



SRS Document SEM II 20192020

Group Name

1. Kam Jian Wei
2. Yamunnawahthi a/p Somasundharam
3. Safinatunnajah Binti Sahrawi
4. Choong Zhen Yuan

CB18137

CB18042

CB18002

CB18059



TABLE OF CONTENTS

1. INTRODUCTION	4
1.1 Purpose	4
1.2 System Identification	4
1.3 System Overview	4
1.4 Referenced Document	5
2. DATA DESIGN	6
2.1 Entity Relationship Diagram (ERD)	6
2.2 Data Dictionary	7
3. GENERAL ARCHITECTURE	11
3.1 Layered Architecture	11
3.1.1 Application Layer	11
3.1.2 Business Services Layer	15
3.1.3 Middleware Layer	17
3.2 MVC Package Relationship	18
4. DETAIL DESIGN	19
4.1 CustomerView [Kam Jian Wei CB18137]	19
4.2 custAddCart [SDD-REQ-100][Kam Jian Wei CB18137]	19
4.3 custMakePayment [SDD-REQ-200] [Choong Zhen Yuan CB18059]	24
4.4 custOrder [SDD-REQ-300][Kam Jian Wei CB18137]	27
4.5 custUpdateInfo [SDD-REQ-400][Kam Jian Wei CB18137]	29
4.6 RunnerView [SDD-REQ-500] [Yamunnawahthi a/p Somasundharam CB18042]	30
4.7 ServiceProviderView [Safinatunnajah Binti Sahrawi CB18002]	33
4.8 spInfo [SDD-REQ-600][Safinatunnajah Binti Sahrawi CB18002]	33
4.9 spProduct [SDD-REQ-700] [Safinatunnaja, Kam, Yamunna]	35
4.10 spConfirmOrder [SDD-REQ-800][Safinatunnajah Binti Sahrawi]	43
4.11 UserView [SDD-REQ-900][Choong Zhen Yuan]	44
4.12 Customercontroller[SDD-REQ-1000][Kam Jian Wei CB18137]	46
4.13 Customermodel [SDD-REQ-1100][Kam Jian Wei CB18137]	48
4.14 Runnercontroller [SDD-REQ-1200] [Yamunnawahthi a/p Somasundharam CB18042]	50

4.15 Runnermodel [SDD-REQ-1300] [Yamunnawahthi a/p Somasundharam CB18042]	52
4.16 Admin_M [SDD-REQ-1400]	53
4.17 spController [SDD-REQ-1500][Safinatunnajah Binti Sahrawi CB18002]	54
4.18 spModel [SDD-REQ-1600] [Safinatunnajah Binti Sahrawi CB18002]	58
4.19 LoginController [SDD-REQ-1700] [CHOONG ZHEN YUAN CB18059]	64
4.20 RegisterController [SDD-REQ-1800] [CHOONG ZHEN YUAN CB18059]	65
4.21 model [SDD-REQ-1900] [CHOONG ZHEN YUAN CB18059]	66
5. REQUIREMENT TRACEABILITY	70
5.1 Manage Food module [SRS-REQ-100] [Kam Jian Wei CB18137]	70
5.2 Manage Good Module [SRS-REQ-200] Yamunnawahthi A/P Somasundharam [CB18042]	71
5.3 Manage Medical module [SRS-REQ-300] [Safinatunnajah Binti Sahrawi CB18002]	72
5.4 Payment module [SRS-REQ-400] [Choong Zhen Yuan CB18059]	73

1. INTRODUCTION

1.1 Purpose

The Software Design Description (SDD) document describes the design of the WeServe Management System, to include system-wide design results, architectural designs and method to implement the one stop center runner services capabilities specified in the Software Requirements Specification (SRS) document.

1.2 System Identification

This Software Design Document (SDD) belongs to the “WeServe Management System” (WSMS).

System title	: WeServe Management System
System abbreviation	: WSMS
System identification number	: WSMS-2020-V1

1.3 System Overview

The system to be developed shall meet the requirement to handle various kind of service provided as one stop center for runner services. It includes the modules of:

- Manage Food Delivery
- Manage Good Delivery
- Manage Medical Delivery
- Manage Payment

The Manage Food Delivery will sell various kind of food products. Next, The manage good will sell many kind of goods such as beauties, fashion and tools. The Medical delivery will provide medical products.

This web-based systems are required to reach the quality to support concurrent users, robust database, fast response time, easy backup and information recovery. There are three main users of the system which is Service Provider, who sell various kind of services, Runner who is responsible for getting the service from the service provider to the customer and the Customer, who will buy the service. This system will be able to facilitate the process of selling, buying and delivering services for these users.

From Service provider point of view, they will need to register an account as a supplier where they will be able to choose the type of services they will be selling, such as food delivery, good delivery and medical delivery. The service provider must provide the details of their service as well as the service pickup location. Service Provider also can track the orders information and status of delivery by using this system.

From customer point of view, they can preview the services that are available without log in or register. The user must register or login to their account to be able to purchase the service. The system shall segregate the type of services that are available and the service provider of that service. That way, the customer can know where and who, is offering the service. For the good delivery, Customer can add the item to cart to purchase more items. Customers also can make payment in this system by using debit or credit card. Other than that, customer also can track the status of delivery and order information. The system should also support multiple concurrent user, so that customers can use the system at the same time.

From runner point of view, they can preview the services that are available but need to register or login as a runner to accept the delivery job. Runners can choose to be which service's runner and they can also choose more than one services.

1.4 Referenced Document

This section lists all the reference used in this Software Design Document.	
1. Software Requirement Specification of WeServe Management System	SRS_WESERVE_V3.pdf
2. IEEE Standard For Requirement Specification	IEEE Std 829-1998
3. Request For Proposal	BCS2233-SEM1-2019/2020

2. DATA DESIGN

2.1 Entity Relationship Diagram (ERD)

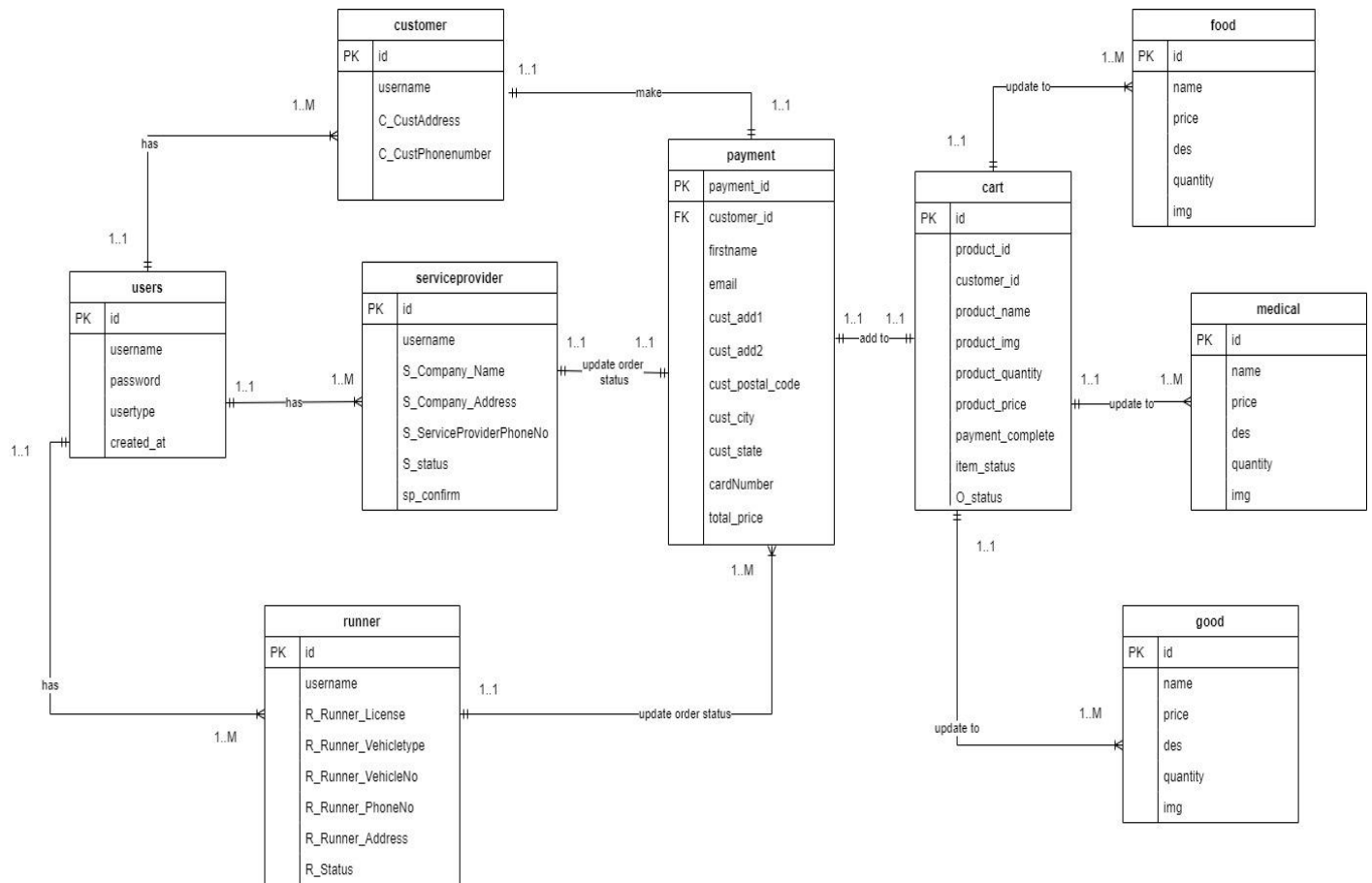


Figure 2.1 Entity Relationship Diagram for WeServe Management System

2.2 Data Dictionary

2.2.1 food table

Field Name	Description	Data Type	Constraint
id	Food id	Int(11)	PK
name	Food name	Varchar(200)	-
price	Price of Food	Decimal(7,2)	-
des	Food description	text	-
quantity	Food quantity	Int(11)	-
img	Food image	text	-

2.2.2 good table

Field Name	Description	Data Type	Constraint
id	good id	Int(11)	PK
name	good name	Varchar(200)	-
price	price of good	Decimal(7,2)	-
des	good description	text	-
quantity	good quantity	Int(11)	-
img	good image	Varchar(255)	-

2.2.3 medical table

Field Name	Description	Data Type	Constraint
id	medical id	Int(11)	PK
name	medical name	Varchar(200)	-
price	medical of good	Decimal(7,2)	-
des	medical description	text	-
quantity	medical quantity	Int(11)	-
img	medical image	Varchar(255)	-

2.2.4 payment table

Field Name	Description	Data Type	Constraint
payment_id	Id for payment	Int(11)	PK
customer_id	Id for customer	text	FK
firstname	Customer name for payment	text	-
email	Customer email for payment	text	-
cust_add1	Customer address	text	-
cust_add2	Customer address	text	-
cust_postal_code	Customer postcode	text	-
cust_city	Customer city	text	-
cust_state	Customer state	text	-
cardNumber	Customer card number	text	-
total_price	Total price of the purchased	double	-

2.2.5 Customer table

Field Name	Description	Data Type	Constraint
id	ID of customer	int(2)	PK
username	Username of Customer	varchar(50)	-
C_CustAddress	Address of Customer	varchar(100)	-
C_CustPhoneNumber	Phone number of Customer	varchar(30)	-

2.2.6 Runner table

Field Name	Description	Data Type	Constraint
id	Id for runner	Int(2)	PK
username	Username for runner	varchar(50)	-
R_Runner_License	License of Runner	varchar(50)	-
R_Runner_VehicleType	Vehicle type of Runner	varchar(50)	-
R_Runner_VehicleNo	Vehicle Number of the Runner	varchar(50)	-
R_Runner_PhoneNo	Phone Number of the Runner	varchar(50)	-
R_Runner_Address	Address of the runner	varchar(50)	-

R_Status	Runner approval status	Int(11)	-
----------	------------------------	---------	---

2.2.7 cart table

Field Name	Description	Data Type	Constraint
id	ID for cart	Int(3)	PK
product_id	Id for product	Int(3)	FK
customer_id	Id for customer	Int(3)	FK
product_name	Name of product	text	-
product_img	Image of product	text	-
product_quantity	Quantity of product	Int(3)	-
product_price	Price of product	double	-
payment_complete	Payment status	tinyint(4)	-
item_status	Pickup order status	Int(11)	-
O_status	Status update of order	varchar(12)	-

2.2.8 serviceprovider table

Field Name	Description	Data Type	Constraint
id	ID for service provider	Int(2)	PK
username	Name of Service Provider	varchar(50)	-
S_Company_Name	Company name of the Service Provider	varchar(100)	-
S_CompanyAddress	Company Address of the Service Provider	varchar(100)	-
S_ServiceProviderPhoneNo	Phone number for Service provider	varchar(50)	-
S_status	Service Provider approval status	Int(11)	-

