SDD Document

SEM II 20192020

BEEP BEEP DELIVERY SYSTEM

Group 4

1. FARAH HIDAYAH BINTI MOHD FADZIL [CB20065]
2. NUR AININ SOFIYA BINTI ABU KASIM [CB20076]
3. NUR DARWISYAH FAQIHAH BINTI LUTFI [CB20080]

TABLE OF CONTENTS

1. INTRODUCTION

1.1 Purpose

The purpose of the Software Design Document (SDD) is to deliver a detail description of the requirement for our software product – Beep Beep Delivery System. It also includes the complete declaration for the development of the system. This SDD is intended for both stakeholders – customers, runners and service providers, and the system developer as a detailed reference to develop the first version of this system. It provides a clear view of how our stakeholders and team to see either the software product's functionality or non-functionality. It also helps the designers and developers to make decision in software development life cycle (SDLC) process.

1.2 System Identification

System Title: Beep Beep Delivery System

Abbreviation: BBDS

System Identification Number: BBDS_v1_2020

Use Case Identification Number: SRS-REQ-XXX

Package Identification Number: SDD-REQ-XXX

The version of the system will be recorded as the format, BBDS_v1_2020. “BBDS” is a term of our software product name – Beep Beep Delivery System, first release of the system. Use Case Identification Number is SRS-REQ-XXX, that XXX stands according to the diagrams for SRS documentation. Package Identification Number is SDD-REQ-XXX, that XXX stands according to the diagrams for SDD.

1.3 System Overview

Beep Beep Delivery System (BBDS) is a web-based system that in order to improve the delivery system service to another higher level. BBDS can replace the older version of delivery system which processing work is still in a miserable manual way. BBDS has the good functionalities that will ease the stakeholder – customer, service provider and runner, in many ways. For instance, all the payments are made cashless that can avoid the miscalculation transaction. Everyone can register as customer to enjoy the services which provided by the system. For those who have the company/restaurant license can be one of the service providers meanwhile to be a runner, a vehicle number plate is needed with a minimum requirement as motorbike to run the task.

The BBDS provides 6 major functions – manage food delivery, manage good delivery, manage pet assist, manage medical delivery, manage payment and provide tracking and analytic. The constraint of the system is that all the services can only be run within diameter of 20 kilometers from customer location and runner location.

Firstly, functions of manage food delivery, good delivery, pet delivery and medical delivery, it allows customer to choose the provided services in the system which are registered to the system by the service providers. System also welcome the service providers to take part of the system in order to increase the number of offered food on the food menu.

Secondly, function of manage payment, customer and payment gateway, PayPal are involved in this function. Customer can make their payment via PayPal online banking or credit/debit card that directed by payment gateway and customers can also review their past orders.

Lastly, function of tracking and analytic. For tracking function in this system, once the customer has done an order(s), service provider would get notification to accept the orders and then the runner would be notified after service provider. Besides, for analytic function, service provider and runner can view the analysis report.

1.4 Referenced Document

1- UMP KALAM. (2020, 2 15). Retrieved from https://kalam.ump.edu.my/pluginfile.php/400313/mod_folder/content/0/SDD%20Document/SDD%20Template%20Document.pdf?forcedownload=1

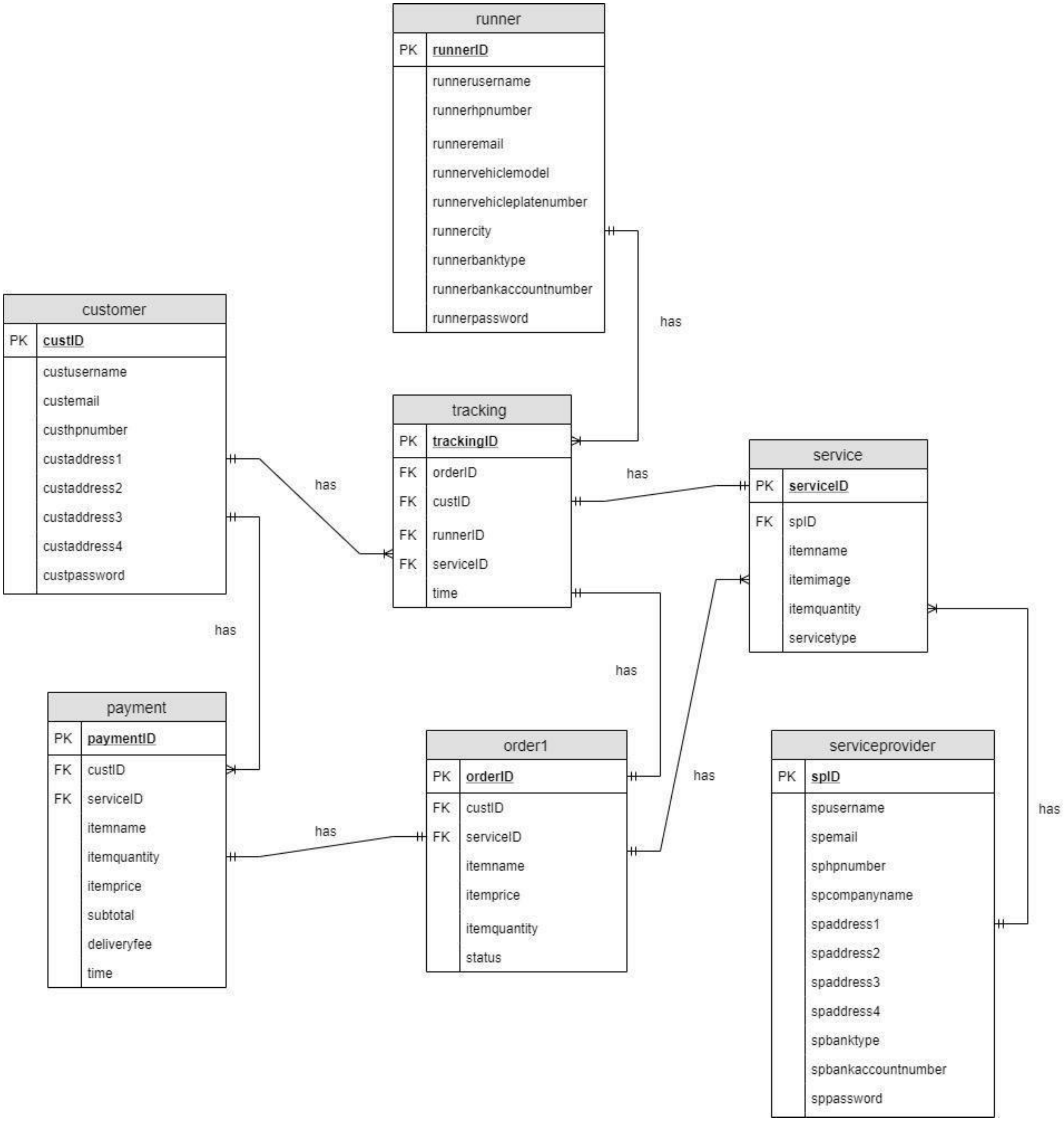
2- UMP KALAM. (2020, 2 12). Retrieved from https://kalam.ump.edu.my/pluginfile.php/400313/mod_folder/content/0/SRS%20Document/SRS%20Template%20Document.pdf?forcedownload=1

3- FOX, C. J. (2020, 2 22). Why Writing Software Design Documents Matters. Retrieved from Developers: https://www.toptal.com/freelance/why-design-documents-matter

4- AKTA PENGANGKUTAN JALAN 1987, 2013

2. DATA DESIGN

2.1 Entity Relationship Diagram (ERD)



2.2 Data Dictionary

2.2.1 Customer

Field Name	Description	Data Type	Constraint
custID	Customer ID	INT (11)	PK
custuserame	Customer username	VARCHAR (100)	
custpassword	Customer login password	VARCHAR (50)	
custhpnumber	Customer contact number	VARCHAR (50)	
custemail	Customer email	VARCHAR (50)	
custaddress1	Customer address line 1	VARCHAR (50)	
custaddress2	Customer address line 2	VARCHAR (50)	
custaddress3	Customer address line 3	VARCHAR (50)	
custaddress4	Customer address line 4	VARCHAR (50)	

2.2.2 Service Provider

Field Name	Description	Data Type	Constraint
spID	Service provider ID	INT (11)	PK
spusername	Service provider username	VARCHAR (100)	
spemail	Service provider email	VARCHAR (100)	
sphpnumber	Servicer provider contact number	VARCHAR (50)	
spcompanyname	Service provider company name	VARCHAR (100)	
spbankaccountnumber	Service provider bank account number	VARCHAR (50)	
Sppassword	Service provider login password	VARCHAR (50)	
spaddress1	Service provider address line 1	VARCHAR (50)	
spaddress2	Service provider address line 2	VARCHAR (50)	

spaddress3	Service provider address line 3	VARCHAR (50)	
spaddress4	Service provider address line 4	VARCHAR (50)	
spbanktype	Service provider bank company	VARCHAR (50)	

2.2.3 Runner

Field Name	Description	Data Type	Constraint
runnerID	Runner ID	INT (11)	PK
runnerusername	Runner username	VARCHAR (100)	
runnerhpnumber	Runner contact number	VARCHAR (50)	
runneremail	Runner email	VARCHAR (100)	
runnervehiclemodel	Runner vehicle model	VARCHAR (100)	
runnervehicleplatenumber	Runner vehicle plate number	VARCHAR (50)	
runnercity	Runner current city	VARCHAR (50)	
runnerbanktype	Runner bank company	VARCHAR (50)	
runnerbankaccountnumber	Runner bank account number	VARCHAR (50)	
runnerpassword	Runner login password	VARCHAR (50)	

2.2.4 order1

Field Name	Description	Data Type	Constraint
orderID	Order ID	INT (11)	PK
serviceID	Service ID	INT (11)	FK
custID	Customer ID	INT (11)	FK
itemname	Item name	VARCHAR (50)	
itemprice	Item sold price	INT (50)	
itemquantity	Quantity of item	INT (50)	
status	Order status	INT (11)	

2.2.5 service

Field Name	Description	Data Type	Constraint
serviceID	Service ID	INT (11)	PK
spID	Service provider ID	INT (11)	FK
itemname	Item name	VARCHAR (50)	
itemimage	Item image	LONGBLOB	

itemprice	Price of item	INT (50)	
servicetype	Type of service	VARCHAR (50)	

2.2.6 Tracking

Field Name	Description	Data Type	Constraint
trackingID	Tracking ID	INT (11)	PK
orderID	Order ID	INT (11)	FK
serviceID	Service ID	INT (11)	FK
custID	Customer ID	INT (11)	FK
runnerID	Runner ID	INT (11)	FK
time	Time	TIMESTAMP	

2.2.7 Payment

Field Name	Description	Data Type	Constraint
paymentID	Payment ID	INT (11)	PK
serviceID	Service ID	INT (11)	FK
custID	Customer ID	INT (11)	FK
itemname	Item name	VARCHAR (50)	
itemprice	Item sold price	INT (50)	
itemquantity	Quantity of item	INT (50)	
subtotal	Subtotal of the order	INT (50)	
deliveryfee	Delivery fee	INT (50)	
time	Time of payment	TIMESTAMP	

