



Universiti Malaysia PAHANG

Engineering • Technology • Creativity

SDD Document SEM I 20222023

PETAKOM Mart Management System

B.o.B (Best of the Best) developer:

- MUHAMMAD NAFIZ BIN MANSOR CB20067
- SITI ZULAIKHA BINTI ABDUL MANAP CB20097
- NUR AINUL HAYATI BINTI MOHD SUKOR CB20090
- SITI NAJIAH BINTI DAUD CB20120



TABLE OF CONTENTS

Chapter	Title	Page
	TABLE OF CONTENTS	
1	INTRODUCTION	
	1.1 Purpose	
	1.2 System Identification	
	1.3 System Overview	
	1.4 Module Overview	
	1.5 Document Overview	
2	REFERENCE DOCUMENTS	
3	ARCHITECTURE DESCRIPTION	
	3.1 General Architecture	

1. INTRODUCTION

1.1 Purpose

The purpose of writing this document is to make the process of analysing requirement specifications more convenient as it is a comprehensive description of the system to be developed. All of the required specifications can be simply analysed, and we can spot some of the confusing requirements and try to re-elicite them from the relevant stakeholder by using the SRS (Software Requirement Specifications) formats. This document works as a guideline for the team before proceeding into the system design phase. The paper is designed for both stakeholders and system developers and will be submitted for approval.

1.2 System Identification

This Software Design Document (SDD) belongs to the “PETAKOM Mart Management System” (PETAKOM MMS).

System Title	FKOM E-Donation
System Number	PETAKOM-MMS-2023-v1.0
System Name	PETAKOM Mart Management System
System Abbreviation	PETAKOM MMS

Symbol/Number	Meaning
PETAKOM	It represents the client name which is Persatuan Teknologi Komputer.
B.O.B Developer	It represents the company name which is Best of the Best Developer.
2023	It represents the year where the system has been released.
v1.0	V represent the word “Version” of the system while 1.1 is the general version of the system. The number will increase by 0.1 if there any updates or bugs fixed.

1.3 System Overview

PETAKOM Mart Management System (PETAKOM MM) is an application that is made specifically for people at the Faculty of Computing in University Malaysia Pahang. This application can facilitate the purchase of customers consisting of students and staff. Nevertheless, this application aims to ease the management of the PETAKOM MM System. There are three actors in this management system which are Administrator, Cashier, and Coordinator. PETAKOM MMS consists of five modules including Manage Profile module, Manage Payment module, Manage Duty Rooster, Manage Report, and Manage Inventory.

The PETAKOM MM System should:

1. Manage inventory. The actor involved in this module is the admin. This module is responsible for managing anything related to the product. Among that, which include the admin that can add new items to the PETAKOM MM in addition by being able to see the list and product details, which must be filled in by the admin.
2. Manage profile is the module that is designed for the admin to create a new profile for the PETAKOM MM members by adding the information. Among the information that needs to be filled in in order to create a new profile include first name, last name, role, course, email, and contact number. In addition, the admin can also edit profile information.
3. Manage report module is to generate reports for the sales of the product that is sorted by daily, weekly, monthly, and yearly. The report should be visualized by the motion graphics such as a graph statistic and table data which also displays the item's name, the quantity of the item sold, the total profit, the total investment, and the net price.
4. Manage payment module is the module that needs to be managed by admin and cashier when the user performs payment after they made purchases of the product in the system. The system will automatically add the list of the payments that need to be settled by the customer. The payment available is via cash and QR Pay.

5. Manage duty roster is the module that manages the duty schedule. The system provides a calendar which the admin can play a role by noting the duty on the day that they wanted. Admin can add schedule time by adding the day and date for start and end time, and also the person in charge to be on duty that day. Besides, the admin is also able to update and delete the duty schedule.

1.4 Module Overview

There are five modules for PETAKOM MM system which is

I. Manage Profile

The Manage Profile module allows admin to manage PETAKOM's member profiles by adding new member's profile, editing existing profiles, viewing member information and deleting member profiles. This module also allows cashiers to view their profile. With this module, it will ensure accurate and up-to-date member information, and reduce the risk of security issues.

II. Manage Payment

The Manage Payment module is a module that allows both admin and cashier to manage the payment process. In this module, the system allows cashiers to add new items, edit item quantity and delete items. When a customer makes a purchase, the system automatically adds the payment details to the list of payments that need to be settled. To complete the payments, customers was giving options to pay using cash to cashier at PETAKOM counter or via QR Pay. This module also allows the admin to view payment details.

III. Manage Duty Roster

The Manage Duty Roster module allows admin to manage PETAKOM members schedules. This module enables admin to add new duty rosters for members, edit existing duty rosters, delete outdated or incorrect duty rosters. It also allows admin to view the duty rosters for all PETAKOM members and cashiers to view their duty roster.

IV. Manage Report

The Manage Report module allows the admin to generate sales statistics for products. The module provides valuable insights into the sales performance with reports sorted by daily, weekly, monthly, and yearly sales. The module also allows coordinators to view all recorded reports and each report displays essential details such as the item's name, quantity sold, total profit, total investment, and net price.

V. Manage Inventory

The Manage Inventory is designed to manage products in the PETAKOM MM system and keep track of stock levels. This module allows admin to add new products to the system, view and update product details, and delete selected products. The system also allows the admin to view notification messages related to inventory management, keeping them informed about product stock levels.

1.5 Document Overview

This subsection describes the contents and organizations of this document.

Chapter 1: This section provides an introduction to the system and an overview of the document, including several subsections such as identification, system overview, module overview, and document overview. The identification section provides information about the document, while the system overview explains the purpose and scope of the system. The module overview briefly describes each module in the project and how they work together. The document overview gives an overview of the document's contents chapter by chapter.

Chapter 2: This section lists all references used in order to produce this document.

Chapter 3: This section provides the architecture descriptions. It consists of three sections which are general architecture, package module and data dictionary. In the general architecture section, it provides an explanation about the general architecture definition. It is also divided into three subsections which are application, business services and also middleware.

Chapter 4: This section provides the detailed design of the project. In this section, all modules related to the project will be represented in a detailed way.

Chapter 5: This section provides the requirements traceability allocated in the SRS document as the source of the requirement.

2. REFERENCE DOCUMENTS

This section lists references used in producing this document:

[1] SRS_B.o.B_PETAKOM_MMS.doc

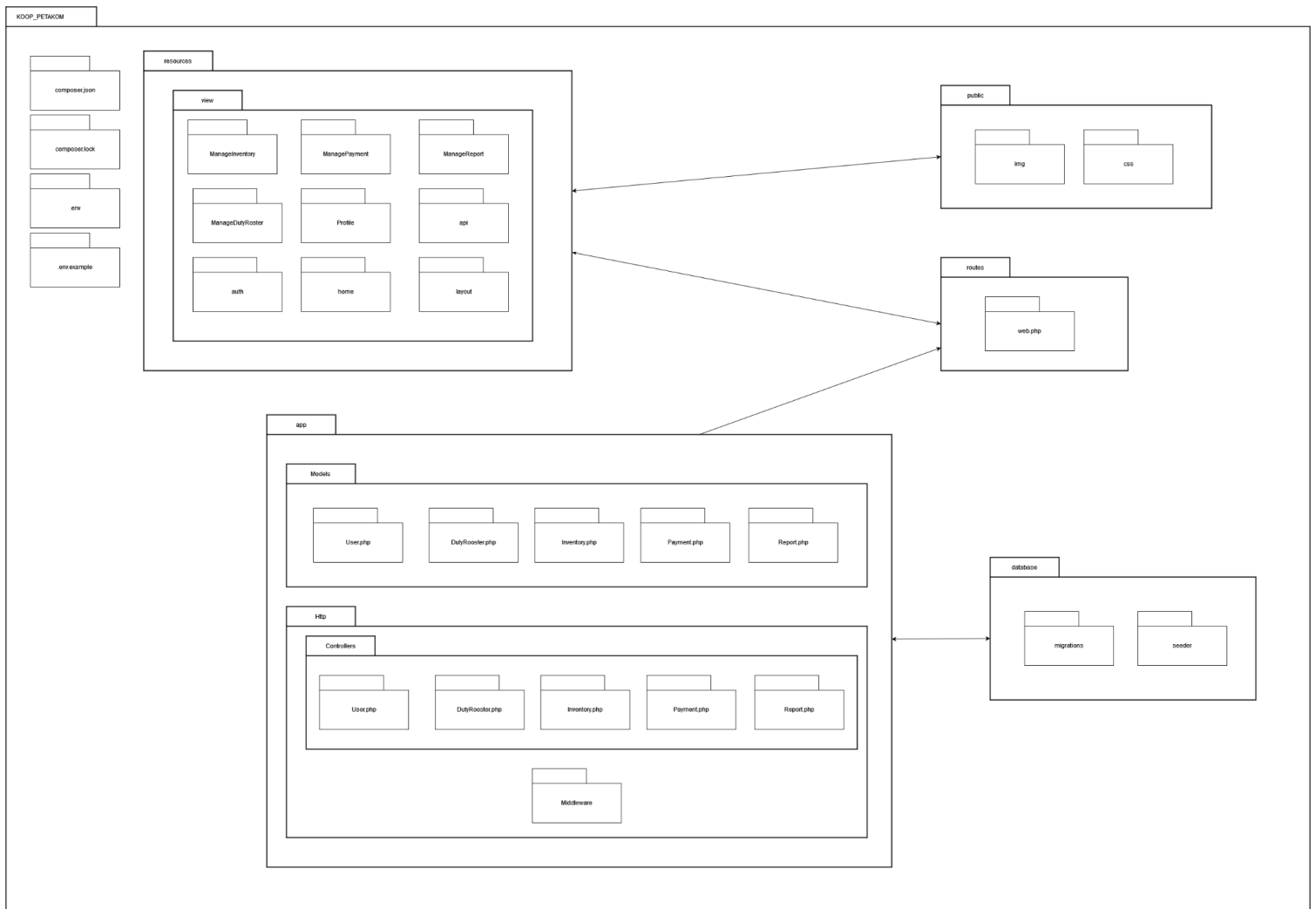
Software Requirement Specification of PETAKOM MMS

[2] IEEE 1016-2009

IEEE Standard for Information Technology--Systems Design--Software Design
Descriptions

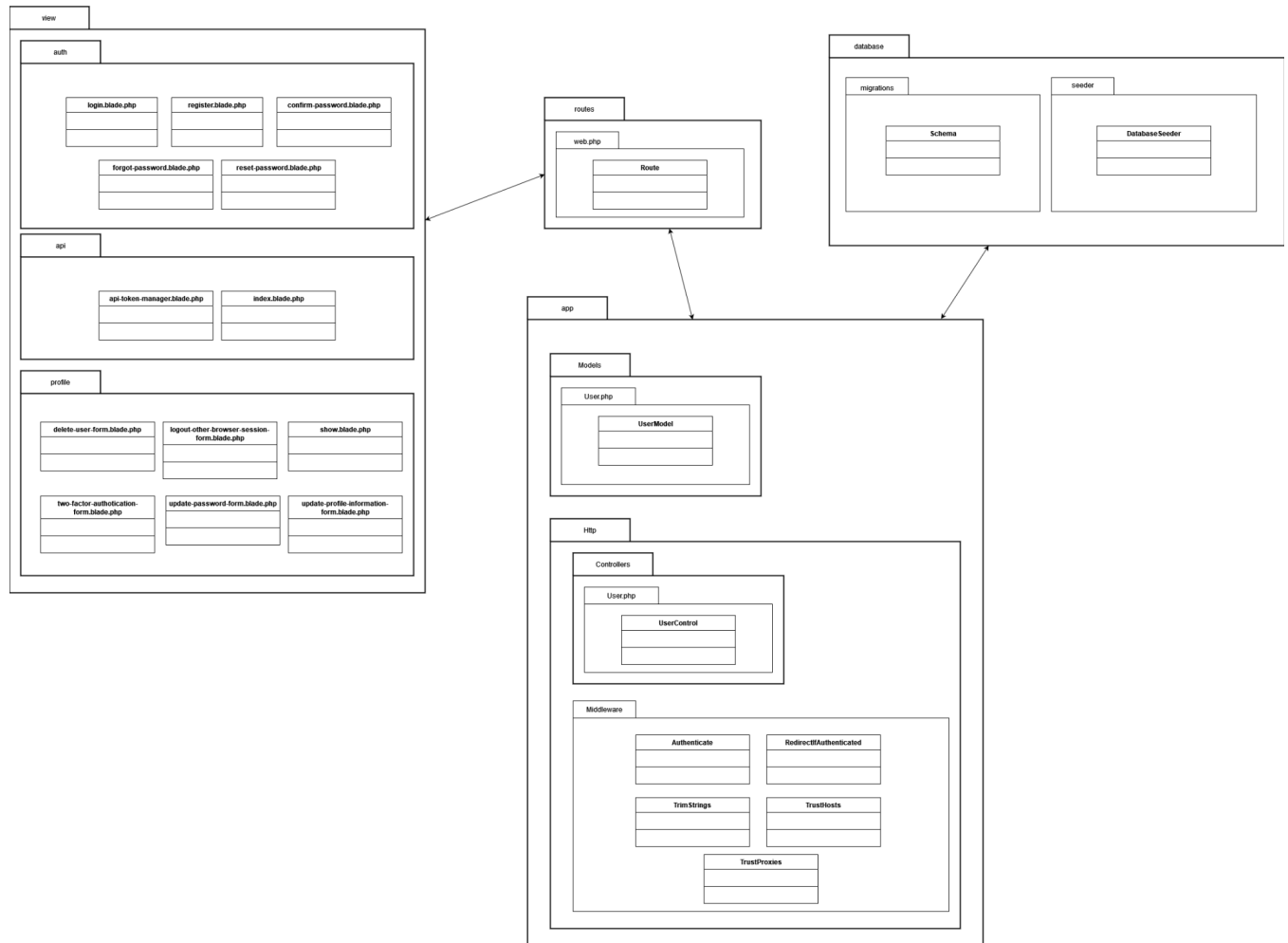
3. ARCHITECTURE DESCRIPTION

3.1 General Architecture



3.1.1 Application Layer

Module 1 (Manage Profile)