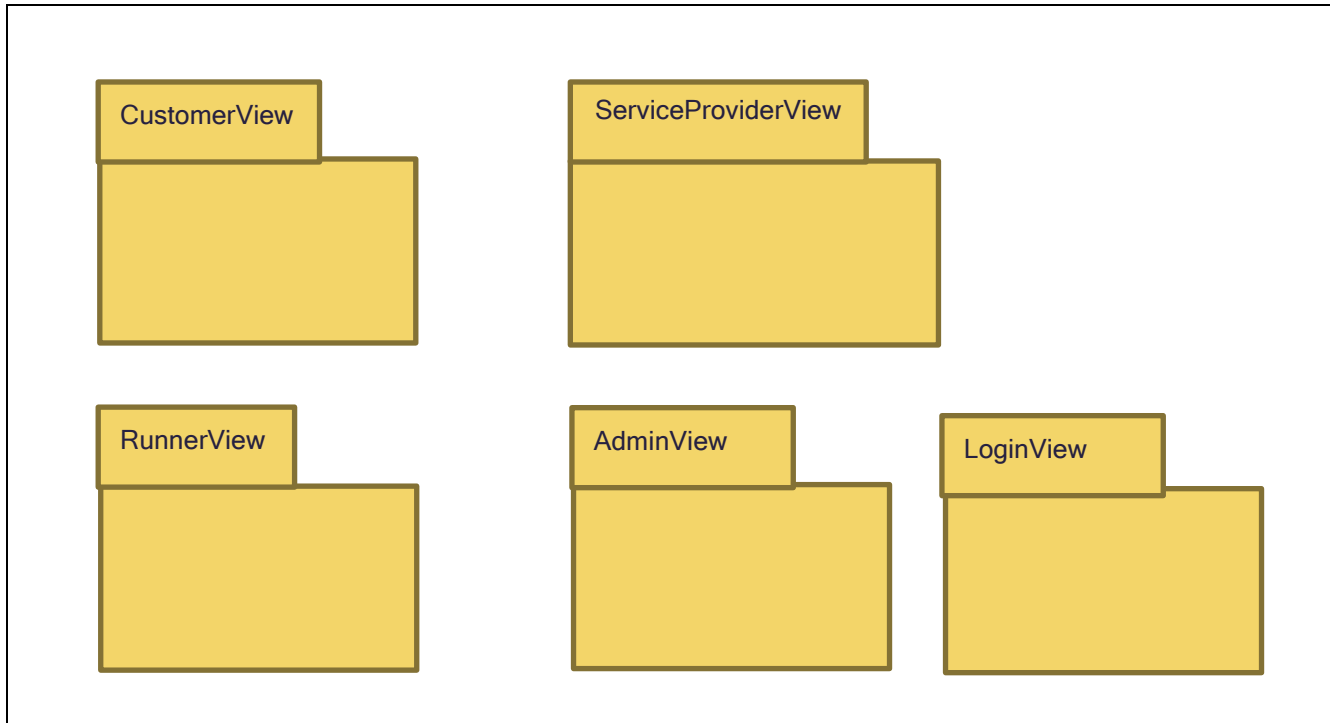
2.2.9 users table

Field Name	Description	Data Type	Constraint
id	Id for users	Int(11)	PK
username	Username for users	varchar(50)	-
password	Password for users	varchar(255)	-
usertype	Usertype of users	varchar(50)	-
created_at	Date and time of created profile	datetime	-

3. GENERAL ARCHITECTURE

3.1 Layered Architecture

3.1.1 Application Layer



3.1.1.1 CustomerView [Kam Jian Wei CB18137]

Folder Name	Description
custAddCart	Allow customer to view the product
custMakePayment	Allow customer to checkout their cart and pay
custOrder	Allow customer to view cart and payment
custUpdateInfo	Allow customer to view and edit the customer information

3.1.1.2 custAddCart [SDD-REQ-100] [Kam Jian Wei CB18137]

Class Name	Description
food	Allow customer to view the food product
good	Allow customer to view the good product
medical	Allow customer to view the medical product
productfood	Allow customer to view the details food product add into cart
productgood	Allow customer to view the details good product add into cart
productmedical	Allow customer to view the details medical product add into cart

3.1.1.3 custMakePayment [SDD-REQ-200] [Choong Zhen Yuan CB18059]

Class Name	Description
custAddress	Allow customer to view the food product
CustPayment	Allow customer to view the good product
custPaymentComplete	Allow customer to view the medical product

3.1.1.4 custAddCart [SDD-REQ-300] [Kam Jian Wei CB18137]

Class Name	Description
cart	Allow customer to view the cart

3.1.1.5 custUpdateinfo[SDD-REQ-400] [Kam Jian Wei CB18137]

Class Name	Description
customerupdate	Allow customer to update customer information
customerview	Allow customer to view customer information

3.1.1.6 RunnerView [SDD-REQ-500] []

Class Name	Description
runnerViewInfo	Runner register, login and update personal information.
runnerDeliverInfo	Runner accept the job and update order status

3.1.1.7 ServiceProviderView [Safinatunnajah Binti Sahrawi CB18002]

Folder Name	Description
spInfo	Interfaces for service provider to view and update information.
spProduct	Interfaces for add, update and delete food, good and medical products
spConfirmOrder	Interface for service provider to confirm order by customer

3.1.1.8 spInfo [SDD-REQ-600] [Safinatunnajah Binti Sahrawi CB18002]

Class Name	Description
spViewInfo	Allow SP to view the account information.
spUpdateInfo	Allow SP to edit and update the information in database.

3.1.1.9 spProduct [SDD-REQ-700]

Class Name	Description
spView	Homepage for Service Provider (SP).
addProduct	Allow SP to select category of product to add.
addFood	Allow SP to add food product into food table.
addGood	Allow SP to add good product into good table.
addMedical	Allow SP to add medical product into medical table.
editFood	Allow SP to update food product information.
editGood	Allow SP to update good product information.
editMedical	Allow SP to update medical product information.
spViewFood	Allow SP to view all food product information
spViewGoods	Allow SP to view all good product information
spViewMedical	Allow SP to view all medical product information
viewFood	Allow SP to view food product details information
viewGoods	Allow SP to view good product details information
viewMedical	Allow SP to view medical product details information

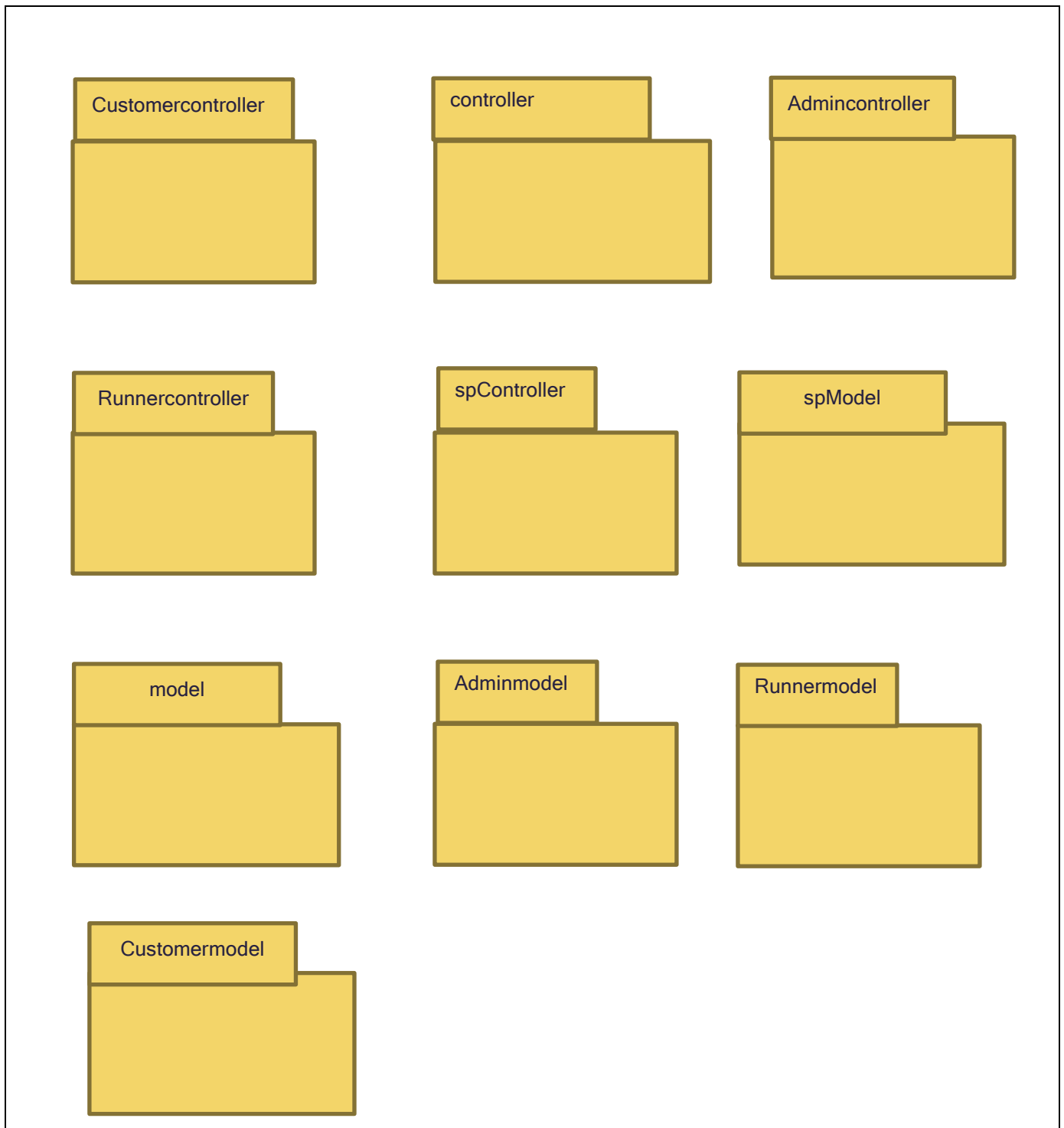
3.1.1.10 spConfirmOrder [SDD-REQ-800] [Safinatunnajah Binti Sahrawi CB18002]

Class Name	Description
sp_confirm	Allow service provider to accept order from customer.

3.1.11 UserView [SDD-REQ-900] [Choong Zhen Yuan CB18059]

Class Name	Description
Cregister	Customer registers into system
Rregister	Runner registers into system
SPregister	Service provider registers into system
login	Login for customer, runner and service provider using username and password

3.1.2 Business Services Layer



3.1.2.1 Customercontroller[SDD-REQ-1000] [Kam Jian Wei CB18137]

Class Name	Description
controller	Manage customer, food,good,medical,payment and cart related function.

3.1.2.2 Customermodel [SDD-REQ-1100] [Kam Jian Wei CB18137]

Class Name	Description
model	Connect into the database to receive data requires for function.

3.1.2.3 Runnercontroller [SDD-REQ-1200]

Class Name	Description
Runnercontroller	Manage accept job, update status of order and process runner personal information.

3.1.2.4 Runnermodel [SDD-REQ-1300]

Class Name	Description
Runnermodel	Runner's personal information Runner approve job information

3.1.2.5 Admin_M [SDD-REQ-1400]

Class Name	Description
Admin_M	Admin's login information

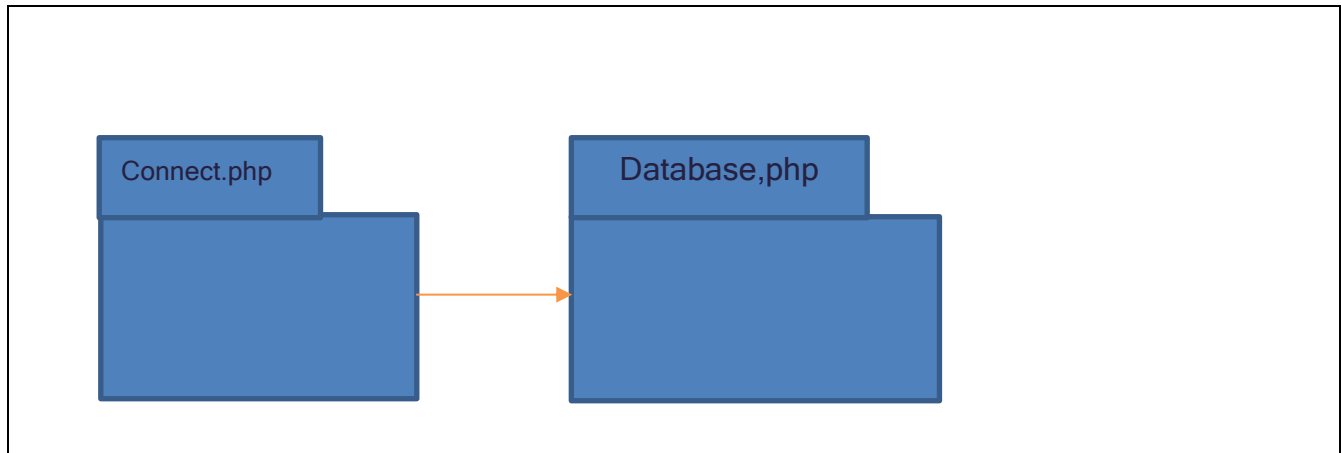
3.1.2.6 spController [SDD-REQ-1500]

Class Name	Description
spController	Manage service provider controller functions and product add, update and delete

3.1.2.7 spModel [SDD-REQ-1600]

