



SDD Document

SEM II 2020/2021

Dercs Computer Repair Shop Management System

Group Name

- | | |
|---------------------------|-----------|
| 1. TAN PEN THE | [CB18037] |
| 2. LIEW YOON HONG | [CB18112] |
| 3. YAP JIA XING | [CB18104] |
| 4. CHOONG KIT HUEY | [CB18099] |
| 5. MATHIARASU A/L GANESAN | [CB18055] |
| 6. NG KOK THAI | [CB18109] |

TABLE OF CONTENTS

1. Chapter 1	3
1.1 Purpose	3
1.2 Module Overview	3
2. Referenced document	5
3. General architecture	6
3.1 Application Layer	6
3.1.1 3-Tier Architecture Layer	6
3.1.1.1 Manage Registration [SDD-REQ-100] (NG KOK THAI CB18109)	6
3.1.1.2 Manage Account Service [SDD-REQ-200] (CHOONG KIT HUEY CB18099)	7
3.1.1.3 Manage Request [SDD-REQ-300] (YAP JIA XING CB18104)	8
3.1.1.4 Manage Delivery [SDD-REQ-400] (MATHIARASU A/L GANESAN CB18055)	9
3.1.1.5 Manage Repair Service [SDD-REQ-500] (LIEW YOON HONG CB18112)	10
3.1.1.6 Manage Payment [SDD-REQ-600] (TAN PEN THE CB18037)	11
3.1.2 Framework	12
3.2 Package Module	13
3.3 Data Dictionary	14
4. DETAIL DESIGN	16
4.1 Manage Registration [SDD-REQ-100] (NG KOK THAI CB18109)	16
4.2 Manage Profile [SDD-REQ-200] (CHOONG KIT HUEY CB18099)	19
4.3 Manage Request Service [SDD-REQ-300] (YAP JIA XING CB18104)	24
4.4 Manage Delivery [SDD-REQ-400] (MATHIARASU A/L GANESAN CB18055)	27
4.5 Manage Repair Service [SDD-REQ-500] (LIEW YOON HONG CB18112)	35
4.6 Manage Payment [SDD-REQ-600]	41
5. REQUIREMENT TRACEABILITY	44
5.1 Manage Registration [SRS-REQ-100] NG KOK THAI (CB18109)	44
5.2 Manage Profile [SRS-REQ-200] CHOONG KIT HUEY [CB8099]	44
5.3 Manage Request Service [SRS-REQ-300] YAP JIA XING [CB18104]	47
5.4 Manage Delivery [SRS-REQ-400] MATHIARASU A/L GANESAN [CB18055]	47
5.5 Manage Repair Service [SRS-REQ-500] LIEW YOON HONG [CB18112]	49
5.6 Manage Payment [SRS-REQ-600] TAN PEN THE [CB18037]	52

1. CHAPTER 1

1.1 Purpose

The purpose of this document is to give a detailed description of the requirements for Dercs Computer Repair Shop Management System where the system will help the Dercs Computer Repair shop owner to better run and monitor the business systematically. In this document, the purpose and complete declaration for the development of Dercs Computer Repair Shop Management System will be illustrated clearly. This document describes the project's targeted audience and its user interfaces, hardware and software requirements.

This document is intended for both the stakeholders and the system developer as a reference to have better understanding regarding the system and help in the development of the first version of this system. It will outline how our client, team and audience see the product's functionality. It also helps any designer and developer to assist in software development life cycle (SDLC) processes.


1.2 Module Overview

Dercs Computer Repair Shop Management System is a web-based application with the main goal of providing devices repair service. The objective of the system is to help the Dercs Computer Repair Shop to better handle the business process, manage the repairing services and offer to pick up the devices from a customer's home and also deliver it directly to the customer. The system will allow the users to create an account on their respective purpose which is customer and rider. There are 6 primary modules in this Dercs Computer Repair Shop Management System which are:

- 1) **Manage Registration:** This module allows the users to register an account in the system. Customer and rider are required to register and login to the system before carrying out any further activity.
- 2) **Manage Profile:** This module allows the users to manage their account profile. For the customer and rider, they are allowed to edit their information after the registration. For staff, they have the right to edit and delete or ban the customer account.
- 3) **Manage Repair Request:** This module allows the customer to request for the service. The customer is the only actor that can request a repair service. They need to provide the required information for making the request.

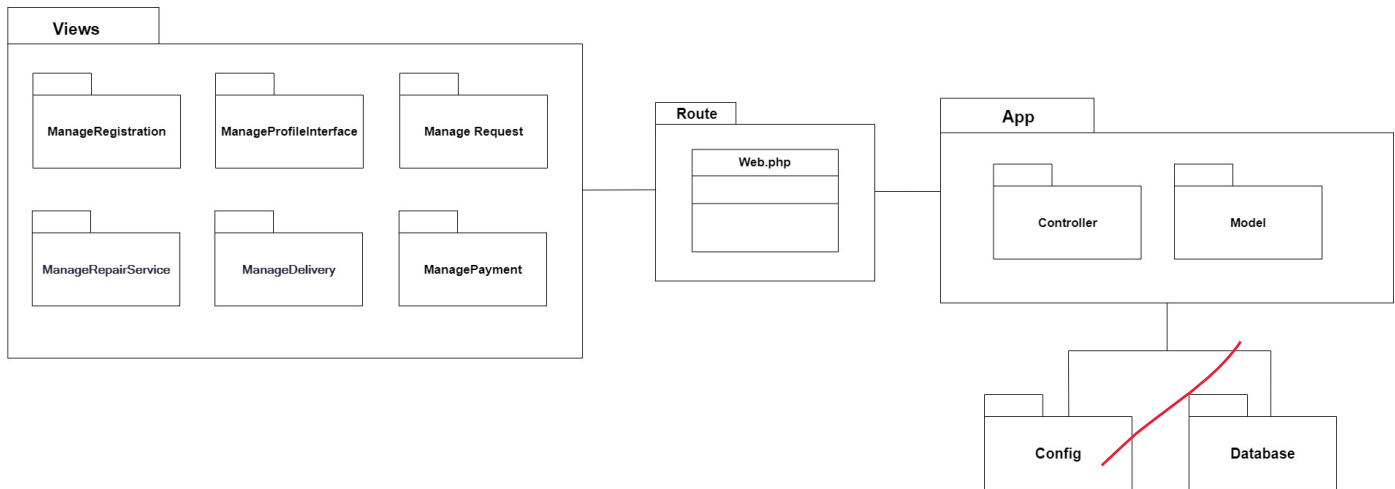
- 4) **Manage Delivery:** This module allows the users to manage the delivery. For customers and staff, they can view the delivery process of the devices. For the rider, they are allowed to access customer information that made a request. So, they can make the pickup and deliver for the devices.
- 5) **Manage Repair Service:** This module allows the users to manage their repairing service status by viewing and updating. For the customer, they can view the status of repairment, the repairment progress details, and the estimated cost for the repairment. For staff, they need to update the status of the repairing progress in the system.
- 6) **Manage Payment:** This module allows the users to proceed with the payment of the repairing cost. In this use case, a payment gateway is used to manage the payment from the customer. The payment gateway will receive the payment detail from the system and wait for the response for further action. The customer has to choose the payment type before proceeding to further action.

2. REFERENCED DOCUMENT

- Brown, D. D. (2002). DOCUMENT TEMPLATES FOR STUDENT PROJECTS IN SOFTWARE ENGINEERING.
- C. F. J. Lange, M. R. (2006). In practice: UML software architecture and design description. *IEEE Software*, vol. 23, no. 2, pp. 40-46.
- Chao-Hsien Hsieh, Z. W.-H. (2020). Development of Laravel Digital Platform Based on MVC Design Pattern for Complicated Data Structure-Take the Bible for Example. *3rd IEEE International Conference on Information Communication and Signal Processing*.
- Christopher Barnes, P. V. (2009). An XML-Based ADL Framework for Automatic Generation of Multithreaded Computer Architecture Simulators.
- Keith Gallagher, M. I. (2008). Software Architecture Visualization: An Evaluation Framework and Its Application. *IEEE TRANSACTIONS ON SOFTWARE ENGINEERING*, VOL. 34, NO. 2.
- Lars Ericson, P. S. (2015). Minutia Deviation Tool: Software Design Description (SDD), (Version 2.0).
- Muhammad Anif, A. D. (2017). Designing Internship Monitoring System Web Based With Laravel Framework . *IEEE International Conference on Communication, Networks and Satellite (Comnetsat)*.
- 

