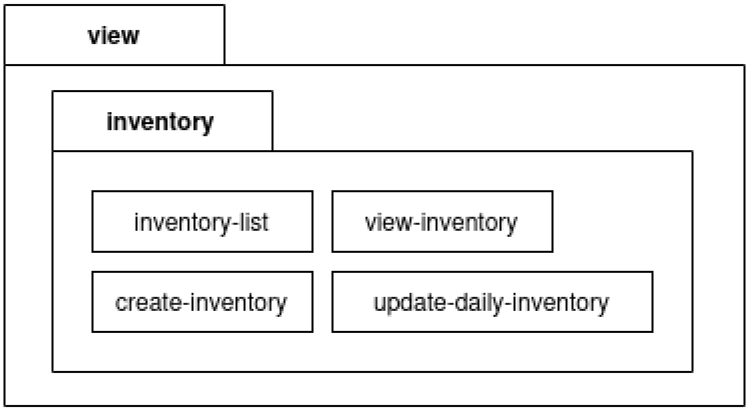


4.1.3.1 User [PMMS-SDD-REQ-301]

Class Type	Model Class	
Responsibility	Model class that is responsible to store and retrieve user data in the database.	
Attributes	Attributes Name	Attributes Type
	user_id	Integer
	username	String
	password	String
	name	String
	email_address	String
	user_type	String
Methods	Method Name	Description
	-	-
Algorithm	1. Start 2. Announcement is defined as a class that extends the “Model” class. 3. Define “$fillable” property that lists the fields such as user_id, username, password, name, email_address, user_type 4. End	

4.2 Manage Inventory (CB20121 - TAN TENG XING)

4.2.1 InventoryView [PMMS-SDD-REQ-400]



4.2.1.1 inventory-list (index) [PMMS-SDD-REQ-401]

Class Type	**Boundary class**	
Responsibility	An interface to display the inventory list.	
Attributes	Attributes Name	Attributes Type
	item_id item_name brand unit_cost item_price	Integer String String Float Float
Methods	Method Name	Description
	create()	Navigates users to “create-inventory” page.
	viewItem()	Navigates users to “view-inventory” page.
	editItem()	Navigates users to “edit-inventory” page.
	openingInventory	Navigates users to “opening-inventory” page.
	closingInventory	Navigates users to “closing-inventory” page.
	deleteItem()	Delete inventory.
Algorithm	None	

4.2.1.2 create-inventory [PMMS-SDD-REQ-402]

Class Type	**Boundary class**	
Responsibility	An interface to display form for creating new inventory.	
Attributes	Attributes Name	Attributes Type
	item_id	Integer
	item_name	String
	brand	String
	unit_cost	Float
	item_price	Float
Methods	Method Name	Description
	createInventory()	Post HTTP request to create new inventory.
Algorithm	1. Validate values of the input inventory attributes. 2. Post HTTP request for saving user inputs in database. 3. End.	

4.2.1.3 view-inventory [PMMS-SDD-REQ-403]

Class Type	**Boundary class**	
Responsibility	An interface to display form for viewing inventory in more detail.	
Attributes	Attributes Name	Attributes Type
	item_id item_name Brand item_photo_path unit_cost item_price	Integer String String String Float Float
Methods	Method Name	Description

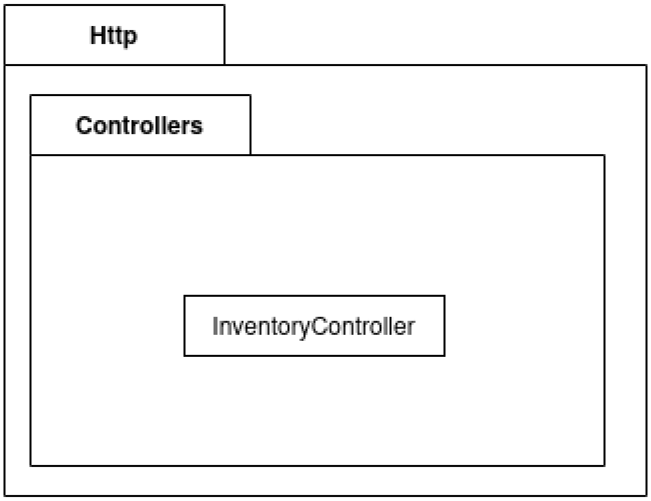
	editInventory()
	deleteInventory()
		A button that navigates users to “edit-inventory” page.
		Delete inventory.

Algorithm	None	

4.2.1.4 update-daily-inventory [PMMS-SDD-REQ-404]

Class Type	**Boundary class**	
Responsibility	An interface to display form for input opening and closing inventory.	
Attributes	Attributes Name	Attributes Type
	item_id	Integer
	item_name	String
	opening _quantity	Integer
	closing_quantity	Integer
	comment	String
Methods	Method Name	Description
	updateDailyInventory()	Submit button that posts a request to update opening and closing inventory.
Algorithm	None	

4.2.2 Controllers [PMMS-SDD-REQ-200]

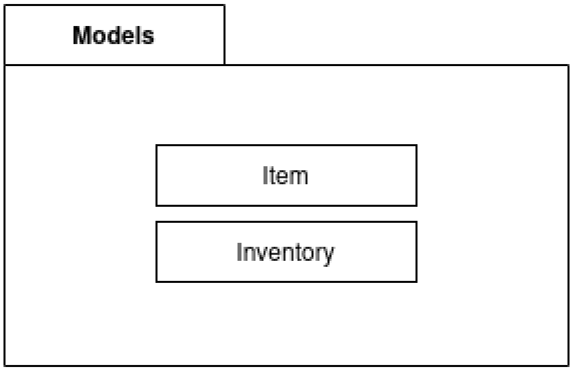


4.2.2.1 InventoryController [PMMS-SDD-REQ-202]

Class Type	**Controller class**	
Responsibility	A controller to handle incoming requests related to inventory, including validate user inputs, create and update inventory attributes in database, search and delete inventory.	
Attributes	Attributes Name	Attributes Type
	item_id item_name item_photo_path unit_cost item_price opening _quantity closing_quantity comment	Integer String String Float Float Integer Integer String
Methods	Method Name	Description
	createItem()	Navigates users to pages for creating new items.
	viewItem()	Navigates users to pages for viewing selected item.

	editItem()	Navigates users to pages for editing selected item.
	updateDailyInventory()	Navigates users to pages for updating daily inventories.
	deleteItem()	Delete row of selected item from database.
Algorithm	createItem() 1. Return blade script of crateItem. 2. End. viewItem() 1. Return blade script of viewItem. 2. End. editItem() 1. Return blade script of editItem. 2. End. updateDailyInventory() 1. Return blade script of updateDailyInventory. 2. End. deleteItem() 1. Return SQL statement: ‘DELETE FROM `item` WHERE item_id = $item_id’. 2. End.	