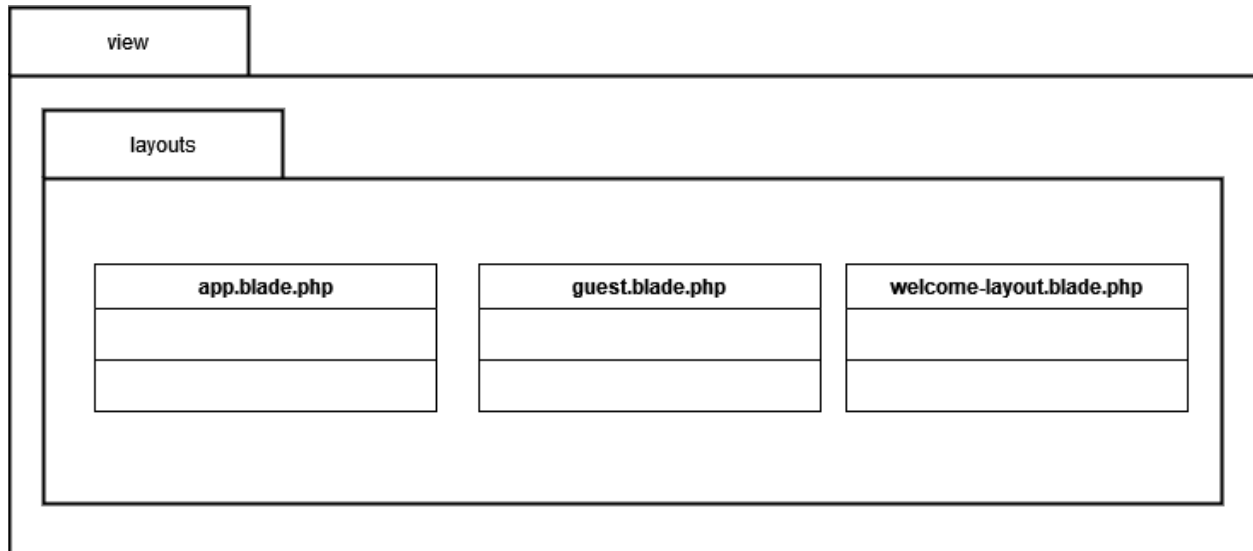
3.1.1.1 Package Authentication: Package auth [SDD-REQ-100-100]

Class Name	Description
login.blade.php [SDD-REQ-100-101]	A class where the system displays the login page for users (Admin, Cashier and Coordinator). The coordinator just clicks on the hyperlink to login to the system.
register.blade.php [SDD-REQ-100-102]	A class where the system displays the

	registration form for the user (Admin).
confirm-password.blade.php [SDD-REQ-100-103]	A class where the system displays the password confirmation page for the user to confirm their password before allowing them to perform any activities.
forgot-password.blade.php [SDD-REQ-100-104]	A class where the system allows the user to reset their passwords if they have forgotten them.
reset-password.blade.php [SDD-REQ-100-105]	A class where the system displays the reset password page where Admin and Cashier can reset their password.
api-token-manager.blade.php [SDD-REQ-100-106]	A class which provides the functionality for managing API tokens associated with the user's profile.
index.blade.php [SDD-REQ-100-107]	A class where the system displays user's profile information.
delete -user-form.blade.php [SDD-REQ-100-108]	A class where the system displays the delete page where Admin can delete any user profile.
logout-other -browser-session-form.blade.php [SDD-REQ-100-109]	A class which provides the functionality for users to logout from all other active browser sessions beside the current one.
show.blade.php [SDD-REQ-100-110]	A class which provides the functionality for displaying data to the user.
two-factor-authentication-form.blade.php [SDD-REQ-100-111]	A class where the system displays the two-factor authentication form to the user.
update-password-form.blade.php [SDD-REQ-100-112]	This interface allows Admin and Cashier to update their password.
update-profile-information-form.blade.php [SDD-REQ-100-113]	A class where the system displays the profile page which Admin can update the profile for all users.

Route [SDD-REQ-100-114]	A class where the system allows developers to define the profile's routes, which determine how the profile file responds to incoming HTTP requests.
UserModel [SDD-REQ-100-115]	A class where the system allows
UserController [SDD-REQ-100-116]	The controller is used to manage all user profile information across the system.
Authenticate [SDD-REQ-100-117]	A class where the system ensures that only authenticated users can access protected resources within the application.
RedirectIfAuthenticated [SDD-REQ-100-118]	A class where the system provides the functionality for redirecting authenticated users to a specific location.
TrimStrings [SDD-REQ-100-119]	A middleware class where the system trimming whitespace from input strings.
TrustHosts [SDD-REQ-100-120]	A middleware class where the system defines a list of trusted hosts for the application.
TrustProxies [SDD-REQ-100-121]	A middleware class where the system configures trusted proxies for the application.
Schema [SDD-REQ-100-122]	A class where the system defines and modifies database schema.
DatabaseSeeder [SDD-REQ-100-123]	To store all data entered by Admin related to user profile such as name, role, course, email, phone number, last login record and last logout record.

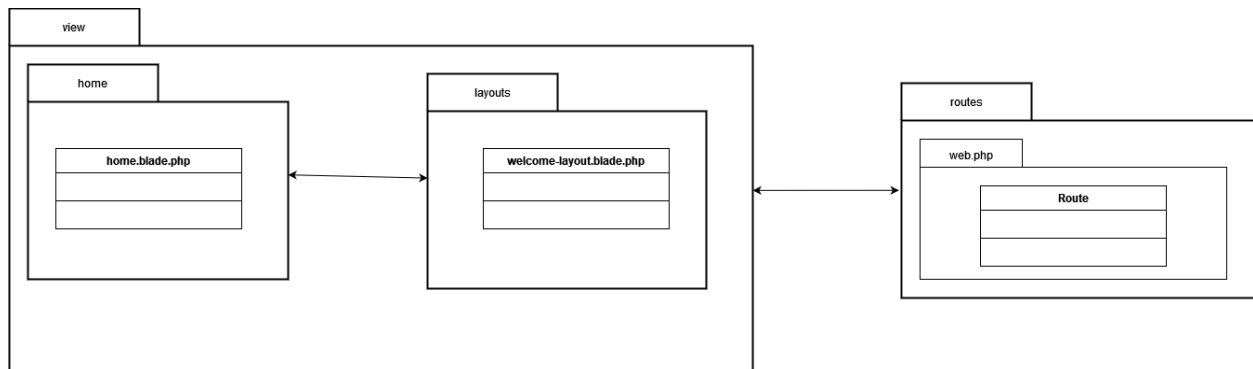
Module 2 (layout)



3.1.1.2 Package Layout: Package layouts [SDD-REQ-200-100]

Class Name	Description
app.blade.php [SDD-REQ-200-101]	A class where the system displays the constant navigation bar by using Jetstream live wire.
guest.blade.php [SDD-REQ-200-102]	A class where the system displays the constant navigation bar by using Jetstream live wire.
welcome-layout.blade.php [SDD-REQ-200-103]	A class where the system displays the constant navigation bar for the dashboard PETAKOM Mart website.

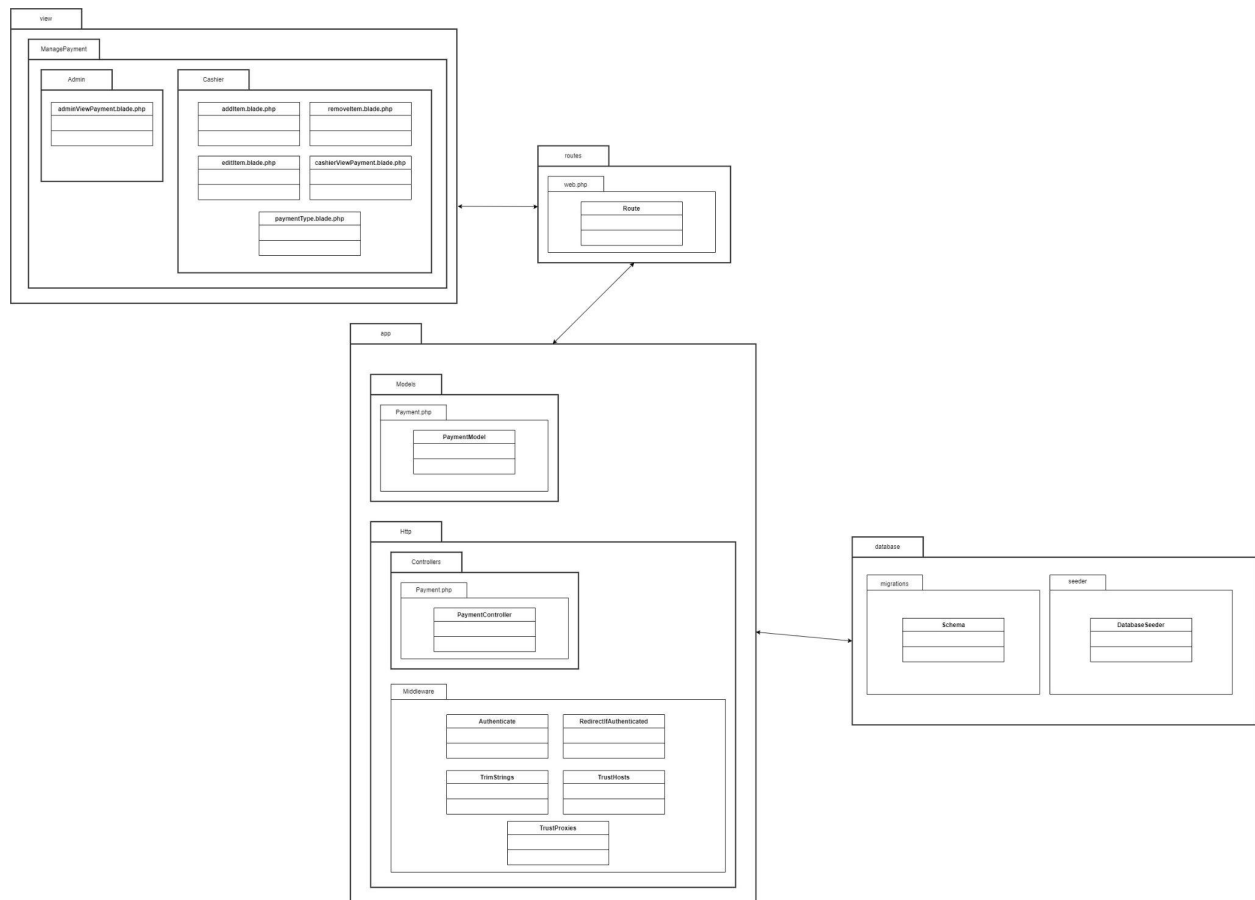
Module 3 (home)



3.1.1.3 Package Dashboard: Package home [SDD-REQ-300-100]

Class Name	Description
home.blade.php [SDD-REQ-300-101]	A class where the system contains the layout of the dashboard PETAKOm Mart to display the interface.
welcome-layout.blade [SDD-REQ-300-102]	A class where the system displays the constant navigation bar for the dashboard PETAKOM Mart website.
Route [SDD-REQ-300-103]	A class where the system allows developers to define application routes, which determine how the application responds to incoming HTTP requests.

Module 4 (Manage Payment)

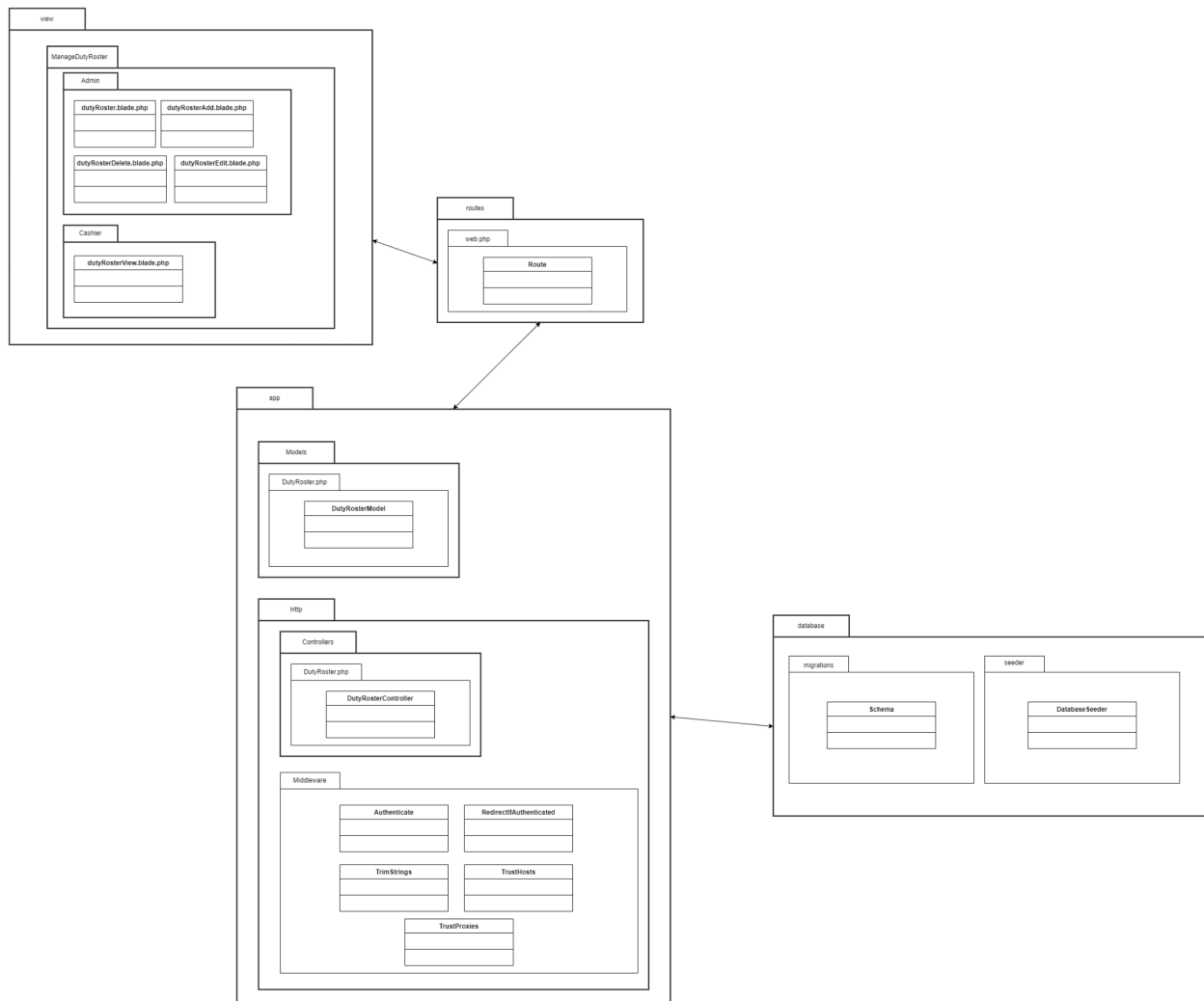


3.1.1.4 Package Manage Payment: Package ManagePayment [SDD-REQ-400-100]

Class Name	Description
payment.blade.php [SDD-REQ-400-101]	A boundary class where the system contains a layout that displays the payment for the cashier of PETAKOM Mart.
Route [SDD-REQ-400-102]	A route class where the system allows developers to define the payment file's routes, which determine how the payment file responds to incoming HTTP requests.

PaymentModel [SDD-REQ-400-103]	An entity class where the system allows developers to interact with the data from a database table.
PaymentController [SDD-REQ-400-104]	A controller class where the system allows developers to control the application's HTTP requests and handle the corresponding responses from the application's backend.
Authenticate [SDD-REQ-400-105]	A middleware class where the system ensures that only authenticated users can access protected resources within the application.
RedirectIfAuthenticated [SDD-REQ-400-106]	A middleware class where the system redirects authenticated users away from the routes that should be accessible only to guests or unauthenticated users.
TrimStrings [SDD-REQ-400-107]	A class where the system trims whitespace from the input data of incoming requests. It is a middleware class that applies to all incoming HTTP requests to the application.
TrustHosts [SDD-REQ-400-108]	A class where the system defines a list of trusted hosts for your application.
TrustProxies [SDD-REQ-400-109]	A class where the system configures trusted proxies for your application.
Schema [SDD-REQ-400-110]	A blueprint class where the system defines and modifies database schema.
DatabaseSeeder [SDD-REQ-400-111]	A seeder class where the system inserts data into database tables to provide a consistent set of data for testing purposes.