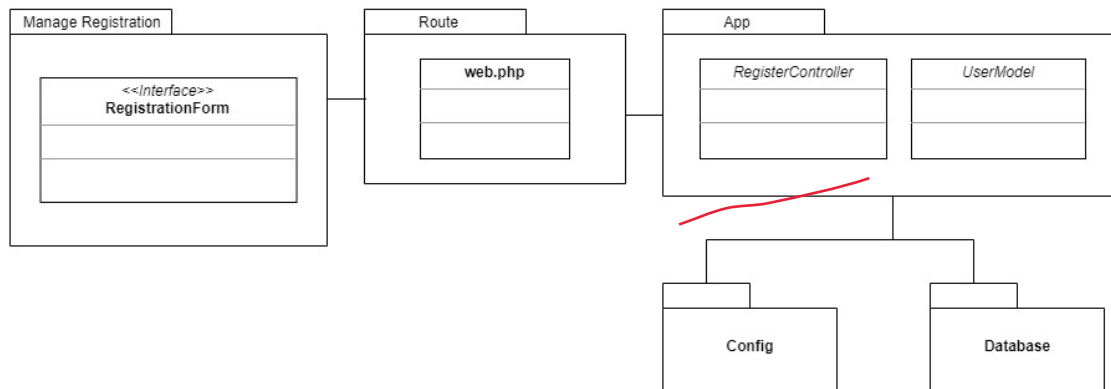
3. GENERAL ARCHITECTURE

3.1 Application Layer



3.1.13-Tier Architecture Layer

3.1.1.1 Manage Registration [SDD-REQ-100] (NG KOK THAI CB18109)



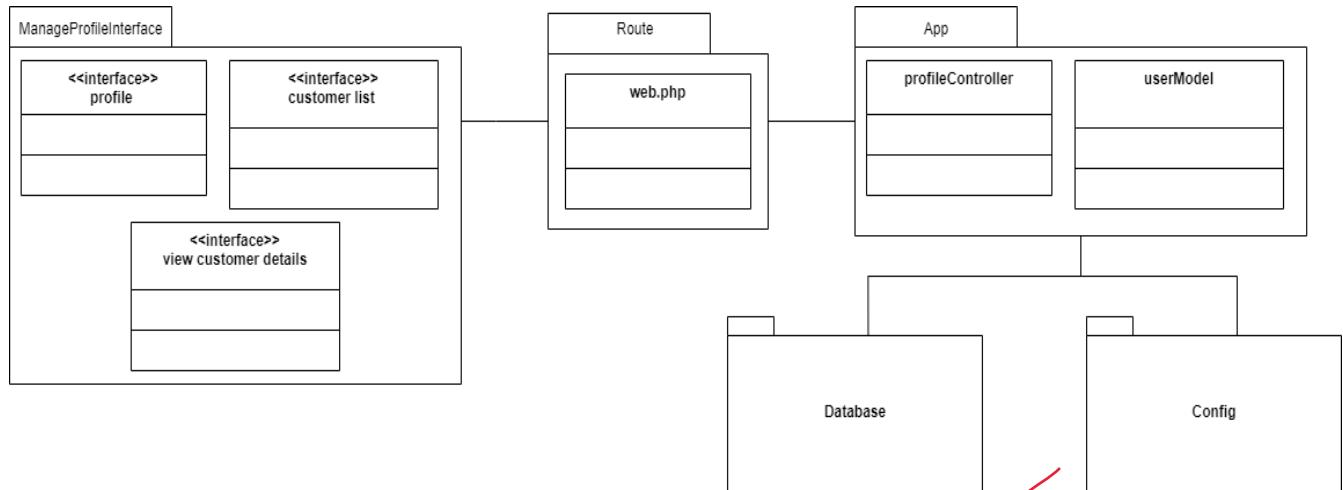
3.1.1.1.1 ManageRegistrationInterface

Class Name	Description
RegistrationForm	This class will allow the user to register as customer or rider by filling the details required in the system.

3.1.1.1.2 App

Class Name	Description
RegisterController	The controller that is used to manage the function for the manage registration interface.
UserModel	To store and retrieve the user information in the database.

3.1.1.2 Manage Account Service [SDD-REQ-200] (CHOONG KIT HUEY CB18099)



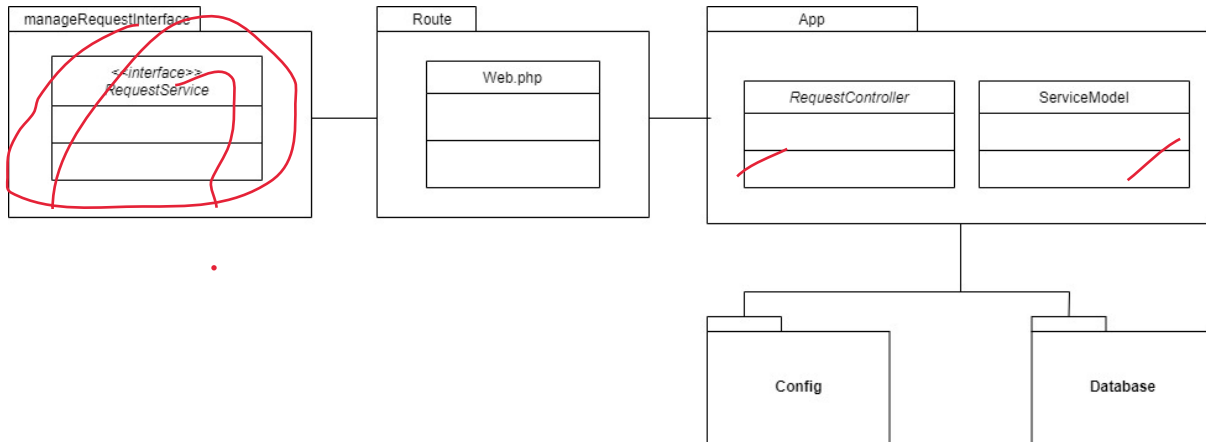
3.1.1.2.1 ManageProfileInterface

Class Name	Description
profile	This class will allow the customer and rider to see their personal information and able to edit their details
customer list	This class will allow the staff to view the list of the customer in that had registered on the system
View customer details	This class will allow the staff to view all the details from the customer and can allow the staff ban and delete the customer

3.1.1.2.2 App

Class Name	Description
profileControlller	The profile controller that is used to manage the functions for the manage profile interface.
userModel	The user model that is used to store and retrieve from the database

3.1.1.3 Manage Request [SDD-REQ-300] (YAP JIA XING CB18104)



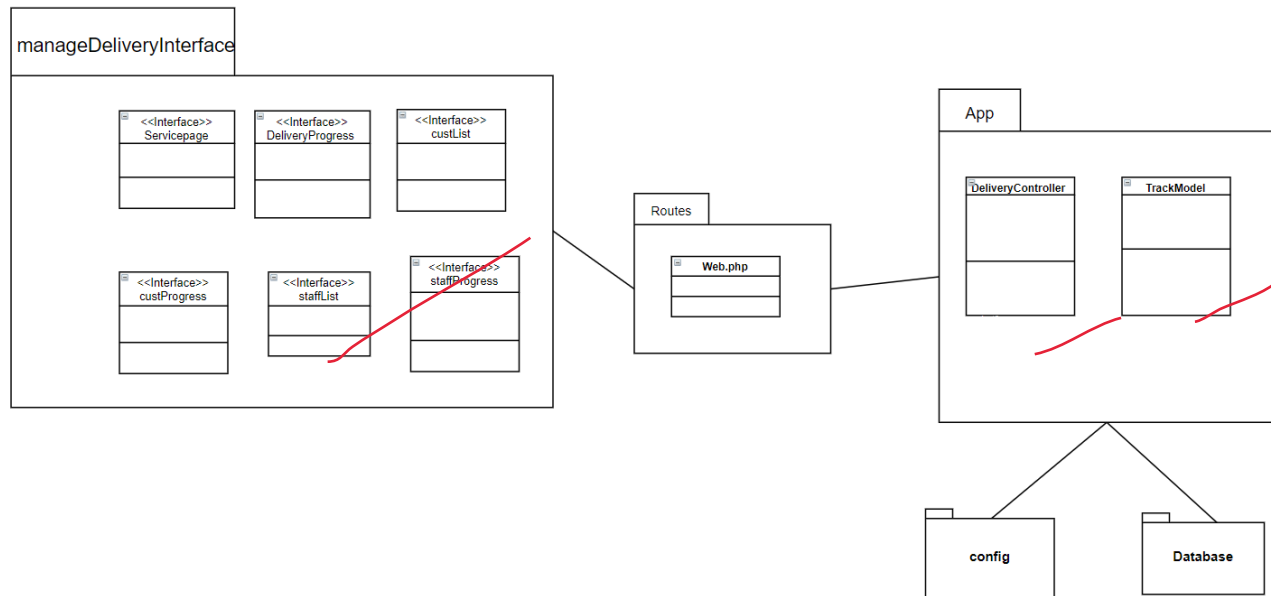
3.1.1.3.1 ManageRequestInterface

Class Name	Description
RequestService	This class will allow the customer for requesting the repair service for their devices.

3.1.1.3.2 App

Class Name	Description
RequestController	The controller that is used to manage the customer's request for repairing their devices.
ServiceModel	To store and retrieve service information.