4.2.3 Models [PMMS-SDD-REQ-300]



4.2.3.1 Item [PMMS-SDD-REQ-302]

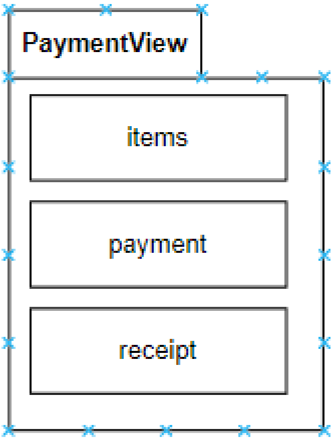
Class Type	**Model class**	
Responsibility	A model to interact with "item" table and other related tables.	
Attributes	Attributes Name	Attributes Type
	item_name brand item_photo_path unit_cost item_price	String String String Float Float
Methods	Method Name	Description
	inventory()	Declare relationship with inventory.
	paymentDetail()	Declare relationship with payment detail.
Algorithm	inventory() 1. Return relationship of item belongs to inventory. 2. End. paymentDetail() 1. Return relationship of item belongs to paymentDetail. 2. End.	

4.2.3.2 Inventory [PMMS-SDD-REQ-303]

Class Type	**Model class**	
Responsibility	A model to interact among “inventory” table and other related tables.	
Attributes	Attributes Name	Attributes Type
	inventory_id	Integer
	item_id	Integer
	daily_roster_id	Integer
	opening_quantity	Integer
	closing_quantity	Integer
	comment	String
Methods	Method Name	Description
	item()	Declare relationship with item.
	dailyRoster()	Declare relationship with daily roster.
Algorithm	item() 1. Return relationship of inventory has many items. 2. End. dailyRoster() 1. Return relationship of inventory belongs to daily roster. 2. End.	

4.3 Manage Payment (CB20114 - LOO CHANG HERNG)

4.3.1 PaymentView [PMMS-SDD-REQ-500]



4.3.1.1 items [PMMS-SDD-REQ-501]

Class Type	Boundary Class	
Responsibility	This class responsible to display an interface of all available items in the mart for cashier to chose from.	
Attributes	Attributes Name	Attributes Type
	photo name price quantity	string string float integer
Methods	Method Name	Description
	-	-
Algorithm	1. Start 2. For each item, display the item's information. 3. End	

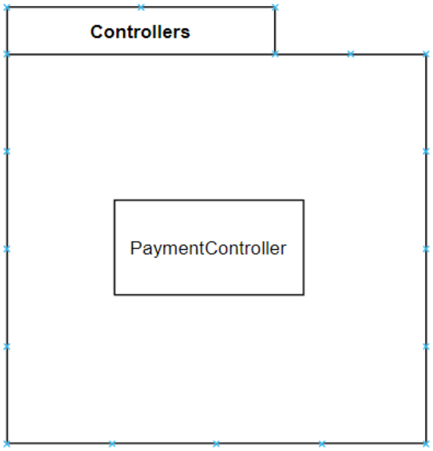
4.3.1.2 payment [PMMS-SDD-REQ-502]

Class Type	Boundary Class	
Responsibility	This class responsible to display an interface to show all the selected items, price of each item, and the total price.	
Attributes	Attributes Name	Attributes Type
	photo name price unit_total_price quantity total_price	string string float float integer float
Methods	Method Name	Description
	-	-
Algorithm	1. Start 2. For each selected item, display its unit price, unit, and total price. 3. Display the total price of all item. 4. End.	

4.3.1.3 receipt [PMMS-SDD-REQ-503]

Class Type	Boundary Class	
Responsibility	This class responsible to display an interface to show a receipt that consists of information like purchase time, total price, purchased items.	
Attributes	Attributes Name	Attributes Type
	photo name price quantity total_price	string string float integer float
Methods	Method Name	Description
	-	-
Algorithm	1. Start 2. For each purchased item, display its unit price, quantity purchased. 3. Display the total price. 4. End	

4.3.2 Controllers [PMMS-SDD-REQ-200]



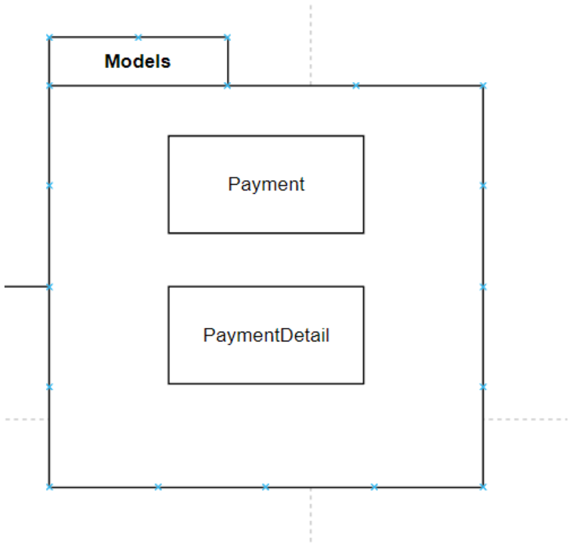
4.3.2.1 PaymentController [PMMS-SDD-REQ-203]

Class Type	Controller Class	
Responsibility	This controller class responsible to control the navigation between payment veiws, store data to database, update data, and also delete data.	
Attributes	Attributes Name	Attributes Type
	-	-
Methods	Method Name	Description
	payment(items)	This function navigates user from items page to payment page.
	receipt(items)	This function navigates user from payment page to receipt page.
	index()	This function return items page to user.
	increaseQuantity(item, inventory)	This function allows user to add the item's quantity by 1.
	decreaseQuantity(item)	This function allows user to decrease the item's quantity by 1.

Algorithm	payment(items)
	1. return payment view of items selected
	2. End
	receipt(items)	

	1. store data to payment, paymentDetail database. 2. return receipt view of items selected and display payment successful message. 3. End index() 1. return items view. 2. End increaseQuantity(item, inventory) 1. Start 2. IF item.quantity < inventory.stock 3. item.quantity += 1 4. End decreaseQuantity(item) 1. Start 2. IF item.quantity >0 3. item.quantity -= 1 4. ELSE 5. return 6. End