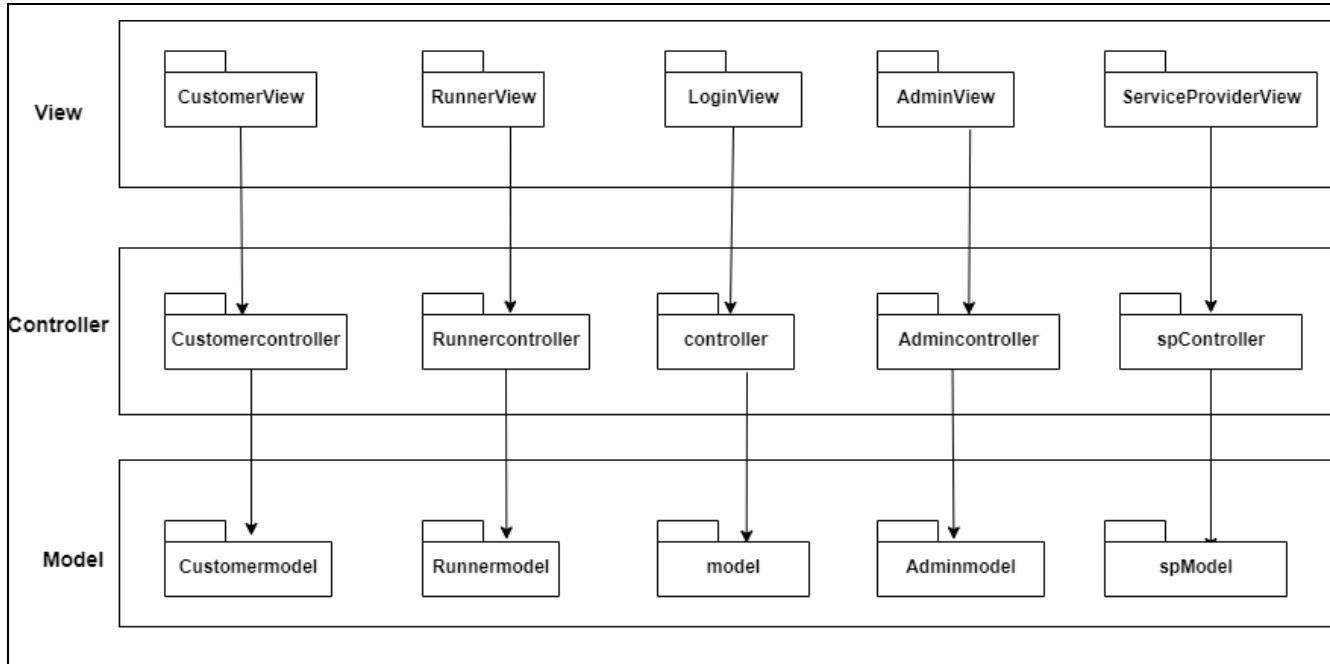
Class Name	Description
spModel	Manage model function and SQL command and connect to database

3.1.3 Middleware Layer



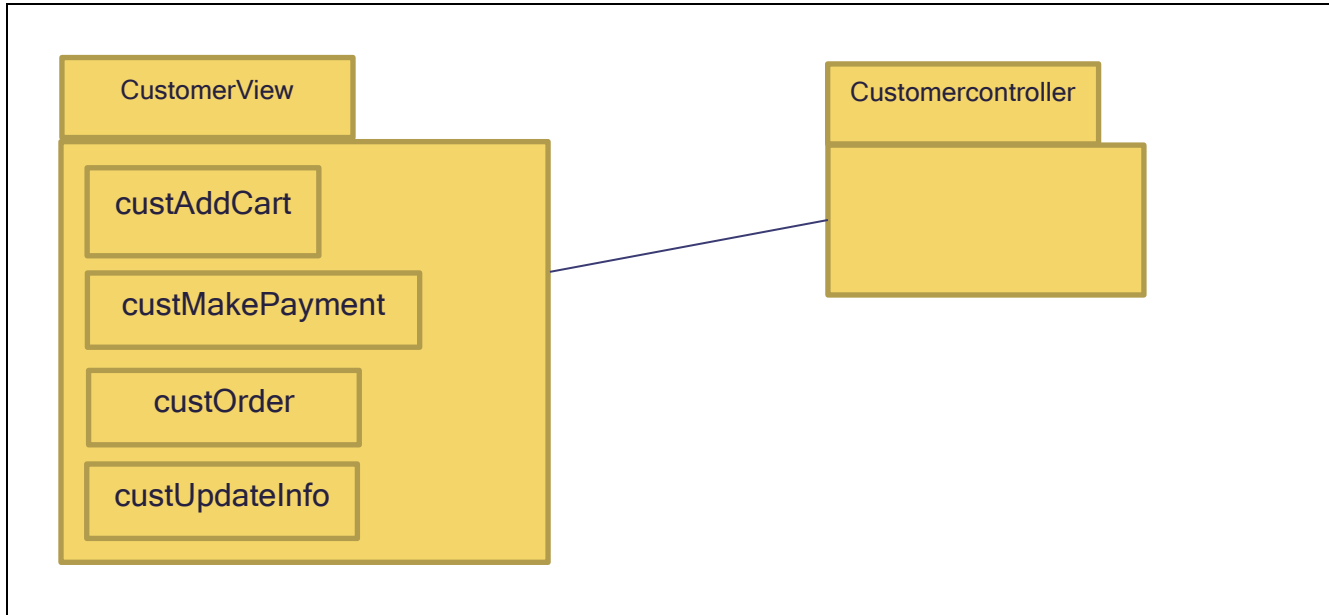
Package Name	Description
config.php	Build database connection between the server.
Database.php	Database information to connect to localhost.

3.2 MVC Package Relationship

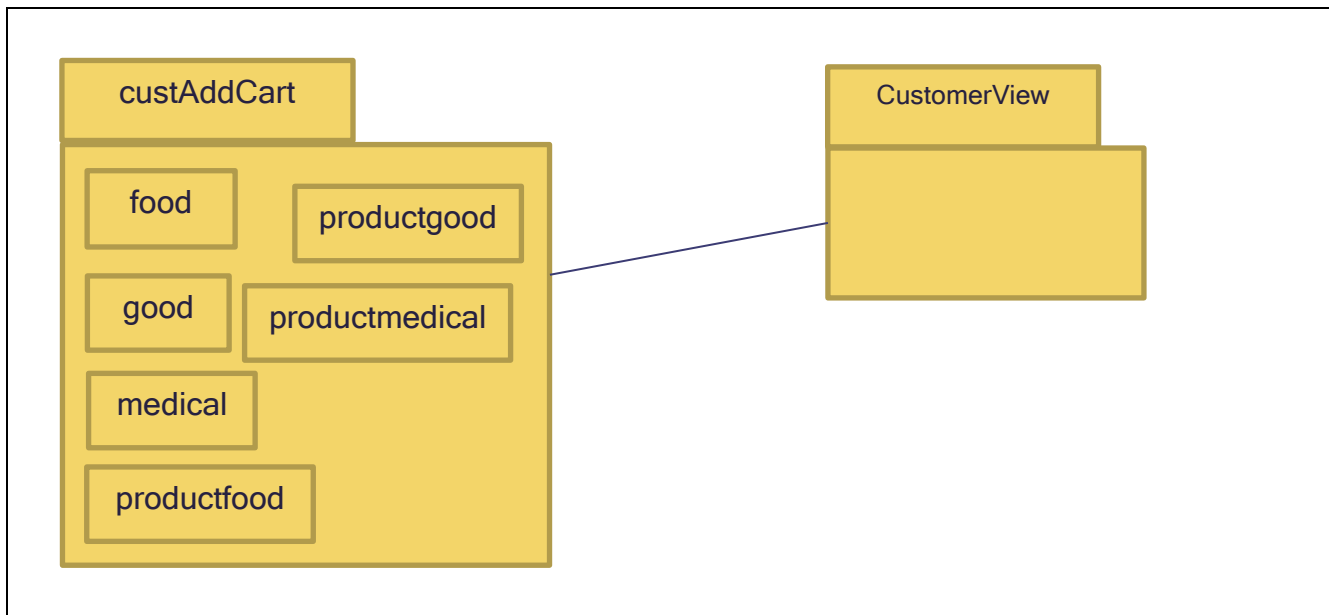


4. DETAIL DESIGN

4.1 CustomerView [Kam Jian Wei CB18137]



4.2 custAddCart [SDD-REQ-100][Kam Jian Wei CB18137]



4.2.1 food [SDD-REQ-101]

Class Type	Boundary class	
Responsibility	An interface to display the customer view the food product page	
Attributes	Attributes Name	Attributes Type

	id name price des quantity img	int(11) varchar(200) decimal(7,2) Int(11) text text
Methods	Method Name	Description
	viewAllFood()	Allow customer to view all the food product
Algorithm	Begin Display the name Diaplay the price Display the des Display img If “img” is clicked Display the specific details food product End	

4.2.2 good [SDD-REQ-102]

Class Type	Boundary class	
Responsibility	An interface to display the customer view the good product page	
Attributes	Attributes Name	Attributes Type
	id name price des img	int(11) varchar(200) decimal(7,2) text text
Methods	Method Name	Description
	viewAllGood()	Allow customer to view all the good product
Algorithm	Begin Display the name Diaplay the price Display the des	

	Display img If “img” is clicked Display the specific details good product End
--	--

4.2.3 medical [SDD-REQ-103]

Class Type	Boundary class	
Responsibility	An interface to display the customer view the medical product page	
Attributes	Attributes Name	Attributes Type
	id	int(11)
	name	varchar(200)
	price	decimal(7,2)
	des	text
	img	text
Methods	Method Name	Description
	viewAllMedical()	Allow customer to view all the medical product
Algorithm	Begin Display the name Diaplay the price Display the des Display img If “img” is clicked Display the specific details good product End	

4.2.4 productfood [SDD-REQ-104]

Class Type	Boundary class	
Responsibility	An interface to display the customer view the details of the specific food product	
Attributes	Attributes Name	Attributes Type
	id	int(11)
	name	varchar(200)

	price des quantity img	decimal(7,2) text int(11) text
Methods	Method Name	Description
	viewCertainFoodDetail() add()	Allow customer to view details of the specific food product Allow customer to add the food product to their shopping cart
Algorithm	Begin Display the name Diaplay the price Display the des Display img If “add to cart” is clicked Add the item into the cart End	

4.2.5 productgood [SDD-REQ-105]

Class Type	Boundary class	
Responsibility	An interface to display the customer view the details of the specific good product	
Attributes	Attributes Name	Attributes Type
	id name price des quantity img	int(11) varchar(200) decimal(7,2) text int(11) text
Methods	Method Name	Description
	viewCertainGoodDetail()	Allow customer to view details of the specific good product

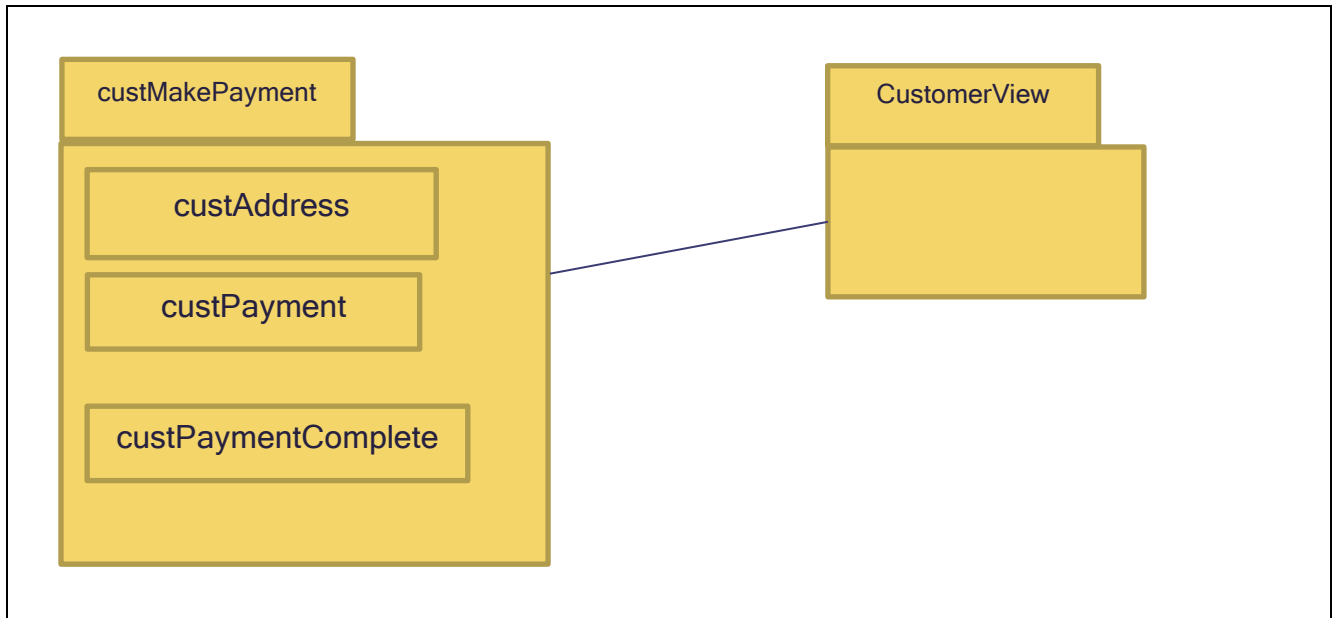
	add()	Allow customer to add the good product to their shopping cart
Algorithm	Begin Display the name Diaplay the price Display the des Display img If “add to cart” is clicked Add the item into the cart End	

4.2.6 productmedical [SDD-REQ-106]

Class Type	Boundary class	
Responsibility	An interface to display the customer view the details of the specific medical product	
Attributes	Attributes Name	Attributes Type
	id	int(11)
	name	varchar(200)
	price	decimal(7,2)
	des	text
	quantity	int(11)
	img	text
Methods	Method Name	Description
	viewCertainGoodDetail()	Allow customer to view details of the specific medical product
	add()	Allow customer to add the medical product to their shopping cart
Algorithm	Begin Display the name Diaplay the price Display the des Display img	

	If “add to cart” is clicked Add the item into the cart End
--	--

4.3 custMakePayment [SDD-REQ-200] [Choong Zhen Yuan CB18059]



4.3.1 custPayment[SDD-REQ-201]