3.1.1.4 Manage Delivery [SDD-REQ-400] (MATHIARASU A/L GANESAN CB18055)



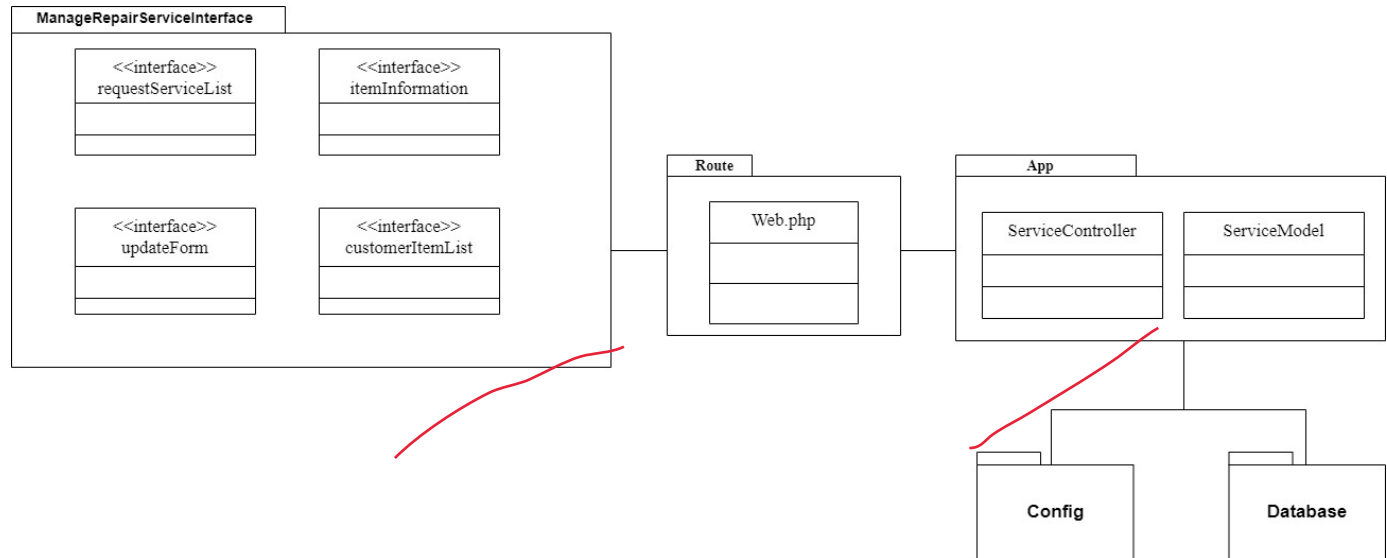
3.1.1.4.1 ManageDeliveryInterface

Class Name	Description
Servicepage	This class will allow rider to view the unaccepted task and accepted task.
Deliveryprogress	This class will allow rider to view the progress and update the delivery progress.
custList	This class will allow to customer to view all tracking their device.
custProgress	This class will allow to customer to view tracking progress.
staffList	This class will allow to staff to view all tracking their device.
staffProgress	This class will allow to staff to view tracking progress.

3.1.1.4.2 App

Class Name	Description
DeliveryController	This controller is used to communicate between the delivery database and interface.
TrackModel	To store and retrieve delivery information in the database.

3.1.1.5 Manage Repair Service [SDD-REQ-500] (LIEW YOON HONG CB18112)



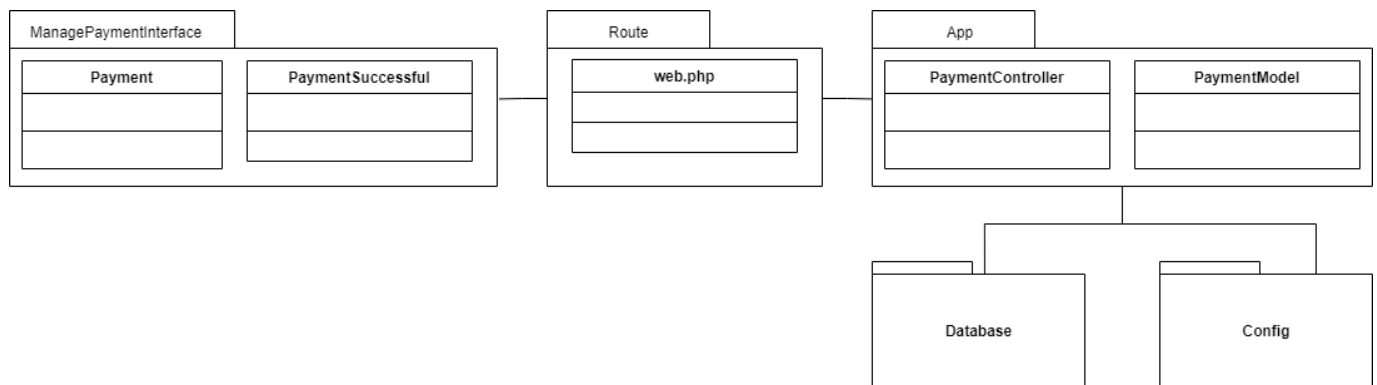
3.1.1.5.1 ManageRepairServiceInterface

Class Name	Description
requestServiceList	This class is to allows the staff to view the customer request in the request service list interface and select the item to view the information.
itemInformation	This class is to allows the staff and customer to view the item information of the device.
updateForm	This class is to allows the staff to update the item information after inspection in the repair service form.
customerItemList	This class is to allows the customer to view their requested service list and select the item to view the information.

3.1.1.5.2 App

Class Name	Description
ServiceController	The controller that is used to manage the functions for the manage repair service interface.
ServiceModel	To store and retrieve repair service information in the database.

3.1.1.6 Manage Payment [SDD-REQ-600] (TAN PEN THE CB18037)



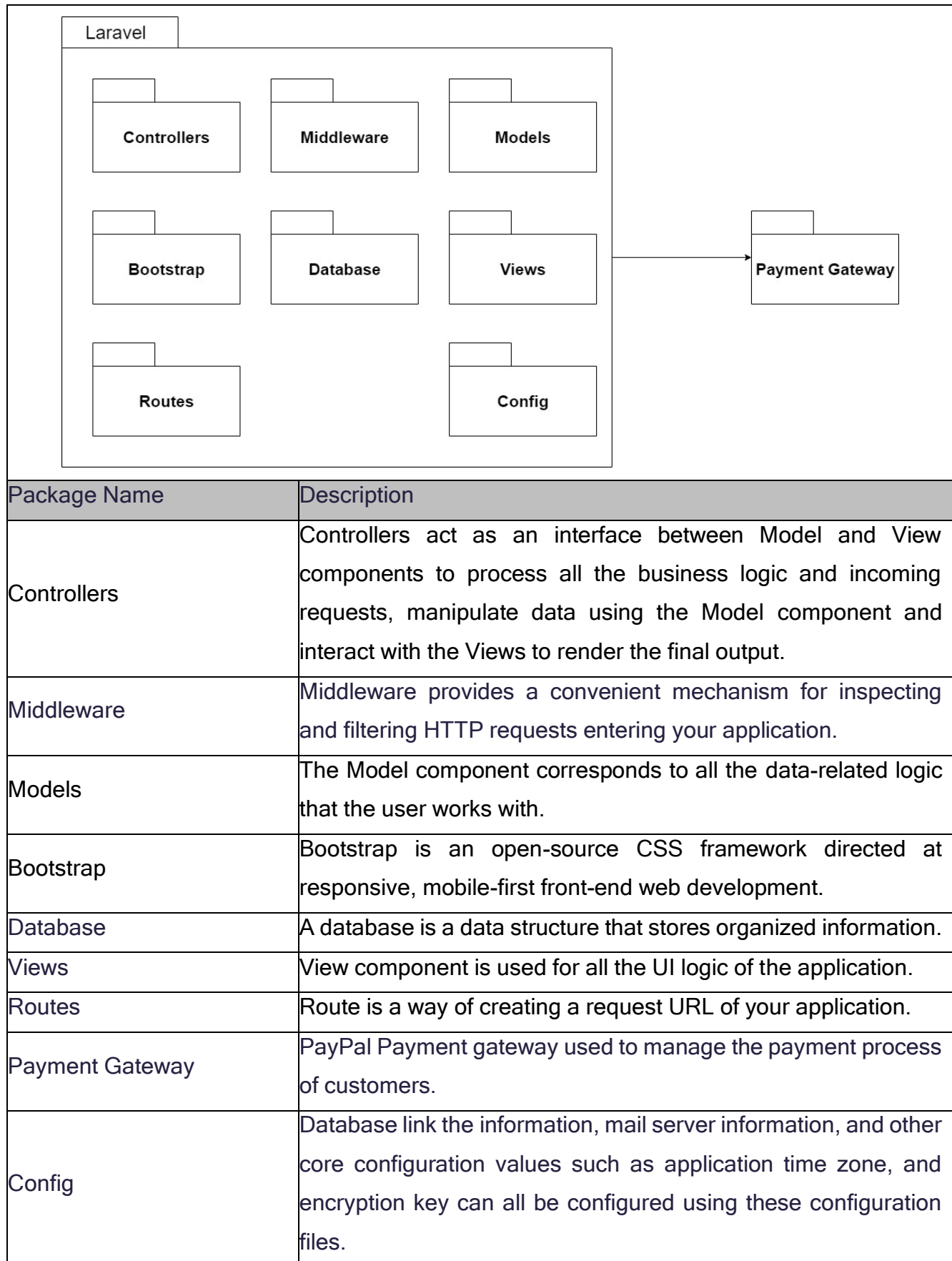
3.1.1.6.1 ManagePaymentInterface

Class Name	Description
PaymentMethod	Customer can view the total price of the service and choose his/her payment method. Customer will be directed to PayPal payment gateway if PayPal payment is chosen.
PaymentSuccessful	Customer will be directed to this page after the payment method is selected or the payment is made.