4.3.3 Models [PMMS-SDD-REQ-300]



4.3.3.1 Payment [PMMS-SDD-REQ-304]

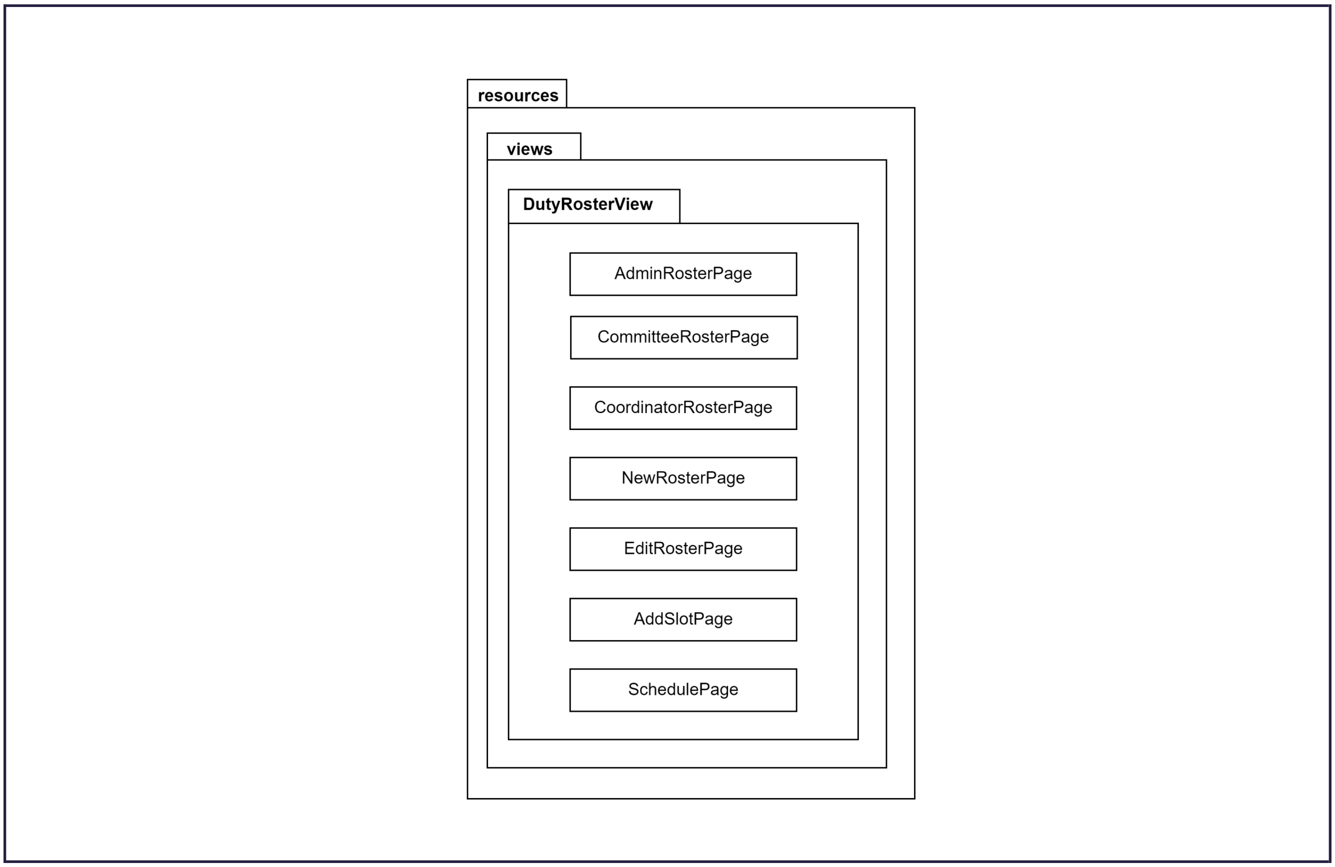
Class Type	Model Class	
Responsibility	This model class responsible to store data of payment, and declare the relationship between itself, user and payment detail.	
Attributes	Attributes Name	Attributes Type
	-	-
Methods	Method Name	Description
	user()	This function declare the relationship between payment and user.
	paymentDetail()	This function declare the relationship between payment and payment details.
Algorithm	user() 1. return this payment belongs to user. 2. End paymentDetail() 1. return this payment can have many payment details. 2. End	

4.3.3.2 PaymentDetail [PMMS-SDD-REQ-305]

Class Type	Model Class	
Responsibility	This model class responsible to store data of payment detail, and declare the relationship between payment and payment details.	
Attributes	Attributes Name	Attributes Type
	-	-
Methods	Method Name	Description
	payment()	This function declare the relationship between payment and payment details.
Algorithm	payment() return this payment detail belongs to payment. End	

4.4 Manage Duty Roster (CB20064 - TAN KHA HUI)

4.4.1 DutyRosterView [PMMS-SDD-REQ-600]



4.4.1.1 AdminRosterPage [PMMS-SDD-REQ-601]

Class Type	Boundary Class	
Responsibility	An interface to display the latest duty roster retrieved from the database to the admin with providing the person-in-charge for each time slot.	
Attributes	Attributes Name	Attributes Type
	-	-
Methods	Method Name	Description
	-	-
Algorithm	1. Start. 2. Display for each time slot with the corresponding person-in-charge of the latest weekly duty roster retrieved from the database. 3. Use @foreach to loop through each time slot of the latest weekly duty roster in the database. 4. Add <<+New>> button which will navigate users to the NewRosterPage once users click it. 5. Add <<Edit>> button, which will navigate users to the EditRosterPage once users click it. 6. Add <<Delete>> button to delete the selected weekly duty roster from the database. 7. End.	

4.4.1.2 CommitteeRosterPage [PMMS-SDD-REQ-602]

Class Type	Boundary Class	
Responsibility	An interface to display the latest duty roster retrieved from the database to the PETAKOM Committee and Secretariat.	
Attributes	Attributes Name	Attributes Type
	-	-
Methods	Method Name	Description
	-	-
Algorithm	1. Start. 2. Display for each time slot of the latest weekly duty roster retrieved from the database. 3. Use @foreach to loop through each time slot of the latest weekly duty roster in the database. 4. End.	

4.4.1.3 CoordinatorRosterPage [PMMS-SDD-REQ-603]

Class Type	Boundary Class	
Responsibility	An interface to display the latest duty roster retrieved from the database to the PETAKOM Coordinator.	
Attributes	Attributes Name	Attributes Type
	-	-
Methods	Method Name	Description
	-	-
Algorithm	1. Start. 2. Display for each time slot of the latest weekly duty roster retrieved from the database. 3. Use @foreach to loop through each time slot of the latest weekly duty roster in the database. 4. End.	

4.4.1.4 NewRosterPage [PMMS-SDD-REQ-604]

Class Type	Boundary Class	
Responsibility	An interface to allow admin to create a new weekly roster and save the data in the database.	
Attributes	Attributes Name	Attributes Type
	-	-
Methods	Method Name	Description
	-	-
Algorithm	1. Start. 2. Create the input field for selecting date, opening and closing time. 3. Add the <<POST>> button to trigger the input data submission and save in the database. 4. End.	

4.4.1.5 EditRosterPage [PMMS-SDD-REQ-605]

Class Type	Boundary class	
Responsibility	An interface to allow admin to edit the weekly roster and update in the database.	
Attributes	Attributes Name	Attributes Type
	-	-
Methods	Method Name	Description
	-	-
Algorithm	1. Start. 2. Use @foreach to loop through each daily roster data of the selected weekly duty roster in the database. 3. Display the input field for date, opening and closing time with pre-filled values which are retrieved from the database. 4. Add a <<UPDATE>> button to trigger the form submission and save in the database. 5. End.	