3. GENERAL ARCHITECTURE

3.1 Layered Architecture

3.1.1 Application Layer

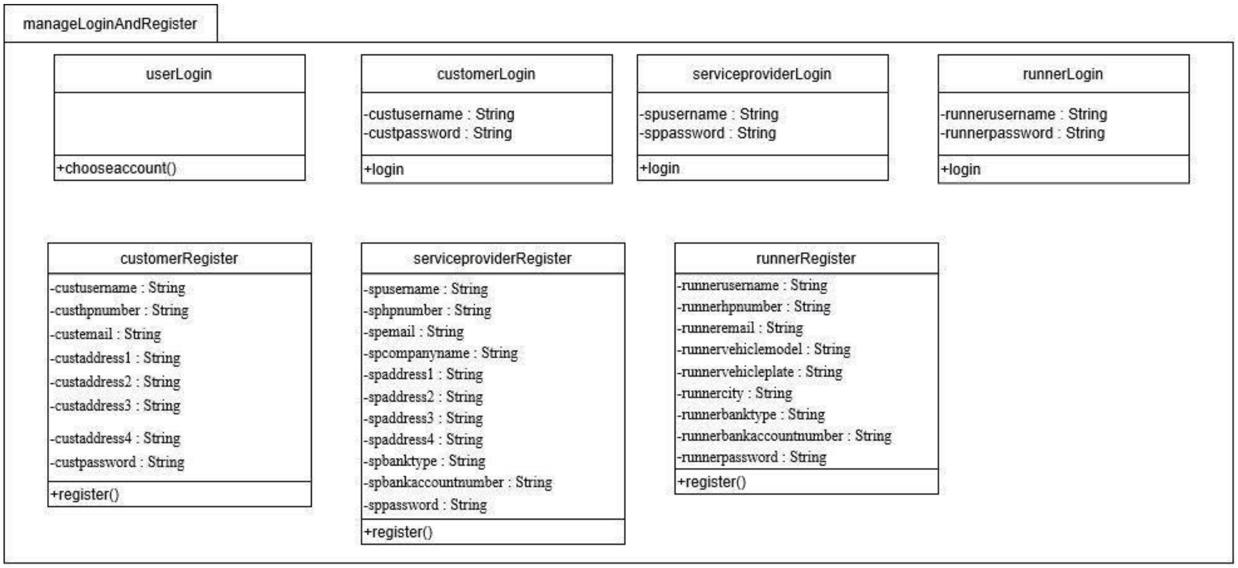


Figure 3.1.1.1: Manage Login and Register Package

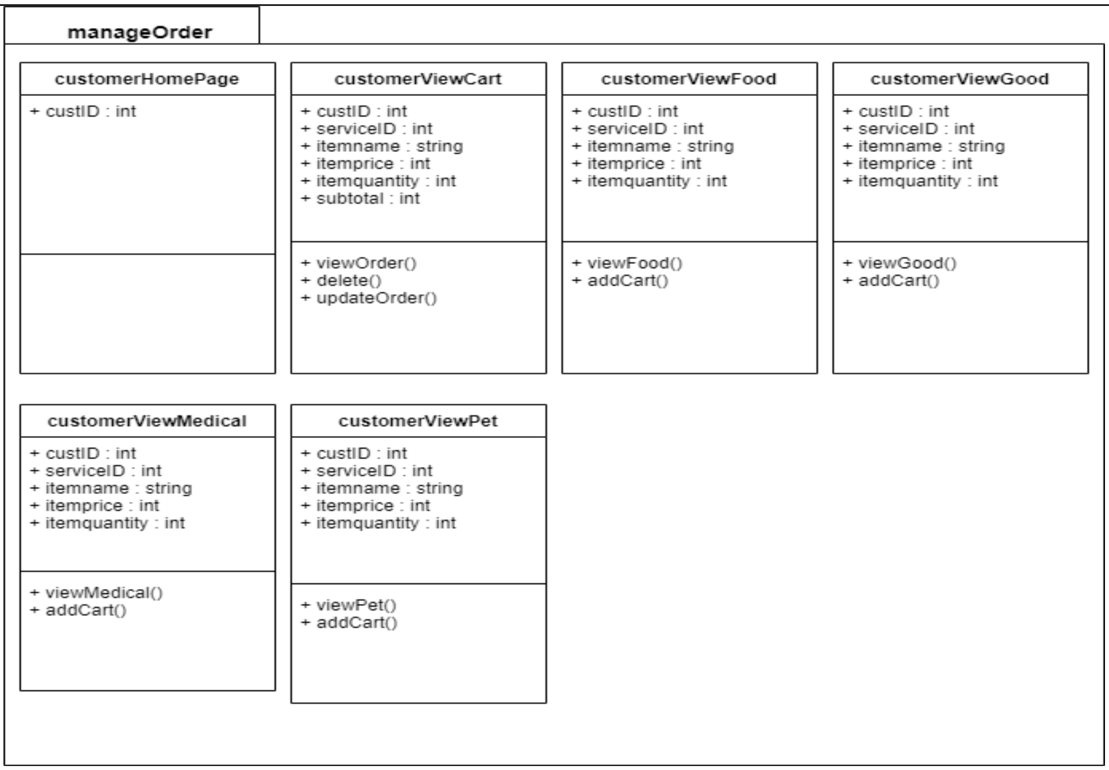


Figure 3.1.1.2: Manage Order Package

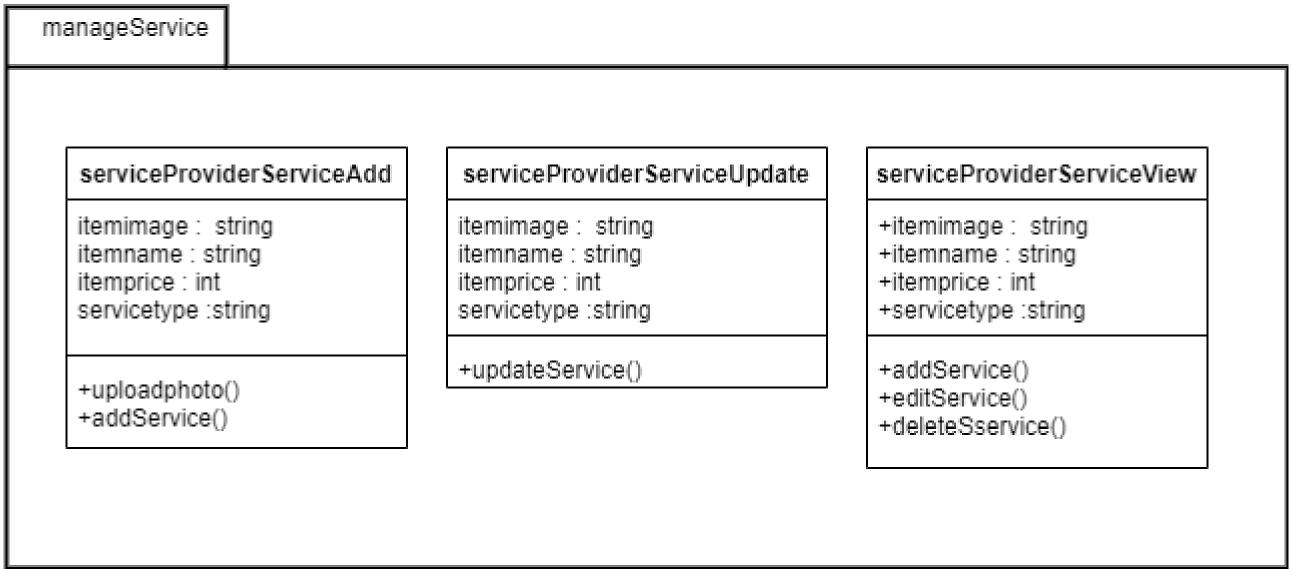


Figure 3.1.1.3: Manage Service Package

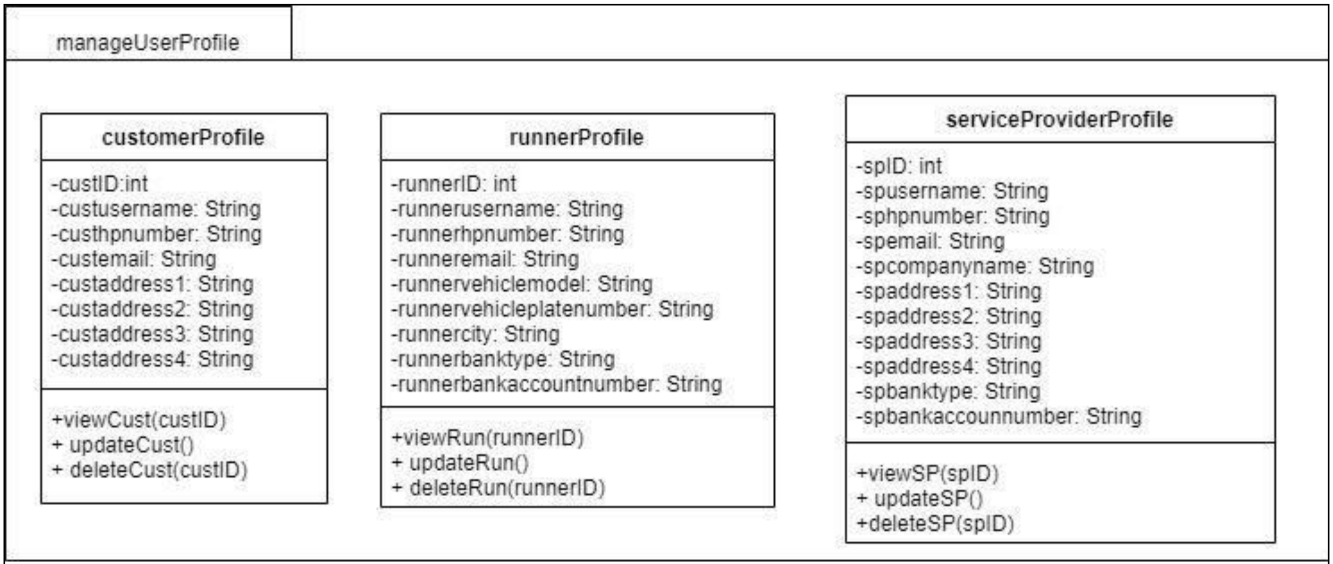


Figure 3.1.1.4: Manage User Profile Package

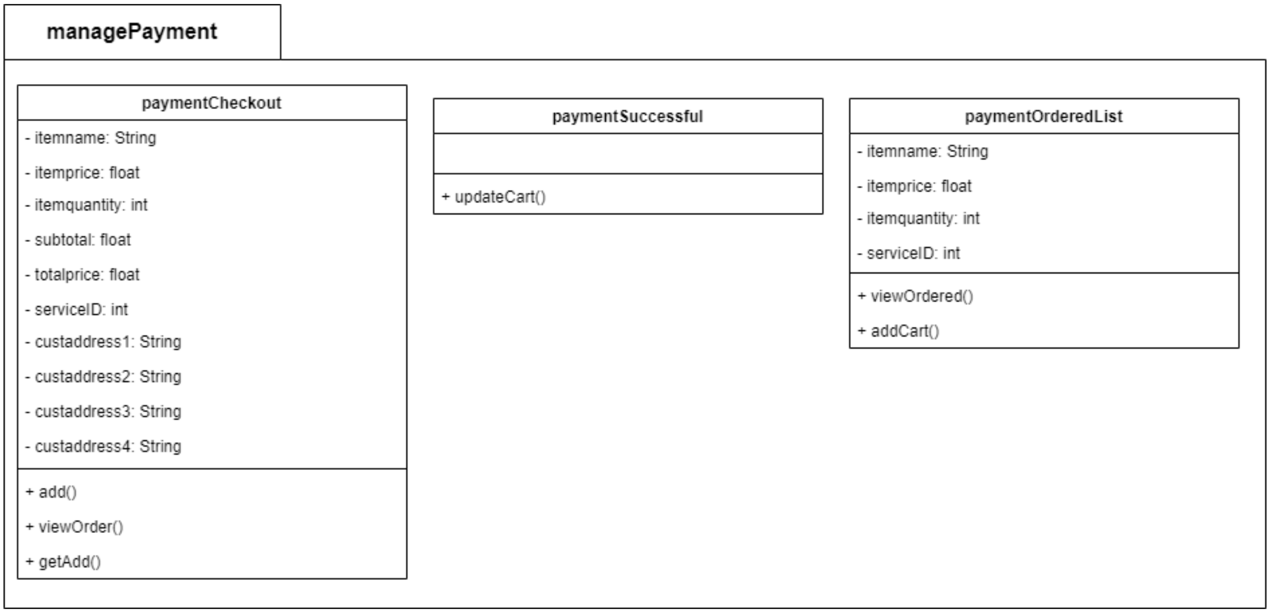


Figure 3.1.1.5: Manage Payment Package

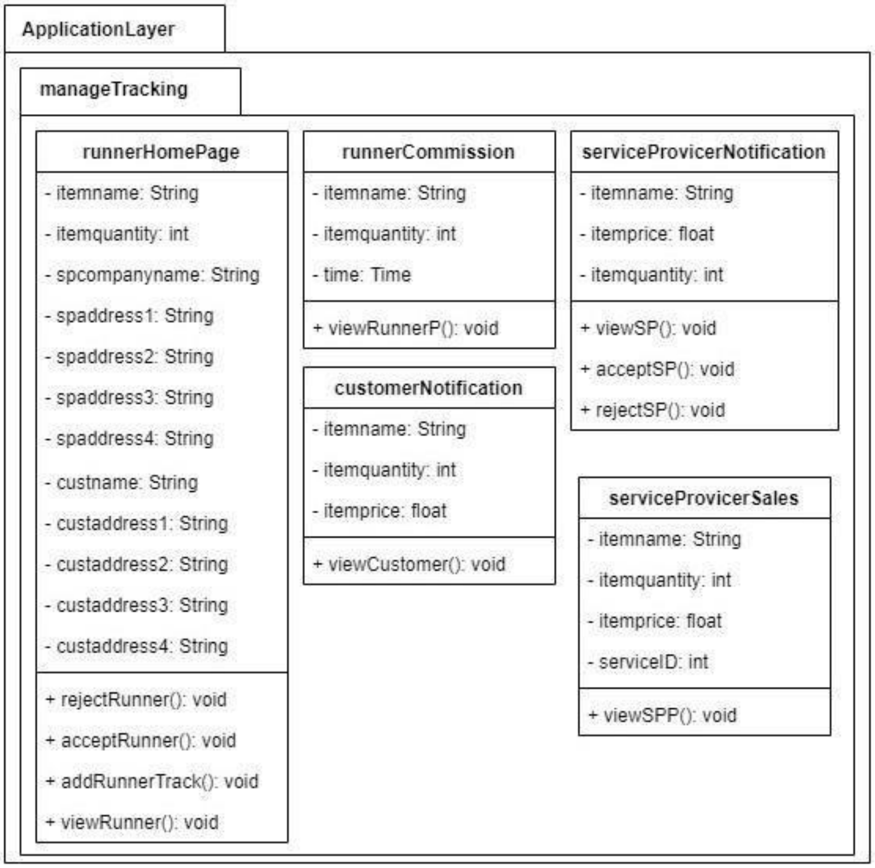


Figure 3.1.1.6: Provide Tracking and Analytic Package

3.1.1.1 Package 1: manageLoginAndRegister [SDD-REQ-100]

Class Name	Description
userRegister	Interface that allow customer to choose type of account that user want to login as.
userLogin	Interface that allow user to login as customer.
customerRegister	Interface that allow user to register as customer if user does not have an account.
customerLogin	Interface that allow user to login as customer.
spRegister	Interface that allow user to register as service provider if user does not have an account.
spLogin	Interface that allow user to login as service provider.
runnerRegister	Interface that allow user to register as runner if user does not have an account.
runnerLogin	Interface that allow user to login as runner.

3.1.1.2 Package 2: manageOrder [SDD-Req-200]

Class Name	Description
customerHomePage	The main interface that can be accessed by customer to perform task in the system
customerViewCart	The interface that can be accessed by customer to view the cart list.
customerViewFood	The interface that can be accessed by customer to view the food service.
customerViewGood	The interface that can be accessed by customer to view the good service.
customerViewMedical	The interface that can be accessed by customer to view the medical service.

customerViewPet	The interface that can be accessed by customer to view the cart service.

3.1.1.3 Package 3: ManageService [SDD-REQ-300]

Class Name	Description
serviceProviderServiceAdd	Interface that allow service provider to add service or item to sell
serviceProviderServiceUpdate	Interface that allow service provider to update the detail of the service
serviceProviderServiceView	Interface that allow service provider to view how much service has in display for sell

3.1.1.4 Package 4: manageUserProfile [SDD-REQ-400]

Class Name	Description
customerProfile	Once the customer registered into the system, the customer can view their profile. In this interface, the customer can either update their details such as phone number, email and their delivery address or delete their respective customer account.
runnerProfile	Once the runner registered into the system, the runner can view their profile. In this interface, the runner can either update their details such as phone number, email, vehicle model, vehicle plate number, their delivery area, bank type and also bank account number. They also can delete their respective runner account.
serviceProviderProfile	Once the service provider registered into the system, the service provider can view their profile. In this interface, the service provider can either update their details such as phone number, email, company name, their address, their bank type and also bank account number. They also can delete their respective service provider account.

3.1.1.5 managePayment [SDD-REQ-500]

Class Name	Description
paymentCheckout	Customer can view their orders and its total amount. Customer can also change their delivery address. Customer can choose their payment method – credit/debit card or PayPal online banking.
paymentSuccessful	If the payment is made successful, then customer will be directed to this page and the system will updater the order status to notify the service provider to prepare their service item.
paymentOrderedList	Once the payment is made, the customer can view their ordered item and they can maybe re-order by placing them into cart.

3.1.1.6 manageTracking [SDD-REQ-600]

Class Name	Description
runnerHomePage	Interface that the runner can know the number of order need to delivery to customer and destination to pick up and drop off the order. In this interface also runner can update the delivery status of the order.
runnerCommission	Interface that the runner can know the number of order that runner had delivered and the total commission which is delivery fee that runner will get.
customerNotification	Interface that customer can know that their order has successfully delivered by the runner.
serviceProviderNotification	Interface that the service provider can know the incoming order and the service provider can accept or reject to do the order.
serviceProviderSales	Interface that the service provider can know the number of order that they had accepted and the total income.

3.1.2 Business Layer

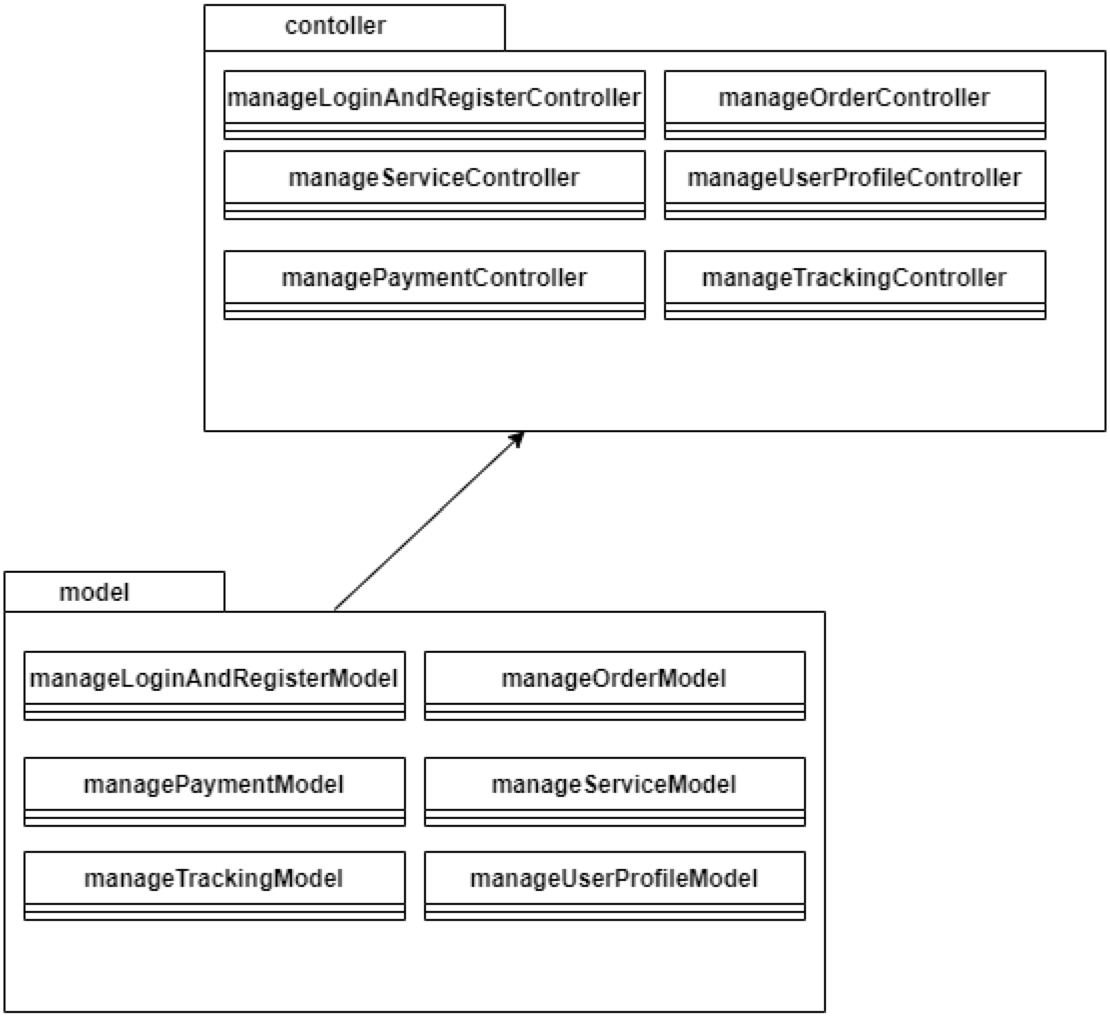


Figure 3.1.2.1: Business Services Layer

3.1.2.1 manageLoginAndRegisterController [SDD-REQ-700]

Class Name	Description
manageLoginAndRegisterController	The controller that used to communicate the manageLoginAndRegisterController class with entity class include ServiceData class, CustomerData class, RunnerData class and ServiceProviderData class.

3.1.2.2 manageOrderController [SDD-REQ-800]

Class Name	Description

manageOrderController	The controller that used to communicate between the customer database and interface.

3.1.2.3 manageServiceController [SDD-REQ-900]

Class Name	Description
manageServiceController	Service controller is used to communicate between service provider interface and database

3.1.2.4 manageUserProfileController [SDD-REQ-1000]

Class Name	Description
manageUserProfileController	User Profile Controller is being used to communicate the customer, service provider and runner database and the interfaces.

3.1.2.5 managePaymentController [SDD-REQ-1100]

Class Name	Description
managePaymentController	Payment Controller is used to communicate between the payment database and the interfaces.

3.1.2.6 manageTrackingController [SDD-REQ-1200]

Class Name	Description
manageTrackingController	The controller that used to communicate the interface and the database.

3.1.2.7 FunctionalData [SDD-REQ-1300]

Class Name	Description
manageLoginAndRegister Model	Database which store and retrieve information of customer, service provider and runner.
manageOrderModel	The database that use to store and retrieved the information for customer.

manageServiceModel	The database that use to store and retrieved the information for service provider
manageUserProfileModel	The database used to store any changes to the users details and retrieve the information about the user which consists of three which are the customer, service provider and runner.
managePaymentModel	Database to store and retrieve the record of ordered services (payment).
manageTrackingModel	The database that used to store and retrieve the delivery information.