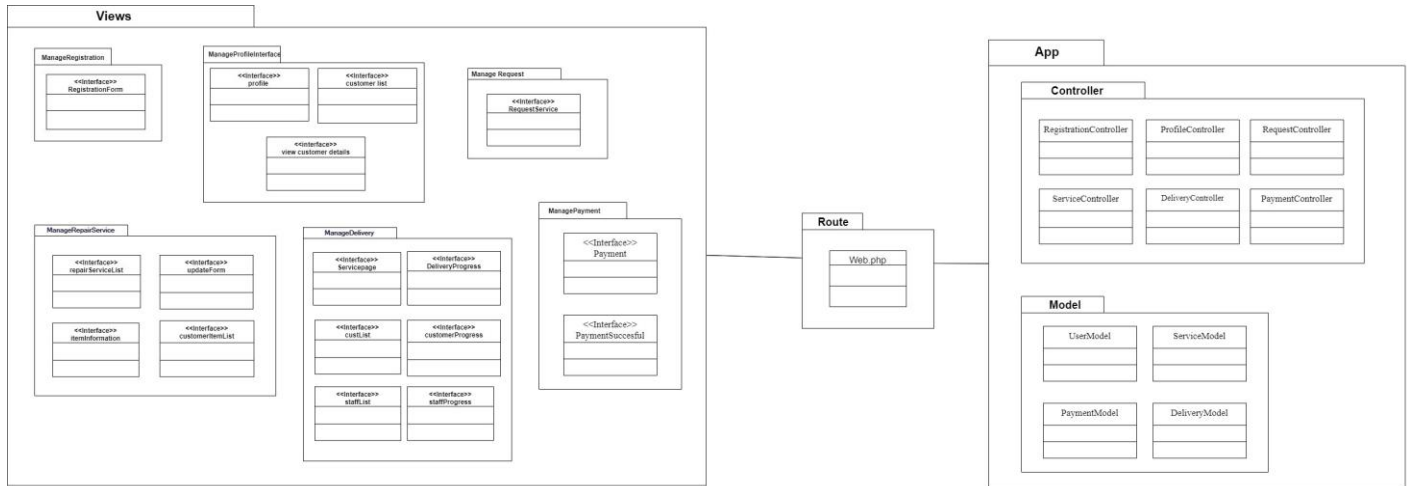
3.1.1.6.2 App

Class Name	Description
PaymentController	The controller that is used to manage the functions for the manage payment interface.
PaymentModel	To retrieve and store repair service information and payment information in the database.

3.1.2 Framework



3.2 Package Module



3.3 Data Dictionary

3.3.1 users

Field Name	Description	Data Type	Constraint
userID	User ID	INTEGER	PK
username	Username	VARCHAR2(50)	
password	password	VARCHAR2(50)	
email	Email	VARCHAR2(50)	
userType	User Type	VARCHAR2(50)	
phoneNo	Phone Number	VARCHAR2(50)	
address	Address	VARCHAR2(50)	
licensephoto	License's photo	LONGBLOB	

3.3.2 services

Field Name	Description	Data Type	Constraint
serviceID	Service ID	INTEGER	PK
userID	User ID	INTEGER	FK
device	Chosen device from customer	VARCHAR2(100)	
symptom	Symptom of the device	VARCHAR2(100)	
repairStatus	Is the device repairable	VARCHAR2(100)	
repairProgress	The progress of the repairing device	VARCHAR2(100)	
detail	Detail of the repairing device provided by repairing staff	VARCHAR2(100)	
reason	Reason of the repair progress	VARCHAR2(100)	
estimateCost	Cost estimated by repairing staff	VARCHAR2(100)	

3.3.3 trackings

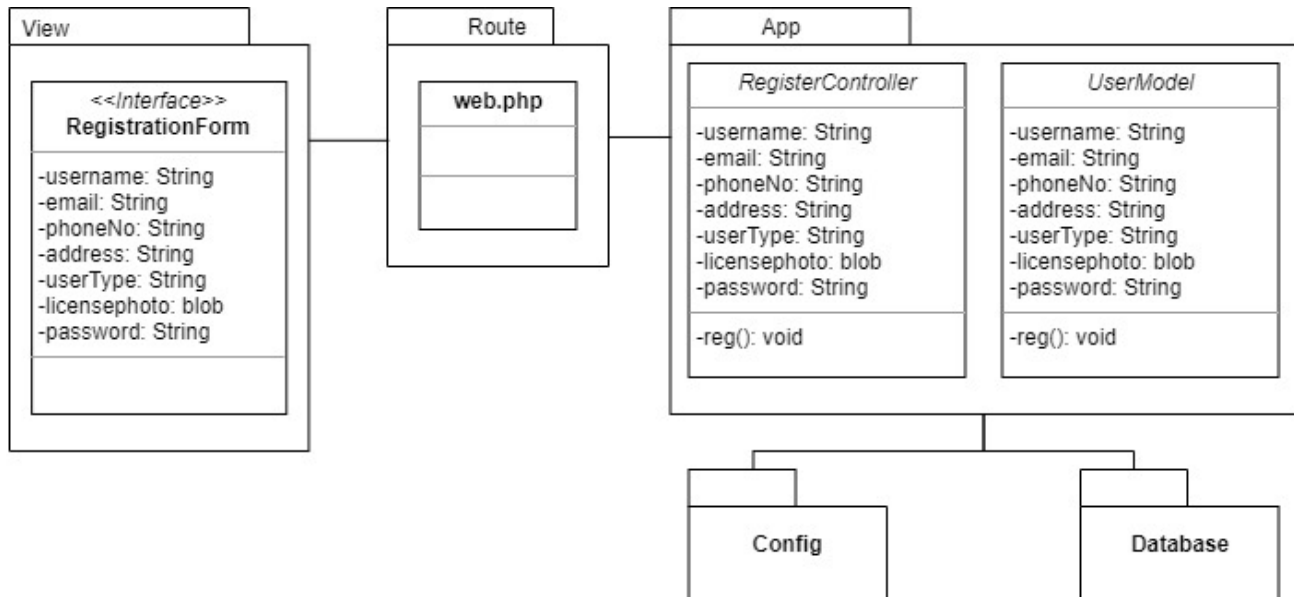
Field Name	Description	Data Type	Constraint
trackingID	Tracking ID of the service	INTEGER	PK
userID	User ID	INTEGER	FK
serviceID	Service ID	INTEGER	FK
pickupAddress	Address to pick up device	VARCHAR2(100)	
trackdate	Date of tracking	DATETIME	
tracktime	Time of tracking	DATETIME	
trackprogress	Progress of tracking	VARCHAR2(100)	

3.3.4 payments

Field Name	Description	Data Type	Constraint
paymentID	Payment ID of the service	INTEGER	PK
serviceID	Service ID	INTEGER	FK
userID	User ID	INTEGER	FK
paymentMethod	Chosen payment method	VARCHAR2(50)	
total	Total of payment	DOUBLE	
time	Time of made payment	DATETIME	

4. DETAIL DESIGN

4.1 Manage Registration [SDD-REQ-100] (NG KOK THAI CB18109)



4.1.1 RegistrationForm [SDD-REQ-101]

Class Type	Boundary class	
Responsibility	An interface to display the registration form for the user to fill in during registration.	
Attributes	Attributes Name	Attributes Type
	username	String
	email	String
	phoneNo	String
	address	String
	userType	String
	licensephoto	blob
Methods	Method Name	Description
	Not applicable	Not applicable
Algorithm	Not applicable	

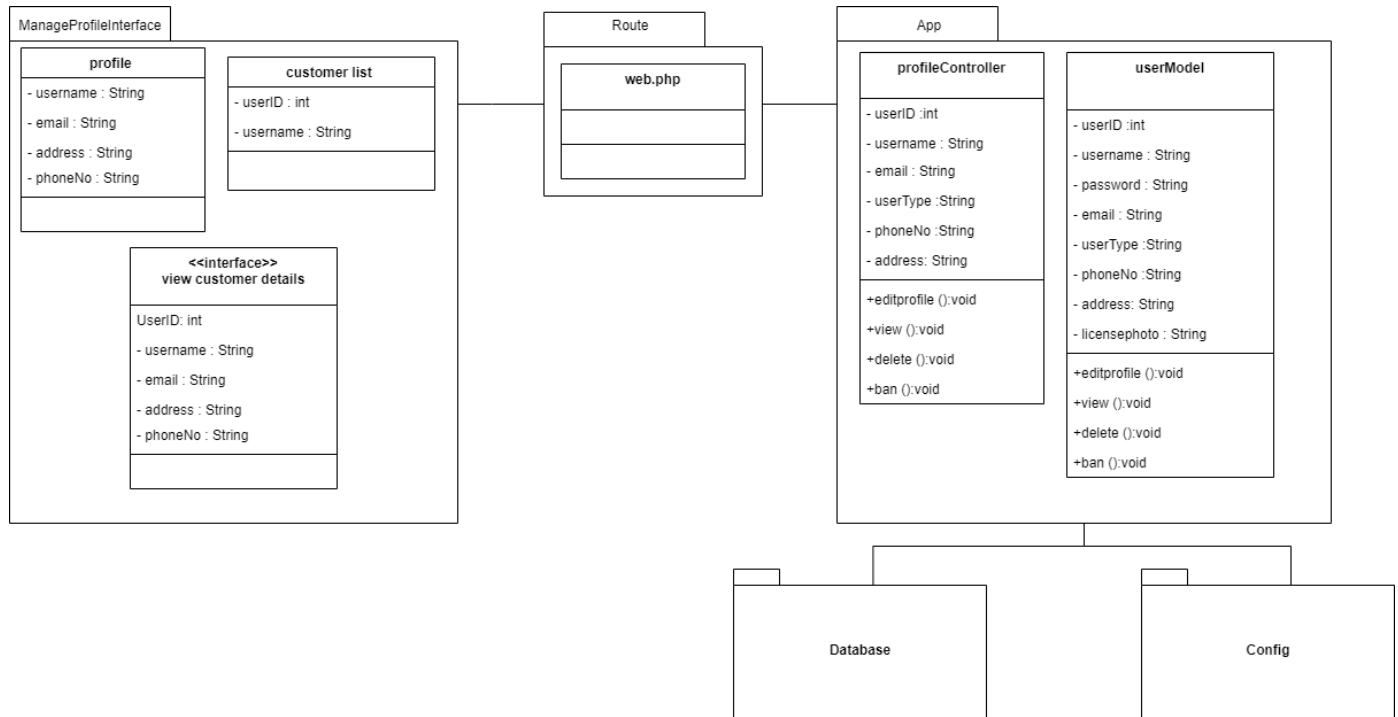
4.1.2 RegisterController [SDD-REQ-102]