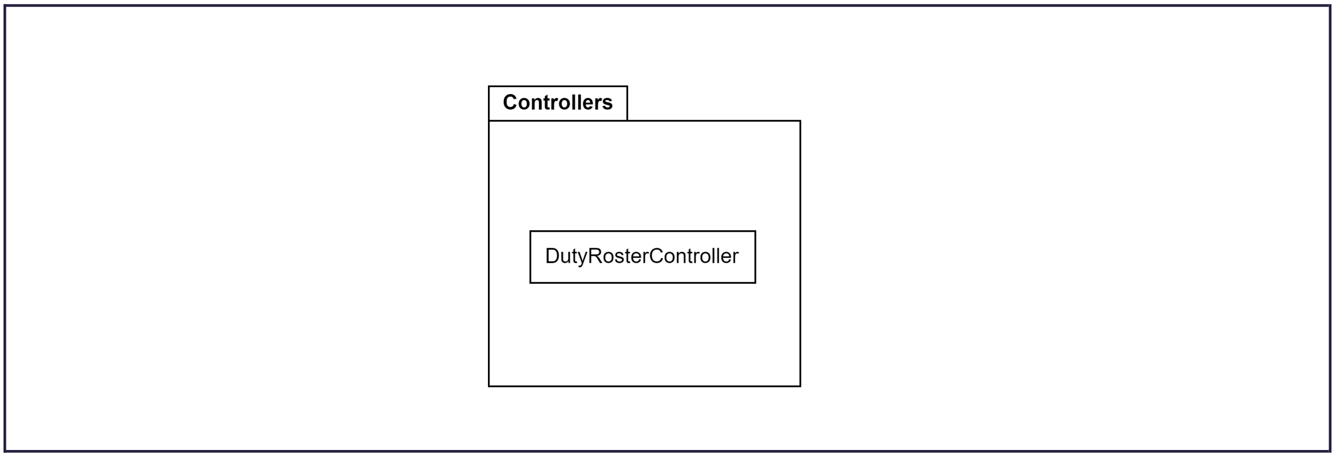
4.4.1.6 AddSlotPage [PMMS-SDD-REQ-606]

Class Type	Boundary Class	
Responsibility	An interface to allow the PETAKOM Committee to add slots to his/her duty schedule and save in the database.	
Attributes	Attributes Name	Attributes Type
	-	-
Methods	Method Name	Description
	-	-
Algorithm	1. Start. 2. Create the input field for selecting date and time slot. 3. Add the <<ADD>> button to trigger the input data submission and save in the database. 4. End.	

4.4.1.7 SchedulePage [PMMS-SDD-REQ-607]

<table>
<tr><td>Class Type</td><td colspan="2">Boundary Class</td></tr>
<tr><td>Responsibility</td><td colspan="2">An interface to allow the PETAKOM Committee to view or delete his/her added slot.</td></tr>
<tr><td rowspan="2">Attributes</td><td>Attributes Name</td><td>Attributes Type</td></tr>
<tr><td>-</td><td>-</td></tr>
<tr><td rowspan="2">Methods</td><td>Method Name</td><td>Description</td></tr>
<tr><td>-</td><td>-</td></tr>
<tr><td>Algorithm</td><td colspan="2">1. Start.
2. IF time slot is not empty
3. Use @foreach to loop through each time slot of the user in the database and display it in a list form.
4. Add a <<delete>> button for each time slot displayed to allow the user to remove the corresponding time slot from the database.
5. End.</td></tr>
</table>

4.4.2 Controllers [PMMS-SDD-REQ-200]

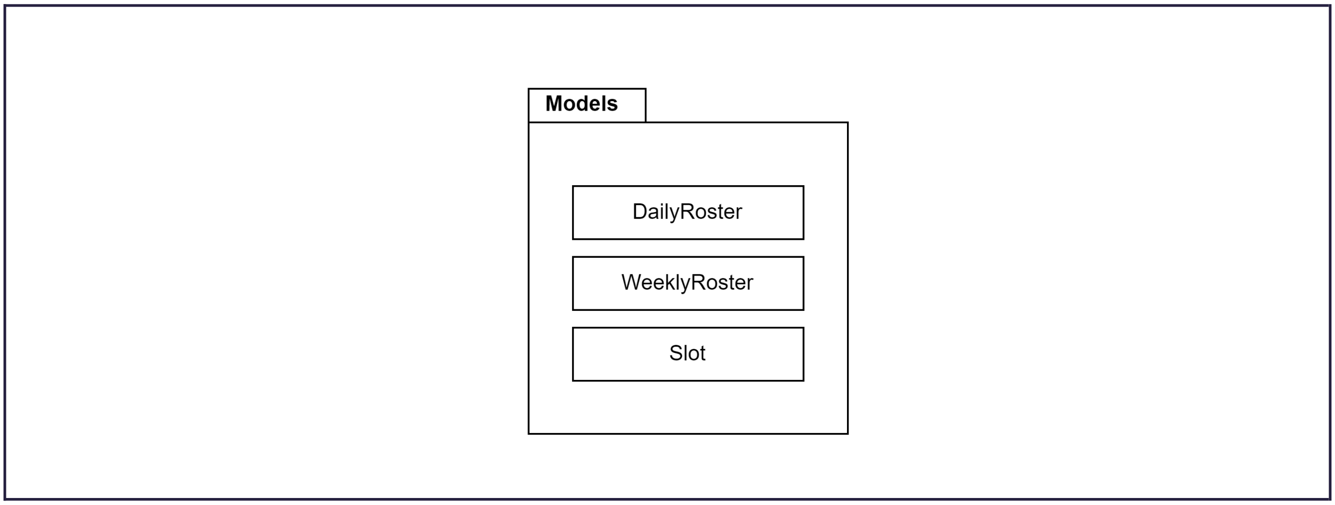


4.4.2.1 DutyRosterController [PMMS-SDD-REQ-204]

Class Type	Controller Class	
Responsibility	Controller class that controls the flow of duty roster module, including navigation between duty roster interfaces, creating and adding slots.	
Attributes	Attributes Name	Attributes Type
	-	-
Methods	Method Name	Description
	showAdminRoster()	Retrieve the latest duty roster from the database and return to the "AdminRosterPage" view for display.
	showCommitteeRoster()	Retrieve the latest duty roster from the database and return to the "CommitteeRosterPage" view for display.
	showCoordinatorRoster()	Retrieve the latest duty roster from the database and return to the "CoordinatorRosterPage" view for display.
	newRoster()	Return to the "AddRosterPage" view.
	createRoster()	It handles the creation of a new weekly roster. It validates the incoming request data, creates a new WeeklyRoster instance, and saves it to the database.
	generateTimeSlots()	Generate the time slots for the selected date based on the input of opening time and closing time.
	editRoster()	Retrieve the weekly roster data to be edited and displayed in the "EditRosterPage".
	updateRoster()	It handles the submission of the edited roster

		details by validating the incoming request data, retrieves the existing roster entry to be updated, updates the necessary fields, and saves the changes to the database.
	deleteRoster()	It handles the deletion of the selected weekly roster.
	newSlot()	Return to the "AddSlotPage" view.
	addSlot()	It handles the submission of the selected time slot, creates a new entry in the Slot model and saves it to the database.
	showSchedule()	Retrieve the added time slot of the user from the database and return to the "SchedulePage" view for display.
	deleteTimeSlot()	It handles the deletion of the selected time slot.
Algorithm	1. Start. 2. Create a "showRoster()" method. 3. Create a "newRoster()" method. 4. Create a "createRoster()" method. 5. Create a "generateTimeSlots()" method. 6. Create a "editRoster()" method. 7. Create a "updateRoster()" method. 8. Create a "deleteRoster()" method. 9. Create a "newSlot()" method. 10. Create a "addSlot()" method. 11. Create a "showSchedule()" method. 12. Create a "deleteTimeSlot()" method. 13. End.	