Module 5 (Manage Duty Roster)



3.1.1.5 Package Manage Duty Roster: Package ManageDutyRoster

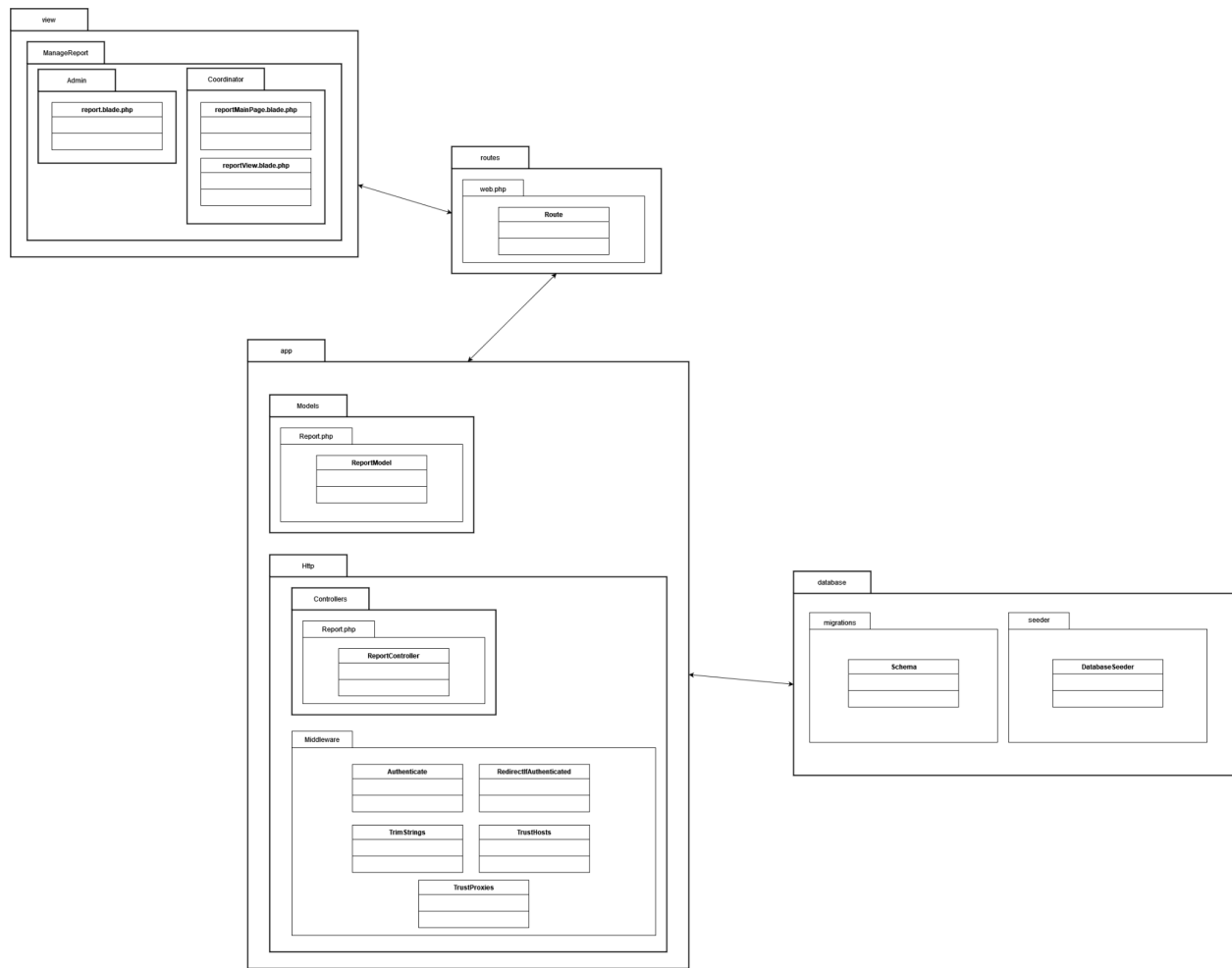
[SDD-REQ-500-100]

Class Name	Description
dutyRoster.blade.php [SDD-REQ-500-101]	A class where the system contains the layout that displays the calendar view of the duty schedule for the Admin and Cashier.
dutyRosterAdd.blade.php	A class where the system displays the

[SDD-REQ-500-102]	duty roster form for Admin to add new schedule duty for Cashiers.
dutyRosterEdit.blade.php [SDD-REQ-500-103]	A class where the system displays the existing duty roster form for Admin to edit the schedule.
dutyRosterDelete.blade.php [SDD-REQ-500-104]	A class where the system displays the delete page for the duty schedule where Admin can delete any existing duty schedule.
dutyRosterView.blade.php [SDD-REQ-500-105]	A class where the system displays the duty schedule page for Cashier's view.
Route [SDD-REQ-500-106]	A class where the system allows the developers to define the duty roster file's routes, which determine how the duty roster file responds to incoming HTTP requests.
DutyRosterModel [SDD-REQ-500-107]	A class where the system allows developers to interact with the data from a database table.
DutyRosterController [SDD-REQ-500-108]	The controller is used to manage all duty schedules across the system.
Authenticate [SDD-REQ-500-109]	A class where the system ensures that only authenticated users can access protected resources within the application.
RedirectIfAuthenticated [SDD-REQ-500-110]	A class where the system provides the functionality for redirecting authenticated users to a specific location.
TrimStrings [SDD-REQ-500-111]	A middleware class where the system trimming whitespace from input strings.
TrustHosts [SDD-REQ-500-112]	A middleware class where the system defines a list of trusted hosts for the

	application.
TrustProxies [SDD-REQ-500-113]	A middleware class where the system configures trusted proxies for the application.
Schema [SDD-REQ-500-114]	A class where the system defines and modifies database schema.
DatabaseSeeder [SDD-REQ-500-115]	To store all data entered by Admin related to duty roster information such as time start, time end, date and person in charge.

Module 6 (Manage Report)



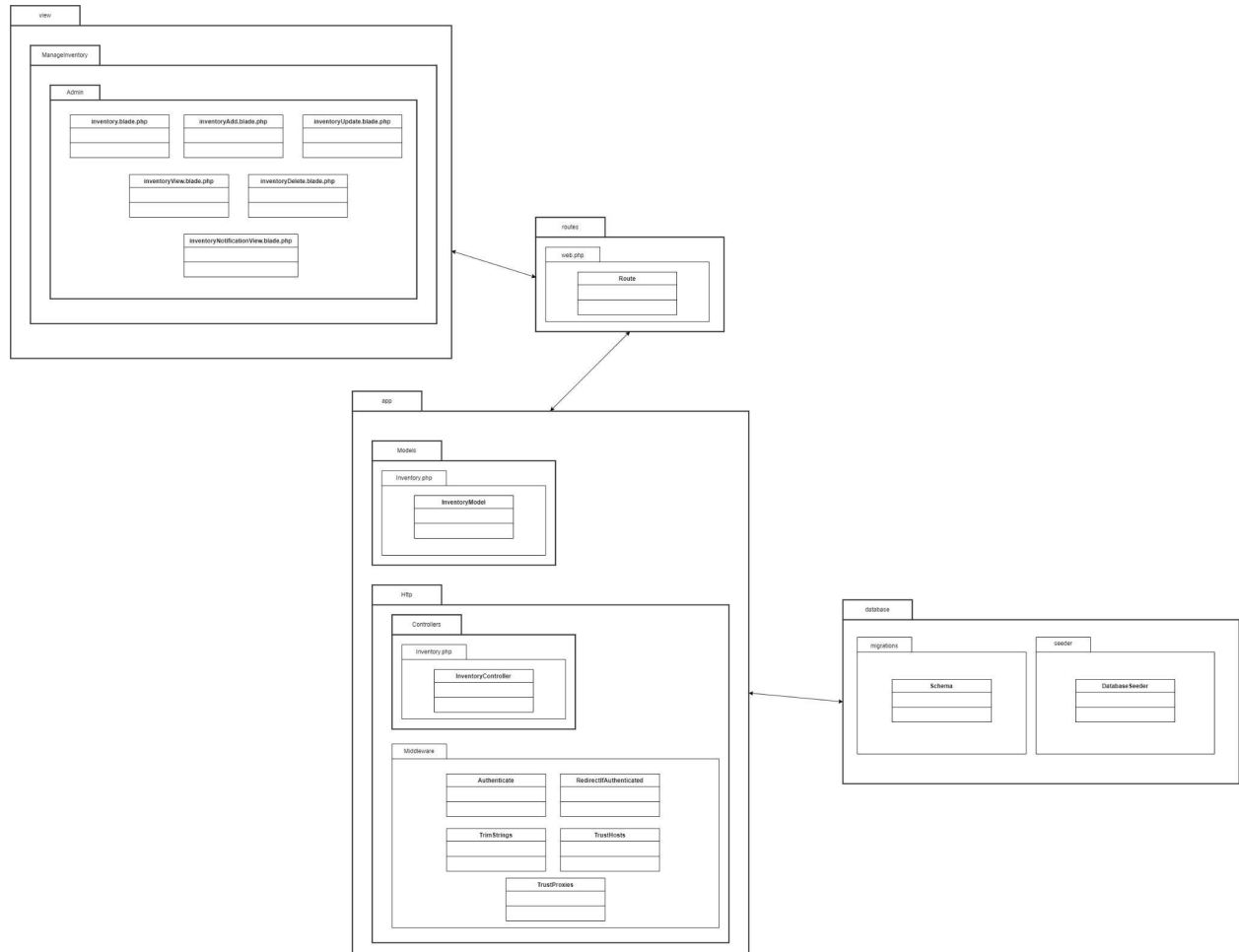
3.1.1.6 Package Manage Report: Package ManageReport [SDD-REQ-600-100]

Class Name	Description
report.blade.php [SDD-REQ-600-101]	A boundary class where the system contains the layout that displays the list of reports PETAKOM Mart for the Admin user.
reportMainPage.blade.php [SDD-REQ-600-102]	A boundary class where the system contains the layouts that display the Main

	Interface of the Report for the Coordinator user.
reportView.blade.php [SDD-REQ-600-103]	A boundary class where the system contains the layout that displays the detail of the report PETAKOM Mart.
Route [SDD-REQ-600-102]	A route class where the system allows developers to define the report file's routes, which determine how the report file responds to incoming HTTP requests.
ReportModel [SDD-REQ-600-103]	An entity class where the system allows developers to interact with the data from a database table.
ReportController [SDD-REQ-600-104]	A controller class where the system allows developers to control the application's HTTP requests and handle the corresponding responses from the application's backend.
Authenticate [SDD-REQ-600-105]	A middleware class where the system ensures that only authenticated users can access protected resources within the application.
RedirectIfAuthenticated [SDD-REQ-600-106]	A middleware class where the system redirects authenticated users away from the routes that should be accessible only to guests or unauthenticated users.
TrimStrings [SDD-REQ-600-107]	A middleware class where the system trims whitespace from the input data of incoming requests. It is a middleware class that applies to all incoming HTTP requests to the application.
TrustHosts [SDD-REQ-600-108]	A middleware class where the system defines a list of trusted hosts for your application.

TrustProxies [SDD-REQ-600-109]	A middleware class where the system configures trusted proxies for your application.
Schema [SDD-REQ-600-110]	A blueprint class where the system defines and modifies database schema.
DatabaseSeeder [SDD-REQ-600-111]	A seeder class where the system inserts data into database tables to provide a consistent set of data for testing purposes.

Module 7 (Manage Inventory)



3.1.1.7 Package Manage Inventory: Package ManageInventory

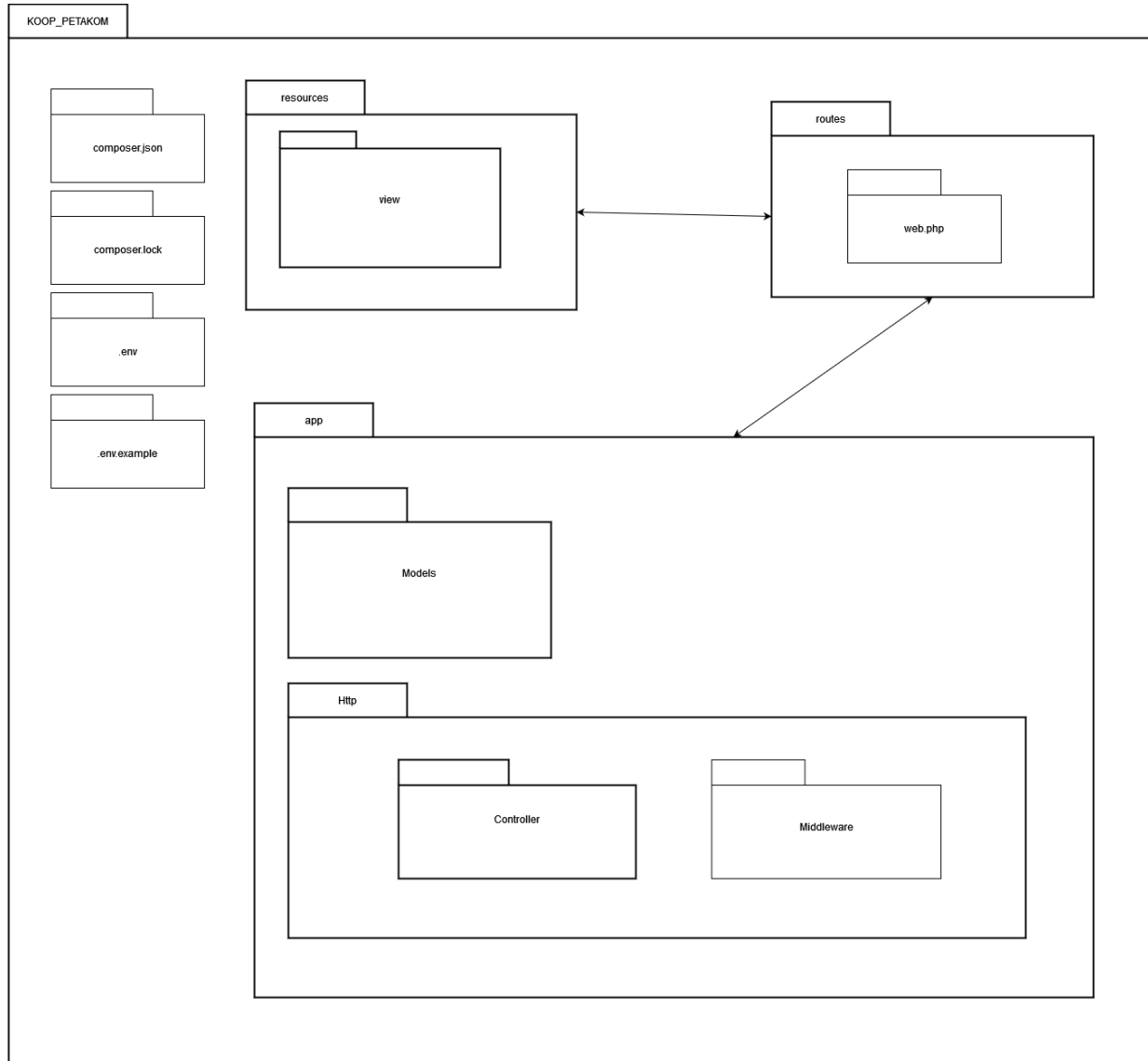
[SDD-REQ-700-100]