Class Type	Controller class	
Responsibility	To connect UserModel and ManageRegistrationInterface.	
Attributes	Attributes Name	Attributes Type
	username	String
	email	String
	phoneNo	String
	address	String
	userType	String
	licensephoto	blob
	password	String
Methods	Method Name	Description
	reg()	This method allows the user to register as customer or rider in the system.
Algorithm	<pre> START IF reg() is called IF usertype == cust THEN create object customer to connect to UserModel RETURN to registration form ELSE IF usertype == rider THEN create object rider to connect to UserModel RETURN to registration form ELSE System displays homepage END IF END </pre>	

4.1.3 UserModel [SDD-REQ-102]

Class Type	Model class	
Responsibility	To connect, store, retrieve and manage the database of user information.	
Attributes	Attributes Name	Attributes Type
	username	String
	email	String
	phoneNo	String
	address	String
	userType	String
	licensephoto	blob
	password	String
Methods	Method Name	Description
	reg()	This method allows the user to register as customer or rider in the system.
Algorithm	<pre> START IF reg() is called IF usertype == cust Insert username, email, phoneNo, address, password into customer table. Create object cust to connect and create SQL Execute object cust RETURN cust ELSE IF usertype == rider Insert username, email, phoneNo, address, licensephoto, password into rider table. Create object rider to connect and create SQL Execute object rider RETURN rider ELSE System displays homepage END IF END </pre>	

4.2 Manage Profile [SDD-REQ-200] (CHOONG KIT HUEY CB18099)



4.2.1 profile [SDD-REQ-201]

Class Type	Boundary class	
Responsibility	An interface to display the username, email, address and phone number of users.	
Attributes	Attributes Name	Attributes Type
	username	String
	email	String
	address	String
	phoneNo	String
Methods	Method Name	Description
	Not applicable	Not applicable
Algorithm	Not applicable	

4.2.2 customer list [SDD-REQ-202]

Class Type	Boundary class	
Responsibility	This class is responsible to display the list of all customer for the staff to view	
Attributes	Attributes Name	Attributes Type
	userID	int
	username	String
Methods	Method Name	Description
	Not applicable	Not applicable
Algorithm	Not applicable	

4.2.3 view customers details [SDD-REQ-203]

Class Type	Boundary class	
Responsibility	This class is responsible to display the customer personal details to the staff to view	
Attributes	Attributes Name	Attributes Type
	userID	int
	username	String
	email	String
	address	String
	phoneNo	String
Methods	Method Name	Description
	Not applicable	Not applicable
Algorithm	Not applicable	

4.2.3 profileController [SDD-REQ-204]

Class Type	Controller class	
Responsibility	This class is used to communicate between the interface and model, used to handle the edit profile, view profile details, delete account, ban account, retrieve data and return responses.	
Attributes	Attributes Name	Attributes Type
	userID	int
	username	String
	email	String
	userType	String
	address	String
	phoneNo	String
Methods	Method Name	Description
	Editprofile ()	The user can make edit or update their profile
	View ()	To display the details of the customer to staff
	Delete ()	The staff can delete the account of the customer
	Ban ()	The staff can ban the account of the customer
Algorithm	EditProfile () START IF EditProfile () is called THEN update and store data of username, email, phoneNo, address in the model ELSE send error message END IF END View () START IF View () is called THEN display the details of the customer END IF END	

	<p>Delete ()</p> <p>START</p> <p>IF Delete () is called</p> <p>THEN send a confirmation message</p> <p>delete data of the customer</p> <p>ELSE</p> <p>Return to the view customer details interface</p> <p>END IF</p> <p>END</p> <p>Ban ()</p> <p>START</p> <p>IF Ban () is called</p> <p>THEN send a confirmation message</p> <p>ban the customer</p> <p>ELSE</p> <p>Return to the view customer details interface</p> <p>END IF</p> <p>END</p>
--	---

4.2.5 profileModel [SDD-REQ-205]

Class Type	Model class	
Responsibility	This model is used to save data to the database and retrieve request data from the database.	
Attributes	Attributes Name	Attributes Type
	userID	int
	Username	String
	password	String
	Email	String
	userType	String
	address	String

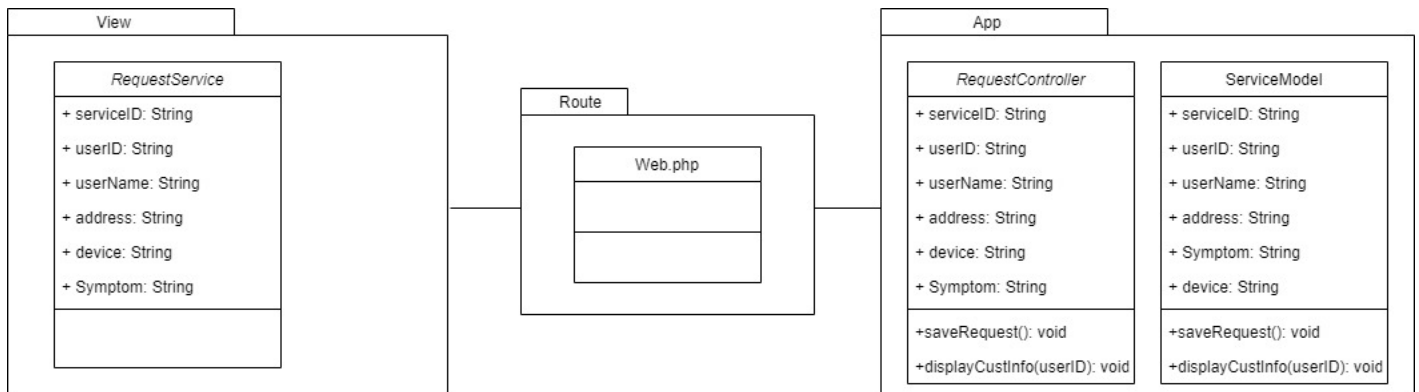
	phoneNo liensephoto	String
Methods	Method Name	Description
	Editprofile ()	The user can make edit or update their profile
	View ()	To display the details of the customer to staff
	Delete ()	The staff can delete the account of the customer
	Ban ()	The staff can ban the account of the customer
Algorithm	EditProfile () START IF EditProfile () is called THEN update and store data of username, email, phoneNo, address in the model ELSE send error message END IF END View () START IF View () is called THEN display the details of the customer END IF END Delete () START IF Delete () is called THEN send a confirmation massage delete data of the customer ELSE Return to the view customer details interface END IF END	

```

Ban ()
START
IF Ban () is called
THEN send a confirmation message
ban the customer
ELSE
Return to the view customer details interface
END IF
END

```

4.3 Manage Request Service [SDD-REQ-300] (YAP JIA XING CB18104)



4.3.1 RequestService [SDD-REQ-301]

Class Type	Boundary Class	
Responsibility	This class is responsible for customer to request repair service request.	
Attributes	Attributes Name	Attributes Type
	serviceID	String
	userID	String
	userName	String
	address	String
	device	String
	symptom	String
Methods	Method Name	Description
	Not applicable	Not applicable
Algorithm	Not applicable	

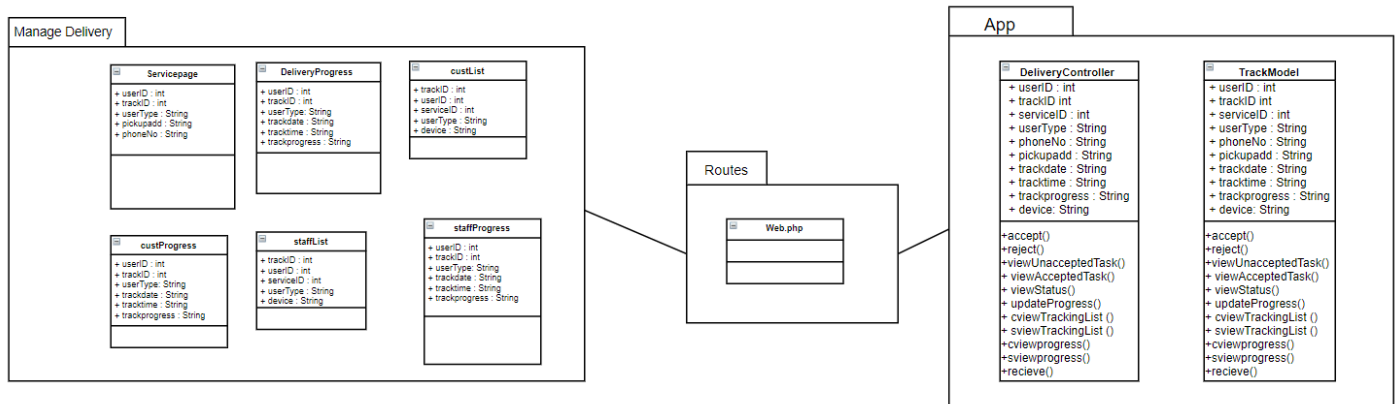
4.3.2 RequestController [SDD-REQ-302]