4.4.3 Models [PMMS-SDD-REQ-300]



4.4.3.1 DailyRoster [PMMS-SDD-REQ-306]

Class Type	Model Class	
Responsibility	Model class that declares data type of daily roster details according to data dictionary.	
Attributes	Attributes Name	Attributes Type
	daily_roster_id weekly_roster_id opening_amount closing_amount roster_date roster_start_time roster_end_time day_of_week	Integer Integer Float Float TimeStamp TimeStamp TimeStamp String
Methods	Method Name	Description
	weeklyRoster()	Declare the relationships between the daily roster and the weekly roster.
	slot()	Declare the relationships between the daily roster and the time slot.
Algorithm	1. Start. 2. Create a “weeklyRoster()” method. 3. Create a “slot()” method. 4. End.	

4.4.3.2 WeeklyRoster [PMMS-SDD-REQ-307]

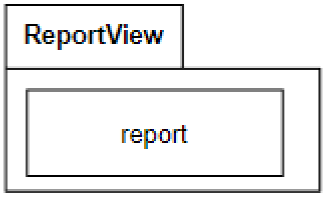
Class Type	Model Class	
Responsibility	Model class that declares data type of weekly roster according to data dictionary.	
Attributes	Attributes Name	Attributes Type
	weekly_roster_id user_id	Integer Integer
Methods	Method Name	Description
	dailyRoster()	Declare the relationships between the weekly roster and the daily roster.
	user()	Declare the relationships between the weekly roster and the user.
Algorithm	1. Start. 2. Create a “dailyRoster()” method. 3. Create a “user()” method. 4. End.	

4.4.3.3 Slot [PMMS-SDD-REQ-308]

Class Type	Model Class	
Responsibility	Model class that declares data type of slots according to the data dictionary.	
Attributes	Attributes Name	Attributes Type
	slot_id daily_roster_id user_id start_slot end_slot	Integer Integer Integer TimeStamp TimeStamp
Methods	Method Name	Description
	dailyRoster()	Declare the relationships between the time slot and the daily roster.
	user()	Declare the relationships between the time slot and the user.
Algorithm	1. Start. 2. Create a “dailyRoster()” method. 3. Create a “user()” method. 4. End.	

4.5 Manage Report (CB20114 - LOO CHANG HERNG)

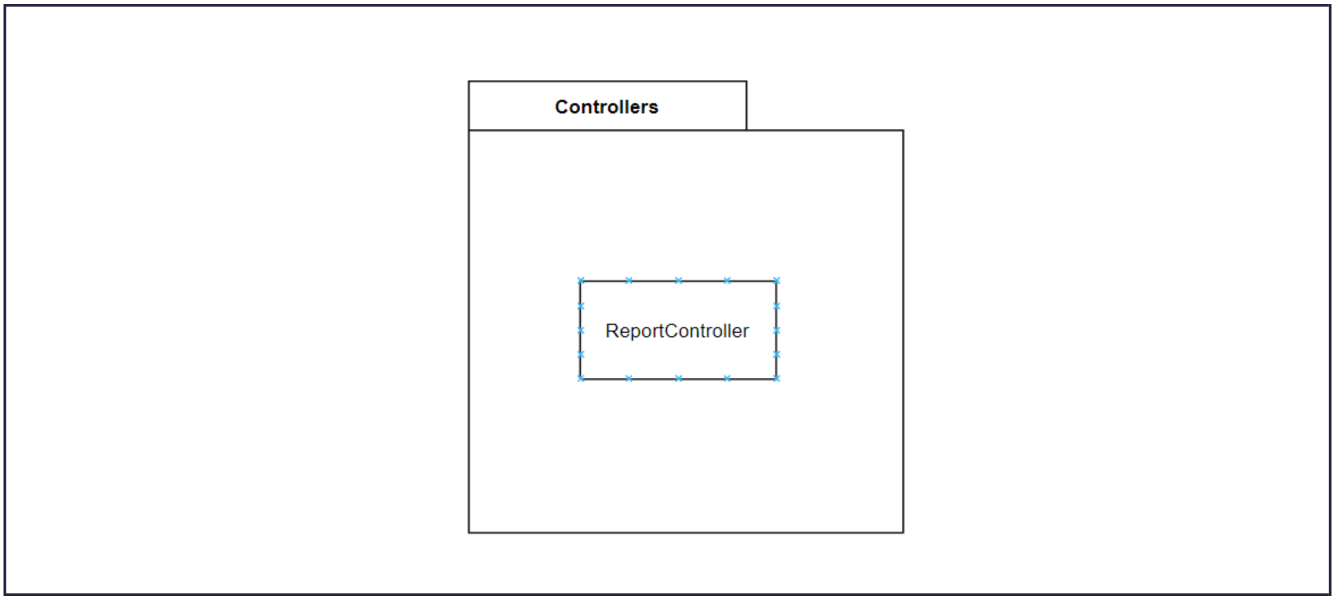
4.5.1 ReportView [PMMS-SDD-REQ-700]



4.5.1.1 report [PMMS-SDD-REQ-701]

Class Type	Boundary Class	
Responsibility	This class responsible to display an interface that shows all the chart, graph, and data to user.	
Attributes	Attributes Name	Attributes Type
	name item_sold quantity profit	string integer integer float
Methods	Method Name	Description
	-	-
Algorithm	1. Start 2. For each item, display its name and quantity in inventory. 3. Display item inventory graph, and profit graph. 4. End.	