Class Name	Description
inventory.blade.php [SDD-REQ-700-101]	A boundary class where the system contains the layouts that display inventory of PETAKOM Mart for Admin.
inventoryAdd.blade.php [SDD-REQ-700-102]	A boundary class where the system contains the layouts that display form to add new product details.

inventoryUpdate.blade.php [SDD-REQ-700-103]	A boundary class where the system contains the layout that displays form to update details of selected products.
inventoryView.blade.php [SDD-REQ-700-104]	A boundary class where the system contains the layout that displays selected product details.
inventoryDelete.blade.php [SDD-REQ-700-105]	A boundary class where the system contains the layout that displays selected products to be deleted.
inventoryNotificationView.blade.php [SDD-REQ-700-106]	A boundary class where the system contains the layout that displays notification messages for inventory.
Route [SDD-REQ-700-107]	A route class where the system allows developers to define the inventory file's routes to determine how the inventory file responds to incoming HTTP requests.
InventoryModel [SDD-REQ-700-108]	An entity class where the system allows developers to store and retrieve inventory data from a database table.
InventoryController [SDD-REQ-700-109]	A controller class where the system allows developers to control the application's HTTP requests and handle the corresponding responses from the application's backend.
Authenticate [SDD-REQ-700-110]	A middleware class where the system ensures that only authenticated users can access protected resources within the application.
RedirectIfAuthenticated [SDD-REQ-700-111]	A middleware class where the system redirects authenticated users away from the routes that are intended for non-authenticated users or guests.
TrimStrings [SDD-REQ-700-112]	A middleware class where the system trims whitespace from the input data of incoming requests. It is a middleware class that applies

	to all incoming HTTP requests to the application.
TrustHosts [SDD-REQ-700-113]	A middleware class where the system helps prevent malicious HTTP Host header attacks. It ensures that only trusted hosts or domains can make requests to manage inventory.
TrustProxies [SDD-REQ-700-114]	A middleware class where the system handles requests made through trusted proxies and uses the correct request information.
Schema [SDD-REQ-700-115]	A database class where the system defines and modifies database schema.
DatabaseSeeder [SDD-REQ-700-116]	A database class where the system inserts data into database tables to provide a consistent set of data for testing purposes.

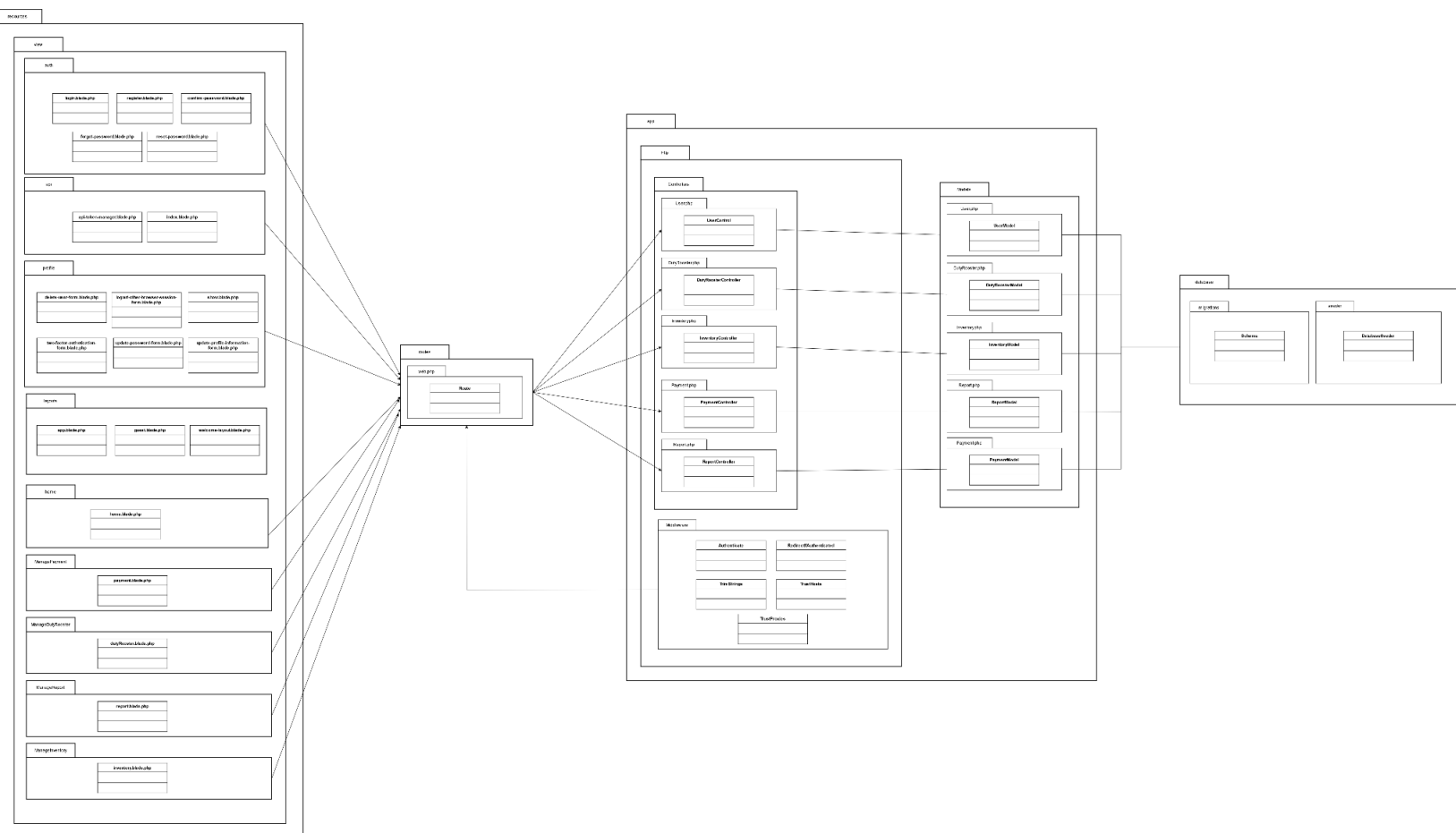
3.1.2 Middleware Layer



Package Name	Description
composer.json	This package is a configuration file used by the Composer package manager to manage the dependencies of a Laravel project.
composer.lock	This package is used to lock down the specific version of each dependency that is installed in the project.

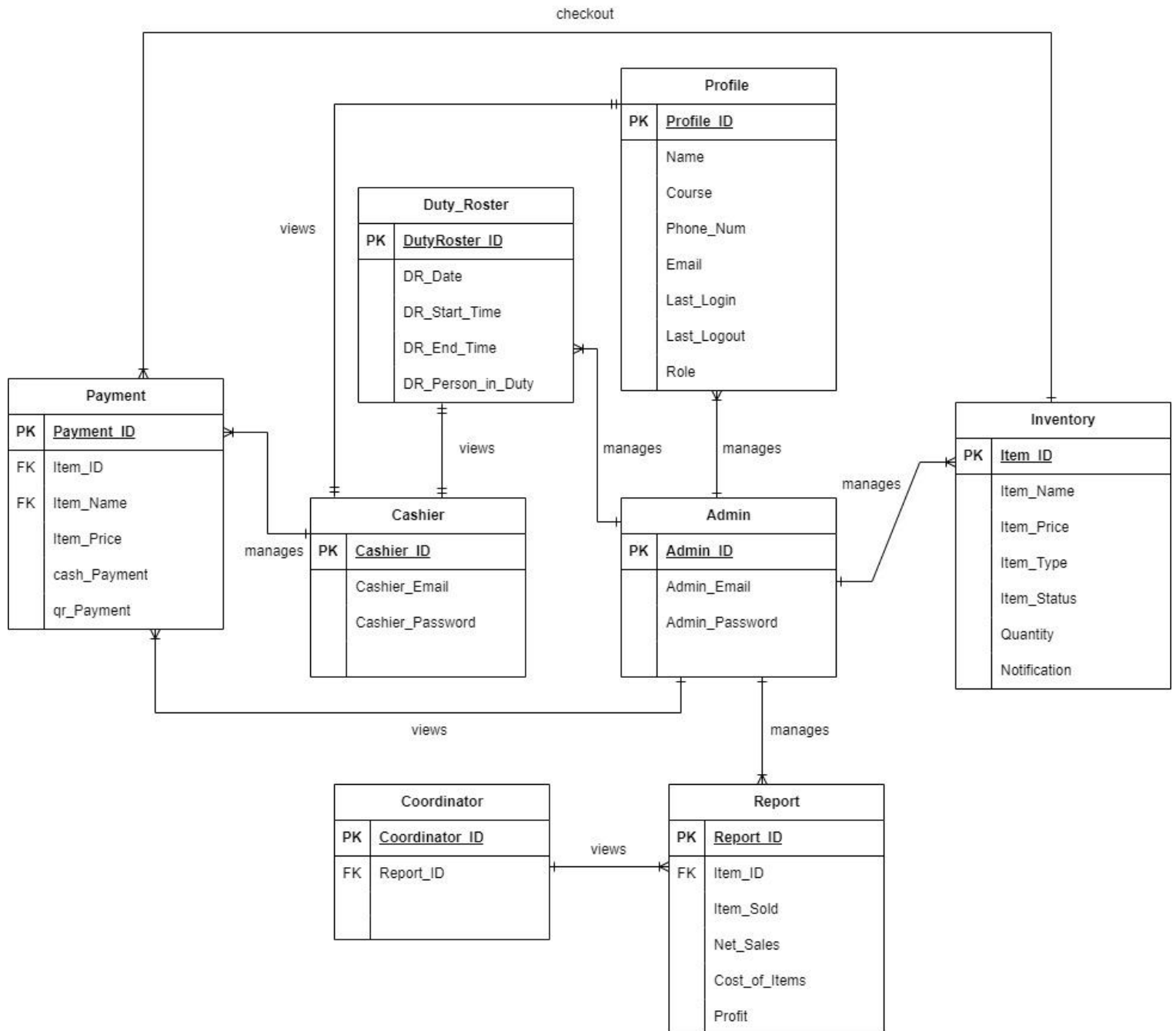
.env	This package is a configuration file used to store environment-specific configuration settings.
.env.example	This package is a sample configuration file that is used to provide an example of the various configuration options that can be used in the '.env' file.
view	This package is to store the HTML templates, which are responsible for generating the output presented to the user.
web.php	This package allows the system to define routes for web-based user interfaces.
Models	These packages allow the system to store Eloquent models, which represent the application's data and handle interactions with the database.
Controller	These packages allow the system to store the controller classes, which handle user requests and generate responses.
Middleware	These packages allow the system to store the middleware classes, which provide a convenient mechanism for filtering HTTP requests and responses entering the application.

3.2 MVC Package Module



3.3 Data Design

3.3.1 Entity Relationship Diagram (ERD)



3.3.1 Explanation of the Entity Relationship Diagram (ERD)

Entities:

- Admin: Represents an admin who manages all of the modules which are profile, inventory, payment, duty roster, and report. Attributes of the admin are Admin_ID as primary key, Admin_Email and Admin_Password.
- Cashier: Represents a cashier who performs payment activity. Attributes of the cashier are Cashier_ID as primary key, Cashier_Email and Cashier_Password.
- Coordinator: Represents a coordinator that is responsible to view the report. Coordinator has only Coordinator_ID as primary key, and foreign key which is Report_ID as attribute.
- Profile: Represents a profile that is responsible for storing the information of the user such as Profile_ID as primary key, Name, Course, Phone_Num, Email, Last_Login, Last_Logout, and Role.
- Payment: Represents a payment that helps to pay the item for the user which contains payment types that can be used by the cashier during the payment activity. Attributes of the payment are Payment_ID as primary key, while for the foreign key are Item_ID, Item_Name, and Item_Price, cash_Payment, and qr_Payment.
- Inventory: Represents an inventory that stores products of PETAKOM Mart. Attributes of the inventory are Item_ID as primary key, Item_Name, Item_Price, Item_Type, Item_Status, Quantity, and Notification.
- Duty Roster: Represents a schedule of duty for the user of the PETAKOM MM system whose attributes are DutyRoster_ID as primary key, DR_Time, DR_Start_Time, DR_End_Time, and DR_Person_in_Duty.
- Report: Represents a report that generates sales statistics of products. Attributes of the report are Report_ID as primary key, Item_ID as foreign key, Item_Sold, Net_Sales, Cost_of_Items, and Profit.

Relationships:

- Admin - Profile: Admin can view, add, delete, and edit the profile. This is a one-to-many relationship.
- Admin - Inventory: Admin can view, add, update, and delete products, and view notification messages. This is a one-to-many relationship.
- Admin - Report: Admin can view report details. This is a one-to-many relationship.
- Admin - Payment: Admin can view payment details. This is a one-to-many relationship.
- Admin - Duty Roster: Admin can add, view, delete, and edit duty schedule. This is a one-to-many relationship.
- Cashier - Payment: Cashier can add, edit, and remove items. This is a one-to-many relationship.
- Cashier - Profile: Cashier can view profile. This is a one-to-one relationship.
- Cashier - Duty Roster: Cashier can view duty schedule. This is a one-to-one relationship.
- Coordinator - Report: Coordinator can view report details. This is a one-to-many relationship.