Class Type	Controller Class	
Responsibility	A controller class that handles the saveRequest, displayCustInfo, retrieves model data and return appropriate responses	
Attributes	Attributes Name	Attributes Type
	serviceID	String
	userID	String
	userName	String
	address	String
	device	String
	symptom	String
Methods	Method Name	Description
	saveRequest()	To save the information of the repair request submit by the user.
	displayCustInfo()	To display the name and address that register by the user
Algorithm	displayCustInfo () START displayCustInfo() is called create object service to connect ServiceModel RETURN to requestService interface END saveRequest() START IF saveRequest() is called THEN create object service to connect ServiceModel END IF RETURN to requestService interface END	

4.3.3 ServiceModel [SDD-REQ-303]

Class Type	Model Class	
Responsibility	A model class to store the details of the request made by customer.	
Attributes	Attributes Name	Attributes Type
	serviceID	String
	userID	String
	userName	String
	address	String
	device	String
	symptom	String
Methods	Method Name	Description
	saveRequest()	To save the information of the repair request submit by the user.
	displayCustInfo()	To display the name and address that register by the user
Algorithm	<pre> displayCustInfo () START IF User userID == userID THEN send data of userName, address, userID PRINT in RequestService interface page END saveRequest() START READ device READ symptom WHILE device.length == 0 symptom.length == 0 PRINT "Please fill in all required information!" END WHILE PRINT "Your request is submitted" SAVE serviceID, userID, address, device, and symptom END </pre>	

4.4 Manage Delivery [SDD-REQ-400] (MATHIARASU A/L GANESAN CB18055)



4.4.1 ServicePage [SDD-REQ-401]

Class Type	Boundary class	
Responsibility	An interface to display the all task that accepted and unaccepted. And need to accept task or reject task.	
Attributes	Attributes Name	Attributes Type
	userID	Int
	trackID	Int
	userType	String
	pickupadd	String
	phoneNo	String
Methods	Method Name	Description
	No applicable	No applicable
Algorithm	No applicable	

4.4.2 DeliveryProgress [SDD-REQ-402]

Class Type	Boundary class	
Responsibility	An interface to display the delivery status and update delivery progress.	
Attributes	Attributes Name	Attributes Type
	userID	Int
	trackID	Int
	userType	String
	trackdate	String
	tracktime	String

	trackprogress	String
Methods	Method Name	Description
	No applicable	No applicable
Algorithm	No applicable	

4.4.3 custList [SDD-REQ-403]

Class Type	Boundary class	
Responsibility	An interface to display the item list that want to track and need to select item need to track.	
Attributes	Attributes Name	Attributes Type
	trackID	int
	userID	int
	serviceID	int
	userType	String
	device	String
Methods	Method Name	Description
	No applicable	No applicable
Algorithm	No applicable	

4.4.4 custProgress [SDD-REQ-404]

Class Type	Boundary class	
Responsibility	An interface the system will display delivery progress for customer.	
Attributes	Attributes Name	Attributes Type
	userID	Int
	trackID	Int
	userType	String
	trackdate	String
	tracktime	String
	trackprogress	String
Methods	Method Name	Description

	No applicable	No applicable
Algorithm	No applicable	

4.4.5 staffList [SDD-REQ-405]

Class Type	Boundary class	
Responsibility	An interface to display the item list that want to track and need to select device need to track.	
Attributes	Attributes Name	Attributes Type
	trackID	int
	userID	int
	serviceID	int
	userType	String
	device	String
Methods	Method Name	Description
	No applicable	No applicable
Algorithm	No applicable	

4.4.6 staffProgress [SDD-REQ-406]

Class Type	Boundary class	
Responsibility	An interface to display the delivery process and need to confirm after receive order.	
Attributes	Attributes Name	Attributes Type
	userID	Int
	trackID	Int
	userType	String
	trackdate	String
	tracktime	String
	trackprogress	String
Methods	Method Name	Description
	No applicable	No applicable
Algorithm	No applicable	

4.4.7 DeliveryController [SDD-REQ-407]

Class Type	Controller class	
Responsibility	It is a connection between interface and manageDeliveryModel.	
Attributes	Attributes Name	Attributes Type
	userID	Int
	trackID	Int
	serviceID	Int
	userType	String
	phoneNo	String
	pickupadd	String
	trackdate	String
	tracktime	String
	trackprogress	String
	device	String
Methods	Method Name	Description
	viewUnacceptedTask()	Rider can view all unaccepted task.
	viewAcceptedTask()	Rider can view all accept task.
	acceptTask()	Rider can accept task to pick up and delivery task.
	rejectTask()	Rider can reject task if not want accept the task
	viewStatus()	Rider can view the status of delivery task.
	UpdateProgress()	Rider also can update delivery of progress.
	cviewTrackingList()	Customer can view all device list to track.
	sviewTrackingList()	Staff can view all device list to track.
	cviewProgress()	Customer can view the delivery progress of selected device.
	sviewProgress()	Staff can view the delivery progress of selected device.
	received()	Staff need to confirm that the device reached.

Algorithm

```

BEGIN
IF viewUnacceptedTask() is called
    THEN create object delivery list to connect TrackModel
    Return to the rider service page
ELSE IF viewAcceptedTask() is called
    THEN create object delivery list to connect TrackModel
    Return to the rider service page
ELSE IF acceptTask() is called
    THEN create object delivery task to connect TrackModel
    Return to the rider service page
ELSE IF rejectTask() is called
    THEN create object delivery task to connect TrackModel
    Return to the rider service page
ELSE IF UpdateProgress() is called
    THEN create object delivery progress to connect TrackModel
    Return to the rider delivery progress
ELSE IF cviewTrackingList() is called
    THEN create object track list to connect TrackModel
    Return to the customer list page
ELSE IF sviewTrackingList() is called
    THEN create object track list to connect TrackModel
    Return to the staff list page
ELSE IF cviewProgress() is called
    THEN create object track progress to connect TrackModel
    Return to the customer progress page
ELSE IF sviewProgress() is called
    THEN create object track progress to connect TrackModel
    Return to the staff progress page
ELSE IF received() is called
    THEN create object track progress to connect TrackModel
    Return to the staff progress page
END IF
END

```

4.4.8 TrackModel [SDD-REQ-408]

Class Type	Model class	
Responsibility	It is a model need to save data to database and retrieve data from database.	
Attributes	Attributes Name	Attributes Type
	userID	Int
	trackID	Int
	serviceID	Int
	userType	String
	phoneNo	String
	pickupadd	String
	trackdate	String
	tracktime	String
	trackprogress	String
	device	String
Methods	Method Name	Description
	viewUnacceptedTask()	Rider can view all unaccepted task.
	viewAcceptedTask()	Rider can view all accept task.
	AcceptTask()	Rider can accept task to pick up and delivery task.
	rejectTask()	Rider can reject task if not want accept the task
	viewStatus()	Rider can view the status of delivery task.
	UpdateProgress()	Rider also can update delivery of progress.
	cviewTrackingList()	Customer can view all device list to track.
	svviewTrackingList()	Staff can view all device list to track.
	cviewProgress()	Customer can view the delivery progress of selected device.
	svviewProgress()	Staff can view the delivery progress of selected device.
	received()	Staff need to confirm that the device reached.

Algorithm

viewUnacceptedTask()

Begin

SELECT * from track join user on track.userID = user.userID where
status= "Unaccepted"

Return data to interface

End

viewAcceptedTask()

Begin

SELECT * from track join user on track.userID = user.userID where
status= "Accepted"

Return data to interface

End

AcceptTask()

Begin

Update track set status = status, userID=userID where trackID trackID

Return data to interface

End

rejectTask()

Begin

Update track set status = status, userID=userID where trackID trackID

Return data to interface

End

ViewStatus()

Begin

Select * from track where trackID=trackID

Return data to interface

End

updateProgress()

Begin

Insert into track(trackID, trackDate, trackTime, trackPogress) value
(trackID, trackDate, trackTime, trackPogress)

Retrun data to interface

End

cviewTrackingList()

Begin

Select * from (track inner join service on
track.serviceID=service.serviceID) innner join serviceID where
trackprogress like 'On Delivery' and track.userID=userID.

Return data to interface

End

svviewTrackingList()

Begin

Select * from (track inner join service on
track.serviceID=service.serviceID) innner join serviceID where
trackprogress like 'On Delivery' and track.userID=userID.

Return data to interface

End

cviewProgress()

Begin

Select * from track where trackID = trackID

Return data to interface

End

svviewProgress()

Begin

Select * from track where trackID = trackID

Return data to interface

End

received()

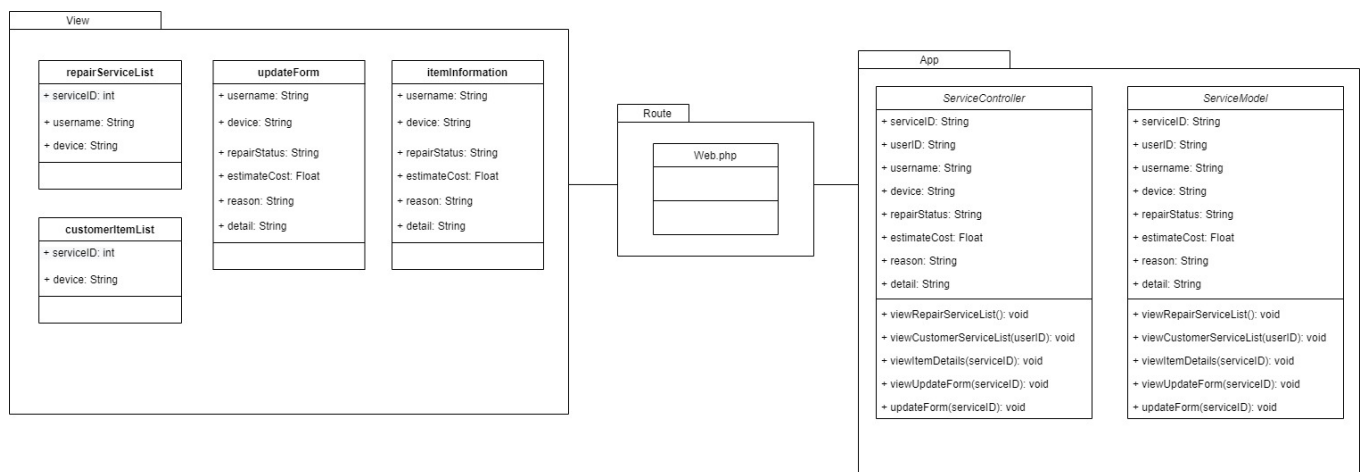
Begin

Update track set trackprogress = trackprogress where trackID = trackID

Return data to interface

End

4.5 Manage Repair Service [SDD-REQ-500] (LIEW YOON HONG CB18112)



4.5.1 repairServiceList [SDD-REQ-501]