3.1.3 Middleware Layer

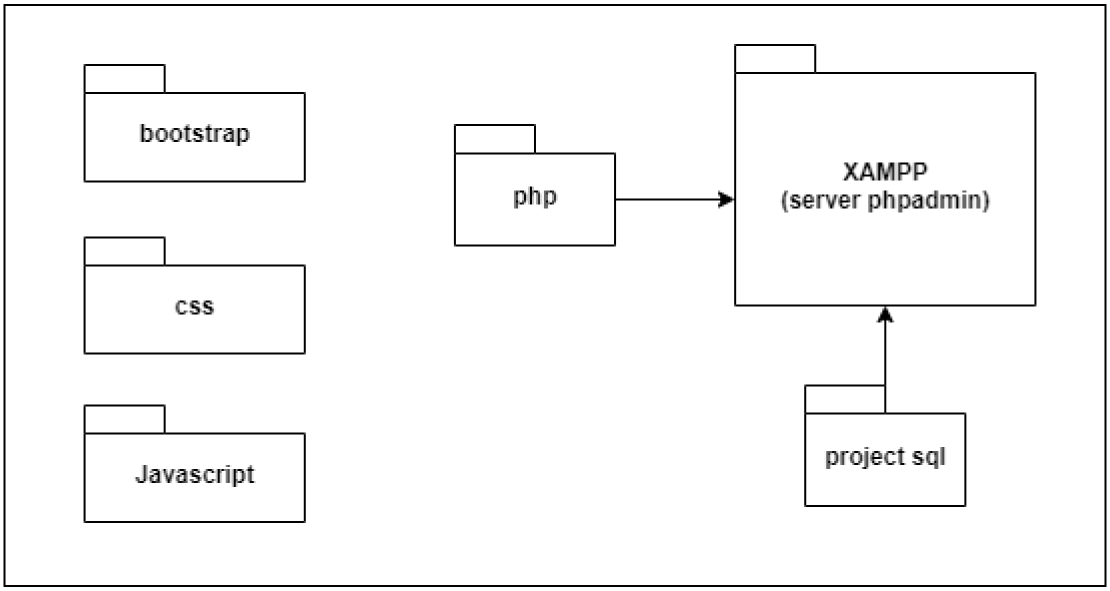


Figure 3.1.3.1: Middleware Layer Package

Package Name	Description
bootstrap	It is the most popular CSS framework for developing mobile-first websites.
css	For describing how HTML elements should be displayed.

php	For connecting website to the server database.
Server phpadmin	For manipulating the sql database.
project sql	The SQL file of the database and table information that needed to transport to the server phpadmin.
Javascript	JavaScript is the Programming Language for the Web.

3.2 MVC Package Relationship

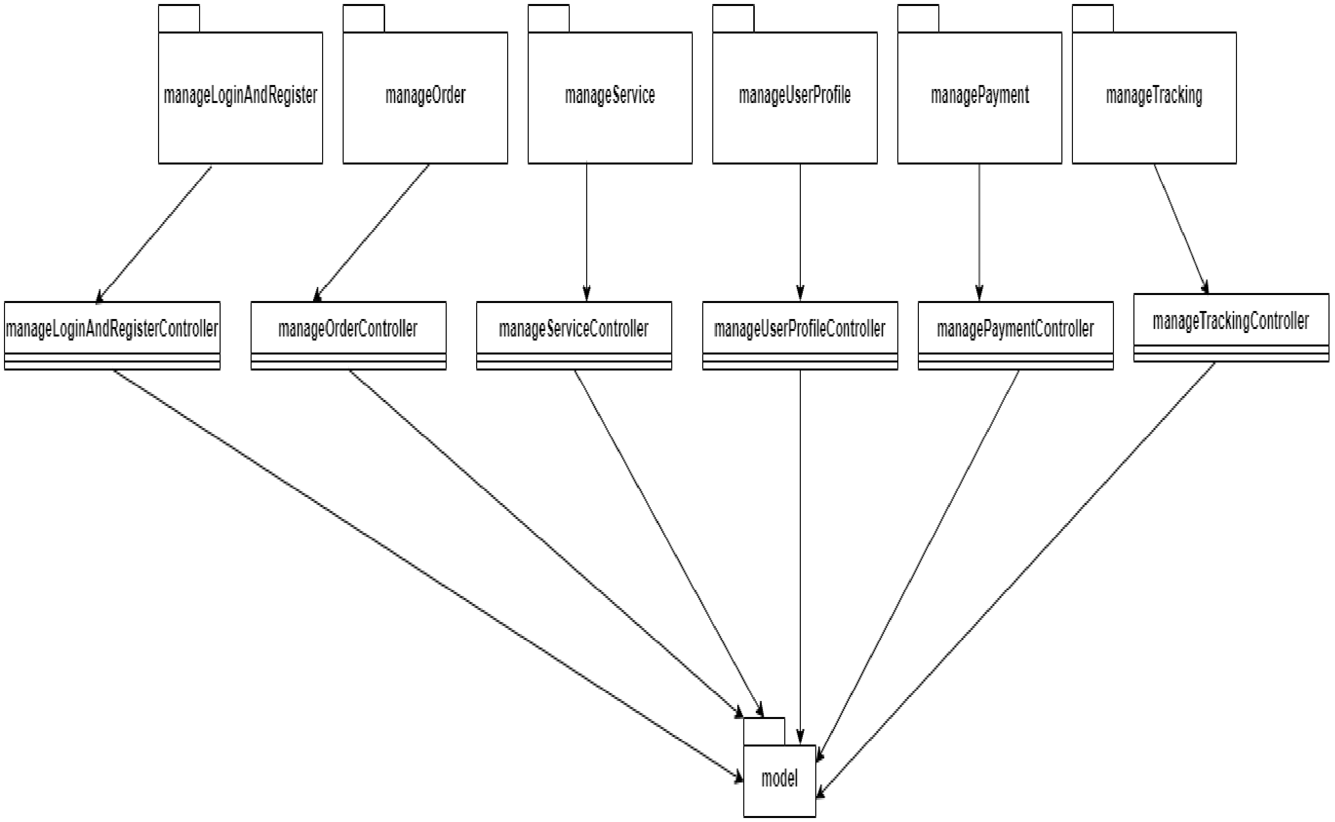


Figure 3.2.1: MVC Package Relationship

4. DETAIL DESIGN

4.1 manageLoginAndRegister [SDD-REQ-100]

FARAH HIDAYAH BINTI MOHD FADZIL

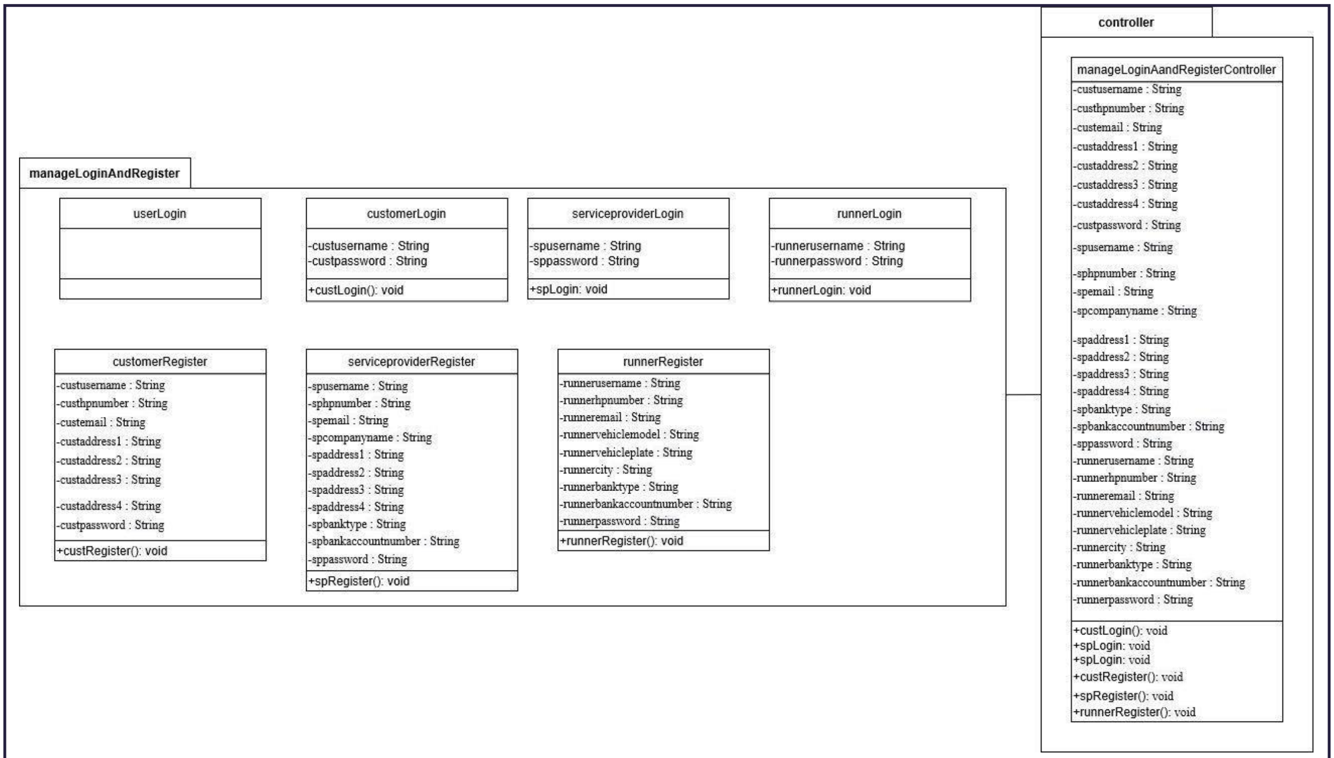


Figure 4.1.1: LoginAndRegister Package

4.1.1 Class 1: userLogin [SDD-REQ-101]

Class Type	Boundary class	
Responsibility	An interface to allow users to log in the system according to the type of user such as customer, service provider and runner.	
Attributes	Attributes Name	Attributes Type
	None	None
Methods	Method Name	Description
	None	None
Algorithm	START System display options for several login interface. User can choose type of account to login	

	IF user click button “Customer”
	THEN
	System display customer login interface
	ELSE IF user click button “Service provider”
	THEN
	System display service provider login interface
	ELSE user click button “Runner”
	THEN
	System display runner login interface
	END IF
	END

4.1.2 Class 2: userRegister [SDD-REQ-102]

Class Type	Boundary class	
Responsibility	An interface to allow users to register in the system according to the type of user such as customer, service provider and runner.	
Attributes	Attributes Name	Attributes Type
	None	None
Methods	Method Name	Description
	None	None

Algorithm	START
	System display
	User can choose which type of account to register
	IF user click button “Customer”
	System display customer register interface
	ELSE IF user click button “Service provider”
	THEN
	System display service provider register interface
	ELSE user click button “Runner”
	THEN
	System display runner login interface
	END IF	

	END

4.1.3 Class 3: customerLogin [SDD-REQ-103]

Class Type	Boundary Class	
Responsibility	An interface to display the customer login page.	
Attributes	Attributes Name	Attributes Type
	custusername	String
	custpassword	String
Methods	Method Name	Description
	customerLogin()	To login into the system as customer.
Algorithm	START System display customer login interface User fill in username and password User click button “Login” IF login successful THEN system display customer interface ELSE System display error message END IF END	

4.1.4 Class 4: customerRegister [SDD-REQ-104]

Class Type	Boundary Class	
Responsibility	An interface to display the customer register page.	
Attributes	Attributes Name	Attributes Type
	custusername	String
	custhpnumber	String
	custemail	String
	custaddress1	String
	custaddress2	String
	custaddress3	String
	custaddress4	String
	custpassword	String

Methods	Method Name	Description
	customerRegister ()	To register into the system.
Algorithm	START System display customer register interface User fill in username, phone number,email address, address and password User click button “Register” IF register successful THEN system display customer login interface ELSE THEN system display error message END IF END	

4.1.5 Class 5: spLogin [SDD-REQ-105]

Class Type	Boundary Class	
Responsibility	An interface to display the service provider login page.	
Attributes	Attributes Name	Attributes Type
	spusername sppassword	String String
Methods	Method Name	Description
	spLogin()	To login into the system as service provider.
Algorithm	START System display service provider login interface User fill in username and password User click button “Login” IF login successful THEN system display service provider interface ELSE System display error message END IF	

	END

4.1.6 Class 6: spRegister [SDD-REQ-106]

Class Type	Boundary Class	
Responsibility	An interface to display the service provider register page.	
Attributes	Attributes Name	Attributes Type
	spusername	String
	sphpnumber	String
	spemail	String
	spcompanyname	String
	spaddress1	String
	spaddress2	String
	spaddress3	String
	spaddress4	String
	spbanktype	String
	spbankaccountnumber	String
	sppassword	String
Methods	Method Name	Description
	spRegister()	To register into the system as service provider.
Algorithm	START System display service provider register interface User fill in username, phone number,email address, company name, address, bank type, bank account number and password User click button “Register” IF register successful THEN system display service provider login interface ELSE THEN system display error message END IF END	

4.1.7 Class 7: runnerLogin [SDD-REQ-107]

Class Type	Boundary Class	
Responsibility	An interface to display the runner login page.	
Attributes	Attributes Name	Attributes Type
	runnerusername runnerpassword	String String
Methods	Method Name	Description
	runnerLogin()	To login into the system.
Algorithm	START System display runner login interface User fill in username and password User click button “Login” THEN system display runner interface END	

4.1.8 Class 8: runnerRegister [SDD-REQ-108]

Class Type	Boundary Class	
Responsibility	An interface to display the runner register page.	
Attributes	Attributes Name	Attributes Type
	runnerusername runnerhpnumber runneremail runnervehiclemodel runnervehicleplate runnercity runnerbanktype runnerbankaccountnumber runnerpassword	String String String String String String String String String
Methods	Method Name	Description
	runnerRegister()	To register into the system as runner.

Algorithm	START System display runner register interface User fill in username, phone number,email address, vehicle model, vehicle plate number, city, bank type, bank account number and password User click button “Register” IF register successful THEN system display service provider login interface ELSE THEN system display error message END IF END

4.2 manageOrder [SDD-REQ-200]

NUR AININ SOFIYA BINTI ABU KASIM

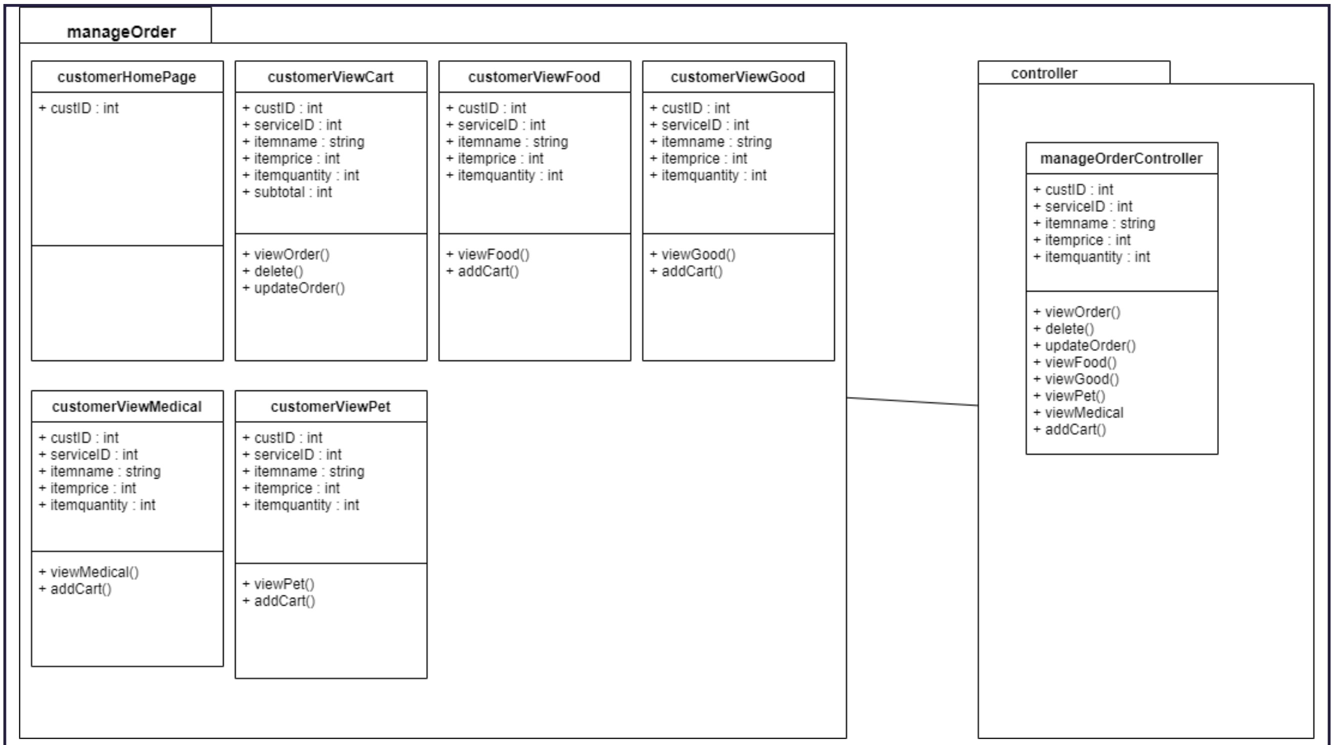


Figure 4.2.1: Manage Order Package

4.2.1 customerHomePage [SDD-REQ-201]

Class Type	Boundary class	
Responsibility	An interface to allow users to view the homepage	
Attributes	Attributes Name	Attributes Type
	custID	int
Methods	Method Name	Description
	None	None

Algorithm	START
	System display Customer Home page
	IF user click button “Good”
	THEN
	System display Customer View Good page
	ELSE IF user click button “Food”	