3.3.2 Data Dictionary

3.3.2.1 Admin

Field Name	Description	Data Type	Constraint
Admin_ID	Admin Identification	VARCHAR(20)	PK
Admin_Email	Admin Email Address	VARCHAR(30)	
Admin_Password	Admin Password	VARCHAR(20)	

3.3.2.2 Cashier

Field Name	Description	Data Type	Constraint
Cashier_ID	Cashier Identification	VARCHAR(20)	PK
Cashier_Email	Cashier Email Address	VARCHAR(30)	
Cashier_Password	Cashier Password	VARCHAR(20)	

3.3.2.3 Coordinator

Field Name	Description	Data Type	Constraint
Report_ID	Report Identification	VARCHAR(10)	FK

3.3.2.4 Profile

Field Name	Description	Data Type	Constraint
Profile_ID	Profile Identification	VARCHAR(20)	PK
Name	User Name	VARCHAR(50)	

Course	User Course	VARCHAR(10)	
Phone_Num	User Phone Number	VARCHAR(15)	
Email	User Email Address	VARCHAR(30)	
Last_Login	Last Time User Login to The System	DATETIME	
Last_Logout	Last Time User Logout from the System	DATETIME	
Role	User Role	VARCHAR(30)	

3.3.2.5 Inventory

Field Name	Description	Data Type	Constraint
Item_ID	Item Identification	VARCHAR(20)	PK
Item_Name	Item Name	VARCHAR(30)	
Item_Price	Item Price	FLOAT	
Item_Type	Item Type	VARCHAR(20)	
Item_Status	Item Status	VARCHAR(20)	
Quantity	Item Quantity	INT	
Notification	Notification	VARCHAR(100)	

3.3.2.6 Report

Field Name	Description	Data Type	Constraint
Report_ID	Report Identification	VARCHAR(10)	PK
Item_ID	Item ID	VARCHAR(20)	FK

Item_Sold	Item Sold	INT	
Net_Sales	Total Sales	FLOAT	
Cost_of_Items	Cost Items	FLOAT	
Profit	Profit	FLOAT	

3.3.2.7 Payment

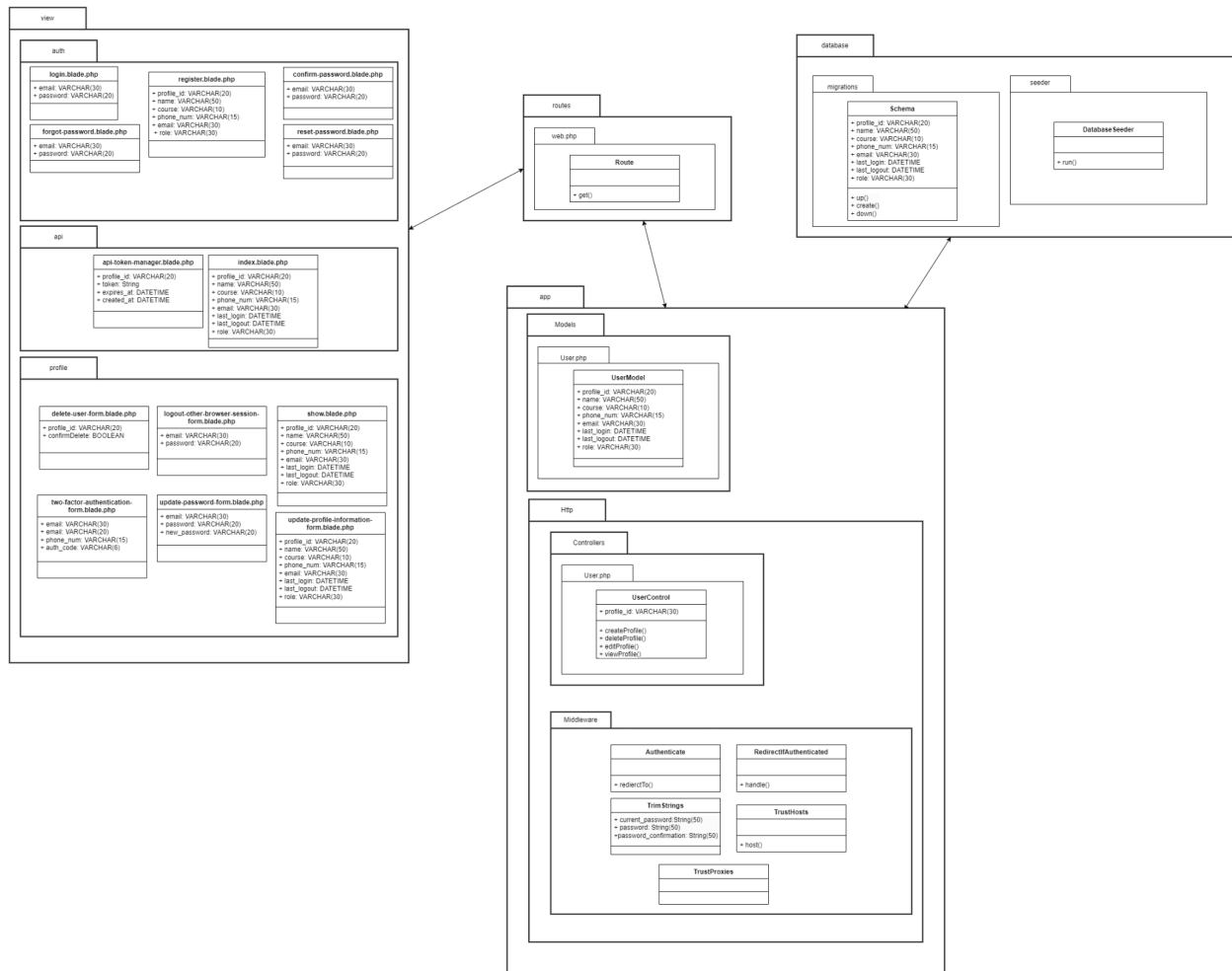
Field Name	Description	Data Type	Constraint
Payment_ID	Payment Identification	VARCHAR(20)	PK
Item_ID	Item ID	VARCHAR(20)	FK
Item_Name	Item Name	VARCHAR(30)	
Item_Price	Item Price	FLOAT	
cash_Payment	Payment Type	FLOAT	
qr_Payment	Payment Type	VARCHAR(10)	

3.3.2.8 Duty Roster

Field Name	Description	Data Type	Constraint
DutyRoster_ID	Duty Roster ID	VARCHAR(30)	PK
DR_Date	Duty Date	DATE	
DR_Start_Time	Duty Start Time	TIME	
DR_End_Time	Duty End Time	TIME	
DR_Person_in_Duty	Person in Charge	VARCHAR(50)	

4. DETAILED DESIGN

4.1 Package Manage Profile [SDD-REQ-100-100]



4.1.1 login.blade.php [SDD-REQ-100-101]

Class Type	Boundary Class	
Responsibility	An interface to display the login page for Admin and Cashier.	
Attributes	Attributes Name	Attributes Type
	row/column/name	class

	name CSS/JScript link number/String	id href scr value
Methods	Method Name	Description
	-	-
Algorithm	Start User clicks on the <<Sign In>> button If (Authentication == user) { The system will navigate to the PETAKOM main page } Else { The system will navigate to login page }	

4.1.2 register.blade.php [SDD-REQ-100-102]

Class Type	Boundary Class	
Responsibility	An interface to display the register page for Admin.	
Attributes	Attributes Name	Attributes Type
	row/column/name name CSS/JScript link number/String	class id href scr value
Methods	Method Name	Description
	-	-

Algorithm	Start User clicks on the <<Register>> button If (Authentication == user) { The system will navigate to the register page form } Else { The system will navigate to login page }
------------------	--

4.1.3 confirm-password.blade.php [SDD-REQ-100-103]

Class Type	Boundary Class	
Responsibility	An interface to display the password confirmation page for the user to confirm their password before allowing them to perform any activities.	
Attributes	Attributes Name	Attributes Type
	row/column/name name CSS/JScript link number/String	class id href scr value
Methods	Method Name	Description
	-	-
Algorithm	Start User enter password and clicks on <<Sign In >> button If (Authentication == user) { The system will navigate to confirm password page } Else	

	<pre> { The system will navigate to login page } </pre>
--	---

4.1.4 forgot-password.blade.php [SDD-REQ-100-104]

Class Type	Boundary Class	
Responsibility	An interface to allow users to reset passwords if they have forgotten them.	
Attributes	Attributes Name	Attributes Type
	row/column/name name CSS/JScript link number/String	class id href scr value
Methods	Method Name	Description
	-	-
Algorithm	<pre> Start User clicks on the <<Forgot Password>> button If (Authentication == user) { The system will navigate to the reset password page } Else { The system will navigate to login page } </pre>	

4.1.5 reset-password.blade.php [SDD-REQ-100-105]

Class Type	Boundary Class
-------------------	----------------

Responsibility	An interface to display reset password forms for users.	
Attributes	Attributes Name	Attributes Type
	row/column/name name CSS/JScript link number/String	class id href scr value
Methods	Method Name	Description
	-	-
Algorithm	<pre> Start User clicks on the <<Reset Password>> button If (Authentication == user) { The system will display reset password form } Else { The system will navigate to login page } </pre>	

4.1.6 api-token-manager.blade.php [SDD-REQ-100-106]

Class Type	Boundary Class	
Responsibility	An interface to manage API tokens associated with the user's profile	
Attributes	Attributes Name	Attributes Type
	row/column/name name CSS/JScript link number/String	class id href scr value

Methods	Method Name	Description
	-	-
Algorithm	<p>Start</p> <p>User clicks on the <<Sign In>> button</p> <p>If (Authentication == user)</p> <p>{</p> <p> The system will generate unique token associate with user account</p> <p>}</p> <p>Else</p> <p>{</p> <p> The system will navigate to login page</p> <p>}</p>	

4.1.7 index.blade.php [SDD-REQ-100-107]

Class Type	Boundary Class	
Responsibility	An interface to display the list of user profile information for Admin but for Cashier, the system only displays their own profile.	
Attributes	Attributes Name	Attributes Type
	row/column/name name CSS/JScript link number/String	class id href scr value
Methods	Method Name	Description
	-	-
Algorithm	<p>Admin:</p> <p>Start</p>	

	<p>Admin clicks on the <<Staff>> button</p> <pre> If (Authentication == user) { The system will navigate to the list of profile information } Else { The system will navigate to PETAKOM main page } End </pre> <p>Cashier:</p> <p>Start</p> <p>User clicks on the profile icon</p> <pre> If (Authentication == user) { The system will display the profile information } Else { The system will navigate to PETAKOM main page } End </pre>
--	--

4.1.8 delete-user-form.blade.php [SDD-REQ-100-108]

Class Type	Boundary Class	
Responsibility	An interface for Admin to delete any user profile information	
Attributes	Attributes Name	Attributes Type
	row/column/name name CSS/JScript link	class id href scr

	number/String	value
Methods	Method Name	Description
	-	-
Algorithm	Start User clicks on the <<Delete Staff>> button If (Authentication == user) { The system will navigate to confirmation message interface } Else { The system will navigate to profile information page }	

4.1.9 logout-other-browser-session-form.blade.php [SDD-REQ-100-109]

Class Type	Boundary Class	
Responsibility	An interface to manage the process of logging out users from all other browser sessions.	
Attributes	Attributes Name	Attributes Type
	row/column/name name CSS/JScript link number/String	class id href scr value
Methods	Method Name	Description
	-	-
Algorithm	Start User clicks on the <<Log Out>> button	

	<pre> If (Authentication == user) { The system will navigate to the login page } Else { The system will navigate to PETAKOM main page } </pre>
--	--

4.1.10 show.blade.php [SDD-REQ-100-110]

Class Type	Boundary Class	
Responsibility	An interface to display the profile data to the user.	
Attributes	Attributes Name	Attributes Type
	row/column/name name CSS/JScript link number/String	class id href scr value
Methods	Method Name	Description
	-	-
Algorithm	Admin: Start Admin clicks on the <<Staff>> button <pre> If (Authentication == user) { The system will navigate to the list of profile information } Else { The system will navigate to PETAKOM main page } </pre>	

	<pre> } End Cashier: Start User clicks on the profile icon If (Authentication == user) { The system will display the profile information } Else { The system will navigate to PETAKOM main page } End </pre>
--	--

4.1.11 two-factor-authentication-form.blade.php [SDD-REQ-100-111]

Class Type	Boundary Class	
Responsibility	An interface to display the two-factor authentication form to the user.	
Attributes	Attributes Name	Attributes Type
	row/column/name name CSS/JScript link number/String	class id href scr value
Methods	Method Name	Description
	-	-
Algorithm	Start User enter email address	

	<pre> If (Authentication == user) { The system will display form to the user requesting OTP } Else { The system will navigate to login page } </pre>
--	--

4.1.12 update-password-form.blade.php [SDD-REQ-100-112]

Class Type	Boundary Class	
Responsibility	An interface to allows Admin and Cashier to update their password	
Attributes	Attributes Name	Attributes Type
	row/column/name name CSS/JScript link number/String	class id href scr value
Methods	Method Name	Description
	-	-
Algorithm	<pre> Start User clicks on <<Update Password>> button If (Authentication == user) { The system will navigate to update password form page } Else { The system will navigate to PETAKOM main page } </pre>	

	} End
--	----------

4.1.13 update-profile-information.blade.php [SDD-REQ-100-113]

Class Type	Boundary Class	
Responsibility	An interface to display the profile page which Admin can update the profile for all users	
Attributes	Attributes Name	Attributes Type
	row/column/name name CSS/JScript link number/String	class id href scr value
Methods	Method Name	Description
	-	-
Algorithm	Start User clicks on <<Update>> button If (Authentication == user) { The system will display update profile form } Else { The system will navigate to PETAKOM main page }	

4.1.14 Route [SDD-REQ-100-114]

Class Type	Route Class
-------------------	-------------

Responsibility	A routing that provides a powerful and flexible way to define routes for the Laravel application's web-based user interfaces.	
Attributes	Attributes Name	Attributes Type
	-	-
Methods	Method Name	Description
	get()	A method provided by the 'Route' class in Laravel that registers a new route for HTTP GET requests.
Algorithm	<p>Start</p> <p>The system receives the information navigation from the boundary class interface.</p> <p>The system validates the data, if the data needs to be sent to the Controller, it will be sent to the controller first. Otherwise, it will display the data to the interface.</p> <p>End</p>	

4.1.15 UserModel [SDD-REQ-100-115]

Class Type	Entity Class	
Responsibility	The structure and behaviour of the entity interacting with the database through an associated Eloquent model.	
Attributes	Attributes Name	Attributes Type
	profile_id name course phone_num email last_login last_logout role	VARCHAR VARCHAR VARCHAR VARCHAR VARCHAR DATETIME DATETIME VARCHAR

Methods	Method Name	Description
	-	-
Algorithm	-	

4.1.16 UserController [SDD-REQ-100-116]

Class Type	Control Class	
Responsibility	A package that controls the flow of data between the model and the view.	
Attributes	Attributes Name	Attributes Type
	profile_id	VARCHAR
Methods	Method Name	Description
	addProfile() deleteProfile() editProfile() viewProfile()	To connect the profile id between the database and view.
Algorithm	<p>Start</p> <p>The system directly navigates the interface from Route to the Controller.</p> <p>The system validates the data and displays the data to the interface.</p> <p>End</p>	

4.1.17 Authenticate [SDD-REQ-100-117]

Class Type	Middleware Class
Responsibility	A package that authenticates incoming requests to ensure that the user is logged in before allowing them access to protected routes or resources.

Attributes	Attributes Name	Attributes Type
	request	Request
Methods	Method Name	Description
	redirectTo()	To Get the path for users that should be redirected to when they are not authenticated.
Algorithm	-	

4.1.18 RedirectIfAuthenticated [SDD-REQ-100-118]

Class Type	Middleware Class	
Responsibility	A package that checks if a user is already authenticated and redirects them to a specified path if they are.	
Attributes	Attributes Name	Attributes Type
	Request next guard	Request Closure String
Methods	Method Name	Description
	handle()	To handle an incoming request.
Algorithm	-	

4.1.19 TrimStrings [SDD-REQ-100-119]

Class Type	Middleware Class	
Responsibility	A package that automatically trims whitespace from the beginning and end of request input data.	