Class Type	Boundary class	
Responsibility	An interface to allow customer to fill in customer payment information before proceed to payment gateway.	
Attributes	Attributes Name	Attributes Type
	payment_id	int(11)
	customer_id	text
	firstname	text
	email	text
	cust_add1	text
	cust_add2	text
	cust_postal_code	text
	cust_city	text
	cust_state	text

	cardNumber	text
	total_price	double
Methods	Method Name	Description
	viewAllCart()	Allow customer to view the product and subtotal from the cart.
	addCustomerPaymentInfo()	Allow customer to fill in customer payment information.
Algorithm	Begin Display cart Display total delivery fee Display subtotal Display input field for full name Display input field for Email Display input field for Address1 Display input field for Address2 Display input field for City Display input field for State Display input field for poskod Display input field for cardNumber Display input field for exp Display input field for cvv Display submit button Make Payment End	

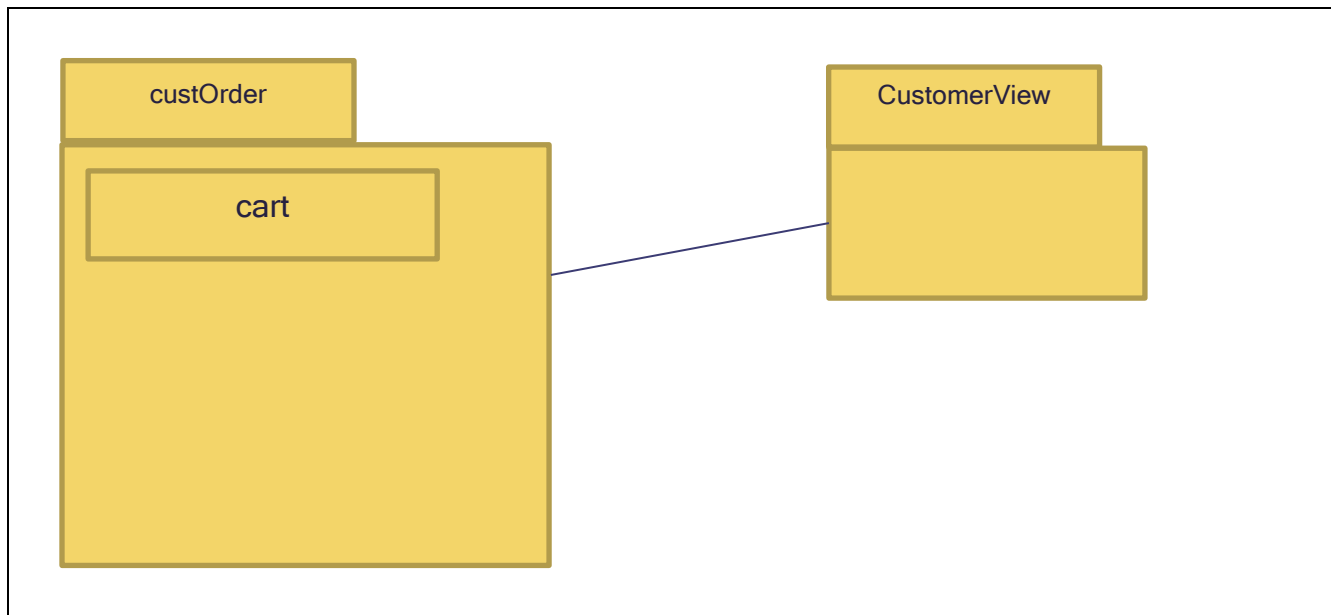
4.3.2 custAddress[SDD-REQ-202]

Class Type	Boundary class	
Responsibility	An interface to allow Paypal API to store address information and subtotal of the products from customer to enable Paypal API.	
Attributes	Attributes Name	Attributes Type
	cust_add1	text
	cust_add2	text
	cust_postal_code	text
	cust_city	text
	cust_state	text
	total_price	double
Methods	Method Name	Description
	viewAllCart() viewAddress()	Allow customer to view the subtotal from the cart. Retrieve the customer address information.
Algorithm	Begin Display cart and subtotal Generate Paypal API button Display the Paypal API interface Display the customer address Display the subtotal for payment Display the payment option END	

4.3.3 custPaymentComplete[SDD-REQ-203]

Class Type	Boundary class	
Responsibility	An interface to notify customer that the customer has successfully make payment..	
Attributes	Attributes Name	Attributes Type
Methods	Method Name	Description
	updatePaymentStatus()	Update the payment status after customer has success payment.
Algorithm	Begin Display “Your Payment has been confirmed” Display “ok” button END	

4.4 custOrder [SDD-REQ-300][Kam Jian Wei CB18137]

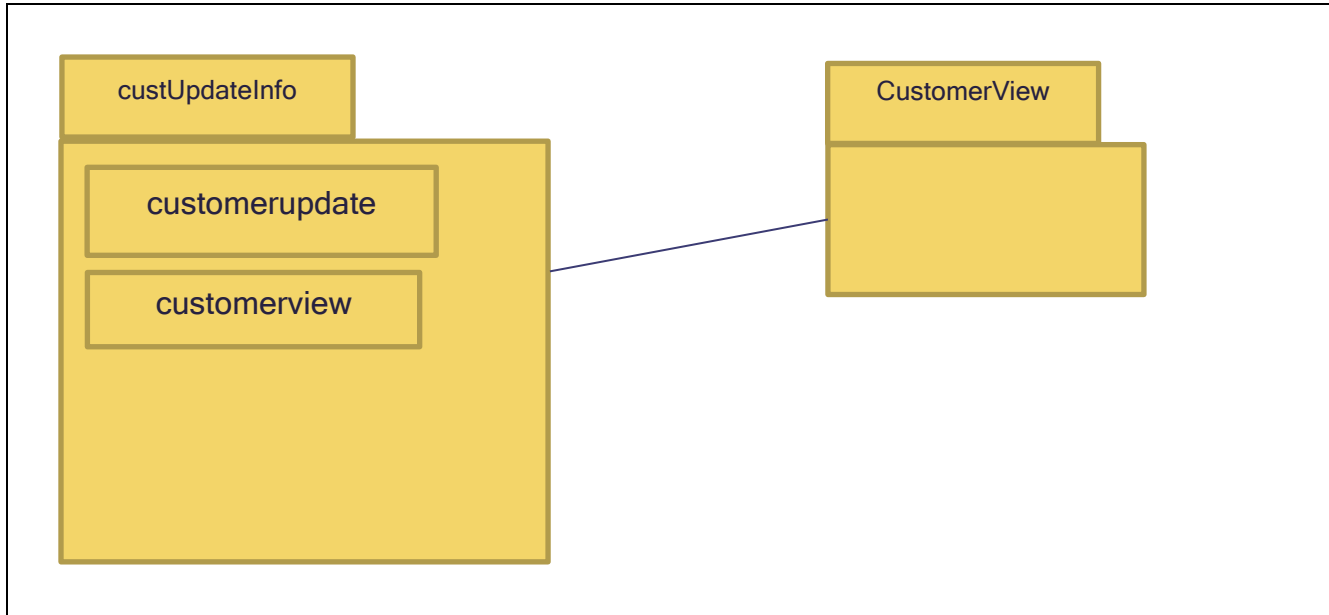


4.4.1 cart [SDD-REQ-301]

Responsibility	An interface to display the shopping cart for the customer	
Attributes	Attributes Name	Attributes Type
	id	int(3)

	product_id customer_id product_name product_img product_quantity product_price payment_complete sp_confirm item_status O_ststus	int(3) int(3) text text int(3) double tinyint(4) tinyint(4) int(11) varchar(12)
Methods	Method Name	Description
	viewAllcart() delete() Checkout()	Allow customer to view cart product Allow customer to delete the item that has been added into the cart Direct customer to the payment gateway page
Algorithm	Begin Display itemid Display itemname Display itemtype Display itemprice Display img Display quantity If “remove” is clicked Remove the product from the cart If “checkout” is clicked Save into database End	

4.5 custUpdateInfo [SDD-REQ-400][Kam Jian Wei CB18137]



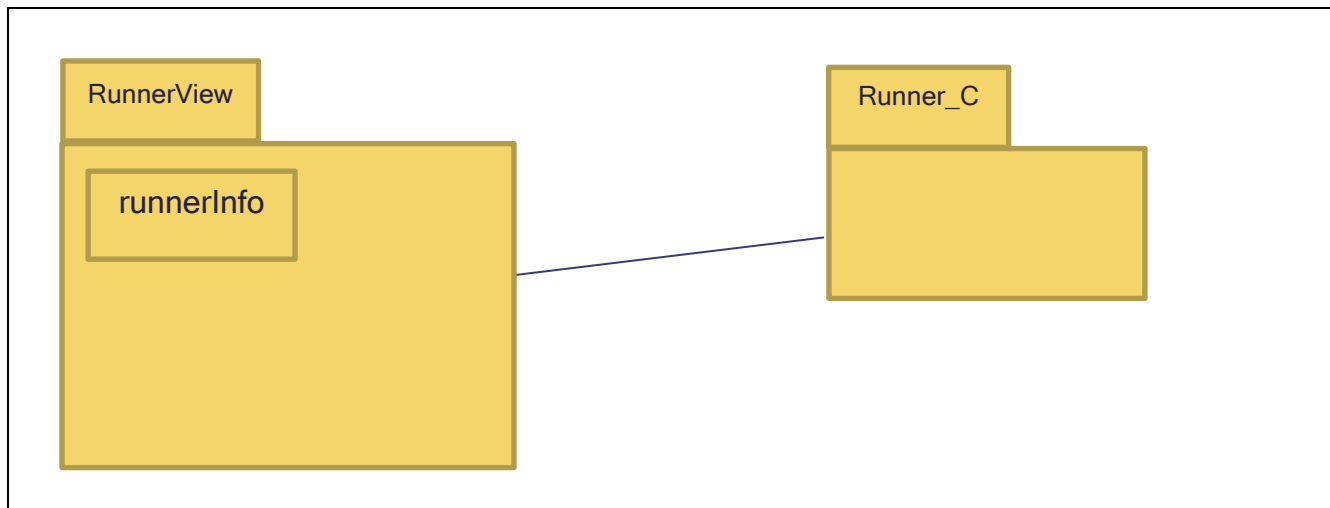
4.5.1 customerupdate [SDD-REQ-401]

Class Type	Boundary class	
Responsibility	An interface to allow customers edit their profile.	
Attributes	Attributes Name	Attributes Type
	Username	varchar(50)
	C_CustAddress	varchar(100)
	C_CustPhoneNumber	varchar(30)
Methods	Method Name	Description
Algorithm	Begin Display Username Display C_CustAddress Display C_CustPhoneNumber READ input IF “submit” is clicked Update into database End	

4.5.2 customerview [SDD-REQ-402]

Class Type	Boundary class	
Responsibility	An interface to display customer profile	
Attributes	Attributes Name	Attributes Type
	Username C_CustAddress C_CustPhoneNumber	varchar(50) varchar(100) varchar(30)
Methods	Method Name	Description
	update()	Allow customer to edit their profile
Algorithm	Begin Display Username Display C_CustAddress Display C_CustPhoneNumber IF “Edit Profile” is clicked Go to class “customerupdate ” End	

4.6 RunnerView [SDD-REQ-500] [Yamunnawahthi a/p Somasundharam CB18042]



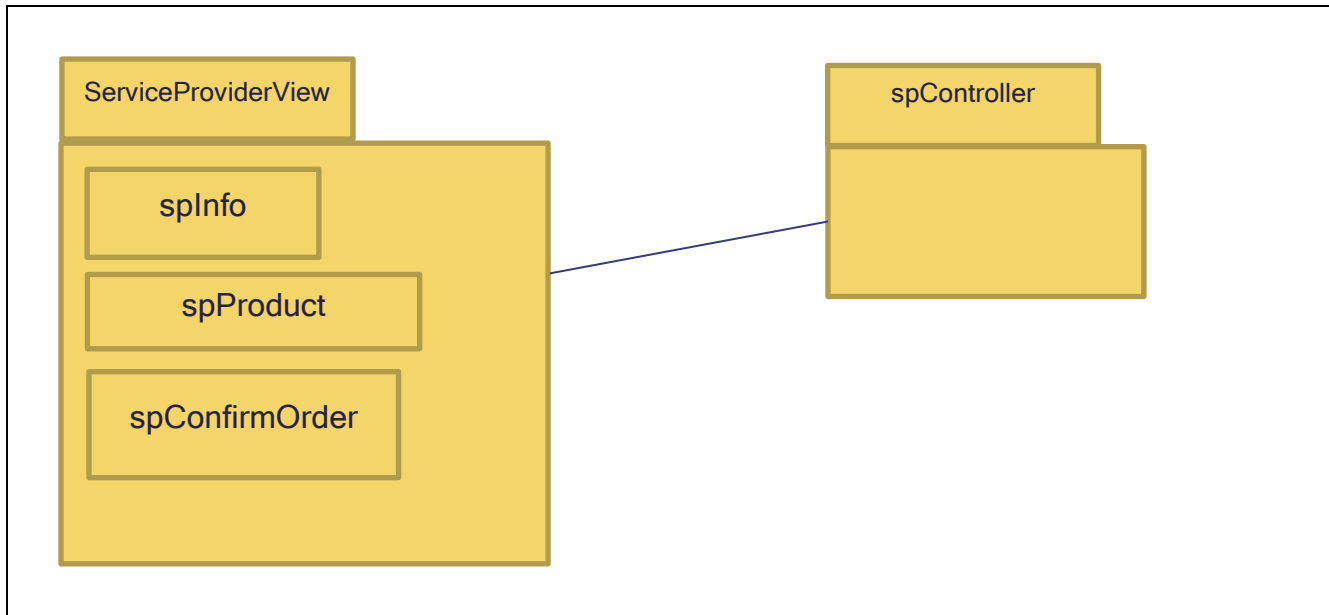
4.6.1 RunnerViewInfo [SDD-REQ-501]