	THEN
	System display Customer View Food Service page
	ELSE IF user click button “Pet”
	THEN
	System display Customer View Pet Service page
	ELSE IF user click button “Medical”
	THEN
	System display Customer View Medical Service page
	END IF
	END

4.2.2 customerViewCart [SDD-REQ-202]

Class Type	Boundary class	
Responsibility	An interface to allow users to view the cart	
Attributes	Attributes Name	Attributes Type
	custID	int
	serviced	int
	itemname	string
	itemprice	int
	itemquantity	int
	subtotal	int
Methods	Method Name	Description
	viewOrder()	To view the customer order
	delete()	To delete the customer order
	updateOrder()	To update the customer order

Algorithm	START
	System display Customer View Cart page
	IF user click button “delete”
	THEN
	System display popup “Delete Successful!”
	ELSE IF user click button “Checkout”
	THEN	

	System display Payment Checkout page
	END IF
	END

4.2.3 customerViewFood [SDD-REQ-203]

Class Type	Boundary class	
Responsibility	An interface to allow users to view the food service	
Attributes	Attributes Name	Attributes Type
	custID	int
	serviced	int
	itemname	string
	itemprice	int
	itemquantity	int
Methods	Method Name	Description
	viewFood()	To view the food service
	addCart	To add item to the cart

Algorithm	START
	System display Customer View Food Service page
	IF user click button “Add to Cart”
	THEN
	System display popup “Confirm add to cart?”
	THEN the user click button “OK”
	ELSE IF user click button “Homepage”
	THEN
	System display Homepage
	END IF
	END	

4.2.4 customerViewGood [SDD-REQ-204]

Class Type	Boundary class	
Responsibility	An interface to allow users to view the good service	
Attributes	Attributes Name	Attributes Type
	custID	int
	serviced	int
	itemname	string
	itemprice	int
	itemquantity	int
Methods	Method Name	Description
	viewGood()	To view the good service
	addCart	To add item to the cart

Algorithm	START
	System display Customer View Good Service page
	IF user click button “Add to Cart”
	THEN
	System display popup “Confirm add to cart?”
	THEN the user click button “OK”
	ELSE IF user click button “Homepage”
	THEN
	System display Homepage
	END IF
	END	

4.2.5 customerViewMedical [SDD-REQ-205]

Class Type	Boundary class	
Responsibility	An interface to allow users to view the medical service	
Attributes	Attributes Name	Attributes Type
	custID	int
	serviced	int
	itemname	string
	itemprice	int
	itemquantity	int
Methods	Method Name	Description
	viewMedical()	To view the medical service
	addCart	To add item to the cart

Algorithm	START
	System display Customer View Medical Service page
	IF user click button “Add to Cart”
	THEN
	System display popup “Confirm add to cart?”
	THEN the user click button “OK”
	ELSE IF user click button “Homepage”
	THEN
	System display Homepage
	END IF
	END	

4.2.6 customerViewPet [SDD-REQ-205]

Class Type	Boundary class	
Responsibility	An interface to allow users to view the pet service	
Attributes	Attributes Name	Attributes Type
	custID	int
	serviced	int
	itemname	string
	itemprice	int
	itemquantity	int
Methods	Method Name	Description
	viewPet()	To view the pet service
	addCart	To add item to the cart

Algorithm	START
	System display Customer View Pet Service page
	IF user click button “Add to Cart”
	THEN
	System display popup “Confirm add to cart?”
	THEN the user click button “OK”
	ELSE IF user click button “Homepage”
	THEN
	System display Homepage
	END IF
	END	

4.3 manageService [SDD-REQ-300]

NUR DARWISYAH FAQIHAH BINTI LUTFI

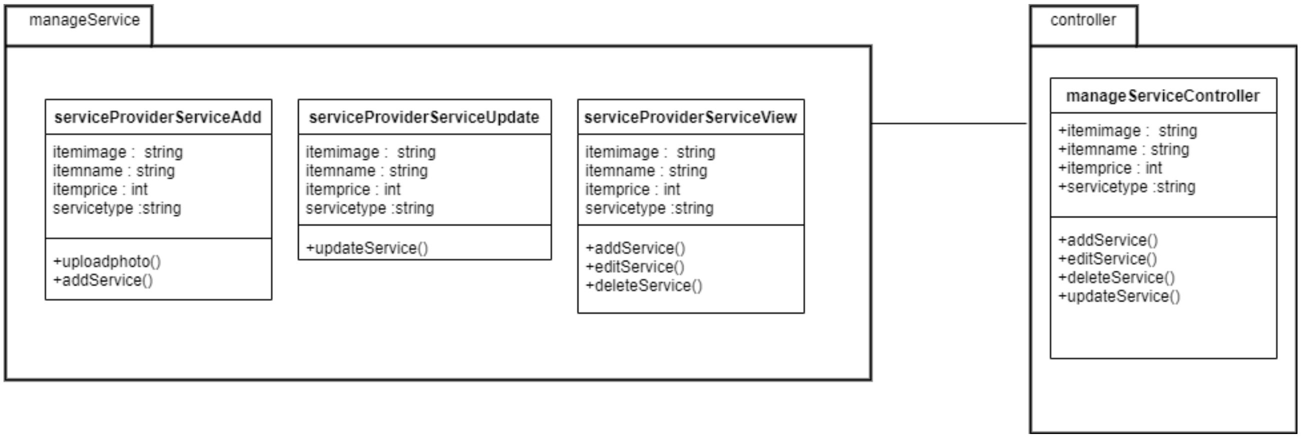


Figure 4.3.1: Manage Service Package

4.3.1 serviceProviderServiceView [SDD-REQ-301]

Class Type	Boundary Class	
Responsibility	An interface to display service provide by service provider	
Attributes	Attributes Name	Attributes Type
	itemimage itemname itemprice servicetype	String String int String
Methods	Method Name	Description

	addService()
	editService()
	deleteService()
		To add new service or item for display.
		To update detail of service or item to sell.
		To delete item or service that already exist.

Algorithm	addService() START When service provider click the "Click Here" button Then the service provider will go to the page service add	

	END editService() START When service provider clicks the “update icon” button Then the service provider will go to update page END deleteService() START When service provider click the “Delete icon” button Then the service provider item service will be deleted END

4.3.2 serviceProviderServiceAdd [SDD-REQ-302]

Class Type	Boundary Class	
Responsibility	An interface to allow service provider to add service	
Attributes	Attributes Name	Attributes Type
	itemimage itemname itemprice servicetype	String String int String
Methods	Method Name	Description

	uploadphoto()
	addService()
		To upload the image of the service
		To add the service into the databse

Algorithm	uploadphoto() START When customer already choose file for picture then customer click the “Upload Photo” button	

	Then the file will upload to the database END addService() START System display service provider add page Then service provider fills in the blank detail or service to provide Then the service provider will click the “Add” button Then detail will be saved to the database displays in service view page END

4.3.3 serviceProviderServiceUpdate [SDD-REQ-303]

Class Type	Boundary Class	
Responsibility	An interface to allow service provider to add service	
Attributes	Attributes Name	Attributes Type
	itemimage	String
	itemname	String
	itemprice	int
	servicetype	String
Methods	Method Name	Description
	updateService()	To update the existing service
Algorithm	updateService() START System display service provider update page Then service provider fills in new detail for service to change Then the service provider will click the “Update Service” button Then new detail will be saved to the database displays in service view page END	

4.4 manageUserProfile [SDD-REQ-400]

FARAH HIDAYAH BINTI MOHD FADZIL

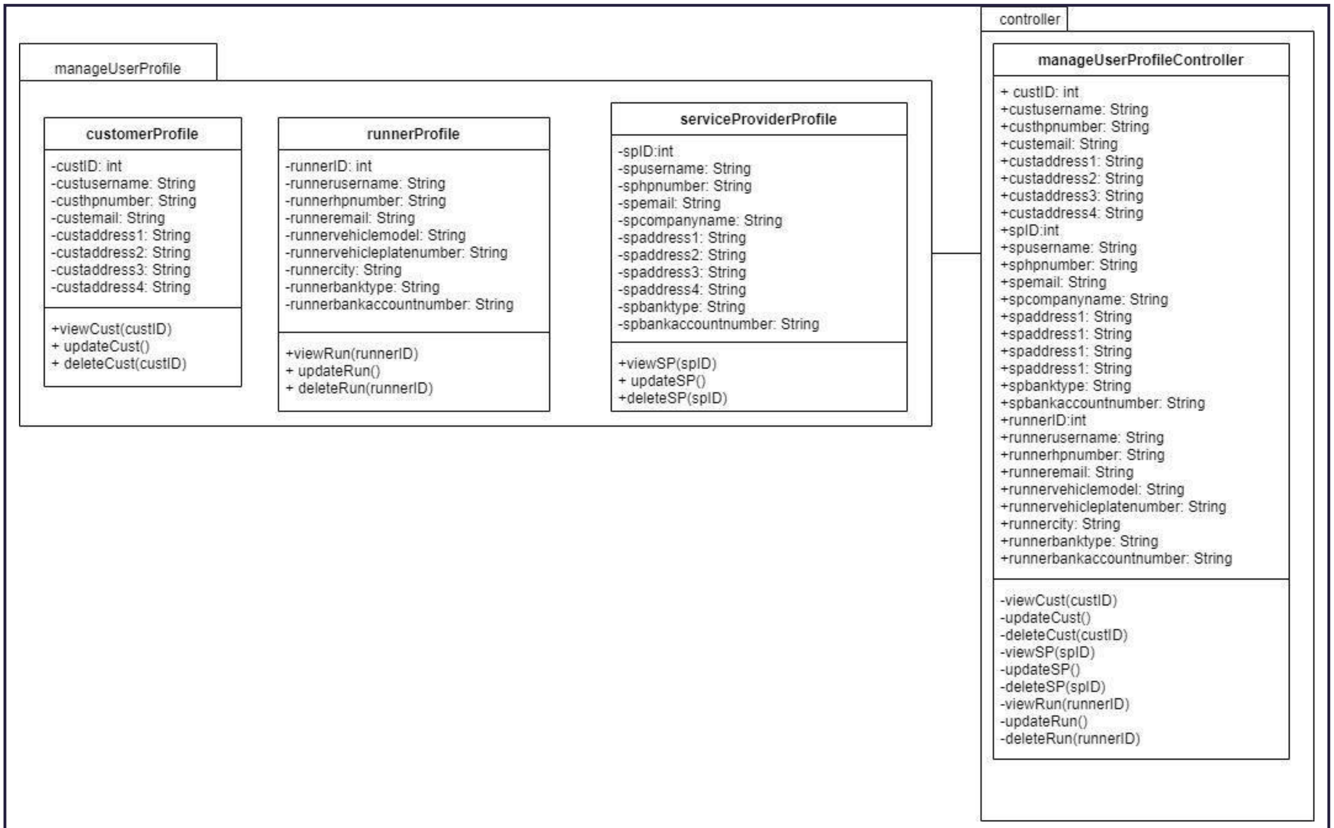


Figure 4.4.1: Manage User Profile Package

4.4.1 customerProfile [SDD-REQ-401]

Class Type	Boundary Class	
Responsibility	It displays customer profile with credentials used during customer registration process.	
Attributes	Attributes Name	Attributes Type
	custID	int
	custusername	String
	custphnumber	String
	custemail	String
	custaddress1	String
	custaddress2	String
	custaddress3	String

	custaddress4	String
Methods	Method Name	Description
	viewCust(custID)	To view the details about the customer
	updateCust()	To update the details about the customer.
	deleteCust(custID)	To delete the customer account.

Algorithm	viewCust(custID)
	START
	  When customer click the “User Profile” button
	  System display customer profile with details
	END
	
	updateCust() & deleteCust(custID)
	START
	    Customer click the “User Profile” button
	    System display customer profile with details
	    Customer fill in back all the details
	    IF the customer click “Update Profile” button
	      THEN
	      The customer account details will be updated
	    ELSE IF customer click “Delete Profile” button
	      THEN
	      The customer account will be delete
	END	

4.4.2 runnerProfile [SDD-REQ-402]

Class Type	Boundary Class	
Responsibility	It displays runner profile with credentials used during runner account registration process.	
Attributes	Attributes Name	Attributes Type
	runnerID	int
	runnerusername	String
	runnerphnumber	String
	runneremail	String
	runnervehiclemodel	String
	runnervehicleplatenumber	String
	runnercity	String
	runnerbanktype	String
	runnerbankaccountnumber	String
Methods	Method Name	Description
	viewRun(runnerID)	To view the details about the runner
	updateRun()	To update the details about runner.
	deleteRun(runnerID)	To delete the runner account.
Algorithm	viewRun(runnerID) START When runner click the "User Profile" button System display runner profile with details END updateRun() & deleteRun(runnerID) START Runner click the "User Profile" button System display runner profile with details Runner fill in back all the details IF the runner click "Update Profile" button	

```
        THEN
        The runner account details will be updated
      ELSE IF runner click “Delete Profile” button
        THEN
        The runner account will be delete.
END
```

4.4.3 serviceProviderProfile [SDD-REQ-403]

Class Type	Boundary Class	
Responsibility	It displays service provider profile with credentials used during service provider registration process.	
Attributes	Attributes Name	Attributes Type
	spID	int
	spusername	String
	spphnumber	String
	spemail	String
	spcompanyname	String
	spaddress1	String
	spaddress2	String
	spaddress3	String
	spaddress4	String
	spbanktype	String
	spbankaccountnumber	String
Methods	Method Name	Description
	viewSP(spID)	To view the details about service provider.
	updateSP()	To update the details about service provider.
	deleteSP(spID)	To delete the service provider account.

Algorithm	viewSP(spID) START When service provider click the “User Profile” button System display service provider profile with details END updateSP() & deleteSP(spID) START Service provider click the “User Profile” button System display service provider profile with details Service provider fill in back all the details IF the service provider click “Update Profile” button THEN The service provider account details will be updated ELSE IF service provider click “Delete Profile” button THEN The service provider account will be delete. END

4.5 managePayment [SDD-REQ-500]

NUR AININ SOFIYA BINTI ABU KASIM

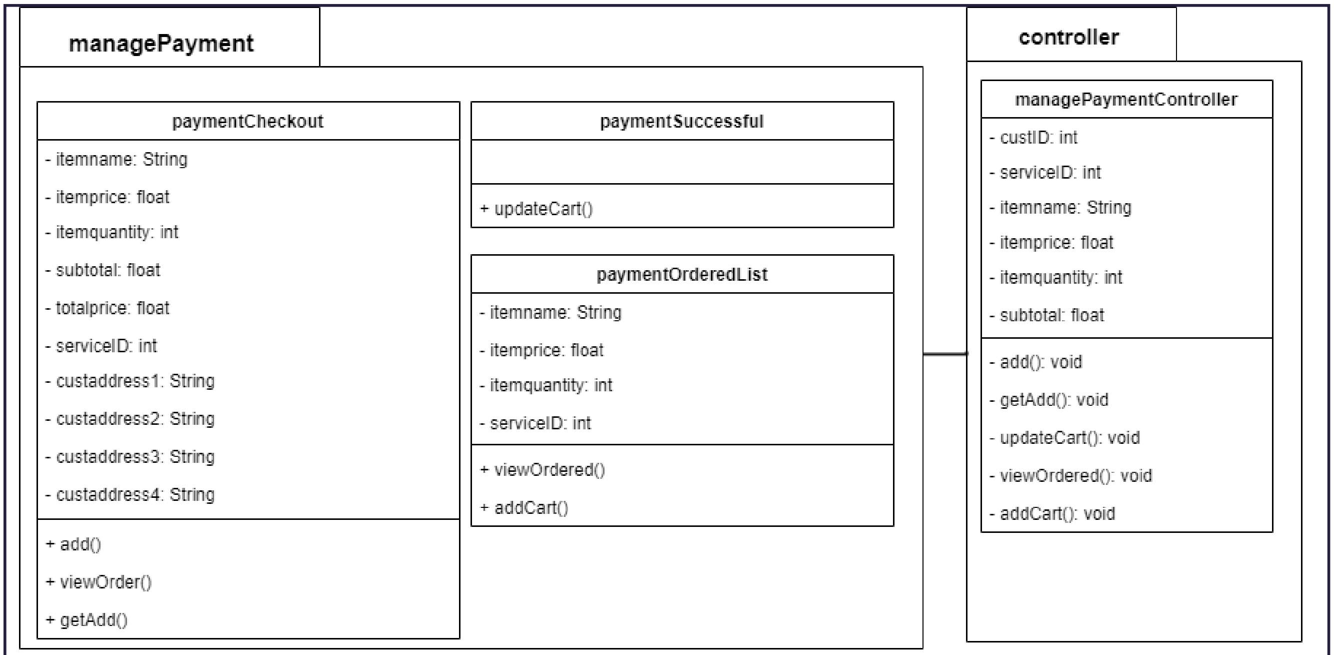


Figure 4.5.1: Manage Payment Package

4.5.1 paymentCheckout [SDD-REQ-501]

Class Type	Boundary Class	
Responsibility	It displays final decision of orders and total amount, the payment methods and the changeable delivery address.	
Attributes	Attributes Name	Attributes Type
	itemname	String
	itemprice	float
	itemquantity	int
	subtotal	float
	totalprice	float
	serviceID	int
	custaddress1	String
	custaddress2	String
	custaddress3	String
	custaddress4	String

	Method Name	Description
Methods	add()	To add the confirmed item to the payment list before making payment.
	viewOrder()	To display the ordered item from cart for the customer.
	getAdd()	To display the address of the customer.