Attributes	Attributes Name	Attributes Type
	current_password password password_confirmation	String String String
Methods	Method Name	Description
	-	-
Algorithm	-	

4.1.20 TrusHosts [SDD-REQ-100-120]

Class Type	Middleware Class	
Responsibility	A package that verifies that the HTTP request's host is trusted before allowing the request to proceed to the application.	
Attributes	Attributes Name	Attributes Type
	-	-
Methods	Method Name	Description
	host()	To get the host patterns that should be trusted.
Algorithm	-	

4.1.21 TrustProxies [SDD-REQ-100-121]

Class Type	Middleware Class	
Responsibility	A package that configures trusted proxies that are used to retrieve the client's IP address when the application is behind a load balancer or reverse proxy.	
Attributes	Attributes Name	Attributes Type

	headers	Request
Methods	Method Name	Description
	-	-
Algorithm	-	

4.1.22 Schema [SDD-REQ-100-122]

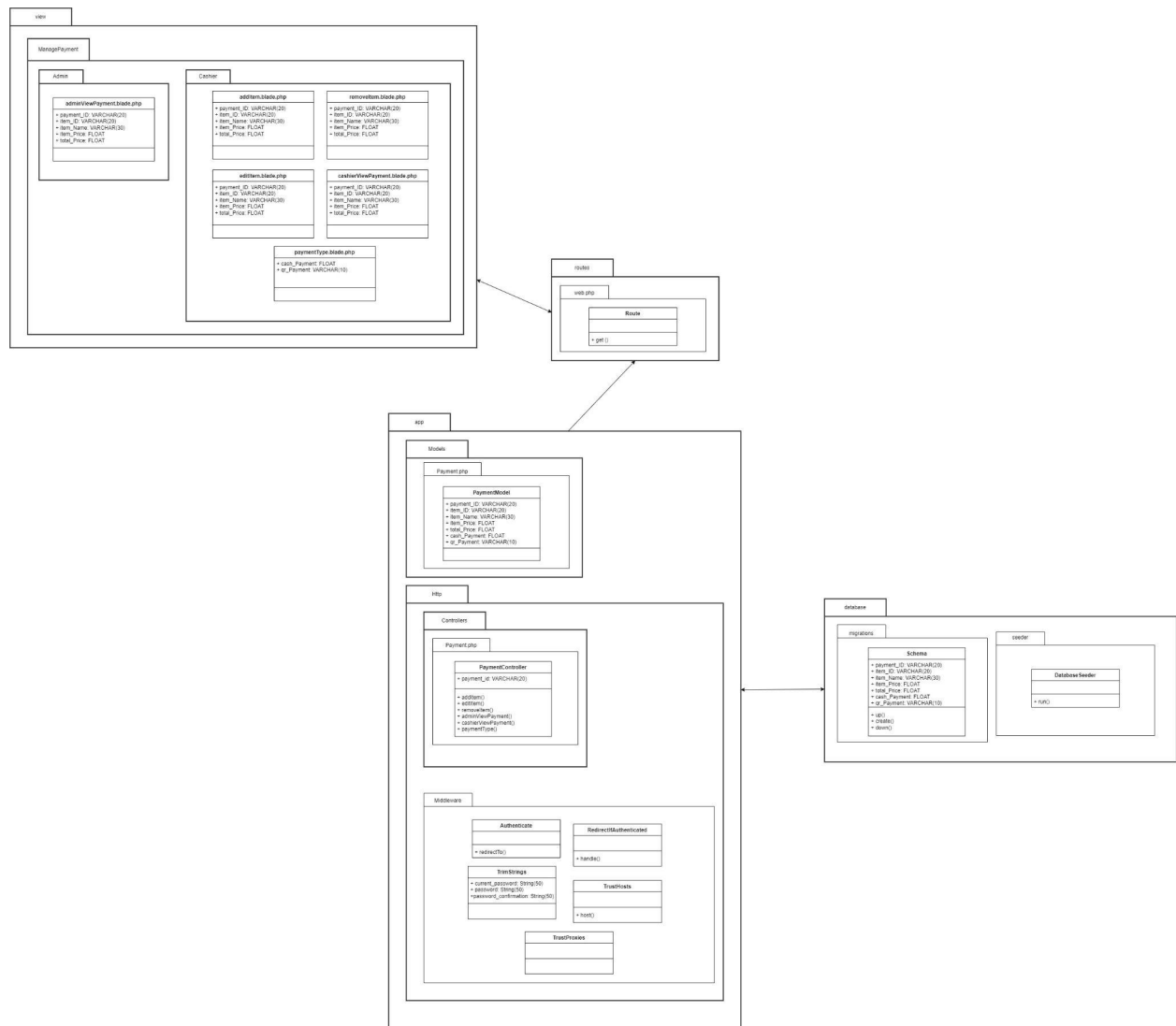
Class Type	Blueprint Class	
Responsibility	A package that defines and modifies database tables through the creation of migration files.	
Attributes	Attributes Name	Attributes Type
	table	Blueprint
Methods	Method Name	Description
	up() create() down()	Run migration to database To create a table in database To reverse the migration to database.
Algorithm	-	

4.1.23 DatabaseSeeder [SDD-REQ-100-123]

Class Type	Seeder Class	
Responsibility	A package that populates the database with initial or dummy data when the 'php artisan db:seed' command is executed.	
Attributes	Attributes Name	Attributes Type
	-	-

Methods	Method Name	Description
	run()	To seed the application's database.
Algorithm	-	

4.2 Package Manage Payment [SDD-REQ-400-100]



4.2.1 addItem.blade.php [SDD-REQ-400-101]

Class Type	Boundary Class	
Responsibility	An interface to display the add item page for the cashier.	
Attributes	Attributes Name	Attributes Type
	row/column/name name	class id

	CSS/JScript link number/String	href scr value
Methods	Method Name	Description
	-	-
Algorithm	Start User clicks on the <<Add to cart>> button The system directly adds the item. The system displays the page with the added items. End	

4.2.2 removeItem.blade.php [SDD-REQ-400-102]

Class Type	Boundary Class	
Responsibility	An interface to display the removed item on the page for the cashier.	
Attributes	Attributes Name	Attributes Type
	row/column/name name CSS/JScript link number/String	class id href scr value
Methods	Method Name	Description
	-	-
Algorithm	Start User clicks on the <<Removes>> button The system directly removes the item.	

	The system displays the page with the current items. End
--	---

4.2.3 editItem.blade.php [SDD-REQ-400-103]

Class Type	Boundary Class	
Responsibility	An interface to display the edited item on the page for the cashier.	
Attributes	Attributes Name	Attributes Type
	row/column/name name CSS/JScript link number/String	class id href scr value
Methods	Method Name	Description
	-	-
Algorithm	<p>Start</p> <p>User clicks on the <<quantity>> button</p> <p>The system directly increases or decreases the quantity of the item.</p> <p>The system displays the page with the updated items.</p> <p>End</p>	

4.2.4 adminViewPayment.blade.php [SDD-REQ-400-104]

Class Type	Boundary Class	
Responsibility	An interface to display the payment to the admin.	
Attributes	Attributes Name	Attributes Type

	row/column/name name CSS/JScript link number/String	class id href scr value
Methods	Method Name	Description
	-	-
Algorithm	Start User clicks on the <<Payment>> button The system displays the page with the payment details. End	

4.2.5 cashierViewPayment.blade.php [SDD-REQ-400-105]

Class Type	Boundary Class	
Responsibility	An interface to display the payment to the cashier.	
Attributes	Attributes Name	Attributes Type
	row/column/name name CSS/JScript link number/String	class id href scr value
Methods	Method Name	Description
	-	-
Algorithm	Start User clicks on the <<Payment>> button The system displays the page with payment details. End	

4.2.6 paymentType.blade.php [SDD-REQ-400-106]

Class Type	Boundary Class	
Responsibility	An interface to display the type of payment.	
Attributes	Attributes Name	Attributes Type
	row/column/name name CSS/JScript link number/String	class id href scr value
Methods	Method Name	Description
	-	-
Algorithm	Start User clicks on the <<Checkout>> button The system displays the page with the type of payments. End	

4.2.7 Route [SDD-REQ-400-107]

Class Type	Route Class	
Responsibility	A routing that provides a powerful and flexible way to define routes for the Laravel application's web-based user interfaces.	
Attributes	Attributes Name	Attributes Type
	-	-
Methods	Method Name	Description
	get()	A method provided by the 'Route' class in Laravel that registers a new route for HTTP GET requests.

Algorithm	Start
	The system receives the information navigation from the boundary class interface.
	The system validates the data, if the data needs to be sent to the Controller, it will be sent to the controller first. Otherwise, it will display the data to the interface.
	End

4.2.8 PaymentModel [SDD-REQ-400-108]

Class Type	Entity Class	
Responsibility	The structure and behaviour of the entity interacting with the database through an associated Eloquent model.	
Attributes	Attributes Name	Attributes Type
	payment_ID item_ID item_Name item_Price payment_Type	VARCHAR VARCHAR VARCHAR Float VARCHAR
Methods	Method Name	Description
	-	-
Algorithm	-	

4.2.9 PaymentController [SDD-REQ-400-109]

Class Type	Control Class
Responsibility	A package that controls the flow of data between the model and the view.

Attributes	Attributes Name	Attributes Type
	payment_id item_id	VARCHAR VARCHAR
Methods	Method Name	Description
	addItem() removeItem() editItem() viewPayment()	To connect the payment id between the database and view.
Algorithm	<p>Start</p> <p>The system directly navigates the interface from Route to the Controller.</p> <p>The system validates the data and displays the data to the interface.</p> <p>End</p>	

4.2.10 Authenticate [SDD-REQ-400-110]

Class Type	Middleware Class	
Responsibility	A package that authenticates incoming requests to ensure that the user is logged in before allowing them access to protected routes or resources.	
Attributes	Attributes Name	Attributes Type
	request	Request
Methods	Method Name	Description
	redirectTo()	To Get the path for users that should be redirected to when they are not authenticated.
Algorithm	-	

4.2.11 RedirectIfAuthenticated [SDD-REQ-400-111]

Class Type	Middleware Class	
Responsibility	A package that checks if a user is already authenticated and redirects them to a specified path if they are.	
Attributes	Attributes Name	Attributes Type
	Request next guard	Request Closure String
Methods	Method Name	Description
	handle()	To handle an incoming request.
Algorithm	-	

4.2.12 TrimStrings [SDD-REQ-400-112]

Class Type	Middleware Class	
Responsibility	A package that automatically trims whitespace from the beginning and end of request input data.	
Attributes	Attributes Name	Attributes Type
	current_password password password_confirmation	String String String
Methods	Method Name	Description
	-	-
Algorithm	-	

4.4.13 TrusHosts [SDD-REQ-400-113]

Class Type	Middleware Class	
Responsibility	A package that verifies that the HTTP request's host is trusted before allowing the request to proceed to the application.	
Attributes	Attributes Name	Attributes Type
	-	-
Methods	Method Name	Description
	host()	To get the host patterns that should be trusted.
Algorithm	-	

4.4.14 TrustProxies [SDD-REQ-400-114]

Class Type	Middleware Class	
Responsibility	A package that configures trusted proxies that are used to retrieve the client's IP address when the application is behind a load balancer or reverse proxy.	
Attributes	Attributes Name	Attributes Type
	headers	Request
Methods	Method Name	Description
	-	-
Algorithm	-	

4.4.15 Schema [SDD-REQ-400-115]

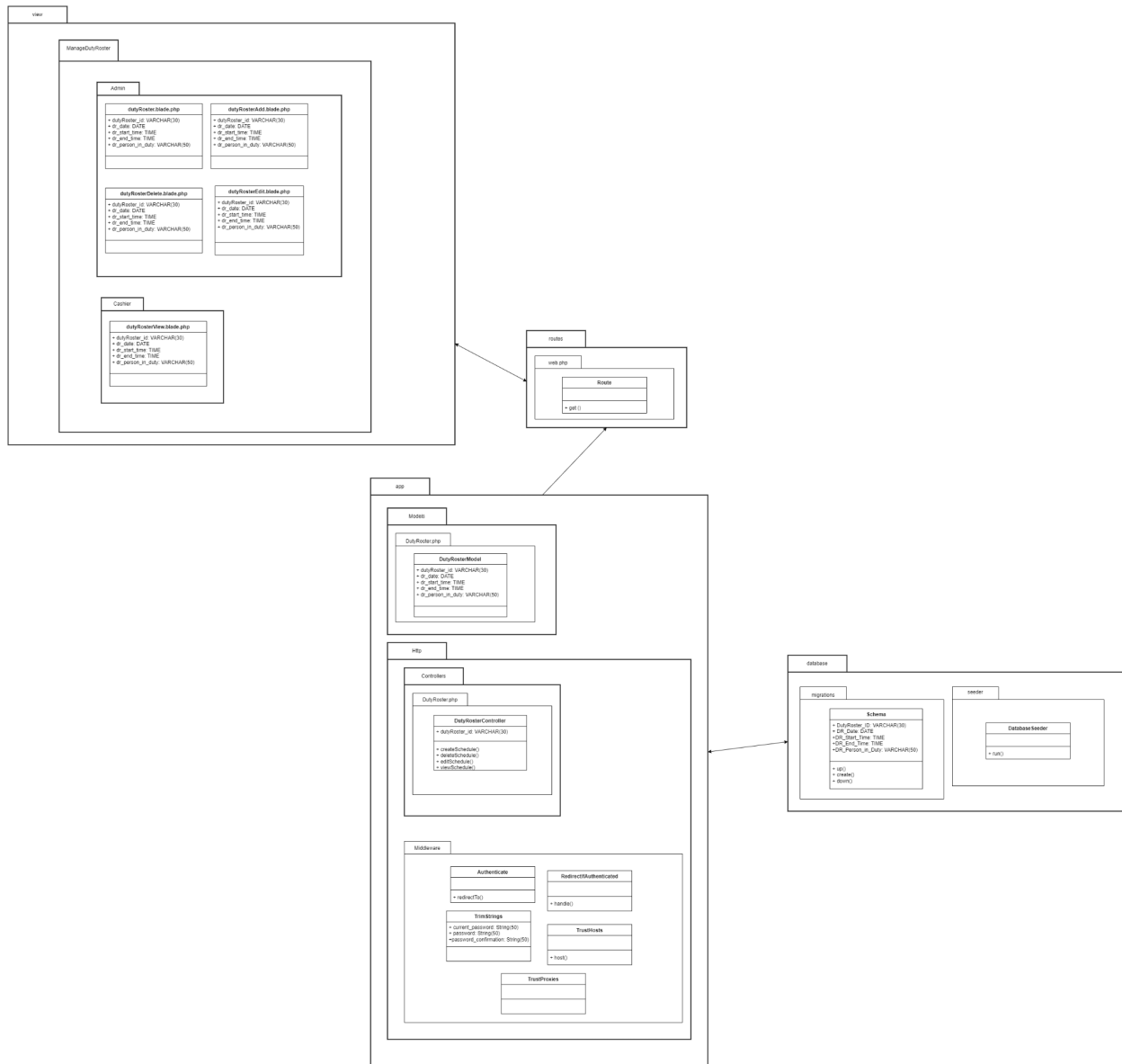
Class Type	Blueprint Class	
Responsibility	A package that defines and modifies database tables through the creation of migration files.	