4.5.2 Controllers [PMMS-SDD-REQ-200]

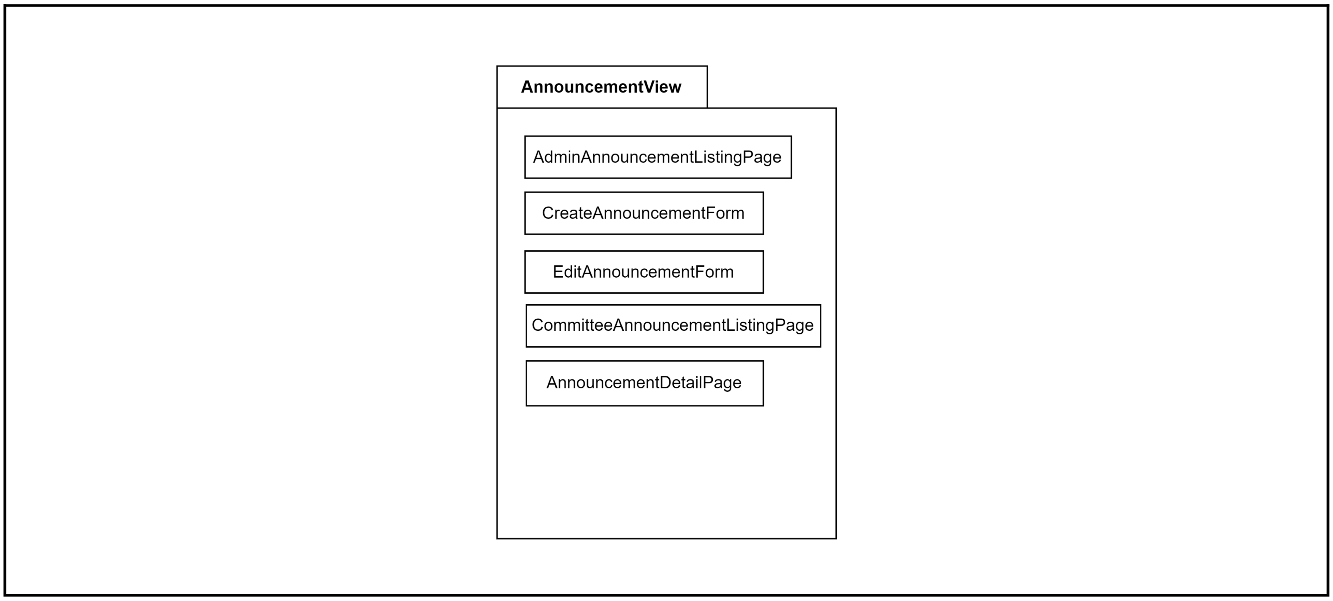


4.5.2.1 ReportController [PMMS-SDD-REQ-205]

Class Type	Controller Class	
Responsibility	This controller class responsible to control the navigation of report interfaces and process the data to show as graph to user.	
Attributes	Attributes Name	Attributes Type
	-	-
Methods	Method Name	Description
	report()	This function navigates user from main page to report page and retrieve data from database, process the data and display in a graph.
Algorithm	report() 1. retrieve data from database. 2. calculate profit using unit price, item sold, and unit cost. 3. return inventory report and profit report view. 4. End	

4.6 Manage Announcement (CB20068 - TEE JIA MEI)

4.6.1 AnnouncementView [PMMS-SDD-REQ-800]



4.6.1.1 AdminAnnouncementListingPage [PMMS-SDD-REQ-801]

Class Type	Boundary class	
Responsibility	An interface to display the list of announcements for admin.	
Attributes	Attributes Name	Attributes Type
	-	-
Methods	Method Name	Description
	-	-
Algorithm	1. Start 2. Add <<Create New Announcement>> button and once click the button, send a GET request to the “create-announcement” route to display the CreateAnnouncementForm. 3. Create a table to display the list of announcements, including columns for the announcement title, description and date. 4. Use @foreach to iterate over the list of announcements and display each one in the appropriate row and column of the table. 5. For each announcement row, add <<Edit>> button and once click the button, send a GET request to the “edit-announcement/{id}” route to display the EditAnnouncementForm. 6. For each announcement row, add <<Delete>> button and once click the button, send a GET request to the “delete-announcement/{id}” route to delete the corresponding announcement record from the database . 7. End	

4.6.1.2 CreateAnnouncementForm [PMMS-SDD-REQ-802]

Class Type	Boundary class	
Responsibility	An interface for admin to create a new announcement.	
Attributes	Attributes Name	Attributes Type
	-	-
Methods	Method Name	Description
	-	-
Algorithm	1. Start 2. Create a form that uses the HTTP method ‘POST” and has an action attribute that points to a route that maps to the “store” method of the AnnouncementController. 3. Include input fields such as announcement title and description inside the form. 4. Add a CSRF token field to the form, using the “@csrf” directive. 5. Add <<Save>> button to trigger form submission. 6. End	

4.6.1.3 EditAnnouncementForm [PMMS-SDD-REQ-803]

Class Type	Boundary class	
Responsibility	An interface for admin to edit an announcement.	
Attributes	Attributes Name	Attributes Type
	-	-
Methods	Method Name	Description
	-	-
Algorithm	1. Start 2. Create a form that uses the HTTP method ‘POST” and has an action attribute that points to a route that maps to the “update” method of the AnnouncementController. 3. Include input fields such as announcement title and description inside the form 4. Add a CSRF token field to the form, using the “@csrf” directive. 5. Add <<Save>> button to trigger form submission. 6. End	

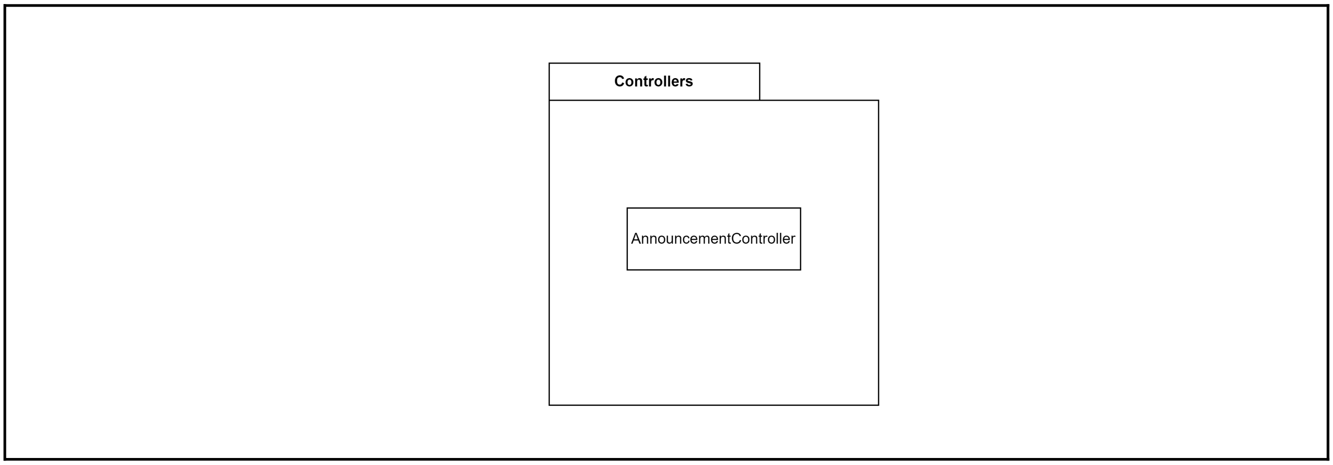
4.6.1.4 CommitteeAnnouncementListingPage [PMMS-SDD-REQ-804]

Class Type	Boundary class	
Responsibility	An interface to display the list of announcements for the PETAKOM Committee, Secretariat and the PETAKOM Coordinator.	
Attributes	Attributes Name	Attributes Type
	-	-
Methods	Method Name	Description
	-	-
Algorithm	1. Start 2. Create a table to display the list of announcements, including columns for the announcement title. 3. Use @foreach to iterate over the list of announcements and display each one in the appropriate row and column of the table For each announcement row, add <<View>> button and once click the button, send a GET request to the "view-announcement/{id}" route to display the AnnouncementDetailPage. End	

4.6.1.5 AnnouncementDetailPage [PMMS-SDD-REQ-805]

Class Type	Boundary class	
Responsibility	An interface to display the selected announcement details for the PETAKOM Committee, Secretariat and the PETAKOM Coordinator.	
Attributes	Attributes Name	Attributes Type
	-	-
Methods	Method Name	Description
	-	-
Algorithm	1. Start 2. Display the announcement title in a header tag for the given announcement ID. 3. Display the announcement description in paragraph tag for the given announcement ID. 4. End	