Class Type	Boundary class	
Responsibility	An interface to display the runner's information	
Attributes	Attributes Name	Attributes Type
	Id	Int(2)
	Username	varchar(50)
	R_Runner_License	varchar(50)
	R_Runner_VehicleType	varchar(50)
	R_Runner_VehicleNo	varchar(50)
	R_Runner_PhoneNo	varchar(50)
	R_Runner_Address	varchar(50)
	R_Status	Int(11)
Methods	Method Name	Description
	Runnerviewprofile()	To View Runner's personal info
	RunnerUpdateProfile()	To Update Runner's personal info
Algorithm	Begin Go to RunnerHomepage Display choices Enter Runner Information Display Runner Information Enter Edit Information IF entered information correct THEN Go to Runnerviewprofile with edited information ELSE Page error message prompt END IF End	

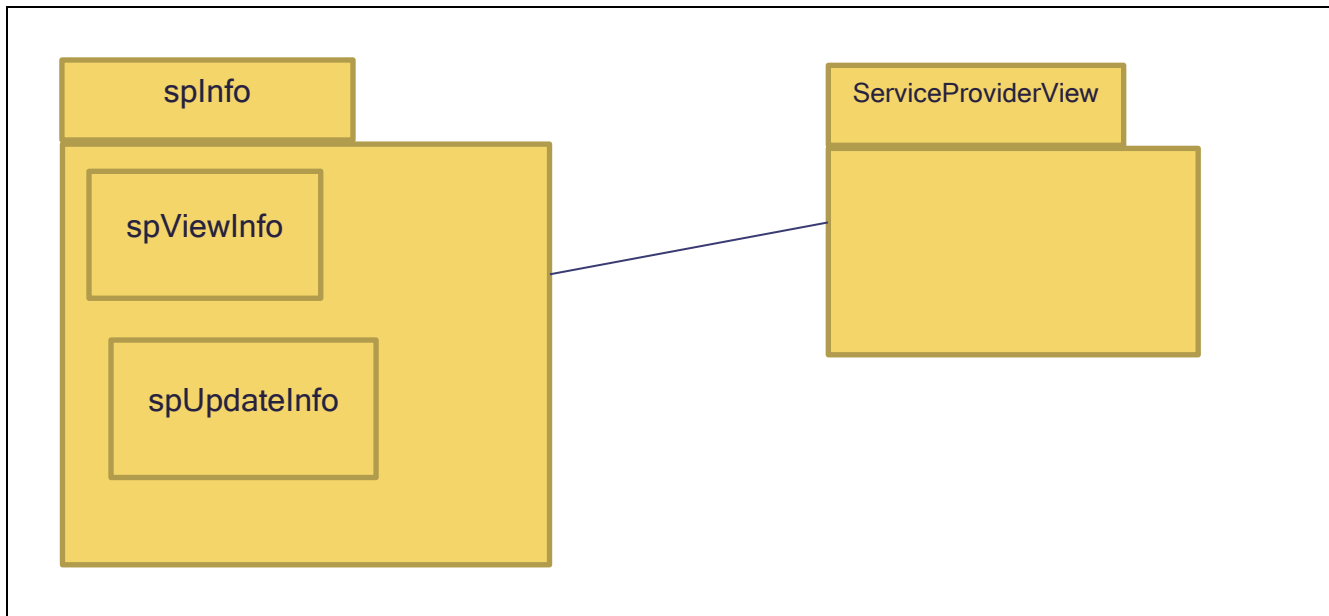
4.6.2 RunnerDeliveryInfo [SDD-REQ-502]

Class Type	Boundary class	
Responsibility	An interface to display runner delivery information.Runners can check their delivery details in this interface.	
Attributes	Attributes Name	Attributes Type
	customer_id	int(3)
	O_status	varchar(12)
	payment_id	int(11)
	Item_status	int(11)
Methods	Method Name	Description
	RunnerOrder()	To display the delivery details to runner
	RunnerDeliveryOrder()	To display the accepted job's order
Algorithm	BEGIN Enter RunnerOrder page View customer_id View firstname View cust_add2 View product_name View product_quantity IF option is AcceptJob THEN Go to RunnerDeliveredOrder page ELSE IF Stay at RunnerOrder page END	

4.7 ServiceProviderView [Safinatunnajah Binti Sahrawi CB18002]



4.8 splInfo [SDD-REQ-600][Safinatunnajah Binti Sahrawi CB18002]



4.8.1 spViewInfo [SDD-REQ-601]

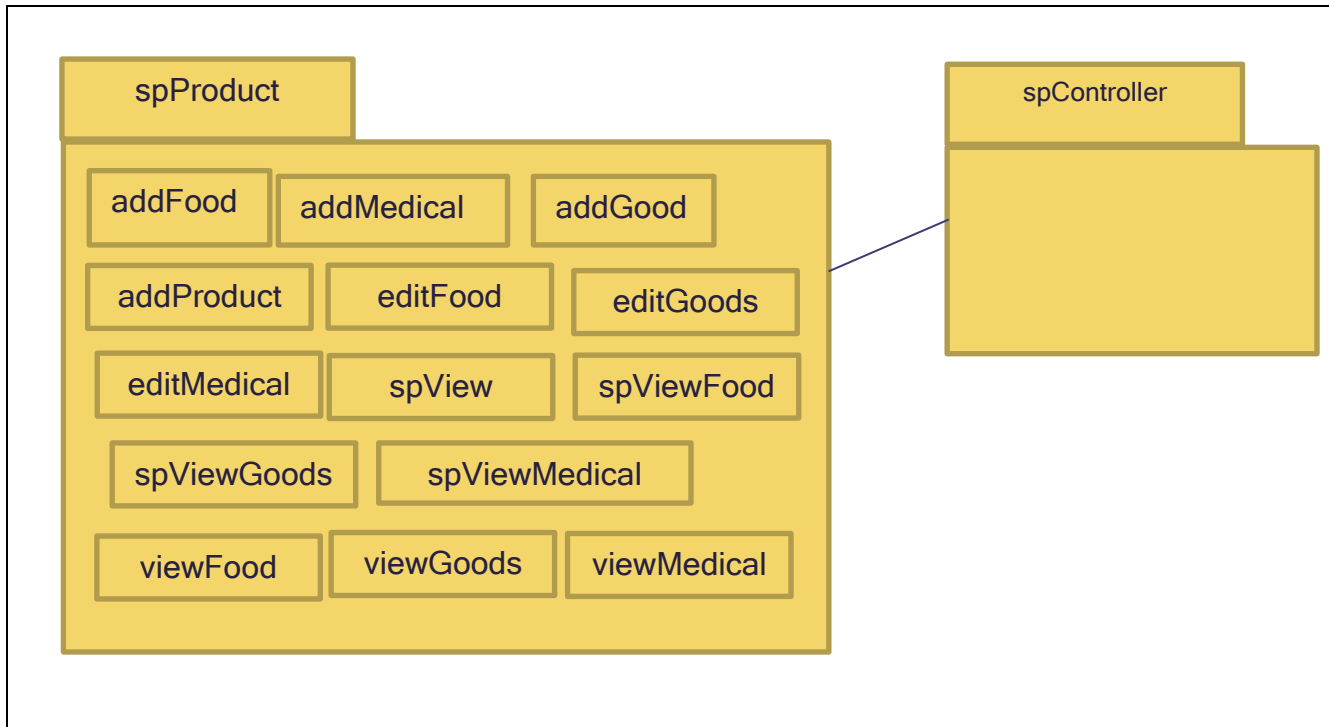
Class Type	Boundary class	
Responsibility	An interface to display the Service Provider account information.	
Attributes	Attributes Name	Attributes Type
	id	int(11)

	username S_CompanyName S_CompanyAddress S_ServiceProviderPhoneNo	varchar(11) varchar(12) varchar(50) varchar(30)
Methods	Method Name	Description
	viewInfo()	To get and call user information through model
Algorithm	Begin Interface call the Service Provider Information, call viewInfo(); End	

4.8.2 spUpdateInfo [SDD-REQ-602]

Class Type	Boundary class	
Responsibility	An interface to display the Service Provider form to update.	
Attributes	Attributes Name	Attributes Type
	id username S_CompanyName S_CompanyAddress S_ServiceProviderPhoneNo	int(11) varchar(11) varchar(12) varchar(50) varchar(30)
Methods	Method Name	Description
	updateInfo()	To get and change user information and update through model
Algorithm	Begin Interface call the Service Provider Information, call updateInfo(); End	

4.9 spProduct [SDD-REQ-700] [Safinatunnaja, Kam, Yamunna]



4.9.1 spView [SDD-REQ-701]

Class Type	Boundary class	
Responsibility	An interface to display the Service Provider homepage.	
Attributes	Attributes Name	Attributes Type
	id	int(11)
	username	varchar(11)
	S_CompanyName	varchar(12)
	S_CompanyAddress	varchar(50)
	S_ServiceProviderPhoneNo	varchar(30)
Methods	Method Name	Description
	viewAllFood()	To get food information and connect to model
	viewAllGood()	To get food information and connect to model
	viewAllMedical()	To get food information and connect to model
Algorithm	Begin Click Food, Good or Medical Category; Call viewAllFood(), viewAllGood() or viewAllMedical() Display product listing; End	

4.9.2 addProduct [SDD-REQ-702]

Class Type	Boundary class	
Responsibility	An interface to display the form for Service Provider to choose product category to add.	
Attributes	Attributes Name	Attributes Type
Methods	Method Name	Description
	addProduct()	To make sure Service Provider choose the correct table
Algorithm	Begin Choose product category; Go to addFood, addGood or addMedical; End	

4.9.3 addFood [SDD-REQ-703]

Class Type	Boundary class	
Responsibility	An interface to display the form for Service Provider to add food product into database.	
Attributes	Attributes Name	Attributes Type
	id name price des quantity img	Int(11) varchar(200) decimal(7,2) Int(11) int(3) text
Methods	Method Name	Description
	addFood()	To receive product information and pass to model
Algorithm	Begin Display form to insert data; Click <<ADD>> button; Connect to addFood();	

	End
--	-----

4.9.4 addGood [SDD-REQ-704]

Class Type	Boundary class	
Responsibility	An interface to display the form for Service Provider to add good product into database.	
Attributes	Attributes Name	Attributes Type
	id name price des quantity img	Int(11) varchar(200) decimal(7,2) Int(11) int(3) text
Methods	Method Name	Description
	addGood()	To receive product information and pass to model
Algorithm	Begin Display form to insert data; Click <<ADD>> button; Connect to addGood(); End	

4.9.5 addMedical [SDD-REQ-705]

Class Type	Boundary class	
Responsibility	An interface to display the form for Service Provider to add medical product into database.	
Attributes	Attributes Name	Attributes Type
	id name price des quantity img	Int(11) varchar(200) decimal(7,2) Int(11) int(3) text
Methods	Method Name	Description

	addMedical()	To receive product information and pass to model
Algorithm	Begin Display form to insert data; Click <<ADD>> button; Connect to addMedical(); End	

4.9.6 editFood [SDD-REQ-706]

Class Type	Boundary class	
Responsibility	An interface to display the food information and get information to update to database	
Attributes	Attributes Name	Attributes Type
	id name price des quantity img	Int(11) varchar(200) decimal(7,2) Int(11) int(3) text
Methods	Method Name	Description
	editFood()	To get product detail to update and connect to model
Algorithm	Begin Get product data and display form to edit data; Click <<UPDATE>> button; Connect to editFood(); End	

4.9.7 editGood [SDD-REQ-707]

Class Type	Boundary class	
Responsibility	An interface to display the goods information and get information to update to database	
Attributes	Attributes Name	Attributes Type

	Id name price Des quantity Img	Int(11) varchar(200) decimal(7,2) Int(11) int(3) text
Methods	Method Name	Description
	editGood()	To get product detail to update and connect to model
Algorithm	Begin Get product data and display form to edit data; Click <<UPDATE>> button; Connect to editGood(); End	

4.9.8 editMedical [SDD-REQ-708]

Class Type	Boundary class	
Responsibility	An interface to display the medical information and get information to update to database	
Attributes	Attributes Name	Attributes Type
	id name price des quantity img	Int(11) varchar(200) decimal(7,2) Int(11) int(3) text
Methods	Method Name	Description
	editMedical()	To get product detail to update and connect to model
Algorithm	Begin Get product data and display form to edit data; Click <<UPDATE>> button; Connect to editMedical (); End	

4.9.9 spViewFood [SDD-REQ-709]

Class Type	Boundary class	
Responsibility	An interface to display the food listings from food table.	
Attributes	Attributes Name	Attributes Type
	id name price des quantity img	Int(11) varchar(200) decimal(7,2) Int(11) int(3) text
Methods	Method Name	Description
	viewAllFood()	To get food information and connect to model
Algorithm	Begin Get product data and display; Connect to viewAllFood(); End	

4.9.10 spViewGoods [SDD-REQ-710]

Class Type	Boundary class	
Responsibility	An interface to display the goods listings from goods table.	
Attributes	Attributes Name	Attributes Type
	id name price des quantity img	Int(11) varchar(200) decimal(7,2) Int(11) int(3) text
Methods	Method Name	Description
	viewAllGood()	To get goods information and connect to model
Algorithm	Begin Get product data and display; Connect to viewAllFood(); End	