Attributes	Attributes Name	Attributes Type
	table	Blueprint
Methods	Method Name	Description
	up() create() down()	Run migration to database To create a table in database To reverse the migration to database.
Algorithm	-	

4.2.16 DatabaseSeeder [SDD-REQ-400-116]

Class Type	Seeder Class	
Responsibility	A package that populates the database with initial or dummy data when the 'php artisan db:seed' command is executed.	
Attributes	Attributes Name	Attributes Type
	-	-
Methods	Method Name	Description
	run()	To seed the application's database.
Algorithm	-	

4.3 Package Manage Duty Roster [SDD-REQ-500-100]



4.3.1 dutyRoster.blade.php [SDD-REQ-500-101]

Class Type	Boundary Class	
Responsibility	An interface to display the main page of duty schedule for Admin.	
Attributes	Attributes Name	Attributes Type
	row/column/name name CSS/JScript link number/String	class id href scr value
Methods	Method Name	Description
	-	-
Algorithm	<pre> Start User clicks on the <<Duty Roster>> button If (Authentication == user) { The system will navigate to the duty schedule main page } Else { The system will navigate to login page } End </pre>	

4.3.2 dutyRosterAdd.blade.php [SDD-REQ-500-102]

Class Type	Boundary Class	
Responsibility	An interface to display the add schedule form page for Admin.	
Attributes	Attributes Name	Attributes Type

	row/column/name name CSS/JScript link number/String	class id href scr value
Methods	Method Name	Description
	-	-
Algorithm	Start User clicks on the <<Add Schedule>> button If (Authentication == user) { The system will navigate to the add duty schedule form } Else { The system will display the main page for duty roster. } End	

4.3.3 dutyRosterEdit.blade.php [SDD-REQ-500-103]

Class Type	Boundary Class	
Responsibility	An interface to display the edit schedule form page for Admin.	
Attributes	Attributes Name	Attributes Type
	row/column/name name CSS/JScript link number/String	class id href scr value
Methods	Method Name	Description

	-	-
Algorithm	Start User clicks on the <<Edit Schedule>> button If (Authentication == user) { The system will navigate to the edit duty schedule form } Else { The system will display the main page for duty roster } End	

4.3.4 dutyRosterDelete.blade.php [SDD-REQ-500-104]

Class Type	Boundary Class	
Responsibility	An interface to display the delete schedule page for Admin.	
Attributes	Attributes Name	Attributes Type
	row/column/name name CSS/JScript link number/String	class id href scr value
Methods	Method Name	Description
	-	-
Algorithm	Start User clicks on the <<Delete Schedule>> button If (Authentication == user) { The system will navigate to the delete duty schedule }	

	<pre> form } Else { The system will display the main page for duty roster } End </pre>
--	--

4.3.5 dutyRosterView.blade.php [SDD-REQ-500-105]

Class Type	Boundary Class	
Responsibility	An interface to display the edit schedule form page for Admin.	
Attributes	Attributes Name	Attributes Type
	row/column/name name CSS/JScript link number/String	class id href scr value
Methods	Method Name	Description
	-	-
Algorithm	<pre> Start User clicks on the <<Edit Schedule>> button If (Authentication == user) { The system will navigate to the edit duty schedule form } Else { The system will display the main page for duty roster } End </pre>	

4.3.6 Route [SDD-REQ-500-106]

Class Type	Route Class	
Responsibility	An interface to display the edit schedule form page for Admin.	
Attributes	Attributes Name	Attributes Type
	-	-
Methods	Method Name	Description
	get()	A method provided by the 'Route' class in Laravel that registers a new route for HTTP GET requests.
Algorithm	<p>Start</p> <p>The system receives the information navigation from the boundary class interface.</p> <p>The system validates the data, if the data needs to be sent to the Controller, it will be sent to the controller first. Otherwise, it will display the data to the interface.</p> <p>End</p>	

4.3.7 DutyRosterModel [SDD-REQ-500-107]

Class Type	Entity Class	
Responsibility	The structure and behavior of the entity interacting with the database through an associated Eloquent model.	
Attributes	Attributes Name	Attributes Type
	dutyRoster_id dr_date dr_start_time dr_end_time	VARCHAR DATE TIME TIME

	dr_person_in_duty	VARCHAR
Methods	Method Name	Description
	-	-
Algorithm	-	

4.3.8 DutyRosterController [SDD-REQ-500-108]

Class Type	Control Class	
Responsibility	A package that controls the flow of data between the model and the view.	
Attributes	Attributes Name	Attributes Type
	dutyRoster_id	VARCHAR
Methods	Method Name	Description
	createSchedule() deleteSchedule() editSchedule() viewSchedule()	To connect the duty roster id between database and createSchedule, deleteSchedule, editSchedule and viewSchedule.
Algorithm	<p>Start</p> <p>The system directly navigates the interface from Route to the Controller.</p> <p>The system validates the data and displays the data to the interface.</p> <p>End</p>	

4.3.9 Authenticate [SDD-REQ-500-109]

Class Type	Middleware Class	
Responsibility	A package that authenticates incoming requests to ensure that the user is logged in before allowing them access to protected routes or resources.	
Attributes	Attributes Name	Attributes Type
	request	Request
Methods	Method Name	Description
	redirectTo()	To Get the path for users that should be redirected to when they are not authenticated.
Algorithm	-	

4.3.10 RedirectIfAuthenticated [SDD-REQ-500-110]

Class Type	Middleware Class	
Responsibility	A package that checks if a user is already authenticated and redirects them to a specified path if they are.	
Attributes	Attributes Name	Attributes Type
	Request next guard	Request Closure String
Methods	Method Name	Description
	handle()	To handle an incoming request.
Algorithm	-	

4.3.11 TrimStrings [SDD-REQ-500-111]

Class Type	Middleware Class	
Responsibility	A package that automatically trims whitespace from the beginning and end of request input data.	
Attributes	Attributes Name	Attributes Type
	current_password password password_confirmation	String String String
Methods	Method Name	Description
	-	-
Algorithm	-	

4.3.12 TrusHosts [SDD-REQ-500-112]

Class Type	Middleware Class	
Responsibility	A package that verifies that the HTTP request's host is trusted before allowing the request to proceed to the application.	
Attributes	Attributes Name	Attributes Type
	-	-
Methods	Method Name	Description
	host()	To get the host patterns that should be trusted.
Algorithm	-	

4.3.13 TrustProxies[SDD-REQ-500-113]

Class Type	Middleware Class	
Responsibility	A package that configures trusted proxies that are used to	

	retrieve the client's IP address when the application is behind a load balancer or reverse proxy.	
Attributes	Attributes Name	Attributes Type
	headers	Request
Methods	Method Name	Description
	-	-
Algorithm	-	

4.3.14 Schema [SDD-REQ-500-114]

Class Type	Blueprint Class	
Responsibility	A package that defines and modifies database tables through the creation of migration files.	
Attributes	Attributes Name	Attributes Type
	table	Blueprint
Methods	Method Name	Description
	up() create() down()	Run migration to database To create a table in database To reverse the migration to database.
Algorithm	-	

4.3.15 DatabaseSeeder [SDD-REQ-500-115]

Class Type	Seeder Class
Responsibility	A package that populates the database with initial or dummy

	data when the 'php artisan db:seed' command is executed.	
Attributes	Attributes Name	Attributes Type
	-	-
Methods	Method Name	Description
	run()	To seed the application's database.
Algorithm	-	

4.4 Package Manage Report [SDD-REQ-600-100]



4.4.1 report.blade.php [SDD-REQ-600-101]

Class Type	Boundary Class	
Responsibility	An interface to display the main page of the report for users (Admin).	
Attributes	Attributes Name	Attributes Type
	row/column/name name CSS/JScript link number/String	class id href scr value
Methods	Method Name	Description
	-	-
Algorithm	<p>Start</p> <p style="padding-left: 40px;">The user clicks on the <Report> button</p> <pre> if(Authentication == user){ The system will navigate to the main page report. }else{ The system will navigate to the login page. } </pre> <p>End</p>	

4.4.2 reportMainPage.blade.php [SDD-REQ-600-102]

Class Type	Boundary Class	
Responsibility	An interface to display the main page of the report for users (Coordinator).	
Attributes	Attributes Name	Attributes Type
	row/column/name name	class id

	CSS/JScript link number/String	href scr value
Methods	Method Name	Description
	-	-
Algorithm	<p>Start</p> <p>The user clicks on the <View as Coordinator> text link.</p> <p>The system directly automatically creates a random ID for the user.</p> <p>The system displays the main page of the report.</p> <p>End</p>	

4.4.3 reportView.blade.php [SDD-REQ-600-103]

Class Type	Boundary Class	
Responsibility	An interface to display the detail of the report for users (Coordinator).	
Attributes	Attributes Name	Attributes Type
	row/column/name name CSS/JScript link number/String	class id href scr value
Methods	Method Name	Description
	-	-
Algorithm	<p>Start</p> <p>The user clicks one of the button list report on the main interface.</p>	

	<p>The system directly validates the user action and displays the report detail interface for the user.</p> <p>End</p>
--	--

4.4.4 Route [SDD-REQ-600-104]

Class Type	Route Class	
Responsibility	A routing that provides a powerful and flexible way to define routes for the Laravel application's web-based user interfaces.	
Attributes	Attributes Name	Attributes Type
	-	-
Methods	Method Name	Description
	get()	A method provided by the 'Route' class in Laravel that registers a new route for HTTP GET requests.
Algorithm	<p>Start</p> <p>The system receives the information navigation from the boundary class interface.</p> <p>The system validates the data, if the data needs to be sent to the Controller, it will be sent to the controller first. Otherwise, it will display the data to the interface.</p> <p>End</p>	

4.4.5 ReportModel [SDD-REQ-600-105]

Class Type	Entity Class
Responsibility	The structure and behaviour of the entity interacting with the database through an associated Eloquent model.

Attributes	Attributes Name	Attributes Type
	report_id item_id item_Sold net-Sales cost_of_Items profit	VARCHAR VARCHAR INT Float Float Float
Methods	Method Name	Description
	-	-
Algorithm	-	

4.4.6 ReportController [SDD-REQ-600-106]

Class Type	Control Class	
Responsibility	A package that controls the flow of data between the model and the view.	
Attributes	Attributes Name	Attributes Type
	report_id item_id	VARCHAR VARCHAR
Methods	Method Name	Description
	generateReport() viewReport()	To connect the report id between the database and view.
Algorithm	<p>Start</p> <p>The system directly navigates the interface from Route to the Controller.</p> <p>The system validates the data and displays the data to the interface.</p> <p>End</p>	

4.4.7 Authenticate [SDD-REQ-600-107]

Class Type	Middleware Class	
Responsibility	A package that authenticates incoming requests to ensure that the user is logged in before allowing them access to protected routes or resources.	
Attributes	Attributes Name	Attributes Type
	request	Request
Methods	Method Name	Description
	redirectTo()	To Get the path for users that should be redirected to when they are not authenticated.
Algorithm	-	

4.4.8 RedirectIfAuthenticated [SDD-REQ-600-108]

Class Type	Middleware Class	
Responsibility	A package that checks if a user is already authenticated and redirects them to a specified path if they are.	
Attributes	Attributes Name	Attributes Type
	Request next guard	Request Closure String
Methods	Method Name	Description
	handle()	To handle an incoming request.
Algorithm	-	

4.4.9 TrimStrings [SDD-REQ-600-109]

Class Type	Middleware Class	
Responsibility	A package that automatically trims whitespace from the beginning and end of request input data.	
Attributes	Attributes Name	Attributes Type
	current_password password password_confirmation	String String String
Methods	Method Name	Description
	-	-
Algorithm	-	

4.4.10 TrusHosts [SDD-REQ-600-110]

Class Type	Middleware Class	
Responsibility	A package that verifies that the HTTP request's host is trusted before allowing the request to proceed to the application.	
Attributes	Attributes Name	Attributes Type
	-	-
Methods	Method Name	Description
	host()	To get the host patterns that should be trusted.
Algorithm	-	

4.4.11 TrustProxies [SDD-REQ-600-111]

Class Type	Middleware Class	
Responsibility	A package that configures trusted proxies that are used to retrieve the client's IP address when the application is behind a load balancer or reverse proxy.	
Attributes	Attributes Name	Attributes Type
	headers	Request
Methods	Method Name	Description
	-	-
Algorithm	-	

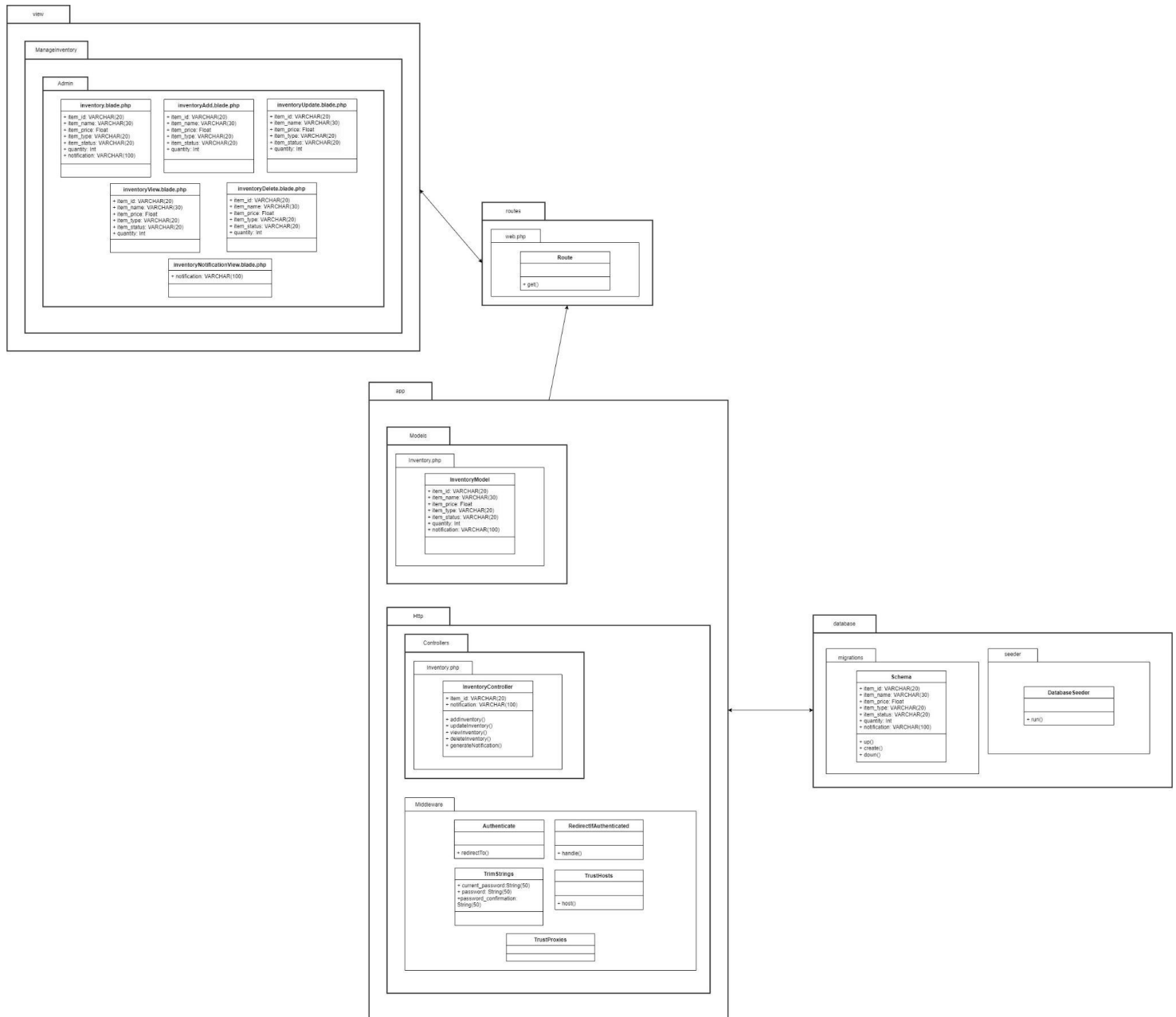
4.4.12 Schema [SDD-REQ-600-112]