Algorithm	add()
	START
	    IF (orderedItem == confirm)
	        Clicks “Confirm”
	        The item data is saved.
	    ELSE
	        Back to cart page to delete ordered item
	    ENDIF
	END
	
	viewOrder()
	START
	    Display the ordered item from the cart.
	END
	
	getAdd()
	START
	    Display customer’s address for delivery purpose.
	END	

4.5.2 paymentSuccessful [SDD-REQ-502]

Class Type	Boundary Class
Responsibility	It displays the payment status.

Attributes	Attributes Name	Attributes Type
Methods	Method Name	Description
	updateCart()	Update the order status
Algorithm	START Update item’s order status for notifying service provider END	

4.5.3 paymentOrderedList [SDD-REQ-503]

Class Type	Boundary Class	
Responsibility	It displays the paid ordered item.	
Attributes	Attributes Name	Attributes Type
	itemname	String
	itemprice	float
	itemquantity	int
	serviceID	int
Methods	Method Name	Description
	viewOrdered()	Display the paid ordered item.
	addCart()	Add the paid ordered item to cart.
Algorithm	viewOrdered() START Display the paid ordered item. END addCart() START Click “Reorder” Add the clicked item to cart. END	

4.6 manageTracking [SDD-REQ-600]

NUR DARWISYAH FAQIHAH BINTI LUTFI

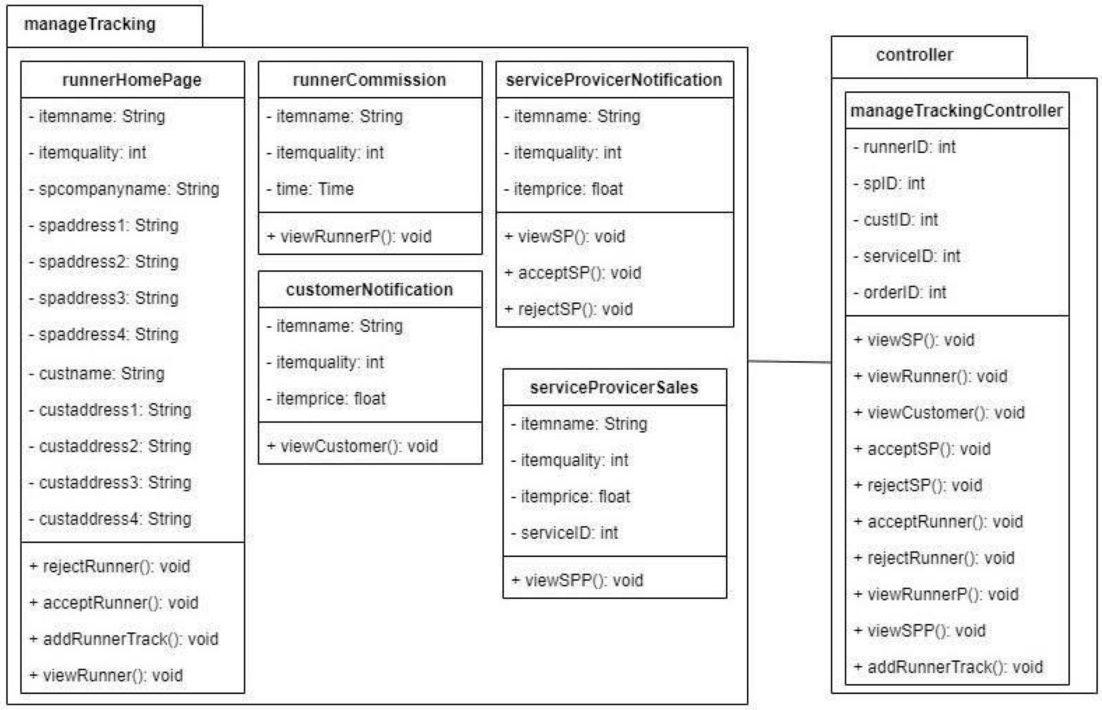


Figure 4.6.1: Provide Tracking and Analytic Package

4.6.1 runnerHomePage [SDD-REQ-601]

Class Type	Boundary Class	
Responsibility	To display the interface that the runner can update the delivery status and pending delivery order.	
Attributes	Attributes Name	Attributes Type
	itemname	String
	itemquantity	int
	spcompanyname	String
	spaddress1	String
	spaddress2	String
	spaddress3	String
	spaddress4	String

	custname	String
	custaddress1	String
	custaddress2	String
	custaddress3	String
	custaddress4	String
Methods	Method Name	Description
	rejectRunner()	To show the message that runner confirm fail to deliver the order.
	acceptRunner()	To show the message that runner confirm fail to deliver the order.
	addRunnerTrack()	Update the delivery status.
	viewRunner()	Display the pending order need to delivery by the customer.
Algorithm	rejectRunner() START When runner click the (X) button Then order status will be fail to delivery END acceptRunner() START When runner click the (√) button Then order status will be success to delivery END addRunnerTrack() START When runner click on the “OK” on the message The status of delivery will be updated. END viewRunner()	

	START Display the pending order that need to delivery END

4.6.2 runnerCommission [SDD-REQ-602]

Class Type	Boundary Class	
Responsibility	To display the commission of runner.	
Attributes	Attributes Name	Attributes Type
	itemname itemquantity time	String int Time
Methods	Method Name	Description
	viewRunnerP()	To display the commission detail.

Algorithm	viewRunnerP()
	START
	Display the runner commission details and total of the commission
	END	

4.6.3 customerNotification [SDD-REQ-603]

Class Type	Boundary Class	
Responsibility	To display the delivery status of order to customer.	
Attributes	Attributes Name	Attributes Type
	itemname itemquantity itemprice	String int float
Methods	Method Name	Description
	viewCustomer()	To display the order delivery status.
Algorithm	viewCustomer () START Display the order delivery status that updated by the runner END	

4.6.4 serviceProviderNotification [SDD-REQ-604]

Class Type	Boundary Class	
Responsibility	To display the incoming order to the service provider.	
Attributes	Attributes Name	Attributes Type
	itemname	String
	itemquantity	int
	itemprice	float
Methods	Method Name	Description
	acceptSP()	To show the message that service provider confirm to accept the order.
	rejectSP()	To show the message that service provider confirm to reject the order.
	viewSP()	Display the incoming order need to prepare by the service provider.
Algorithm	acceptSP() START When service provider click the (√) button Then order will be accepted END rejectSP() START When service provider click the (X) button Then order will be rejected. END viewSP() START Display the incoming order that need to prepare by the service provider END	

4.6.5 serviceProviderSales [SDD-REQ-605]

Class Type	Boundary Class	
Responsibility	To display the incoming order to the service provider.	
Attributes	Attributes Name	Attributes Type
	itemname itemquantity itemprice	String int float
Methods	Method Name	Description
	viewSPP()	To display the sales details to the service provider.

Algorithm	viewSPP()
	START
	Display the service provider sales details and total of the sales
	END	

4.7 LoginAndRegisterService [SDD-REQ-700]

FARAH HIDAYAH BINTI MOHD FADZIL

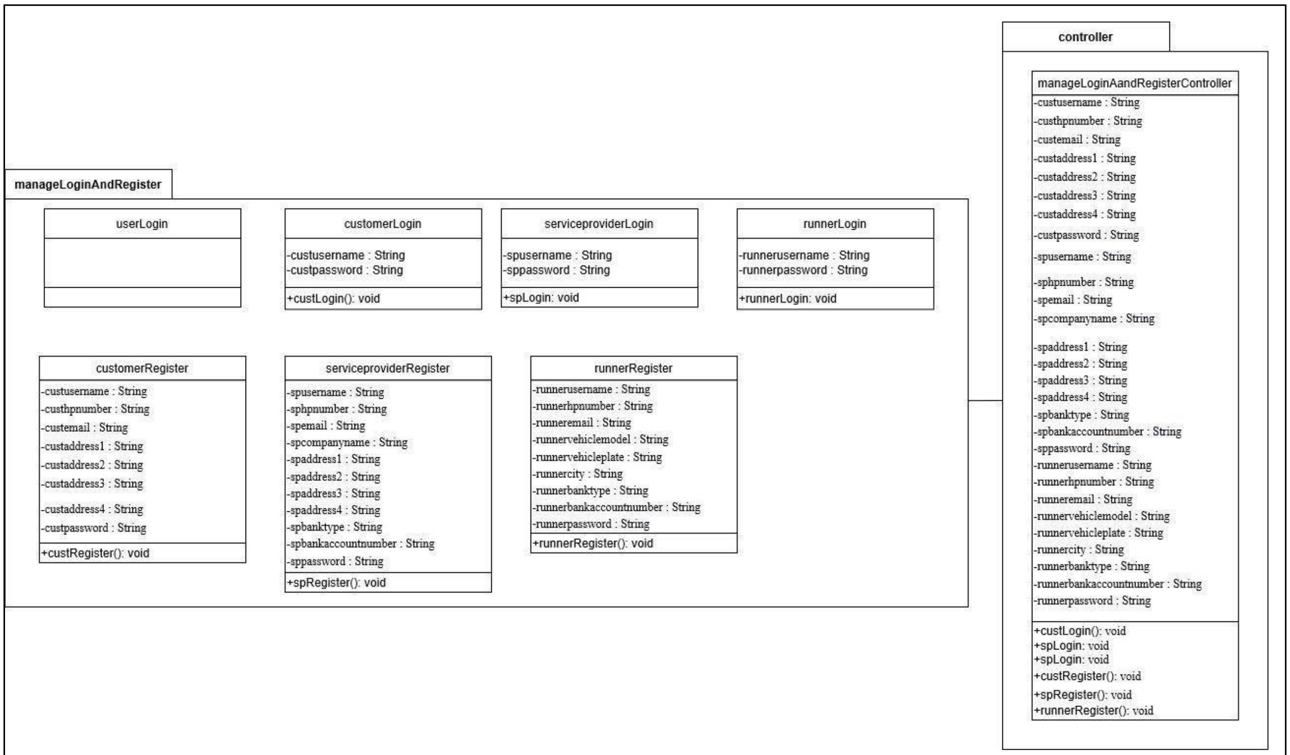


Figure 4.7.1: LoginAndRegisterService Package

4.7.1 manageLoginandRegisterController [SDD-REQ-701]

Class Type	Controller Class	
Responsibility	It manages and store the customer, runner and service provider account information.	
Attributes	Attributes Name	Attributes Type
	custusername	String
	custhpnumber	String
	custemail	String
	custaddress1	String
	custaddress2	String
	custaddress3	String
	custaddress4	String
	custpassword	String

	spusername	String
	sphpnumber	String
	spemail	String
	spcompanyname	String
	spaddress1	String
	spaddress2	String
	spaddress3	String
	spaddress4	String
	spbanktype	String
	spbankaccountnumber	String
	sppassword	String
	runnerusername	String
	runnerhpnumber	String
	runneremail	String
	runnervehiclemodel	String
	runnervehicleplate	String
	runnercity	String
	runnerbanktype	String
	runnerbankaccountnumber	String
	runnerpassword	String
Methods	Method Name	Description
	custRegister()	To enable user register in the system as customer.
	custLogin()	To enable user to login into as customer.
	spRegister()	To enable user register in the system as service provider.
	spLogin()	To enable user to login into as service provider.
	runnerRegister()	

	runnerLogin()	To enable user register in the system as runner. To enable user to login into as runner.

Algorithm	custRegister()
	START
	Create customer data into database.
	END
	
	custLogin()
	START
	Retrieve customer login data from database.
	END
	
	spRegister()
	START
	Create customer data into database.
	END
	
	spLogin()
	START
	Retrieve service provider login data from database.
	END
	
	runnerRegister()
	START
	Create runner data into database.
	END
	
	runnerLogin()
	START
	Retrieve runner login data from database.
	END	

4.8 manageOrderController [SDD-REQ-800]

NUR AININ SOFIYA BINTI ABU KASIM

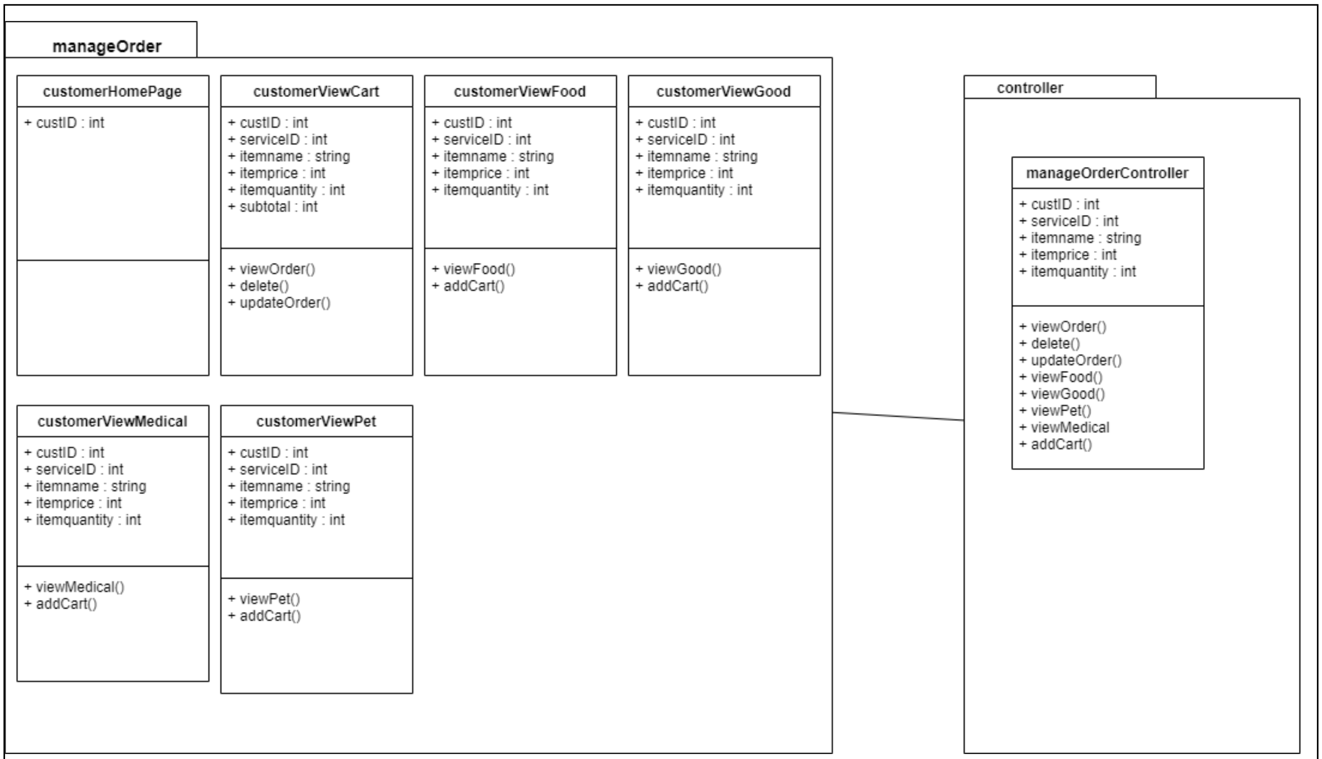


Figure 4.8.1: manageOrderController

4.8.1 manageOrderController [SDD-REQ-801]

Class Type	Controller Class	
Responsibility	It manages and store the customer order information	
Attributes	Attributes Name	Attributes Type
	custID	int
	serviceID	int
	itemname	string
	itemprice	int
	itemquantity	int
Methods	Method Name	Description
	viewOrder()	To enable the customer to view the order details

	delete()	To enable the customer to delete their order details
	updateOrder()	To enable the customer to update their order details
	viewFood()	To enable the customer to view the food services details
	viewGood()	To enable the customer to view the good services details
	viewPet()	To enable the customer to view the pet services details
	viewMedical()	To enable the customer to view the medical services details
	addCart()	To enable the customer to add the order to the cart

Algorithm	viewOrder()
	START
	Retrieve order data from the database and display in customer profile
	END
	delete()
	START
	Delete order data from the database
	profile
	END
	updateOrder()
	START
	Update order data from the database and display in customer profile
	END
	viewFood()	

	START Retrieve food service data from the database and display in customer profile END viewGood() START Retrieve good service data from the database and display in customer profile END viewPet() START Retrieve pet service data from the database and display in customer profile END viewMedical() START Retrieve medical service data from the database and display in customer profile END addCart() START Send the selected item to the database to store in the cart END

4.9 manageServiceController [SDD-REQ-900]