4.9.11 spViewMedical [SDD-REQ-711]

Class Type	Boundary class	
Responsibility	An interface to display the medical listings from medical table.	
Attributes	Attributes Name	Attributes Type
	id name price des quantity img	Int(11) varchar(200) decimal(7,2) Int(11) int(3) text
Methods	Method Name	Description
	viewAllMedical()	To get medical information and connect to model
Algorithm	Begin Get product data and display; Connect to viewAllMedical(); End	

4.9.12 viewFood [SDD-REQ-712]

Class Type	Boundary class	
Responsibility	An interface to display the food product details from food table.	
Attributes	Attributes Name	Attributes Type
	id name price des quantity img	Int(11) varchar(200) decimal(7,2) Int(11) int(3) text
Methods	Method Name	Description
	viewFood()	To get certain food information and connect to model
Algorithm	Begin Get product data and display; Connect to viewFood();	

	End
--	-----

4.9.13 viewGoods [SDD-REQ-713]

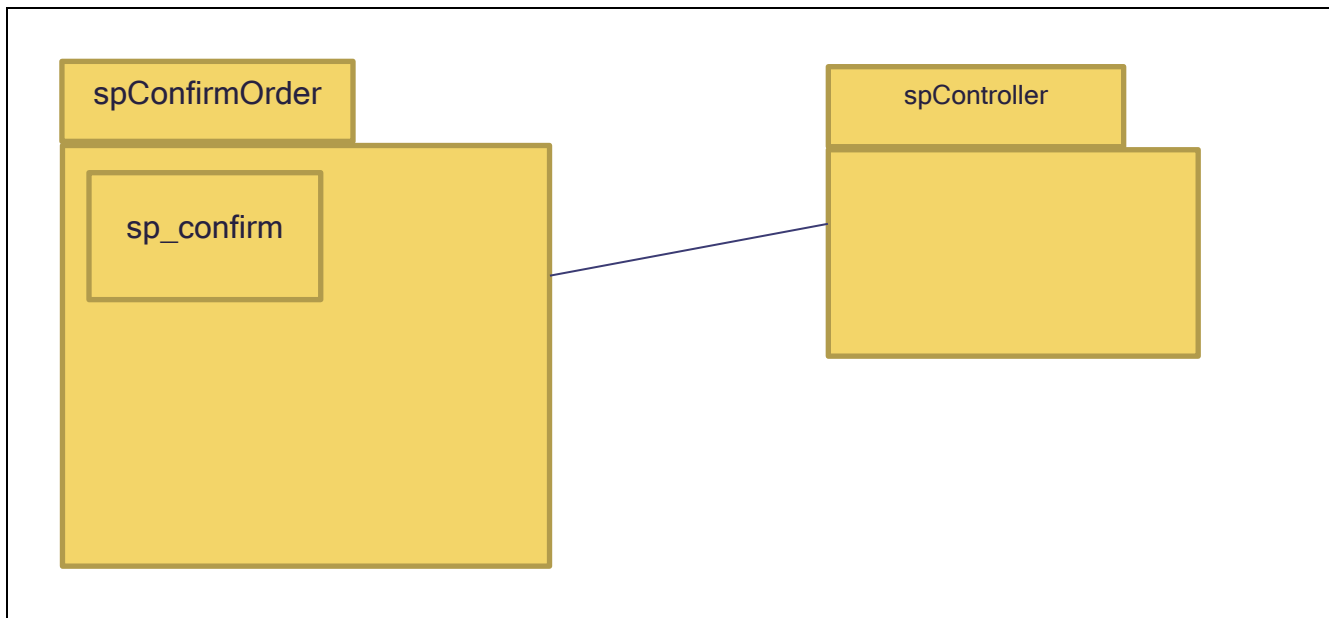
Class Type	Boundary class	
Responsibility	An interface to display the goods product details from goods table.	
Attributes	Attributes Name	Attributes Type
	id name price des quantity img	Int(11) varchar(200) decimal(7,2) Int(11) int(3) text
Methods	Method Name	Description
	viewGood()	To get certain goods information and connect to model
Algorithm	Begin Get product data and display; Connect to viewGoods(); End	

4.9.14 viewMedical [SDD-REQ-714]

Class Type	Boundary class	
Responsibility	An interface to display the medical product details from medical table.	
Attributes	Attributes Name	Attributes Type
	id name price des quantity img	Int(11) varchar(200) decimal(7,2) Int(11) int(3) text
Methods	Method Name	Description
	viewMedical()	To get certain medical information and connect to model
Algorithm	Begin	

	Get product data and display; Connect to viewMedical(); End
--	---

4.10 spConfirmOrder [SDD-REQ-800][Safinatunnajah Binti Sahrawi]

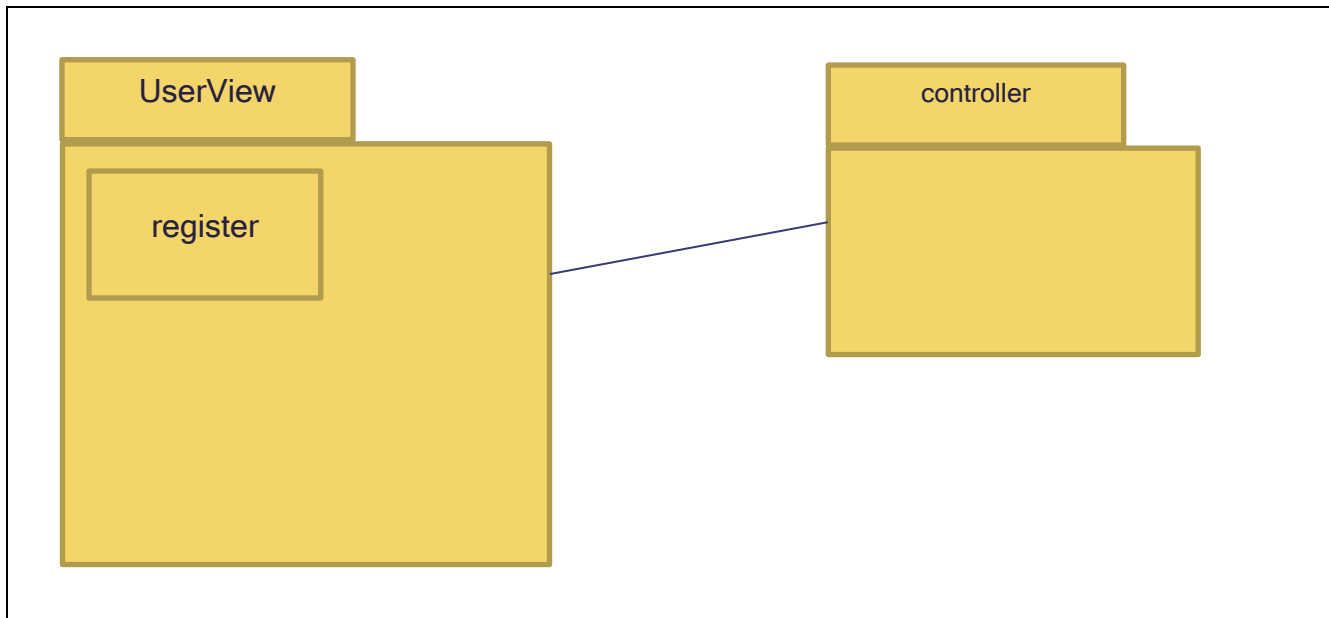


4.10.1 sp_confirm [SDD-REQ-801]

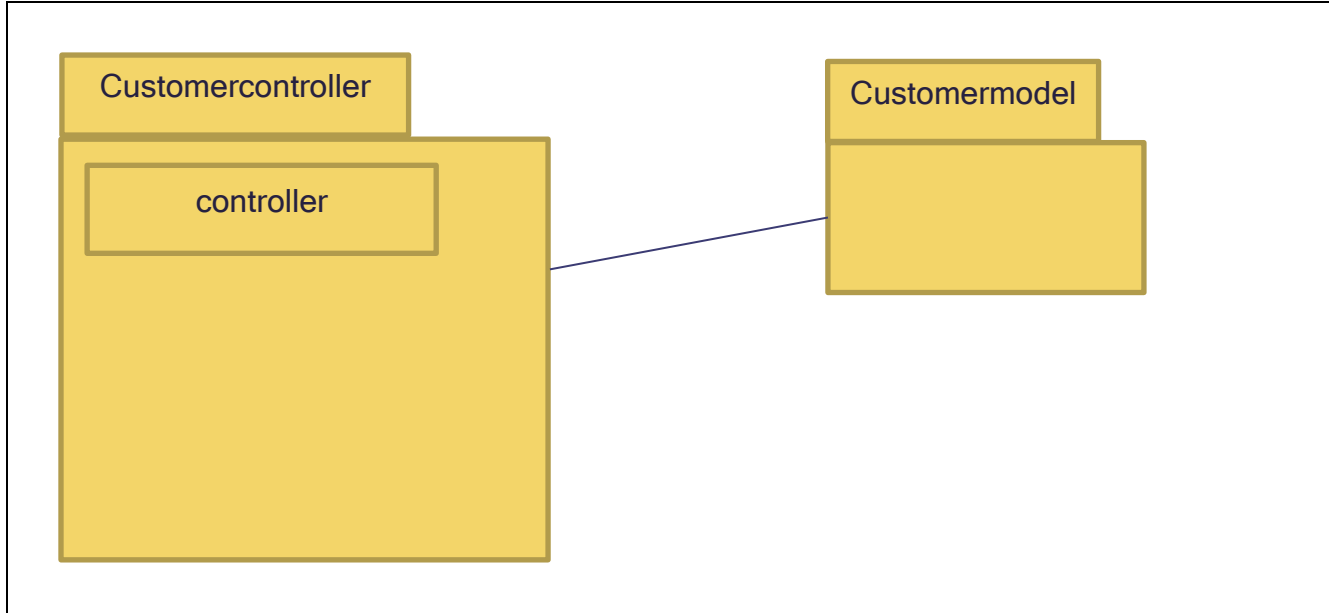
Class Type	Boundary class	
Responsibility	An interface to display the order information and accept the order.	
Attributes	Attributes Name	Attributes Type
	id name price des quantity img	Int(11) varchar(200) decimal(7,2) Int(11) int(3) text
Methods	Method Name	Description
	acceptOrder()	To change order status on the system and connect to model
Algorithm	Begin Get order data from cart and customer table;	

	Display table information and click <<Accept>> to accept; Connect to acceptOrder(); End
--	---

4.11 UserView [SDD-REQ-900][Choong Zhen Yuan]



4.12 Customercontroller[SDD-REQ-1000][Kam Jian Wei CB18137]



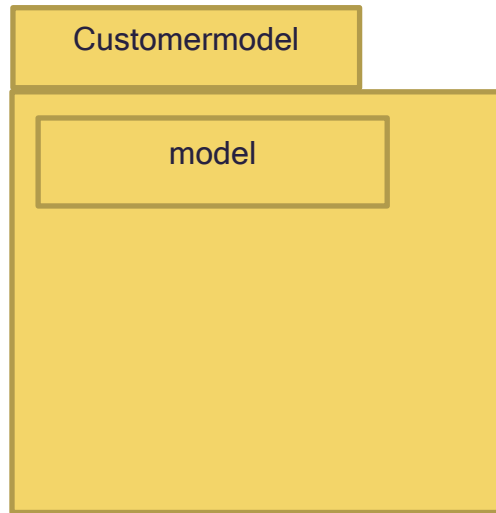
4.12.1 controller[SDD-REQ-1001] [Kam Jian Wei CB18137]

Class Type	Controller	
Responsibility	Manage all function of customer.	
Attributes	Attributes Name	Attributes Type
	id	int(11)
	name	varchar(200)
	price	decimal(7,2)
	des	Int(11)
	quantity	text
	img	text
	product_id	int(3)
	customer_id	int(3)
	product_name	text
	product_img	text
	product_quantity	int(3)
	product_price	double
	payment_complete	tinyint(4)

	sp_confirm item_status O_ststus Username C_CustAddress C_CustPhoneNumber	int(11) varchar(12) varchar(50) varchar(100) varchar(30)
Methods	Method Name	Description
	viewCustomerinfo() editCustomerinfo() viewAllFood() viewCertainFoodDetail() viewAllGood() viewCertainGoodDetail() viewAllMedical() viewCertainMedicalDetail() add() viewAllCart() delete() addCustomerPaymentInfo() viewAddress() updatePaymentStatus()	To allow customer view customer information. To allow customer edit information. For customer view the food product. For customer view the details food product. For customer view the good product. For customer view the details good product. For customer view the medical product. For customer view the details medical product. To allow customer add the product into the cart. To allow customer to view the cart product. To allow customer delete product from cart.
Algorithm	START Received requested function Send requested function to Customermodel Received Result	