Class Type	Boundary Class	
Responsibility	This class is responsible for staff to view the list of repairing service list of customers and select the repairing service to view in detail.	
Attributes	Attributes Name	Attributes Type
	serviceID	Integer
	username	String
	device	String
Methods	Method Name	Description
	Not Applicable	Not Applicable
Algorithm	Not Applicable	

4.5.2 customerItemList [SDD-REQ-502]

Class Type	Boundary Class	
Responsibility	This class is responsible for customer to view their requested services and select the repairing service to view in detail.	
Attributes	Attributes Name	Attributes Type
	serviceID device	Integer String
Methods	Method Name	Description
	Not Applicable	Not Applicable
Algorithm	Not Applicable	

4.5.3 itemInformation [SDD-REQ-503]

Class Type	Boundary Class	
Responsibility	This class is responsible for customer and staff to view selected repairing service request details.	
Attributes	Attributes Name	Attributes Type
	username	String
	device	String
	repairStatus	String
	estimateCost	Float
	reason detail	String String
Methods	Method Name	Description
	Not Applicable	Not Applicable
Algorithm	Not Applicable	

4.5.4 updateForm [SDD-REQ-504]

Class Type	Boundary Class	
Responsibility	This class is responsible for staff to update the information of the repairing item in the form.	
Attributes	Attributes Name	Attributes Type
	username	String
	device	String
	repairStatus	String
	estimateCost	Float
	reason	String
	detail	String
Methods	Method Name	Description
	Not Applicable	Not Applicable
Algorithm	Not Applicable	

4.5.5 serviceController [SDD-REQ-505]

Class Type	Controller Class	
Responsibility	This controller class is responsible to handle the repair service list for staff, customer item list for customers, item information for customer and staff, update form for staff, retrieves data and return insert data.	
Attributes	Attributes Name	Attributes Type
	serviceID	String
	userID	String
	username	String
	device	String
	repairStatus	String
	estimateCost	Float
	reason	String
	detail	String
Methods	Method Name	Description
	viewRepairServiceList()	This method is to display the repair service request list to staff.

	viewCustomerServiceList(userID)	This method is to display their requested service list to customer.
	viewItemDetails(serviceID)	This method is to display the details of the selected item to staff and customer.
	viewUpdateDetailForm(serviceID)	This method is to display the update form to staff.
	updateDetailForm(serviceID)	This method is to update the details entered by staff into the database.
Algorithm	<pre> BEGIN IF viewRepairServiceList() is called THEN create object service list to connect to serviceModel RETURN to staff service request list page ELSE IF viewCustomerServiceList(userID) is called THEN create object customer service list to connect to serviceModel RETURN to customer requested service list page ELSE IF viewItemDetails(serviceID) is called THEN create object item detail to connect to serviceModel RETURN to customer view item details page ELSE IF viewUpdateDetailForm(serviceID) is called THEN create update detail form to connect to serviceModel RETURN to update form page ELSE IF updateDetailForm(serviceID) is called THEN create object update detail form to connect to serviceModel RETURN to staff service request list page END IF END </pre>	

4.5.6 serviceModel [SDD-REQ-506]

Class Type	Model Class	
Responsibility	This model class is responsible to stores, retrieves, cancel, edit and manage the service data for both customer and staff.	
Attributes	Attributes Name	Attributes Type
	serviceID	String
	userID	String
	username	String
	device	String
	repairStatus	String
	estimateCost	Float
	reason	String
	detail	String
Methods	Method Name	Description
	viewRepairServiceList()	This method is to display the repair service request list to staff.
	viewCustomerServiceList(userID)	This method is to display their requested service list to customer.
	viewItemDetails(serviceID)	This method is to display the details of the selected item to staff and customer.
	viewUpdateDetailForm(serviceID)	This method is to display the update form to staff.
	updateDetailForm(serviceID)	This method is to update the details entered by staff into the database.
Algorithm	<pre> viewRepairServiceList() BEGIN SELECT services.serviceID . services.userID . users.username . services.device FROM services LEFT JOIN users ON services.userID = users.userID RETURN data in staff service request list END </pre>	

viewCustomerServiceList(userID)

```

BEGIN
SELECT users.userID . users.username . services.serviceID .
services.device
FROM services LEFT JOIN users ON
services.userID = users.userID
WHERE services.userID = session.userID
RETURN data in customer request service list
END

```

ViewItemDetails(serviceID)

```

BEGIN
SELECT users.userID . users.username . services.serviceID .
services.device . services.repairStatus .
services.estimateCost . services.reason . services.detail
FROM services LEFT JOIN users ON
services.userID = users.userID
WHERE serviceID = services.serviceID
RETURN data in customer service item details
END

```

viewUpdateDetailForm(serviceID)

```

BEGIN
SELECT users.userID . users.username . services.serviceID .
services.device . services.repairStatus .
services.estimateCost . services.reason . services.detail
FROM services LEFT JOIN users ON
Services.userID = users.userID
WHERE serviceID=services.serviceID
RETURN data in detail form
END

```

UpdateDetailForm(serviceID)

```

BEGIN

```

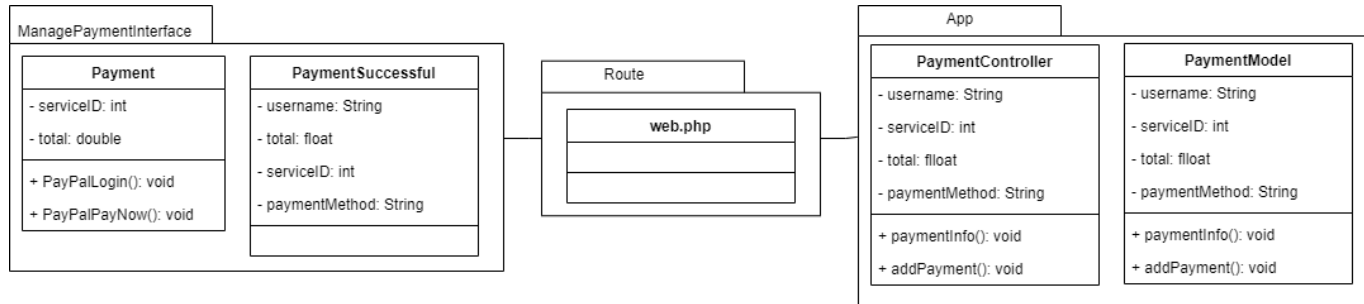

UPDATE services

SET repairStatus = statusUpdate . estimateCost = estimateCostUpdate .
reason = reasonUpdate . detail = detailUpdate

WHERE serviceID = services.serviceID

END

4.6 Manage Payment [SDD-REQ-600]



4.6.1 Payment [SDD-REQ-601]

Class Type	Boundary Class	
Responsibility	It displays the total price of the service and the options of payment method as in cash on delivery and PayPal.	
Attributes	Attributes Name	Attributes Type
	serviceID total	Integer Double
Methods	Method Name	Description
	PayPalLogin() PayPalPayNow()	Customer needs to login to his/her PayPal account. Customer reviews the total price and pays the payment.
Algorithm	PayPalLogin() START IF PayPalusername && PayPalpassword are correct THEN login to PayPal payment gateway ENDIF END PayPalPayNow() START IF Balance > total	

	THEN send payment successful ENDIF END
--	--

4.6.2 PaymentSuccessful [SDD-REQ-602]

Class Type	Boundary Class	
Responsibility	It displays the successful message of making payment.	
Attributes	Attributes Name	Attributes Type
	username	String
	serviceID	Integer
	total	Double
	paymentMethod	String
Methods	Method Name	Description
	Not Applicable	Not Applicable
Algorithm	Not Applicable	

4.6.3 PaymentController [SDD-REQ-603]

Class Type	Controller Class	
Responsibility	It is to connect between PaymentModel and Views of ManagePaymentInterface.	
Attributes	Attributes Name	Attributes Type
	username	String
	serviceID	Integer
	total	Double
	paymentMethod	String
Methods	Method Name	Description
	paymentInfo()	To retrieve the ServiceModel data and send to Payment view.
	addPayment()	To save the Payment view data to PaymentModel.
Algorithm	paymentInfo() START Retrieve username, serviceID, total from ServiceModel	

	END addPayment() START Read data from PaymentSuccessful Send data to PaymentModel END
--	---