NUR DARWISYAH FAQIHAH BINTI LUTFI

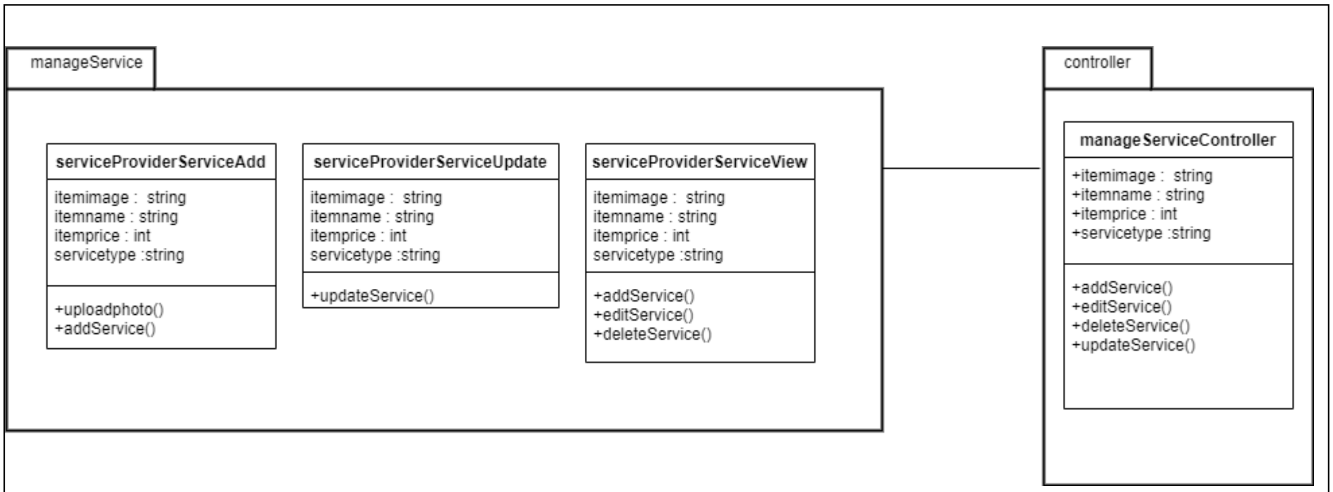


Figure 4.9.1: manageServiceController

4.9.1 manageUserProfileController [SDD-REQ-901]

Class Type	Controller Class	
Responsibility	It connects the serviceProviderServiceView interface, serviceProviderServiceUpdate interface, serviceProviderServiceAdd interface to the database to retrieve and store the data.	
Attributes	Attributes Name	Attributes Type
	itemimage	String
	itemname	String
	itemprice	int
	servicetype	String
Methods	Method Name	Description
	addService()	To enable the service provider to add service
	deleteService()	To allow the service provider to delete their service to sell

	updateService()	To enable the service provider to update their details service used during add service process
	viewAllItem()	To enable the service provider to view all their service with details used during add service process
	viewPerItem()	To enable the service provider to view per item service with details used during add service process
Algorithm	addService() START Send the detail service or item data to the database END updateService() START Service provider fill in service type, item name, item price Service provider click “Update Service” button IF update successful THEN system display success update message and redirect back to service Provider Service View interface ELSE THEN system display failed update message ELSE IF END deleteService() START Service provider click “Delete icon” button Service provider will be redirected to service Provider Service View page END	

	viewAllItem()
	START
	Retrieve service provider service item data from the database and display in service Provider Service View
	END
	
	viewPerItem()
	Start
	Retrieve service item data from the database and service Provider Service View
	END

4.10 manageUserProfileController [SDD-REQ-1000]

FARAH HIDAYAH BINTI MOHD FADZIL

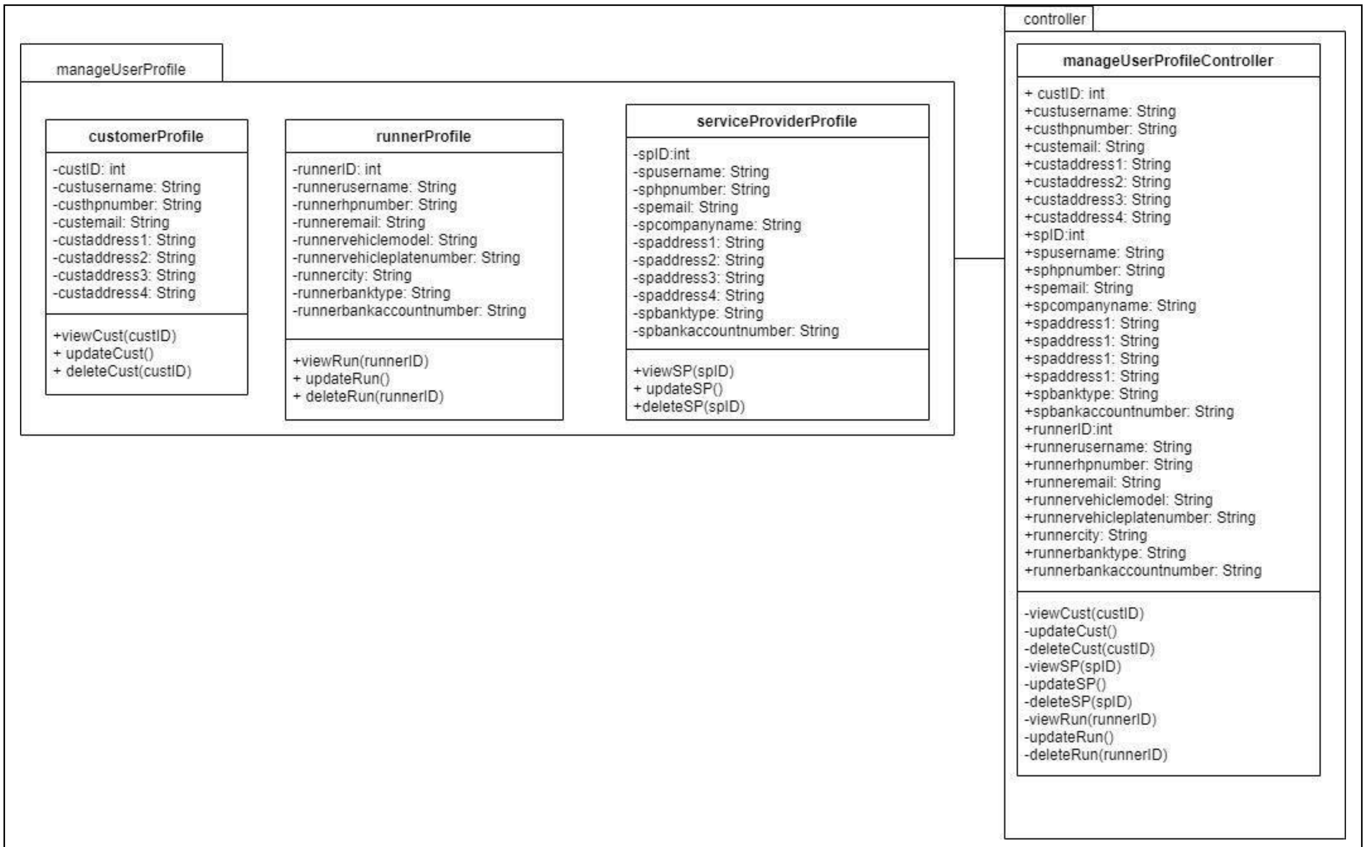


Figure 4.10.1:manageUserProfile Controller

4.10.1 manageUserProfileController [SDD-REQ-1001]

Class Type	Controller Class	
Responsibility	It manages and store any changes made to the users profile data and retrieve the users profile data.	
Attributes	Attributes Name	Attributes Type
	custID	int
	custusername	String
	custphnumber	String
	custemail	String
	custaddress1	String

	custaddress2	String
	custaddress3	String
	custaddress4	String
	spID	int
	spusername	String
	sphpnumber	String
	spemail	String
	spcompanyname	String
	spaddress1	String
	spaddress2	String
	spaddress3	String
	spaddress4	String
	spbanktype	String
	spbankaccountnumber	String
	runnerID	int
	runnerusername	String
	runnerhpnumber	String
	runneremail	String
	runnervehiclemodel	String
	runnervehicleplatenumber	String
	runnercity	String
	runnerbanktype	String
	runnerbankaccountnumber	String
Methods	Method Name	Description
	viewCust(custID)	To enable the customer to view their profile with details used during registration process
	updateCust()	To enable the customer to update their details used during registration process

	deleteCust(custID)	To allow the customer to delete their registered account as customer
	viewSP(spID)	To enable the service provider to view their profile with details used during registration process
	updateSP()	To enable the service provider to update their details used during registration process
	deleteSP(spID)	To allow the service provider to delete their registered account as service provider
	viewRun(runnerID)	To enable the runner to view their profile with details used during registration process
	updateRun()	To enable the runner to update their details used during registration process
	deleteRun(runnerID)	To allow the runner to delete their registered account as runner
Algorithm	viewCust(custID) START Retrieve customer data from the database and display in customer profile END updateCust() START Customer fill in phone number, email, address Customer click “Update Profile” button IF update successful THEN The system display success update message and redirect	

```
                back to customer profile interface
         ELSE IF update failed
            THEN
                 The system display failed update message and redirect back
                 to customer profile interface
END
  deleteCust(custID)
START
   Customer click “Delete Profile” button
   The system will display success delete message and customer will be
    redirected to user login homepage
END
  viewSP(spID)
START
   Retrieve service provider data from the database and display in
   service provider profile
END
  updateSP()
START
    Service provider fill in phone number, email, company name, address,
   bank type, bank account number
   Service provider click “Update Profile” button
       IF update successful
            THEN
             The system display success update message and redirect
             back to service provider profile interface
       ELSE IF update failed
            THEN
             The system display failed update message and redirect back
              to service provider profile interface
```

```
END

deleteSP(spID)
START
  Service provider click “Delete Profile” button
  The system display success delete message and service provider will
  redirected to user login homepage
END
 viewRun(runnerID)
Start
  Retrieve runner data from the database and display in runner profile
END

updateRun()
Start
    Runner fill in phone number, email, vehicle model, vehicle plate
    number, delivery city, bank type, bank account number
    Runner click “Update Profile” button
     IF update successful
         THEN
          The system display success update message and redirect
           back to runner profile interface
     ELSE IF update failed
          THEN
           The system display failed update message and redirect back
           to runner profile interface
END

deleteRun(runnerID)
Start
    Runner click “Delete Profile” button
  The system display success delete message and runner will be
```

	redirected to user login homepage END

4.11 managePaymentController [SDD-REQ-1100]

NUR AININ SOFIYA BINTI ABU KASIM

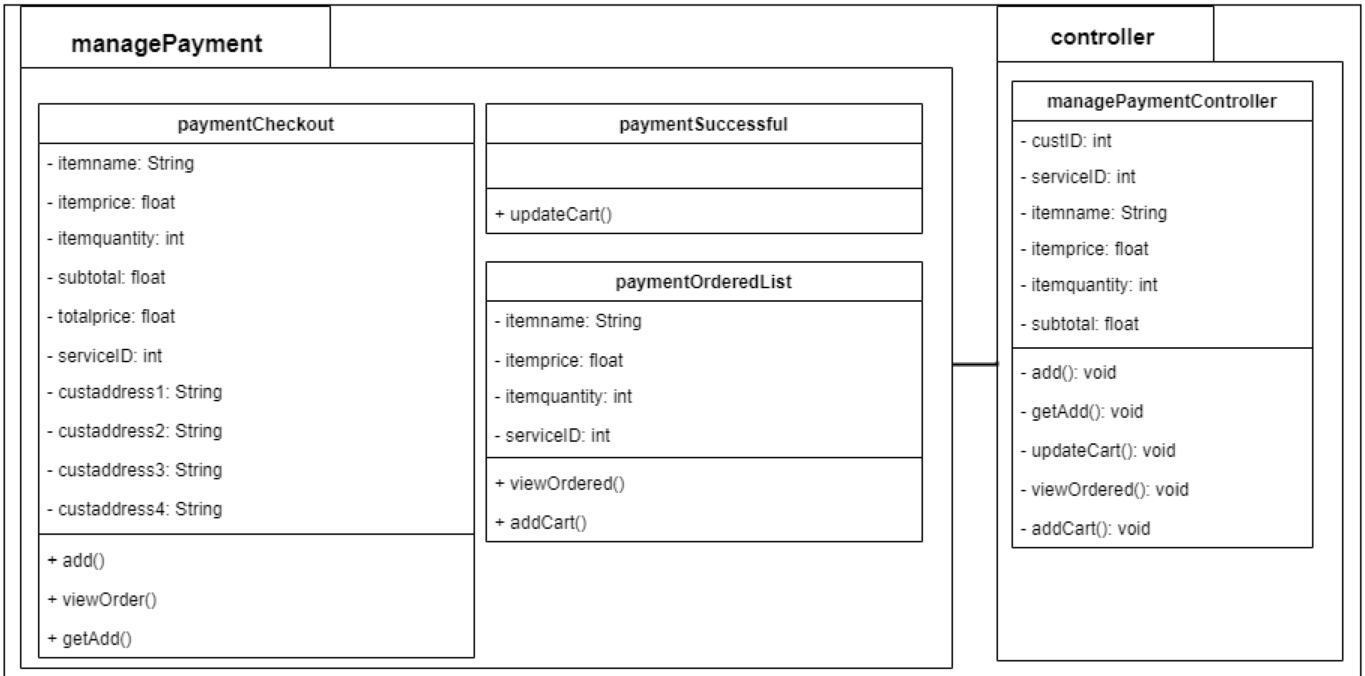


Figure 4.11.1: managePayment Controller

4.11.1 managePaymentController [SDD-REQ-1101]

Class Type	Controller Class	
Responsibility	It manages and processes the customer payment information and generate the paid ordered item of the customer.	
Attributes	Attributes Name	Attributes Type
	custID serviceID itemname	int int String

	itemprice itemquantity subtotal	float int float
Methods	Method Name	Description
	add()	To add the confirmed item to the payment list before making payment.
	viewOrder()	To display the ordered item from cart for the customer.
	getAdd()	To display the address of the customer.
	updateCart()	Update the order status.
	viewOrdered()	Display the paid ordered item.
	addCart()	Add the paid ordered item to cart.

Algorithm	add()
	START
	Send the selected item data to the database
	END
	
	viewOrder()
	START
	Retrieve the ordered item from cart
	END
	
	getAdd()
	START
	Retrieve the address from the customer profile
	END
	
	updateCart()	

```
START
    Send the data to change the status of paid ordered item in the database
END

viewOrdered()
START
    Retrieve the paid ordered item data from database
END

addCart()
START
    Send the selected item to the database to store in the cart
END
```

4.12 manageTrackingController [SDD-REQ-1200]

NUR DARWISYAH FAQIHAH BINTI LUTFI

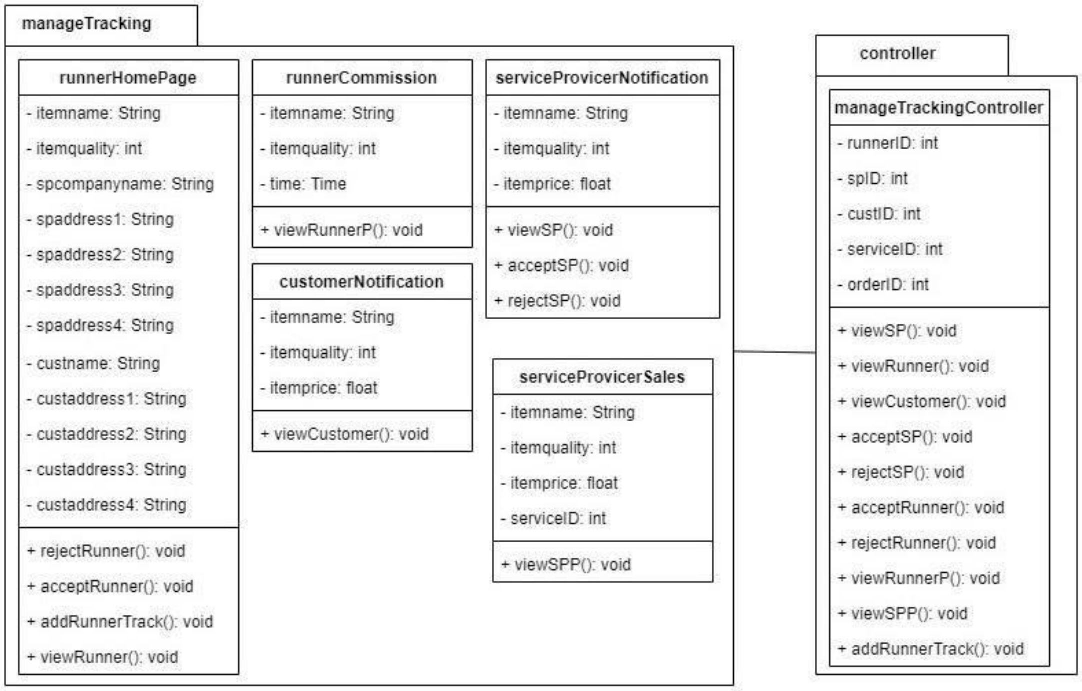


Figure 4.12.1: manageTracking Controller

4.12.1 manageTrackingController [SDD-REQ-1201]

Class Type	Controller Class	
Responsibility	It is connecting the interface and the database to retrieve and store the data.	
Attributes	Attributes Name	Attributes Type
	serviceID orderID runnerID custID spID	int int int int int
Methods	Method Name	Description
	viewSP()	To retrieve the order details from the database.

	viewRunner()
	viewCustomer()
	acceptSP()
	rejectSP()
	acceptRunner()
	rejectRunner()
	viewRunnerP()
	viewSPP()
	addRunnerTrack()
		To retrieve the delivery order details from the database.
		To retrieve the updated delivery status from the database that update by the runner.
		To accept the order that order by customer. To reject the order that order by the customer.
		To update the delivery status of the order when it is success to delivery to customer.
		To update the delivery status of the order when it is failed to delivery to customer.
		To retrieve the commission data from the database.
		To retrieve the sales data from the database.
		To add the delivery details to the database.