	Send Result to requestor END
--	---------------------------------

4.13 Customermodel [SDD-REQ-1100][Kam Jian Wei CB18137]



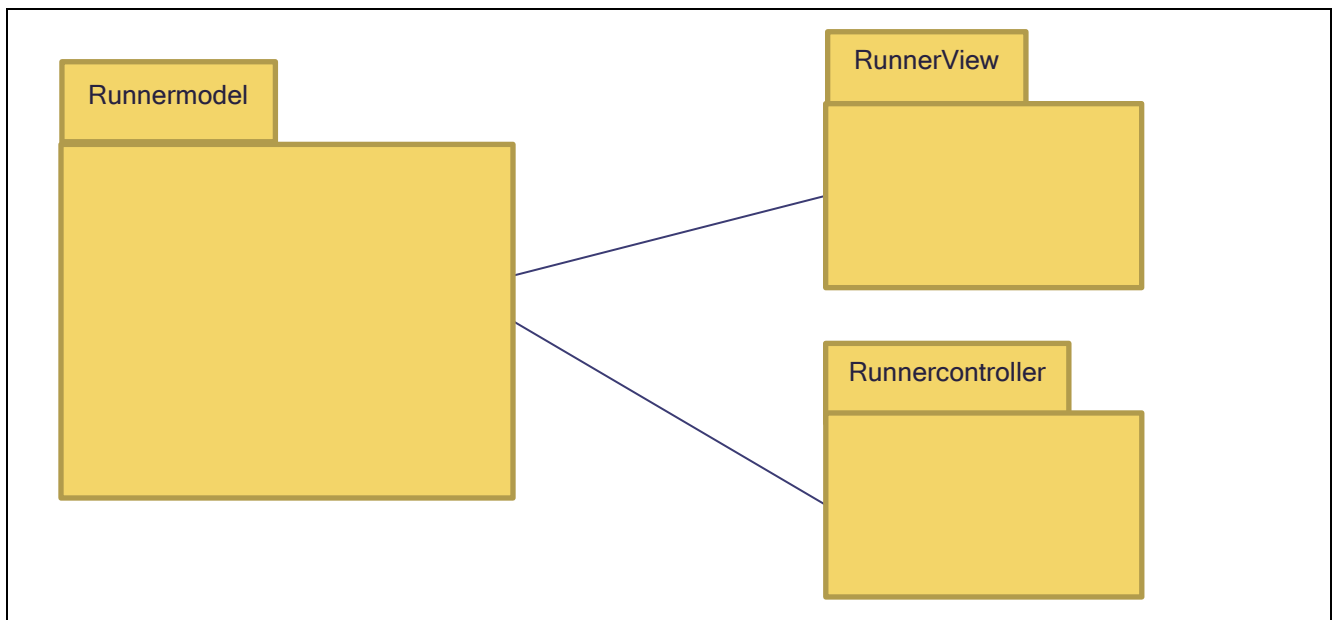
4.13.1 model [SDD-REQ-1101] [Kam Jian Wei CB18137]

Class Type	Controller	
Responsibility	Connect into the database to receive data requires for function.	
Attributes	Attributes Type	
	id	int(11)
	name	varchar(200)
	price	decimal(7,2)
	des	Int(11)
	quantity	text
	img	text
	product_id	int(3)
	customer_id	int(3)
	product_name	text
	product_img	text
	product_quantity	int(3)
	product_price	double

	payment_complete sp_confirm item_status O_ststus Username C_CustAddress C_CustPhoneNumber	tinyint(4) tinyint(4) int(11) varchar(12) varchar(50) varchar(100) varchar(30)
Methods	Method Name	Description
	Connect() updateCustomerinfo() viewCustomerDetail() viewFood() viewFoodDetail() viewGood() viewGoodDetail() viewMedical() viewMedicalDetail() InsertCart() ViewCart() DeleteCart() InsertCustomerPaymentInfo() viewPaymentAddress() updatePaymentStatus()	Connect to Database Update Customer details Fetch customer detail from database Fetch food product from database Fetch specific food product details from database Fetch good product from database Fetch specific good product details from database Fetch medical product from database Fetch specific medical product details from database Add new product into the cart Fetch the product in the cart Delete the profuct from the cart Add the customer payment information Fetch the customer information from database Update the payment status from database.
Algorithm	START Received requested function form Customercontroller	

	Connect to Database Run requested function Send Result END
--	---

4.14 Runnercontroller [SDD-REQ-1200] [Yamunnawahthi a/p Somasundharam CB18042]

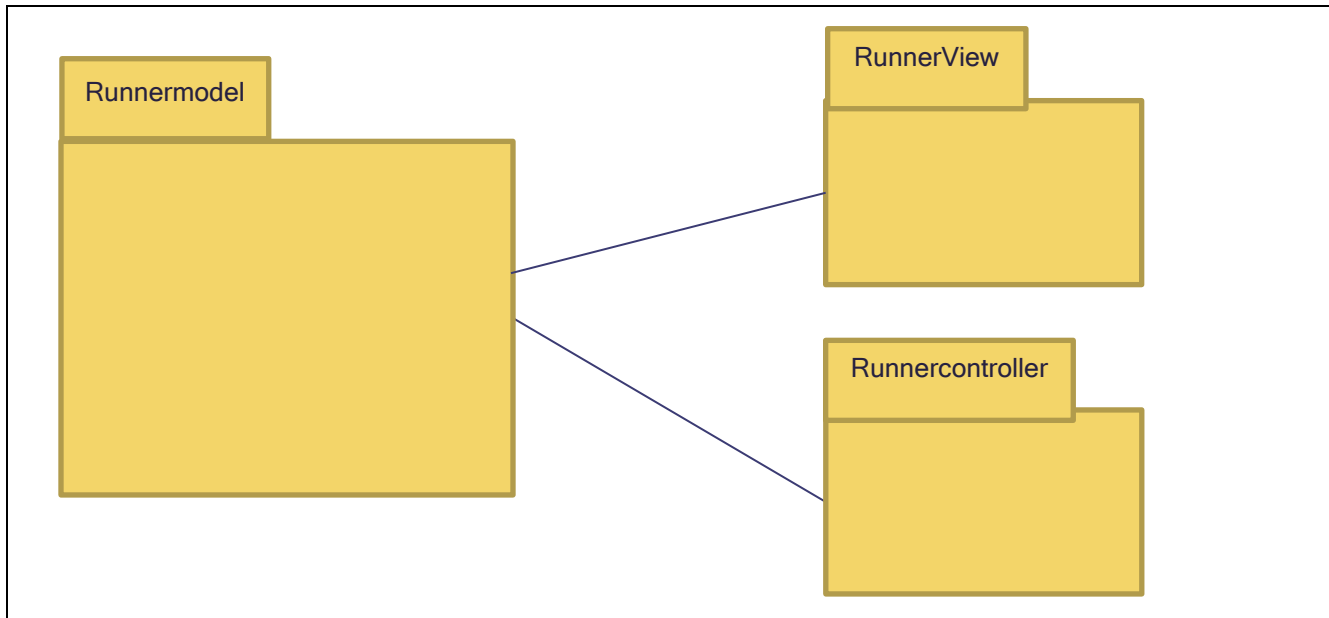


4.14.1 RunnerController [SDD-REQ-1201]

Class Type	Controller class	
Responsibility	To control the runner information and runner delivery information	
Attributes	Attributes Name	Attributes Type
	Id	Int(2)
	Username	varchar(50)
	R_Runner_License	varchar(50)
	R_Runner_Vehicletype	varchar(50)
	R_Runner_VehicleNo	varchar(50)
	R_Runner_PhoneNo	varchar(50)
	R_Runner_Address	varchar(50)
	R_Status	Int(11)

	customer_id O_status payment_id Item_status	int(3) varchar(12) int(11) int(11)
Methods	Method Name	Description
	viewRunnerInfo editRunnerinfo() viewAllDelivery() viewAcceptDelivery() CurrentDellivery() CompleteDelivery() addDelivery()	To send the Runner information. To send the edited Runner information To send all the delivery details to runner order interface To send the accepted job status information To send the currentdelivery details which runner took to update the status To send the delivery status where it is completed to the runner view. To accept the pressed button in the interface
Algorithm	BEGIN Read runner input IF input entered is TRUE Pass the TRUE value to model ELSE Pass the FALSE value to runnerView END	

4.15 Runnermodel [SDD-REQ-1300] [Yamunnawahthi a/p Somasundharam CB18042]

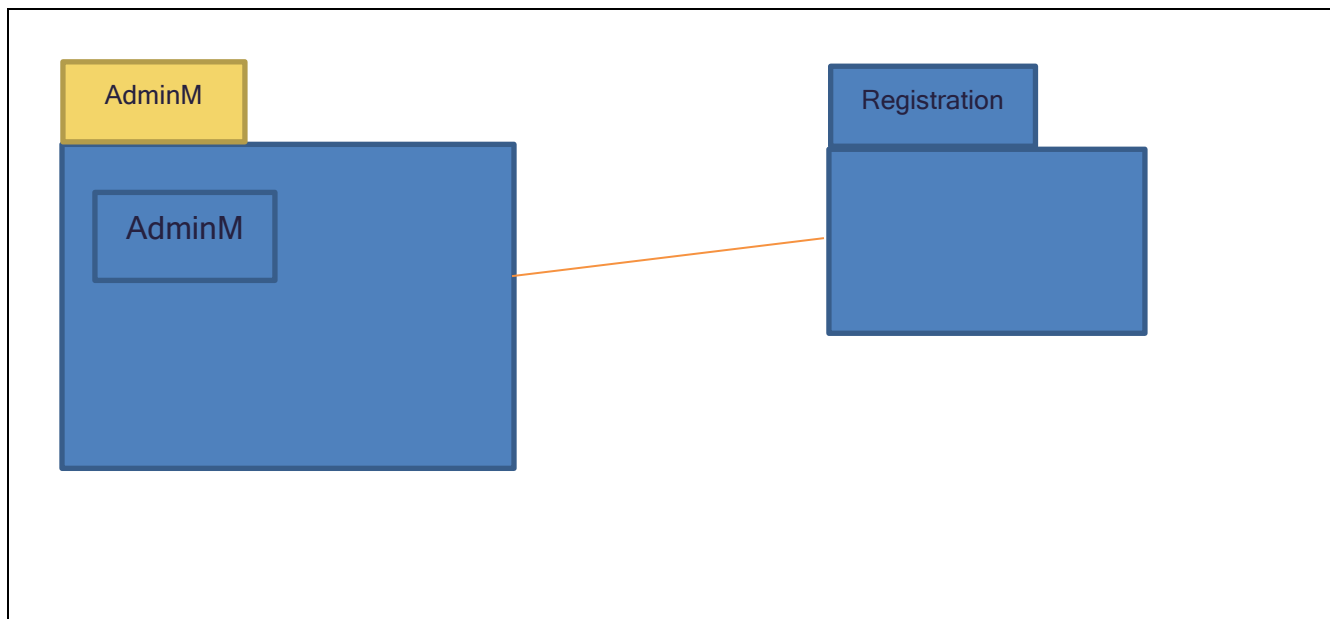


4.15.1 RunnerModel [SDD-REQ-1301]

Class Type	Model class	
Responsibility	Store runner information and runner delivery information.	
Attributes	Attributes Name	Attributes Type
	Id	Int(2)
	Username	varchar(50)
	R_Runner_License	varchar(50)
	R_Runner_Vehicletype	varchar(50)
	R_Runner_VehicleNo	varchar(50)
	R_Runner_PhoneNo	varchar(50)
	R_Runner_Address	varchar(50)
	R_Status	Int(11)
	customer_id	int(3)
	O_status	varchar(12)
	payment_id	int(11)
	Item_status	int(11)
Methods	Method Name	Description

	updateRunnerinfo() viewRunnerDetails() viewAllDelivery() AcceptDelivery() CompleteDelivery() CurrentDelivery() addDelivery()	To store the update edited runner info in database To store runner's personal details To store All the order delivery details To store accepted delivery status details To store completed delivery details To store accepted delivery details To store add button's information for accept job
Algorithm	Begin send the runner's data requested by Controller Store Runner's all information End	

4.16 Admin_M [SDD-REQ-1400]

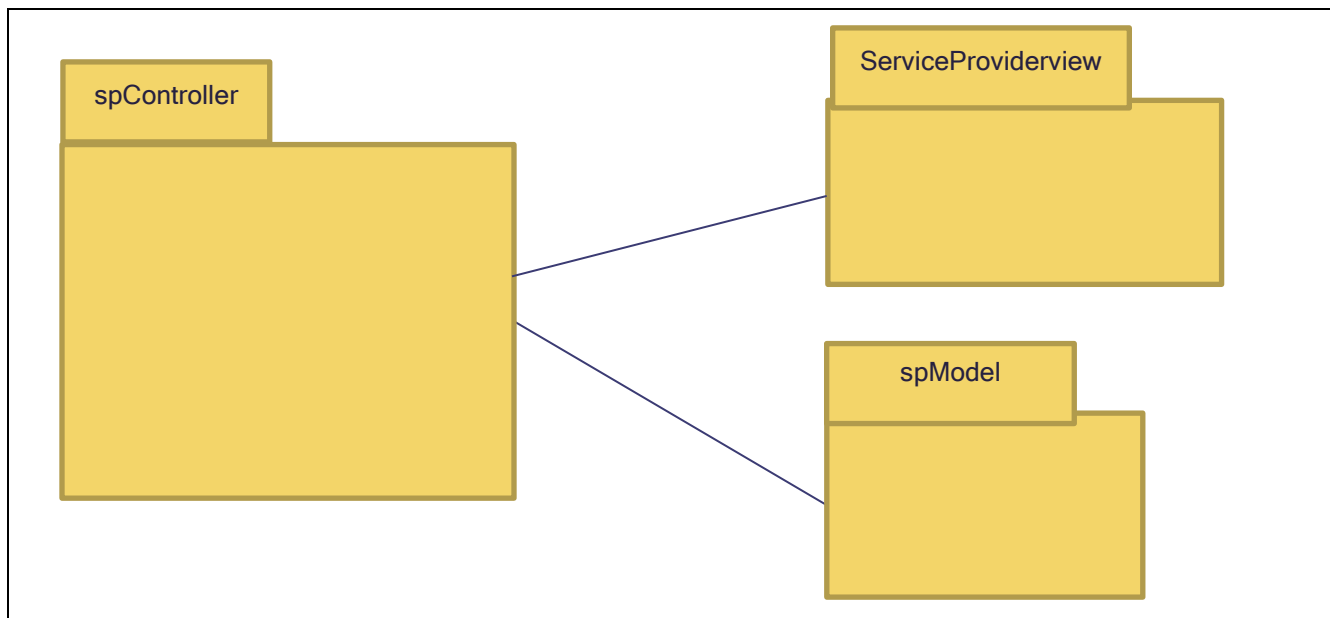


4.16.1 Admin_M [SDD-REQ-1401]

Class Type	Model Class	
Responsibility	Store Admin's account information	
Attributes	Attributes Name	Attributes Type