Class Type	Blueprint Class	
Responsibility	A package that defines and modifies database tables through the creation of migration files.	
Attributes	Attributes Name	Attributes Type
	table	Blueprint
Methods	Method Name	Description
	up() create() down()	Run migration to database To create a table in database To reverse the migration to database.
Algorithm	-	

4.4.13 DatabaseSeeder [SDD-REQ-600-113]

Class Type	Seeder Class	
Responsibility	A package that populates the database with initial or dummy data when the 'php artisan db:seed' command is executed.	
Attributes	Attributes Name	Attributes Type
	-	-
Methods	Method Name	Description
	run()	To seed the application's database.
Algorithm	-	

4.5 Package Manage Inventory [SDD-REQ-700-100]



4.5.1 inventory.blade.php [SDD-REQ-700-101]

Class Type	Boundary Class	
Responsibility	An interface that displays the main page of inventory for users (Admin).	
Attributes	Attributes Name	Attributes Type
	row/column/name name CSS/JScript link number/String	class id href scr value
Methods	Method Name	Description
	-	-
Algorithm	<p>Start</p> <p> The user clicks on the <Store Management> button</p> <p> if (Authentication == user){</p> <p> The system will navigate to the inventory main page.</p> <p> }else{</p> <p> The system will navigate to the login page.</p> <p> }</p> <p>End</p>	

4.5.2 inventoryAdd.blade.php [SDD-REQ-700-102]

Class Type	Boundary Class	
Responsibility	An interface that displays forms for adding new products.	
Attributes	Attributes Name	Attributes Type
	row/column/name name CSS/JScript link number/String	class id href scr value
Methods	Method Name	Description
	-	-
Algorithm	<p>Start</p> <p>The user clicks on the <<Add Product>> button on the main page.</p> <p>The system directly validates the user action and displays product information form interface for the user.</p> <p>The user enters information for the new product and clicks on the <<Save>> button.</p> <p>The system validates input entered.</p> <pre> If (input is valid){ The system will add the product details to the inventory page and store data into DB }else{ The system will display an error message and return to the inventory main page } </pre> <p>End</p>	

4.5.3 inventoryUpdate.blade.php [SDD-REQ-700-103]

Class Type	Boundary Class	
Responsibility	An interface that displays form to update product details.	
Attributes	Attributes Name	Attributes Type
	row/column/name name CSS/JScript link number/String	class id href scr value
Methods	Method Name	Description
	-	-
Algorithm	<p>Start</p> <p>The user enter product name to be updated on the search box</p> <p>The system search the inventory for a product with a matching name and display a list of products</p> <p>The user select the product to be updated</p> <p>The system validates the user action and display current details of the product</p> <p>The user clicks on the <<Edit>> button</p> <p>The system directly validates the user action and display pre-populated product information form</p> <p>The user edit the product details and click on <<Update>> button</p> <p>The system validate the input entered</p> <p>If (input is valid){</p>	

	<p>The system updates the product details in the inventory and stores data into DB.</p> <pre> else{ The system displays an error message and returns to the Inventory main page. } End </pre>
--	---

4.5.4 inventoryView.blade.php [SDD-REQ-700-104]

Class Type	Boundary Class	
Responsibility	An interface that displays product details.	
Attributes	Attributes Name	Attributes Type
	row/column/name name CSS/JScript link number/String	class id href scr value
Methods	Method Name	Description
	-	-
Algorithm	<p>Start</p> <p>The user enter product name to be viewed on the search box</p> <p>The system search the inventory for a product with a matching name and display a list of products</p> <p>The user selects the product to be viewed.</p> <p>The system directly validates the user action and displays the details of the selected product for the user.</p> <p>End</p>	

4.5.5 inventoryDelete.blade.php [SDD-REQ-700-105]

Class Type	Boundary Class	
Responsibility	An interface that allows users to delete products from the inventory page.	
Attributes	Attributes Name	Attributes Type
	row/column/name name CSS/JScript link number/String	class id href scr value
Methods	Method Name	Description
	-	-
Algorithm	<p>Start</p> <p>The user enter product name to be deleted on the search box</p> <p>The system search the inventory for a product with a matching name and display a list of products</p> <p>The user select the product to be deleted and clicks on the <<Delete Product>> button</p> <p>The system validates the user action and display confirmation message</p> <pre> If (user confirms deletion){ The system remove the product from the inventory page and DB }else{ The system return to the inventory main page } </pre> <p>End</p>	

4.5.6 inventoryNotificationView.blade.php [SDD-REQ-700-106]

Class Type	Boundary Class	
Responsibility	An interface that displays notification messages for inventory.	
Attributes	Attributes Name	Attributes Type
	row/column/name name CSS/JScript link number/String	class id href scr value
Methods	Method Name	Description
	-	-
Algorithm	<p>Start</p> <p>The user clicks on the notification icon on the main interface.</p> <p>The system directly validates the user action and displays the notification messages for the user.</p> <p>End</p>	

4.5.7 Route [SDD-REQ-700-107]

Class Type	Route Class	
Responsibility	A routing that provides a powerful and flexible way to define routes for the Laravel application's web-based user interfaces.	
Attributes	Attributes Name	Attributes Type
	-	-
Methods	Method Name	Description
	get()	A method provided by the 'Route' class in Laravel that registers a

		new route for HTTP GET requests.
Algorithm	<p>Start</p> <p>The system receives the information navigation from the boundary class interface.</p> <p>The system validates the data, if data needs to be sent to the Controller, it will be sent to the controller first. Otherwise, it will display the data to the interface.</p> <p>End</p>	

4.5.8 InventoryModel [SDD-REQ-700-108]

Class Type	Entity Class	
Responsibility	The structure and behaviour of the entity interacting with the database through an associated Eloquent model.	
Attributes	Attributes Name	Attributes Type
	item_id item_name item_price item_type item_status quantity notification	VARCHAR VARCHAR Float VARCHAR VARCHAR INT VARCHAR
Methods	Method Name	Description
	-	-
Algorithm	-	

4.5.9 InventoryController [SDD-REQ-700-109]

Class Type	Control Class	
Responsibility	A package that controls the flow of data between the model and the view.	
Attributes	Attributes Name	Attributes Type
	item_id	VARCHAR
Methods	Method Name	Description
	addInventory() updateInventory() viewInventory() deleteInventory() generateNotification()	To connect the item id between the database and view.
Algorithm	<p>Start</p> <p>The system directly navigates the interface from Route to the Controller.</p> <p>The system validates data and displays data to the interface.</p> <p>End</p>	

4.5.10 Authenticate [SDD-REQ-700-110]

Class Type	Middleware Class	
Responsibility	A package that authenticates incoming requests to ensure that the user is logged in before allowing them access to protected routes or resources.	
Attributes	Attributes Name	Attributes Type
	request	Request
Methods	Method Name	Description

	redirectTo()	To Get the path for users that should be redirected to when they are not authenticated.
Algorithm	-	

4.5.11 RedirectIfAuthenticated [SDD-REQ-700-111]

Class Type	Middleware Class	
Responsibility	A package that checks if a user is already authenticated and redirects them to a specified path if they are.	
Attributes	Attributes Name	Attributes Type
	Request next guard	Request Closure String
Methods	Method Name	Description
	handle()	To handle an incoming request.
Algorithm	-	

4.5.12 TrimStrings [SDD-REQ-700-112]

Class Type	Middleware Class	
Responsibility	A package that automatically trims whitespace from the beginning and end of request input data.	
Attributes	Attributes Name	Attributes Type
	current_password password password_confirmation	String String String
Methods	Method Name	Description

	-	-
Algorithm	-	

4.5.13 TrusHosts [SDD-REQ-700-113]

Class Type	Middleware Class	
Responsibility	A package that verifies that the HTTP request's host is trusted before allowing the request to proceed to the application.	
Attributes	Attributes Name	Attributes Type
	-	-
Methods	Method Name	Description
	host()	To get the host patterns that should be trusted.
Algorithm	-	

4.5.14 TrustProxies [SDD-REQ-700-114]

Class Type	Middleware Class	
Responsibility	A package that configures trusted proxies that are used to retrieve the client's IP address when the application is behind a load balancer or reverse proxy.	
Attributes	Attributes Name	Attributes Type
	headers	Request
Methods	Method Name	Description
	-	-
Algorithm	-	

4.5.15 Schema [SDD-REQ-700-115]

Class Type	Blueprint Class	
Responsibility	A package that defines and modifies database tables through the creation of migration files.	
Attributes	Attributes Name	Attributes Type
	table	Blueprint
Methods	Method Name	Description
	up() create() down()	Run migration to database To create a table in database To reverse the migration to database.
Algorithm	-	

4.5.16 DatabaseSeeder [SDD-REQ-700-116]

Class Type	Seeder Class	
Responsibility	A package that populates the database with initial or dummy data when the 'php artisan db:seed' command is executed.	
Attributes	Attributes Name	Attributes Type
	-	-
Methods	Method Name	Description
	run()	To seed the application's database.
Algorithm	-	

5. REQUIREMENTS TRACEABILITY

Requirement		Description
Source	Allocated	
RQ01-PETAKOM-MMS-2023	SDD-REQ-700-100	Manage Inventory
	SDD-REQ-700-101	Inventory Main Page
	SDD-REQ-700-102	Add New Product Page
	SDD-REQ-700-103	Update Product Details Page
	SDD-REQ-700-104	View Product Details Page
	SDD-REQ-700-105	Delete Selected Product Page
	SDD-REQ-700-106	View Inventory Notification Messages
	SDD-REQ-700-107	Route
	SDD-REQ-700-108	Inventory Model
	SDD-REQ-700-109	Inventory Controller
	SDD-REQ-700-110	Authenticate Middleware
	SDD-REQ-700-111	RedirectIfAuthenticated Middleware
	SDD-REQ-700-112	TrimStrings Middleware
	SDD-REQ-700-113	TrustHosts Middleware
	SDD-REQ-700-114	TrustProxies Middleware
	SDD-REQ-700-115	Migration Table

	SDD-REQ-700-116	Inventory Seeder
RQ02-PETAKOM-MMS-2023	SDD-REQ-100-100	Manage Profile
	SDD-REQ-100-101	Login Page
	SDD-REQ-100-102	Register Page
	SDD-REQ-100-103	Confirm Password
	SDD-REQ-100-104	Forgot Password
	SDD-REQ-100-105	Reset Password
	SDD-REQ-100-106	API Token Manager
	SDD-REQ-100-107	Index
	SDD-REQ-100-108	Delete User Form
	SDD-REQ-100-109	Logout From Other Browser Session
	SDD-REQ-100-110	Authenticate
	SDD-REQ-100-111	Two Factor Authentication Form
	SDD-REQ-100-112	Update Password Form
	SDD-REQ-100-113	Update Profile Information Form
	SDD-REQ-100-114	Route
	SDD-REQ-100-115	User Model
	SDD-REQ-100-116	User Controller
	SDD-REQ-100-117	Authenticate Middleware
	SDD-REQ-100-118	RedirectIfAuthenticated Middleware
	SDD-REQ-100-119	TrimStrings Middleware

	SDD-REQ-100-120	TrustHosts Middleware
	SDD-REQ-100-121	TrustProxies Middleware
	SDD-REQ-100-122	Migration Table
	SDD-REQ-100-123	Profile Seeder
RQ03-PETAKOM-MMS-2023	SDD-REQ-600-100	Manage Report
	SDD-REQ-600-101	Report Admin Main Page
	SDD-REQ-600-102	Report Coordinator Main Page
	SDD-REQ-600-103	View Report Detail
	SDD-REQ-600-104	Route
	SDD-REQ-600-105	Report Model
	SDD-REQ-600-106	Report Controller
	SDD-REQ-600-107	Authenticate Middleware
	SDD-REQ-600-108	RedirectIfAuthenticated Middleware
	SDD-REQ-600-109	TrimStrings Middleware
	SDD-REQ-600-110	TrustHosts Middleware
	SDD-REQ-600-111	TrustProxies Middleware
	SDD-REQ-600-112	Migration Table
	SDD-REQ-600-113	Report Seeder
RQ04-PETAKOM-MMS-2023	SDD-REQ-400-100	Manage Payment
	SDD-REQ-400-101	Add item page
	SDD-REQ-400-102	Remove item page
	SDD-REQ-400-103	Edit item page

	SDD-REQ-400-104	Admin view payment page
	SDD-REQ-400-105	Cashier view payment page
	SDD-REQ-400-106	Payment type page
	SDD-REQ-400-107	Route
	SDD-REQ-400-108	Payment Model
	SDD-REQ-400-109	Payment Controller
	SDD-REQ-400-110	Authenticate Middleware
	SDD-REQ-400-111	RedirectIfAuthenticated Middleware
	SDD-REQ-400-112	TrimStrings Middleware
	SDD-REQ-400-113	TrustHosts Middleware
	SDD-REQ-400-114	TrustProxies Middleware
	SDD-REQ-400-115	Migration Table
	SDD-REQ-400-116	Payment Seeder
RQ05-PETAKOM-MMS-2023	SDD-REQ-500-100	Manage Duty Roster
	SDD-REQ-500-101	Duty Roster Main Page
	SDD-REQ-500-102	Duty Roster Add Schedule Page
	SDD-REQ-500-103	Duty Roster Edit Schedule Page
	SDD-REQ-500-104	Duty Roster Delete Schedule Page
	SDD-REQ-500-105	View Duty Roster Detail
	SDD-REQ-500-106	Route

	SDD-REQ-500-107	Duty Roster Model
	SDD-REQ-500-108	Duty Roster Controller
	SDD-REQ-500-109	Authenticate Middleware
	SDD-REQ-500-110	RedirectIfAuthenticated Middleware
	SDD-REQ-500-111	TrimStrings Middleware
	SDD-REQ-500-112	TrustHosts Middleware
	SDD-REQ-500-113	TrustProxies Middleware
	SDD-REQ-500-114	Migration Table
	SDD-REQ-500-115	Duty Roster Seeder