Algorithm	viewSP()
	START
	    Retrieve the order details from the database
	END
	
	viewRunner()
	START
	    Retrieve the delivery order details from the database
	END
	
	viewCustomer()
	START
	    Retrieve the delivery order status from the database
	END
	
	acceptRunner()	

```
START
    Update the success delivery status to the database
END
 rejectRunner()
START
    Update the fail delivery status to the database
END
 acceptSP()
START
    Update the order has been accepted by the service provider to
database
END
 rejectSP()
START
    Update the order has been rejected by the service provider to
database
END
 viewRunnerP()
START
    Retrieve the commission details from the database
END
 viewSPP()
START
    Retrieve the sales details from the database
END
 addRunnerTrack()
```

	START Add the delivery details and delivery status to the database END

4.13 Functional Data [SDD-REQ-1300]

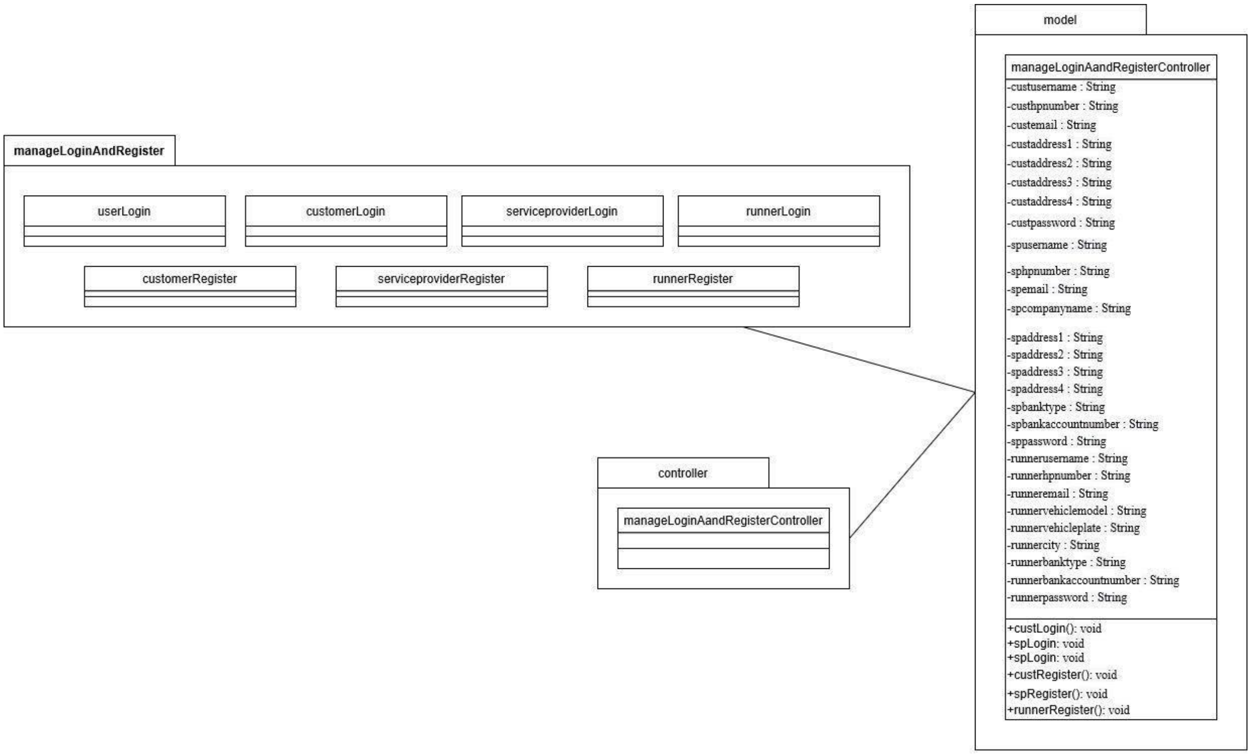


Figure 4.13.1: LoginAndRegisterData Package

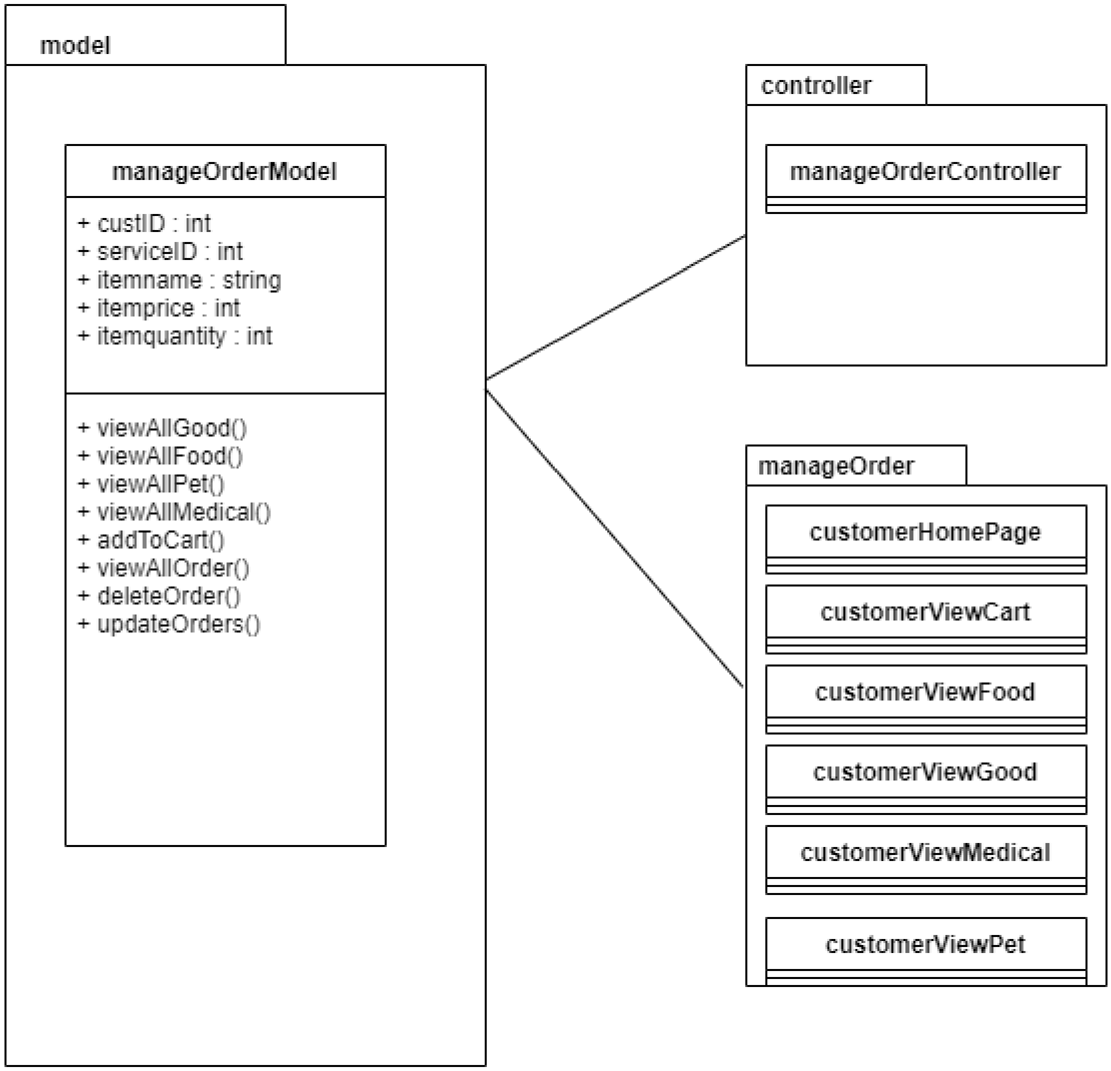


Figure 4.13.2: manageOrderData Package

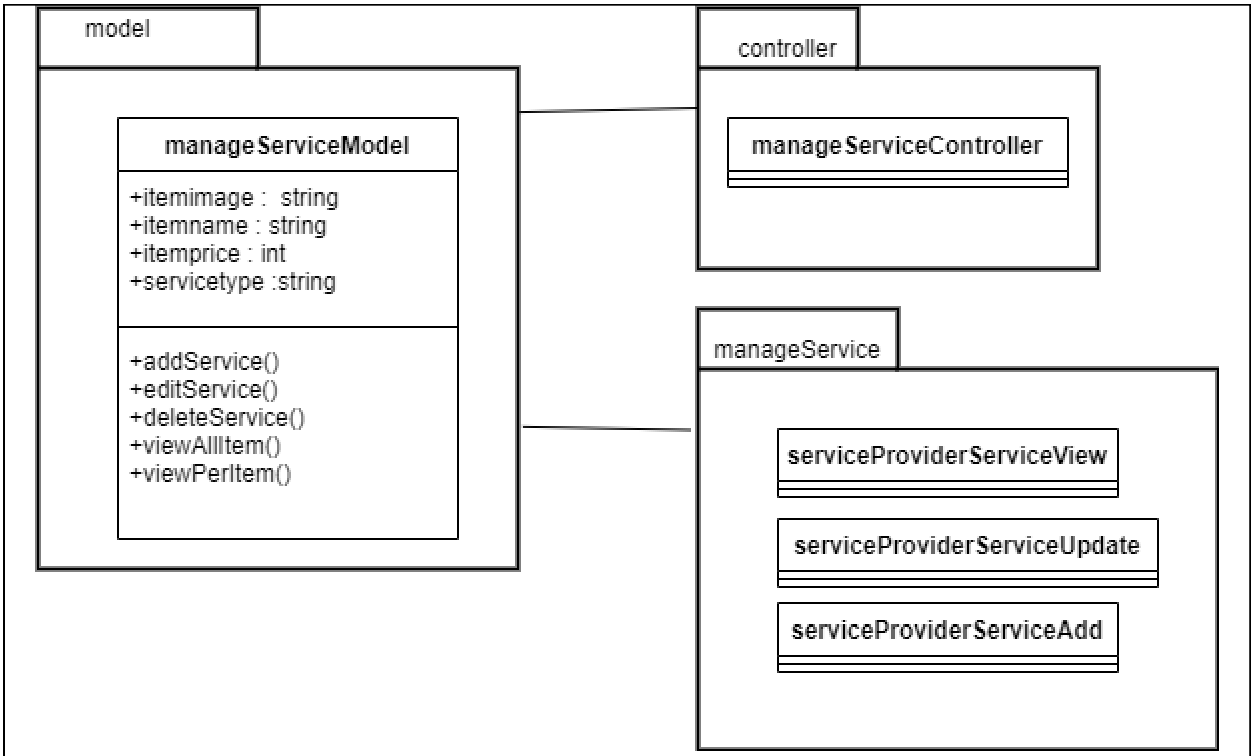


Figure 4.13.3: manageServiceData Package

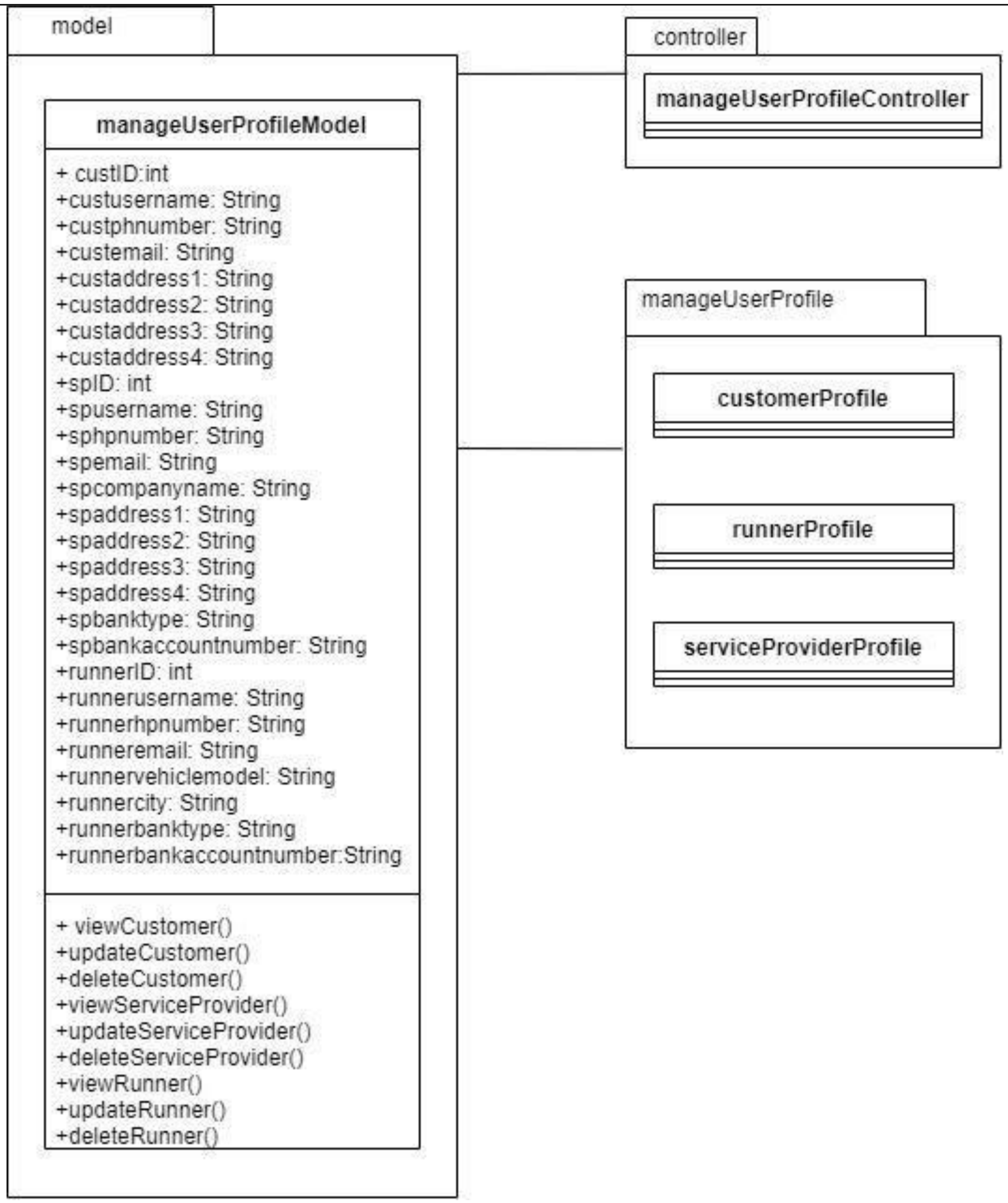


Figure 4.13.4: UserProfileData Package

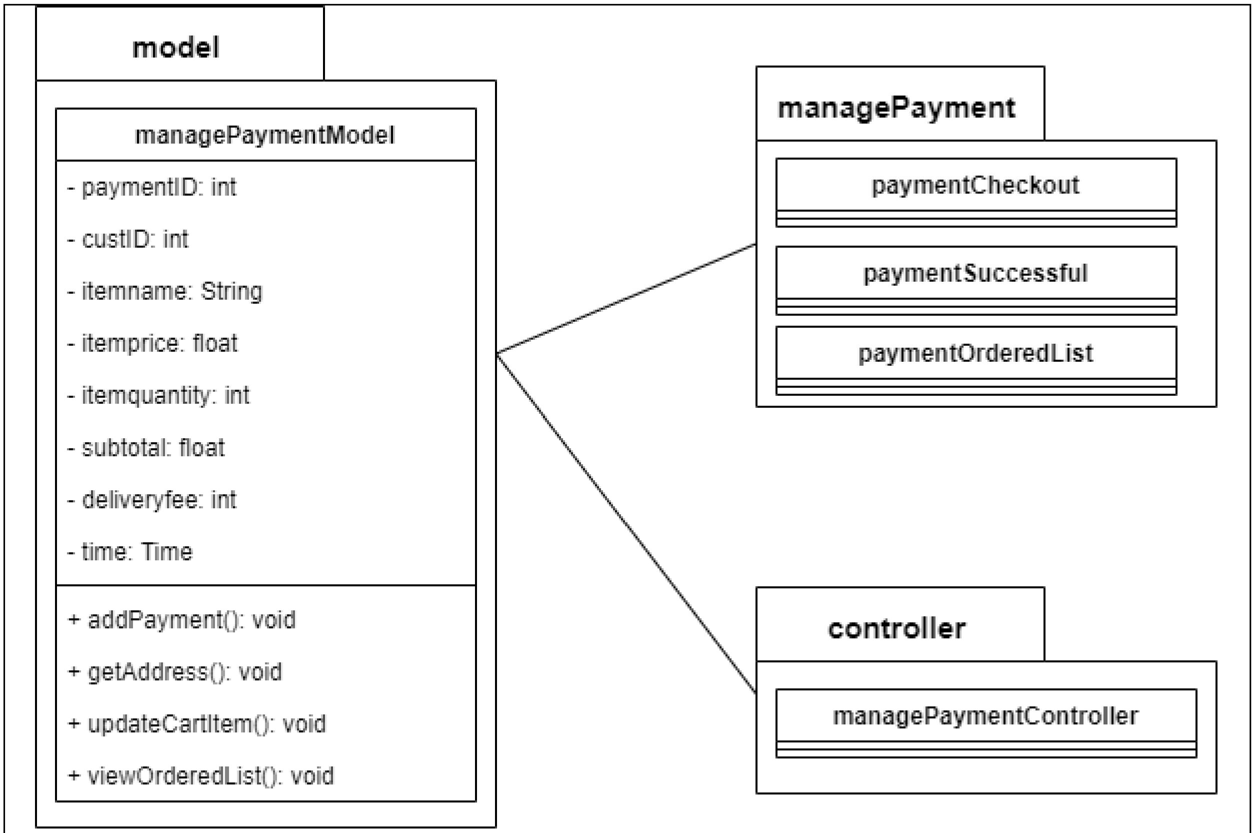


Figure 4.13.5: PaymentData Package

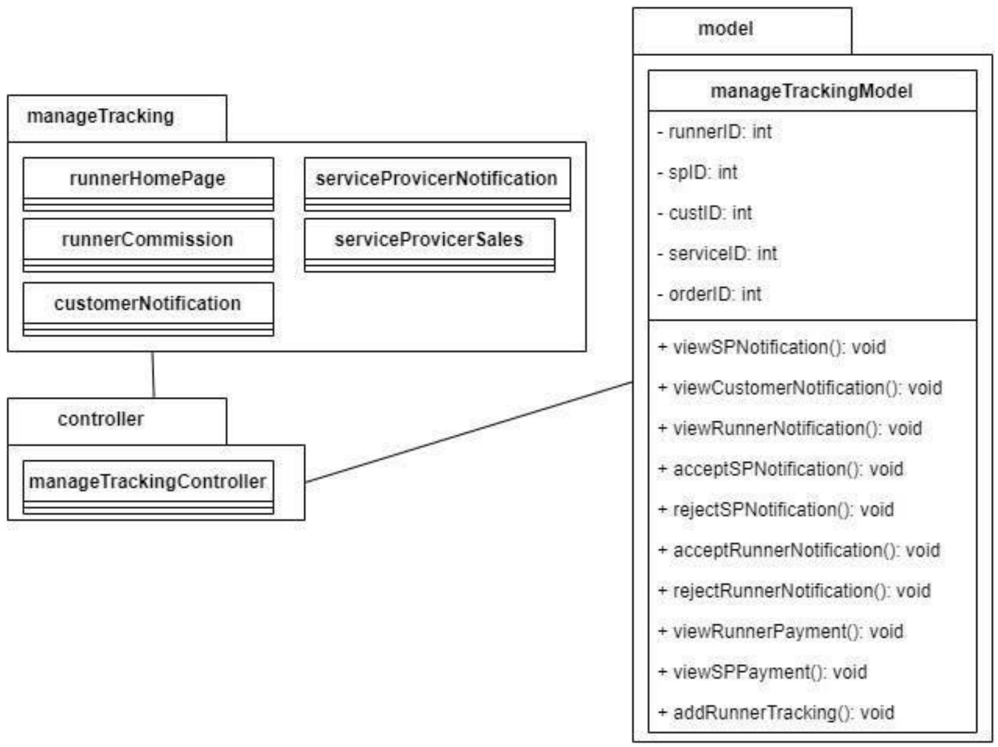

model
manageTrackingModel
- runnerID: int
- spID: int
- custID: int
- serviceID: int
- orderID: int
+ viewSPNotification(): void
+ viewCustomerNotification(): void
+ viewRunnerNotification(): void
+ acceptSPNotification(): void
+ rejectSPNotification(): void
+ acceptRunnerNotification(): void
+ rejectRunnerNotification(): void
+ viewRunnerPayment(): void
+ viewSPPayment(): void
+ addRunnerTracking(): void
manageTracking
runnerHomePage
serviceProvicerNotification
runnerCommission
serviceProvicerSales
customerNotification
controller
manageTrackingController

Figure 4.13.6: manageTrackingModel

4.13.1 manageLoginAndRegisterModel [SDD-REQ-1301]

FARAH HIDAYAH BINTI MOHD FADZIL

Class Type	Model Class	
Responsibility	It stores everything about the customer, service provider and runner information.	
Attributes	Attributes Name	Attributes Type
	custID custusername custhpnumber custemail custaddress1 custaddress2	String String String String String String

	custaddress3	String
	custaddress4	String
	custpassword	String
	spID	String
	spusername	String
	sphpnumber	String
	spemail	String
	spcompanyname	String
	spaddress1	String
	spaddress2	String
	spaddress3	String
	spaddress4	String
	spbanktype	String
	spbankaccountnu	String
	mber	String
	sppassword	String
	runnerID	String
	runnerusername	String
	runnerhpnumber	String
	runneremail	String
	runnervehiclemod	String
	el	String
	runnervehicleplate	String
	runnercity	String
	runnerbanktype	String
	runnerbankaccoun	
	tnumber	
	runnerpassword	
Methods	Method Name	Description
	custRegister()	To store customer data into database.

	custLogin()
	spRegister()
	spLogin()
	runnerRegister()
	runnerLogin()
		To retrieve customer data from database.
		To store service provider data into database.
		To retrieve service provider data from database.
		To store runner data into database.
		To retrieve runner data from database.

Algorithm	custRegister()
	START
	Create customer data into database.
	END
	
	custLogin()
	START
	Retrieve customer login data from database.
	END
	
	spRegister()
	START
	Create customer data into database.
	END
	
	spLogin()
	START
	Retrieve service provider login data from database.
	END	

	runnerRegister()
	START
	Create runner data into database.
	END
	
	runnerLogin()
	START
	Retrieve runner login data from database.
	END

4.13.2 manageOrderModel [SDD-REQ-1302]

NUR AININ SOFIYA BINTI ABU KASIM

Class Type	Model Class	
Responsibility	It stores everything about the customer order.	
Attributes	Attributes Name	Attributes Type
	custID	int
	serviceID	int
	itemname	string
	itemprice	int
	itemquantity	int
Methods	Method Name	Description
	viewAllGood()	To retrieve the good services about customer from the database
	viewAllFood()	To retrieve the food services about customer from the database
	viewAllPet()	To retrieve the pet services about customer from the database
	viewAllMedical()	To retrieve the medical services about customer from the database
	addToCart()	To add item to the database
	viewAllOrder()	

	deleteOrder()
	updateOrders()
		To retrieve the good services about customer from the database
		To delete the customer order to database
		To update the customer order to database

Algorithm	viewAllGood()
	START
	        Retrieve the data about good services from database
	END
	
	viewAllFood()
	START
	        Retrieve the data about food services from database
	END
	
	viewAllPet()
	START
	        Retrieve the data about pet services from database
	END
	
	viewAllMedical()
	START
	        Retrieve the data about medical services from database
	END
	
	addToCart()
	START
	    If(custID==custID, serviceID==serviceID, itemname==itemname, itemprice==itemprice, itemquantity==itemquantity)
	     Store the item details into the database
	END
	
	viewAllOrder()	

```
START
    If( custID==custID)
     Retrieve the data about order details from database
END

deleteOrder()
START
  If (custID==custID, serviceID==serviceID)
          Delete the order details from database
  END IF
END

updateOrders()
START
    If( custID==custID, serviceID==serviceID, itemname==itemname,
    itemprice==itemprice, itemquantity==itemquantity)
          Update the order details into database
  END IF
END
```

4.13.2 manageServiceModel [SDD-REQ-1303]

NUR DARWISYAH FAQIHAH BINTI LUTFI

Class Type	Model Class	
Responsibility	It stores everything about the service provide by service provider.	
Attributes	Attributes Name	Attributes Type
	itemimage	String
	itemname	String
	itemprice	int
	servicetype	String
Methods	Method Name	Description
	addService()	To add service to the database
	deleteService()	To delete the service to database
	updateService()	To update service in database
	viewPerItem()	To retrieve the detail services per item from the database
	viewAllItem()	To retrieve the detail services from the database

Algorithm	addService()
	START
	    If(spID== spID, servicetype== servicetype, itemname==itemname, itemprice==itemprice, itemimage== itemimage)
	     Store the service details into the database
	END
	
	deleteService()
	START
	  If (serviceID==serviceID)	

```
             Delete the service details from database
   END IF
END

updateService()
START
  If (itemname==itemname, itemprice==itemprice,
servicetype==servicetype, serviceID==serviceID)
             Update the service details into database
  END IF
END

viewPerItem()
START
    If(serviceID==serviceID)
     Retrieve the data about service details from database
END

viewAllItem()
START
      Retrieve the data about service details from database
END
```

4.13.4 manageUserProfileModel [SDD-REQ-1304]

FARAH HIDAYAH BINTI MOHD FADZIL

Class Type	Model Class	
Responsibility	It stores everything about the customer, service provider and runner profiles information and also store data if any changes made for their details. It also can retrieve information about that particular user.	
Attributes	Attributes Name	Attributes Type
	custID	int
	custusername	String
	custphnumber	String
	custemail	String
	custaddress1	String
	custaddress2	String
	custaddress3	String
	custaddress4	String
	spID	int
	spusername	String
	sphpnumber	String
	spemail	String
	spcompanyname	String
	spaddress1	String
	spaddress2	String
	spaddress3	String
	spaddress4	String
	spbanktype	String
	spbankaccountnumber	String
	runnerID	int
	runnerusername	String
	runnerhpnumber	String
	runneremail	String
	runnervehiclemodel	String
	runnercity	String

	runnerbanktype runnerbankacoountnumber	String String
Methods	Method Name	Description
	viewCustomer()	To retrieve the details about customer from the database
	updateCustomer()	To update the details about customer in the database
	deleteCustomer()	To delete the customer account in the database
	viewServiceProvider()	To retrieve the details about service provider from the database
	updateServiceProvider()	To update the details about service provider in the database
	deleteServiceProvider()	To delete the service provider account in the database
	viewRunner()	To retrieve the details about runner from the database
	updateRunner()	To update the details about customer in the database
	deleteRunner()	To delete the runner account in the database

Algorithm	viewCustomer ()
	START
	    If(custID==custID)
	            Retrieve the data about customer from database
	    END IF