4.6.4 PaymentModel [SDD-REQ-604]

Class Type	Model Class	
Responsibility	It is to connect the database of payments table.	
Attributes	Attributes Name	Attributes Type
	username	String
	serviceID	Integer
	total	Double
	paymentMethod	String
Methods	Method Name	Description
	paymentInfo()	To send data to Payment view from service table.
	addPayment()	To save the Payment view data to payment table.
Algorithm	paymentInfo() START Retrieve required data from database of service table Send data to PaymentController END addPayment() START Read data from PaymentController Add data to database of payment table END	

5. REQUIREMENT TRACEABILITY

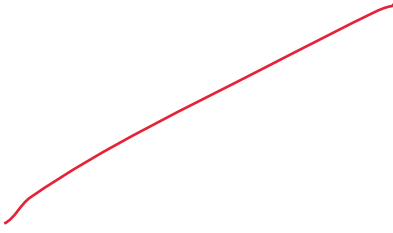
5.1 Manage Registration [SRS-REQ-100] NG KOK THAI (CB18109)

Requirement ID	Description	Design ID
SRS-REQ-101 SRS-REQ-109 SRS-REQ-102 SRS-REQ-110 SRS-REQ-103 SRS-REQ-111	An interface to display the registration form for the user to fill in during registration.	SDD-REQ-101
SRS-REQ-104 SRS-REQ-112 SRS-REQ-105 SRS-REQ-113 SRS-REQ-119 SRS-REQ-121	This class is to connect UserModel and ManageRegistrationInterface.	SDD-REQ-102
SRS-REQ-106 SRS-REQ-114	This class is to connect, store, retrieve and manage the database of user information.	SDD-REQ-103

5.2 Manage Profile [SRS-REQ-200] CHOONG KIT HUEY [CB8099]

Requirement ID	Description	Design ID
SRS-REQ-201 SRS-REQ-204 SRS-REQ-205 SRS-REQ-207 SRS-REQ-221 SRS-REQ-224	An interface to display the username, email, address and phone number of users	SDD-REQ-201

SRS-REQ-225 SRS-REQ-226 SRS-REQ-227		
SRS-REQ-210 SRS-REQ-213 SRS-REQ-213 SRS-REQ-214 SRS-REQ-235 SRS-REQ-240 SRS-REQ-241	This class is responsible to display the list of the customer for the staff to view	SDD-REQ-202
SRS-REQ-217 SRS-REQ-218 SRS-REQ-219 SRS-REQ-220 SRS-REQ-241 SRS-REQ-243 SRS-REQ-244	This is class is responsible to display the customer personal details to the staff to view	SDD-REQ-203
SRS-REQ-202 SRS-REQ-203 SRS-REQ-204 SRS-REQ-205 SRS-REQ-208 SRS-REQ-211 SRS-REQ-212 SRS-REQ-215	The class is used to communicate between the interface and model class which to handle the edit profile, view profile details, delete account, ban account, retrieve data and return responses.	SDD-REQ-204

SRS-REQ-216 SRS-REQ-222 SRS-REQ-223 SRS-REQ-228 SRS-REQ-230 SRS-REQ-231 SRS-REQ-232 SRS-REQ-233 SRS-REQ-236 SRS-REQ-237 SRS-REQ-238		
SRS-REQ-203 SRS-REQ-209 SRS-REQ-212 SRS-REQ-216 SRS-REQ-223 SRS-REQ-229 SRS-REQ-234 SRS-REQ-239	This model is used to save data to the database and retrieve request data from the database	SDD-REQ-205

5.3 Manage Request Service [SRS-REQ-300] YAP JIA XING [CB18104]

Requirement ID	Description	Design ID
SRS-REQ-301 SRS-REQ-302 SRS-REQ-303 SRS-REQ-304 SRS-REQ-309 SRS-REQ-310	This class is responsible for customer to request repair service request.	SDD-REQ-301
SRS-REQ-305 SRS-REQ-306 SRS-REQ-307 SRS-REQ-308	A controller class that handles the saveRequest, displayCustInfo, retrieves model data and return appropriate responses	SDD-REQ-302
SRS-REQ-306 SRS-REQ-307 SRS-REQ-308	A model class to store the details of the request made by customer.	SDD-REQ-303

5.4 Manage Delivery [SRS-REQ-400] MATHIARASU A/L GANESAN [CB18055]

Requirement ID	Description	Design ID
SRS-REQ-401 SRS-REQ-402 SRS-REQ-403 SRS-REQ-405 SRS-REQ-406 SRS-REQ-423 SRS-REQ-424	An interface to display the all task that accepted and unaccepted. And need to accept task or reject task.	SDD-REQ-401

SRS-REQ-408 SRS-REQ-409 SRS-REQ-410 SRS-REQ-412 SRS-REQ-413 SRS-REQ-415	An interface to display the delivery status and update delivery progress.	SDD-REQ-402
SRS-REQ-416	An interface to display the item list that want to track and need to select item need to track.	SDD-REQ-403
SRS-REQ-418	An interface the system will display delivery progress for customer.	SDD-REQ-404
SRS-REQ-419 SRS-REQ-422	An interface to display the item list that want to track and need to select device need to track.	SDD-REQ-405
SRS-REQ-420	An interface to display the delivery process and need to confirm after receive order.	SDD-REQ-406
SRS-REQ-404 SRS-REQ-407 SRS-REQ-411 SRS-REQ-414 SRS-REQ-417 SRS-REQ-421	It is a connection between interface and TrackModel.	SDD-REQ-407
SRS-REQ-404 SRS-REQ-407 SRS-REQ-411 SRS-REQ-414 SRS-REQ-417 SRS-REQ-421	It is a model need to save data to database and retrieve data from database.	SDD-REQ-408

5.5 Manage Repair Service [SRS-REQ-500] LIEW YOON HONG [CB18112]

Requirement ID	Description	Design ID
SRS-REQ-501 SRS-REQ-502 SRS-REQ-503 SRS-REQ-504 SRS-REQ-505	This class is responsible for staff to view the list of the repairing service list of customers and select the repairing service to view in detail.	SDD-REQ-501
SRS-REQ-506 SRS-REQ-507 SRS-REQ-508	This class is responsible for customer to view their requested services and select the repairing service to view in detail.	SDD-REQ-502
SRS-REQ-509 SRS-REQ-510 SRS-REQ-511 SRS-REQ-512 SRS-REQ-513 SRS-REQ-514 SRS-REQ-515 SRS-REQ-516	This class is responsible for customer and staff to view selected repairing service request details.	SDD-REQ-503
SRS-REQ-517 SRS-REQ-518 SRS-REQ-519 SRS-REQ-520 SRS-REQ-521 SRS-REQ-522 SRS-REQ-523 SRS-REQ-524	This class is responsible for staff to update the information of the repairing item in the form.	SDD-REQ-504

SRS-REQ-525		
SRS-REQ-526		
SRS-REQ-501	<p>This controller class is responsible to handle the repair service list for staff, customer item list for customers, item information for customer and staff, update form for staff, retrieves data and return insert data.</p>	SDD-REQ-505
SRS-REQ-502		
SRS-REQ-503		
SRS-REQ-504		
SRS-REQ-505		
SRS-REQ-506		
SRS-REQ-507		
SRS-REQ-508		
SRS-REQ-509		
SRS-REQ-510		
SRS-REQ-511		
SRS-REQ-512		
SRS-REQ-513		
SRS-REQ-514		
SRS-REQ-515		
SRS-REQ-516		
SRS-REQ-517		
SRS-REQ-518		
SRS-REQ-519		
SRS-REQ-520		
SRS-REQ-521		
SRS-REQ-522		
SRS-REQ-523		
SRS-REQ-524		

SRS-REQ-525		
SRS-REQ-526		
SRS-REQ-501	This model class is responsible to stores, retrieves, cancel, edit and manage the service data for both customer and staff.	SDD-REQ-506
SRS-REQ-502		
SRS-REQ-503		
SRS-REQ-504		
SRS-REQ-505		
SRS-REQ-506		
SRS-REQ-507		
SRS-REQ-508		
SRS-REQ-509		
SRS-REQ-510		
SRS-REQ-511		
SRS-REQ-512		
SRS-REQ-513		
SRS-REQ-514		
SRS-REQ-515		
SRS-REQ-516		
SRS-REQ-517		
SRS-REQ-518		
SRS-REQ-519		
SRS-REQ-520		
SRS-REQ-521		
SRS-REQ-522		
SRS-REQ-523		
SRS-REQ-524		

SRS-REQ-525		
SRS-REQ-526		

5.6 Manage Payment [SRS-REQ-600] TAN PEN THE [CB18037]

Requirement ID	Description	Design ID
SRS-REQ-601	Customer can view the payment information of his/her service.	SDD-REQ-601 SDD-REQ-602
SRS-REQ-602		
SRS-REQ-608		
SRS-REQ-610		
SRS-REQ-611		
SRS-REQ-612		
SRS-REQ-614		
SRS-REQ-615		
SRS-REQ-603	Customer can make the payment of his/her service.	SDD-REQ-601
SRS-REQ-604		
SRS-REQ-605		
SRS-REQ-606		
SRS-REQ-607		
SRS-REQ-608		
SRS-REQ-609		
SRS-REQ-613		
SRS-REQ-601	Controller class that handles to retrieve model data, to send payment data and return response according to required interface.	SDD-REQ-601
SRS-REQ-602		SDD-REQ-602
SRS-REQ-608		SDD-REQ-603
SRS-REQ-613		SDD-REQ-604

SRS-REQ-601	Model class that handles to send model data with correct logic based on the requirement by Controller class.	
SRS-REQ-602		SDD-REQ-603
SRS-REQ-608		SDD-REQ-604
SRS-REQ-613		