4.6.2 Controllers [PMMS-SDD-REQ-200]

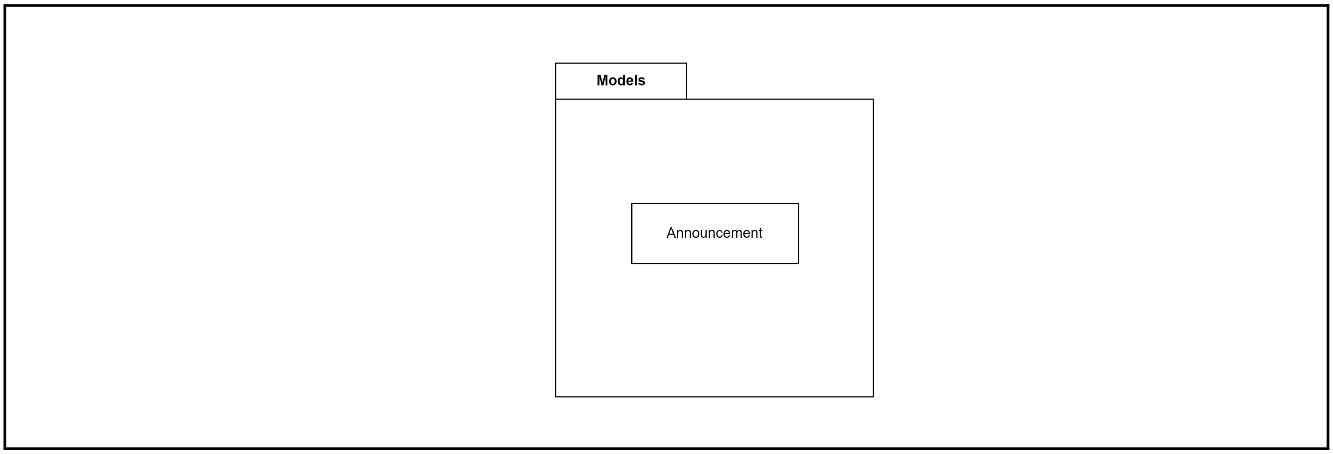


4.6.2.1 AnnouncementController [PMMS-SDD-REQ-206]

Class Type	Controller class	
Responsibility	Controller class that is responsible to manage the flow of manage announcement module, including receiving input from CreateAnnouncementForm or EditAnnouncementForm user interface, validating user input, interacting with Announcement Model to create or update announcement, and returning appropriate responses to the user interface based on the success or failure of the creating and editing announcement process.	
Attributes	Attributes Name	Attributes Type
	-	-
Methods	Method Name	Description
	index()	To retrieve all announcements from the database and return them to the “AdminAnnouncementListingPage” view for display.
	create()	To return a blank “CreateAnnouncementForm” view for the user to fill out and submit.
	store()	To receive the form data from the “CreateAnnouncementForm” view and save a new announcement to the database.
	show()	To retrieve an announcement by ID from the database and return it to the “AnnouncementDetailPage” view for display.

	edit()	To retrieve an announcement by ID from the database and return it to the “EditAnnouncementForm” view for the user to edit and submit.
	update()	To receive the edited form data from the “EditAnnouncementForm” view, validate the data, and update the corresponding announcement in the database.
	indexCommittee Announcement()	To retrieve all announcements from the database and return them to the “CommitteeAnnouncementListingPage” view for display.
	destroy()	To receive a request containing the ID of an announcement to delete, delete the corresponding announcement from the database, and redirect the user back to the “AdminAnnouncementListingPage” view.
Algorithm	1. Start 2. Create a “index()” method. 3. Create a “create()” method. 4. Create a “store()” method. 5. Create a “show()” method. 6. Create a “edit()” method. 7. Create a “update()” method. 8. Create a “indexCommitteeAnnouncement()” method. 9. Create a “destroy()” method. 10. End	

4.6.3 Models [PMMS-SDD-REQ-300]



4.6.3.1 Announcement [PMMS-SDD-REQ-309]

Class Type	Model class	
Responsibility	Model class that is responsible to store and retrieve announcement data in the database.	
Attributes	Attributes Name	Attributes Type
	announcement_id title description announcement_created_at	Integer String String TimeStamps
Methods	Method Name	Description
	-	-
Algorithm	1. Start 2. Announcement is defined as a class that extends the “Model” class. 3. Define “$fillable” property that lists the fields such as announcement_id, title, description and announcement_created_at. 4. End	

5. REQUIREMENT TRACEABILITY

5.1 Manage User Registration [PMMS-SRS-REQ-100] (CB20068 - TEE JIA MEI)

Requirement ID	Description	Design ID
PMMS-SRS-REQ-100	**Manage User Registration** Allow admin to view user listing in list form and manage user registration.	PMMS-SDD-REQ-101 PMMS-SDD-REQ-102 PMMS-SDD-REQ-103 PMMS-SDD-REQ-201 PMMS-SDD-REQ-301
PMMS-SRS-REQ-101	**Create New User Account** Allow admin to create a new user account for the PETAKOM Committee members and PETAKOM Secretariat and Coordinator.	PMMS-SDD-REQ-102 PMMS-SDD-REQ-201 PMMS-SDD-REQ-301
PMMS-SRS-REQ-102	**Delete User Account** Allow admin to delete selected user account for the PETAKOM Committee members and PETAKOM Secretariat and Coordinator.	PMMS-SDD-REQ-101 PMMS-SDD-REQ-201 PMMS-SDD-REQ-301
PMMS-SRS-REQ-103	**Edit User Account** Allow admin to edit user information for selected user account for the PETAKOM Committee members and PETAKOM Secretariat and Coordinator.	PMMS-SDD-REQ-103 PMMS-SDD-REQ-201 PMMS-SDD-REQ-301
PMMS-SRS-REQ-104	**Incomplete or Invalid User Information** Handle the exceptional scenario in which the admin fills in incomplete or invalid user information.	PMMS-SDD-REQ-102 PMMS-SDD-REQ-201 PMMS-SDD-REQ-301
PMMS-SRS-REQ-105	**Complete and Valid User Information** Must fill in all the text fields and the password must be at least 12 characters long with a combination of uppercase and numbers.	PMMS-SDD-REQ-102 PMMS-SDD-REQ-201 PMMS-SDD-REQ-301

5.2 Manage Inventory [PMMS-SRS-REQ-200] (CB20121 - TAN TENG XING)

Requirement ID	Description	Design ID
PMMS-SRS-REQ-200	**Manage Inventory** Provide interaction between system and users to manage inventory, including create, view, edit, delete items, and update daily inventory status.	PMMS-SDD-REQ-401 PMMS-SDD-REQ-402 PMMS-SDD-REQ-403 PMMS-SDD-REQ-404 PMMS-SDD-REQ-202 PMMS-SDD-REQ-203 PMMS-SDD-REQ-302
PMMS-SRS-REQ-201	**Low inventory alert** Alert users about the critical low level status of inventory.	PMMS-SDD-REQ-202 PMMS-SDD-REQ-302 PMMS-SDD-REQ-303
PMMS-SRS-REQ-202	**Failed to save in the database** Failure on saving in database after an operation is executed. Then, system displays error message.	PMMS-SDD-REQ-302 PMMS-SDD-REQ-203
PMMS-SRS-REQ-203	**User permission** Verify the authenticity of users to ensure they own the right permission to view or modify the data.	PMMS-SDD-REQ-202 PMMS-SDD-REQ-302 PMMS-SDD-REQ-303
PMMS-SRS-REQ-204	**Inventory value** Constraints that are related to inventory data. It validates the non-negative value, number of pictures allowed to be uploaded, and the file size.	PMMS-SDD-REQ-402 PMMS-SDD-REQ-403 PMMS-SDD-REQ-404 PMMS-SDD-REQ-302 PMMS-SDD-REQ-303