	END	

```
 updateCustomer()
START
   If (custID==CustID, custusername==custusername, custhpnumber==
   custhpnumber, custemail==custemail, customeraddress1==
   customeraddress1,customeraddress2==customeraddress2,
   customeraddress3==customeraddress3, customeraddress4==
   customeraddress4)
          Update the data about customer in database
  END IF
END
 deleteCustomer()
START
  If (custID==custID)
          Delete the customer account from database
  END IF
END
 viewServiceProvider()
START
    If( spID==spID)
           Retrieve the data about service provider from database
    END IF
END
 updateServiceProvider()
START
  If (spID==spID, spusername==spusername, sphpnumber==
   sphpnumber, spemail==spemail, spcompanyname==
   spcompanyname,spaddress1==spaddress1,
   spaddress2==spaddress2, spaddress3==
    spaddress3,spaddress4==spaddress4,spbanktype==spbanktype,
    spbankaccountnumber==spaccountnumber)
           Update the data about service provider in database
   END IF
END

  deleteServiceProvider()
START
   If (sp==spID)
          Delete service provider account in database
   END IF
END
  viewRunner()
START
     If( runnerID==runnerID)
             Retrieve the data about runner from database
     END IF
END
  updateRunner()
START
  If (runnerID==runnerID, runnerusername==runnerusername,
    runnerhpnumber==runnerhpnumber,runneremail==runneremail,
    runnervehiclemodel==runnervehiclemodel, runnercity==runnercity,
    runnerbanktype==runnerbanktype,runnerbankaccountnumber==
    runnerbankaccountnumber)
          Update the data about runner in database
   END IF
END
```

```
deleteRunner()
START
    If( runnerID==runnerID)
            Delete runner account in database
    END IF
END
```

4.13.5 managePaymentModel [SDD-REQ-1305]

NUR AININ SOFIYA BINTI ABU KASIM

Class Type	Model Class	
Responsibility	It stores everything about the payment transaction activities.	
Attributes	Attributes Name	Attributes Type
	paymentID	int
	custID	int
	itemname	String
	itemprice	float
	itemquantity	int
	subtotal	float
	deliveryfee	int
	time	Time
Methods	Method Name	Description
	addPayment()	To store the confirmed ordered item to the database for calculation
	getAddress()	To retrieve the customer profile’s address
	updateCartItem()	To update the ordered item status

	viewOrderedList()	To retrieve the ordered item from database

4.13.6 manageTrackingModel [SDD-REQ-1306]

NUR DARWISYAH FAQIHAH BINTI LUTFI

Class Type	Model Class	
Responsibility	It stores the data about the delivery status details.	
Attributes	Attributes Name	Attributes Type
	serviceID	int
	orderID	int
	runnerID	int
	custID	int
	spID	int
Methods	Method Name	Description
	viewSPNotification()	To retrieve the ordered item by customer from the database.
	viewCustomerNotification()	To retrieve the status of the order from the database that update by the runner.
	viewRunnerNotification()	To retrieve the order details from the database.
	acceptSPNotification()	To update the status of order by service provider.
	rejectSPNotification()	To update the status of order by service provider.
	acceptRunnerNotification()	To update the updated delivery status of order by runner.
	rejectRunnerNotification()	To update the update delivery status of order by runner.
	viewRunnerPayment()	To retrieve the commission details from the database.
	viewSPPayment()	To retrieve the sales details from the database.
		To store the delivery details into the database.

	addRunnerTracking()	
Algorithm	viewSPNotification() START If (spID==spID) Retrieve the data END IF END viewCustomerNotification() START If (custID==custID) Retrieve the data from database END IF END viewRunnerNotification() START If (runnerID==runnerID) Retrieve the data from database END IF END acceptSPNotification() START If (spID==spID) Update the data to database END IF END rejectSPNotification() START If (spID==spID) Update the data to database	

```
    END IF
END
 acceptRunnerNotification()
START
    If ( runnerID==runnerID)
    If (serviceID==serviceID)
         Update the delivery status to database
    END IF
    END IF
END
 rejectRunnerNotification()
START
    If ( runnerID==runnerID)
    If (serviceID==serviceID)
         Update the delivery status to database
    END IF
    END IF
END
 viewRunnerPayment()
START
    If ( runnerID==runnerID)
         Retrieve the commission details from database
    END IF
END
 viewSPPayment()
START
    If ( spID==spID)
         Retrieve the sales details from database
    END IF
END

addRunnerTracking()
START
    Store the delivery details and status into the database
END
```

5. REQUIREMENT TRACEABILITY

5.1 Manage Login And Register [SRS-REQ-100]

FARAH HIDAYAH BINTI MOHD FADZIL

Requirement ID	Description	Design ID
SRS-REQ-101	System shall enable user to choose type of account to login into the system such as customer, service provider and runner.	SDD-REQ-101
SRS-REQ-102	System shall enable user to choose type of account to register an account in the system such as customer, service provider and runner.	SDD-REQ-102
SRS-REQ-103	System should enable user to enter every data successfully.	SDD-REQ-103
SRS-REQ-104	System should able to display customer home interface if user successfully login into the system as customer.	SDD-REQ-104
SRS-REQ-105	System should able to display service provider home interface if user successfully login into the system as service provider.	SDD-REQ-105
SRS-REQ-106	System should able to display runner home interface if user successfully login into the system as runner.	SDD-REQ-106

5.2 Manage Order [SRS-REQ-200]

NUR AININ SOFIYA BINTI ABU KASIM

Requirement ID	Description	Design ID
SRS-REQ-201	System should allow users view the homepage	SDD-REQ-201
SRS-REQ-202	System should allow users to view the cart	SDD-REQ-202
SRS-REQ-203	System should allow users to view food services	SDD-REQ-203
SRS-REQ-204	System should allow users to view good services	SDD-REQ-204
SRS-REQ-205	System should allow users to view medical services	SDD-REQ-205
SRS-REQ-206	System should allow users to view pet services	SDD-REQ-205

5.3 Manage Service [SRS-REQ-300]

NUR DARWISYAH FAQIHAH BINTI LUTFI

Requirement ID	Description	Design ID
SRS-REQ-301	The system shall allow to view list service provide	SRS-REQ-301
SRS-REQ-302	The system shall allow service provider to add new service	SRS-REQ-302
SRS-REQ-303	The system shall allow service provide to upload image or item of service provide	SRS-REQ-303
SRS-REQ-304	The system shall allow update detail service	SRS-REQ-304
SRS-REQ-305	The system shall allow service provider to edit and delete item or service	SRS-REQ-305

5.4 Manage User Profile [SRS-REQ-400]

FARAH HIDAYAH BINTI MOHD FADZIL

Requirement ID	Description	Design ID
SRS-REQ-401	The customer, service provider and runner can manage their account such as view their account details	SDD-REQ-401 SDD-REQ-402 SDD-REQ-403
SRS-REQ-402	The system will prompt success update message every time changes made to user account details when user pressed “Update Profile” button and redirected the user back to their user profile.	SDD-REQ-1001
SRS-REQ-403	The system will prompt success delete message every time user decided to delete their respective account by pressing “Delete Profile” button and redirected the user back to login page.	SDD-REQ-1001
SRS-REQ-404	The system will record any changes made to the user accounts if they choose to Update Profile or even delete details regarding that particular user if they choose to Delete Profile.	SDD-REQ-1001 SDD-REQ-1304

5.5 Manage Payment [SRS-REQ-500]

NUR AININ SOFIYA BINTI ABU KASIM

Requirement ID	Description	Design ID
SRS-REQ-501	The system should not be able to make transaction with the cash.	SDD-REQ-501 SDD-REQ-1101
SRS-REQ-501	The customer can enter username and password again, if the username and password are wrong	SDD-REQ-501 SDD-REQ-1101
SRS-REQ-501	The customer can choose the payment method either PayPal or credit/debit card	SDD-REQ-501 SDD-REQ-1101
SRS-REQ-503	The customer able to view their past ordered items	SDD-REQ-502 SDD-REQ-1101

5.6 Provide Tracking and Analytic [SRS-REQ-600]

NUR DARWISYAH FAQIHAH BINTI LUTFI

Requirement ID	Description	Design ID
SRS_REQ_601	The customer is able to track the order that order by them after receive.	SDD-REQ-603
SRS_REQ_602	The service provider can decide to accept or reject the order that order by customer	SDD-REQ-604
SRS_REQ_603	The service provider is able to view the report.	SDD-REQ-605
SRS_REQ_604	The runner can update the delivery status when they delivered to customer.	SDD-REQ-601
SRS_REQ_605	The runner can view the commission report.	SDD-REQ-602