5.3 Manage Payment [PMMS-SRS-REQ-300] (CB20114 - LOO CHANG HERNG)

Requirement ID	Description	Design ID
PMMS-SRS-REQ-300	**Manage Payment** Allow cashier to select item and proceed to payment. After payment, the system display a receipt for customer.	PMMS-SDD-REQ-302 PMMS-SDD-REQ-303 PMMS-SDD-REQ-501 PMMS-SDD-REQ-502 PMMS-SDD-REQ-503 PMMS-SDD-REQ-203 PMMS-SDD-REQ-304 PMMS-SDD-REQ-305
PMMS-SRS-REQ-301	**Edit Item Quantity** Allow admin to edit the item quantity of items.	PMMS-SDD-REQ-501 PMMS-SDD-REQ-502 PMMS-SDD-REQ-203
PMMS-SRS-REQ-302	**Cancel Payment** Allow admin to cancel the payment and return to the main page.	PMMS-SDD-REQ-501 PMMS-SDD-REQ-502 PMMS-SDD-REQ-203
PMMS-SRS-REQ-303	**Maximum Quantity of Item Selected** Restrict the quantity of item selected when it reaches the maximum number of unit in inventory.	PMMS-SDD-REQ-501 PMMS-SDD-REQ-203 PMMS-SDD-REQ-303

5.4 Manage Duty Roster [PMMS-SRS-REQ-400] (CB20064 - TAN KHA HUI)

Requirement ID	Description	Design ID
PMMS-SRS-REQ-400	**Manage Duty Roster** Allow the admin to view, create, edit, and delete duty roster and allow the PETAKOM Committee and secretariat to add or delete time slots in his/her duty schedule while PETAKOM coordinator is allowed to view the duty roster.	PMMS-SDD-REQ-601 PMMS-SDD-REQ-602 PMMS-SDD-REQ-603 PMMS-SDD-REQ-604 PMMS-SDD-REQ-605 PMMS-SDD-REQ-606 PMMS-SDD-REQ-607 PMMS-SDD-REQ-204 PMMS-SDD-REQ-306 PMMS-SDD-REQ-307 PMMS-SDD-REQ-308
PMMS-SRS-REQ-401	**Create New Duty Roster** Allow admin to create a new duty roster.	PMMS-SDD-REQ-604 PMMS-SDD-REQ-204 PMMS-SDD-REQ-306 PMMS-SDD-REQ-307
PMMS-SRS-REQ-402	**Edit Duty Roster** Allow admin to edit the selected duty roster.	PMMS-SDD-REQ-605 PMMS-SDD-REQ-204 PMMS-SDD-REQ-306 PMMS-SDD-REQ-307
PMMS-SRS-REQ-403	**Delete Duty Roster** Allow admin to delete the selected duty roster.	PMMS-SDD-REQ-601 PMMS-SDD-REQ-204 PMMS-SDD-REQ-306 PMMS-SDD-REQ-307
PMMS-SRS-REQ-404	**Add Time Slot** Allow the PETAKOM Committee to add time slots to his/her duty schedule.	PMMS-SDD-REQ-606 PMMS-SDD-REQ-204 PMMS-SDD-REQ-308
PMMS-SRS-REQ-405	**View Own Schedule** Allow the PETAKOM Committee to view his/her duty schedule with the added time slots.	PMMS-SDD-REQ-607 PMMS-SDD-REQ-204 PMMS-SDD-REQ-308
PMMS-SRS-REQ-406	**Drop Time Slot** Allow the PETAKOM Committee to drop the selected time slot in his/her duty schedule.	PMMS-SDD-REQ-607 PMMS-SDD-REQ-204 PMMS-SDD-REQ-308
PMMS-SRS-REQ-407	**Cancel Action** Provide the ability for the admin and PETAKOM	PMMS-SDD-REQ-601 PMMS-SDD-REQ-607 PMMS-SDD-REQ-204

	Committee to cancel his/her action such as delete or drop the duty roster.	
PMMS-SRS-REQ-408	**Unavailable Selected Time Slot** Exception when the selected time slot is either full of quota or it has been already added in the user's duty schedule.	PMMS-SDD-REQ-606 PMMS-SDD-REQ-204
PMMS-SRS-REQ-409	**Failed to Save In the Database** Exception when the system is unable to store the data in the database.	PMMS-SDD-REQ-606 PMMS-SDD-REQ-204
PMMS-SRS-REQ-410	**Maximum Quota For Each Time Slot** Constraint to restrict the maximum quota for every time slot.	PMMS-SDD-REQ-606 PMMS-SDD-REQ-204 PMMS-SDD-REQ-308
PMMS-SRS-REQ-411	**Forbidden Repeated Account in One Time Slot** Constraint to avoid one time slot owned by the same account.	PMMS-SDD-REQ-606 PMMS-SDD-REQ-204 PMMS-SDD-REQ-308

5.5 Manage Report [PMMS-SRS-REQ-500] (CB20114 - LOO CHANG HERNG)

Requirement ID	Description	Design ID
PMMS-SRS-REQ-500	**Manage Report** Allow admin, PETAKOM Secretariat and Coordinator to view the report of inventory and profit.	PMMS-SDD-REQ-302 PMMS-SDD-REQ-303 PMMS-SDD-REQ-304 PMMS-SDD-REQ-305 PMMS-SDD-REQ-701 PMMS-SDD-REQ-205

5.6 Manage Announcement [PMMS-SRS-REQ-600] (CB20068 - TEE JIA MEI)

Requirement ID	Description	Design ID
PMMS-SRS-REQ-600	**Manage Announcement** Allow admin to manage user registration and allow PETAKOM Committee Secretariat and PETAKOM Coordinator to view announcements details.	PMMS-SDD-REQ-801 PMMS-SDD-REQ-802 PMMS-SDD-REQ-803 PMMS-SDD-REQ-804 PMMS-SDD-REQ-805 PMMS-SDD-REQ-206 PMMS-SDD-REQ-309
PMMS-SRS-REQ-601	**Create New Announcement** Allow admin to create a new announcement.	PMMS-SDD-REQ-802 PMMS-SDD-REQ-206 PMMS-SDD-REQ-309
PMMS-SRS-REQ-602	**Delete Announcement** Allow admin to delete selected announcement.	PMMS-SDD-REQ-801 PMMS-SDD-REQ-206 PMMS-SDD-REQ-309
PMMS-SRS-REQ-603	**Edit Announcement** Allow admin to edit selected announcement information.	PMMS-SDD-REQ-803 PMMS-SDD-REQ-206 PMMS-SDD-REQ-309
PMMS-SRS-REQ-604	**Incomplete or Invalid Announcement Information** Handle the exceptional scenario in which the admin fills in incomplete or invalid announcement information.	PMMS-SDD-REQ-802 PMMS-SDD-REQ-206 PMMS-SDD-REQ-309
PMMS-SRS-REQ-605	**Complete and Valid Announcement Information** Must fill in both title and description in the text fields.	PMMS-SDD-REQ-802 PMMS-SDD-REQ-206 PMMS-SDD-REQ-309