	Ad_AdminID	VARCHAR(8)
	Ad_Name	VARCHAR(30)
	Ad_Email	String
	Ad_Password	VARCHAR(20)
Methods	Method Name	Description
	Ad_AccountInfo	To store admin's account info
Algorithm	Begin	
	Store admin account information	
	End	

4.17 spController [SDD-REQ-1500][Safinatunnajah Binti Sahrawi CB18002]



4.17.1 spController [SDD-REQ-1500]

Class Type	Controller class	
Responsibility	Put functions to control Service Provider interfaces to connect to spModel.	
Attributes	Attributes Name	Attributes Type
	name	varchar(200)
	price	decimal(7,2)
	des	Int(11)
	quantity	int(3)

	img FoodId GoodId MedicalId sp_id payment_complete sp_confirm Username S_Company_Name S_CompanyAddress S_ServiceProviderPhoneNo	text int(11) int(11) int(11) int(11) tinyint(4) tinyint(4) varchar(11) varchar(12) varchar(50) varchar(30)
Methods	Method Name	Description
	viewAllFood()	To get food information and connect to model
	viewAllGood()	To get food information and connect to model
	viewAllMedical()	To get food information and connect to model
	viewFood()	To get product detail and connect to model
	viewGood()	To get product detail and connect to model
	viewMedical()	To get product detail and connect to model
	addFood()	To receive product information and pass to model
	addGood()	To receive product information and pass to model
	addMedical()	To receive product information and pass to model
	editFood()	To get product detail to update and connect to model
	editGood()	To get product detail to update and connect to model
	editMedical()	To get product detail to update and connect to model
	deleteFood()	To get product ID to delete one product and connect to model
	deleteGood()	To get product ID to delete one product and connect to model
	deleteMedical()	To get product ID to delete one product and connect to model
	acceptOrder()	To change order status on the system and connect to model
	viewInf0()	To get and call user information through model
	updateInfo()	to get and change user information and update through model
	template_header(\$title)	To ensure header has the same template for all system
	template_footer()	To ensure footer has the same template for all system
Algorithm	viewAllFood() begin click Food category; connect to displayAllFood(); show return message and redirect link;	

```

end
viewAllGood()
begin
    click Good category;
    connect to displayAllGood();
    show return message and redirect link;
end
viewAllMedical()
begin
    click Medical category;
    connect to displayAllMedical();
    show return message and redirect link;
end
viewFood()
begin
    post Food information;
    connect to viewFoodDetail();
    show return message and redirect link;
end
viewGood()
begin
    post Good information;
    connect to viewGoodDetail();
    show return message and redirect link;
end
viewMedical()
begin
    post medical information;
    connect to viewMedicalDetail();
    show return message and redirect link;
end
addFood()
begin
    get food information;
    connect to insertFood();
    show return message and redirect link;
end
addGood()
begin
    get goods information;
    connect to insertGood();
    show return message and redirect link;
end
addMedical()
begin
    get medical information;
    connect to insertMedical();
    show return message and redirect link;
end

```



```

editFood()
begin
    get food information;
    connect to modifyFood() or modifyFoodwImg();
    show return message and redirect link;
end

editGood()
begin
    get good information;
    connect to modifyGood() or modifyGoodwImg();
    show return message and redirect link;
end

editMedical()
begin
    get medical information;
    connect to modifyMedical() or modifyMedicalwImg();
    show return message and redirect link;
end

deleteFood()
begin
    delete food product;
    connect to dropFood();
    show return message and redirect link;
end

deleteGood()
begin
    delete goods product;
    connect to dropGood();
    show return message and redirect link;
end

deleteMedical()
begin
    delete medical product;
    connect to dropMedical();
    show return message and redirect link;
end

acceptOrder()
begin
    click Food category;
    connect to accept();
    show return message and redirect link;
end

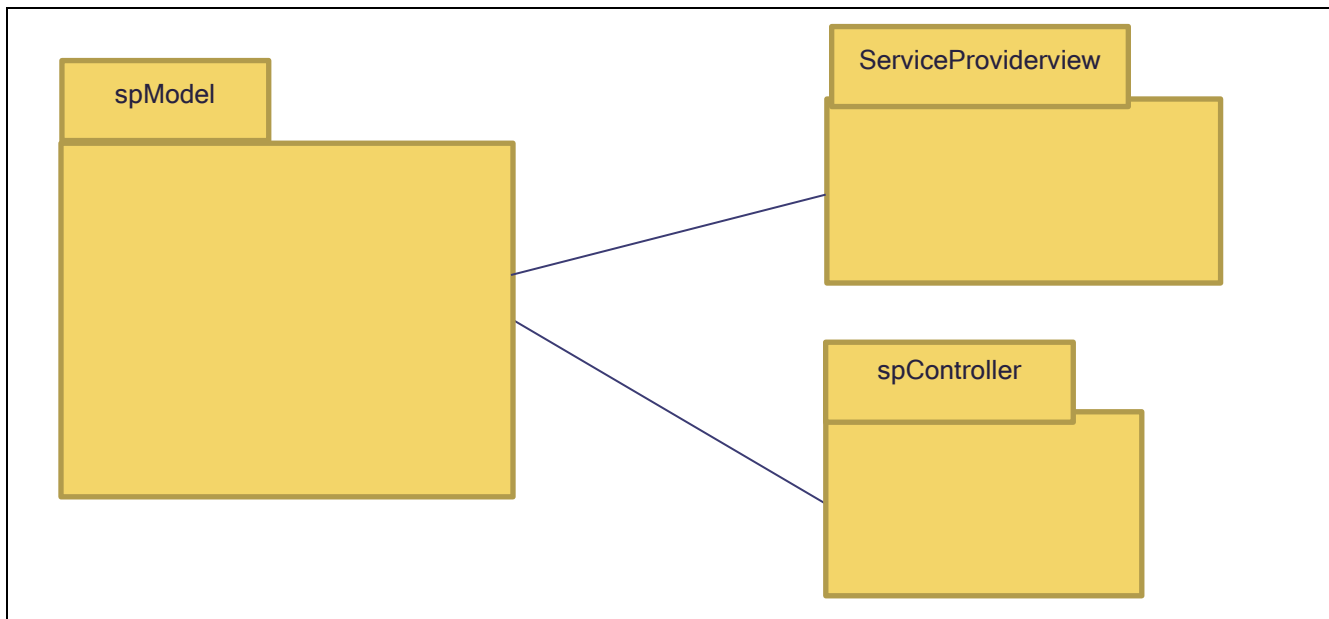
viewInf0()
begin
    Post information to interface;
    connect to viewSPinfo();
    show return message and redirect link;
end

updateInfo()

```

	begin Get information from interface; connect to updateSP(); show return message and redirect link; end
	template_header() begin display header html codes when called end
	template_footer() begin display footer html codes when called end

4.18 spModel [SDD-REQ-1600] [Safinatunnajah Binti Sahrawi CB18002]



4.18.1 spModel [SDD-REQ-1601]

Class Type	Model Class	
Responsibility	To store the information of service provider in the database.	
Attributes	Attributes Name	Attributes Type
	name price des quantity	varchar(200) decimal(7,2) Int(11) int(3)

	img FoodId GoodId MedicalId sp_id payment_complete sp_confirm username S_Company_Name S_CompanyAddress S_ServiceProviderPhoneNo	text int(11) int(11) int(11) int(11) tinyint(4) tinyint(4) varchar(11) varchar(12) varchar(50) varchar(30)
Methods	Method Name	Description
	displayAllFood()	To display all food products in database
	displayAllGood()	To display all good products in database
	displayAllMedical()	To display all medical products in database
	viewFoodDetail()	To display one food product detail
	viewGoodDetail()	To display one food product detail
	viewMedicalDetail()	To display one food product detail
	insertFood()	To add food product into database
	insertGood()	To add food product into database
	insertMedical()	To add food product into database
	modifyFood()	To update food product in database
	modifyGood()	To update good product in database
	modifyMedical()	To update medical product in database
	modifyFoodwImg()	To update food product without changing image in database
	modifyGoodwImg()	To update good product without changing image in database
	modifyMedicalwImg()	To update medical product without changing image in database
	dropFood()	To delete food product from database
	dropGood()	To delete good product from database
	dropMedical()	To delete medical product from database
	accept()	To accept order information from customer
	viewSPinfo()	To view service provider account information
	updateSP()	To update service provider account information
Algorithm	displayAllFood() begin connect to database; retrieve the data information; save the information;	

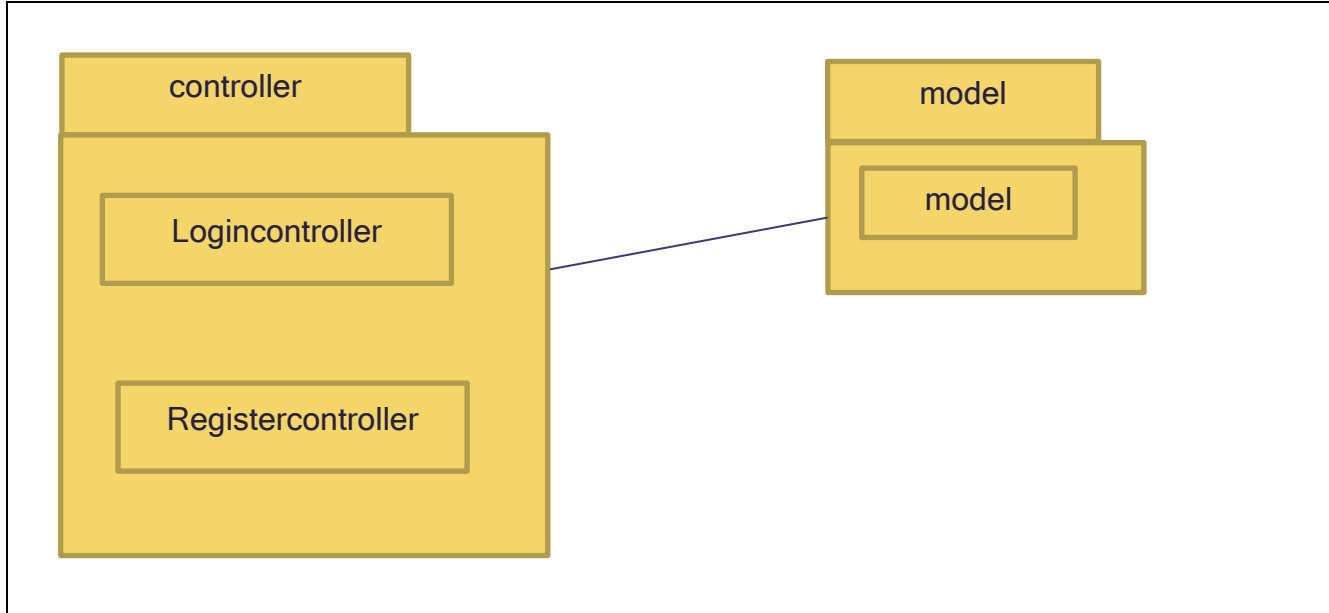
	send the data information for user view; end
	displayAllGood() begin connect to database; retrieve the data information; save the information; send the data information for user view; end
	displayAllMedical() begin connect to database; retrieve the data information; save the information; send the data information for user view; end
	viewFoodDetail() begin connect to database; retrieve the data information; send the data information for user view; end
	viewGoodDetail() begin connect to database; retrieve the data information; send the data information for user view; end
	viewMedicalDetail() begin connect to database; retrieve the data information; send the data information for user view; end
	insertFood() begin

	connect to database; retrieve the data information; send the data information for user view; end
	insertGood() begin connect to database; insert product informations; save the information; send the data information for user view; end
	insertMedical() begin connect to database; insert product information; save the information; send the data information for user view; end
	modifyFood() begin connect to database; update product information; save the information; send the data information for user view; end
	modifyGood() begin connect to database; update product informations; save the information; send the data information for user view; end
	modifyMedical() begin connect to database;

	update product information; save the information; send the data information for user view; end
	modifyFoodwImg() begin connect to database; update product information without changing image; save the information; send the data information for user view; end
	modifyGoodwImg() begin connect to database; update product information without changing image; save the information; send the data information for user view; end
	modifyMedicalwImg() begin connect to database; update product information without changing image; save the information; send the data information for user view; end
	dropFood() begin connect to database; delete product from database; save the information; send the data information for user view; end
	dropGood() begin connect to database;

	delete product from database; save the information; send the data information for user view; end
	dropMedical() begin connect to database; delete product from database; save the information; send the data information for user view; end
	accept() begin connect to database; update sp_confirm in cart table; save the information; send the data information for user view; end
	viewSPinfo() begin connect to database; display SP account info from database; send the data information for user view; end
	updateSP() begin connect to database; update SP account info from database; save the information; send the data information for user view; end

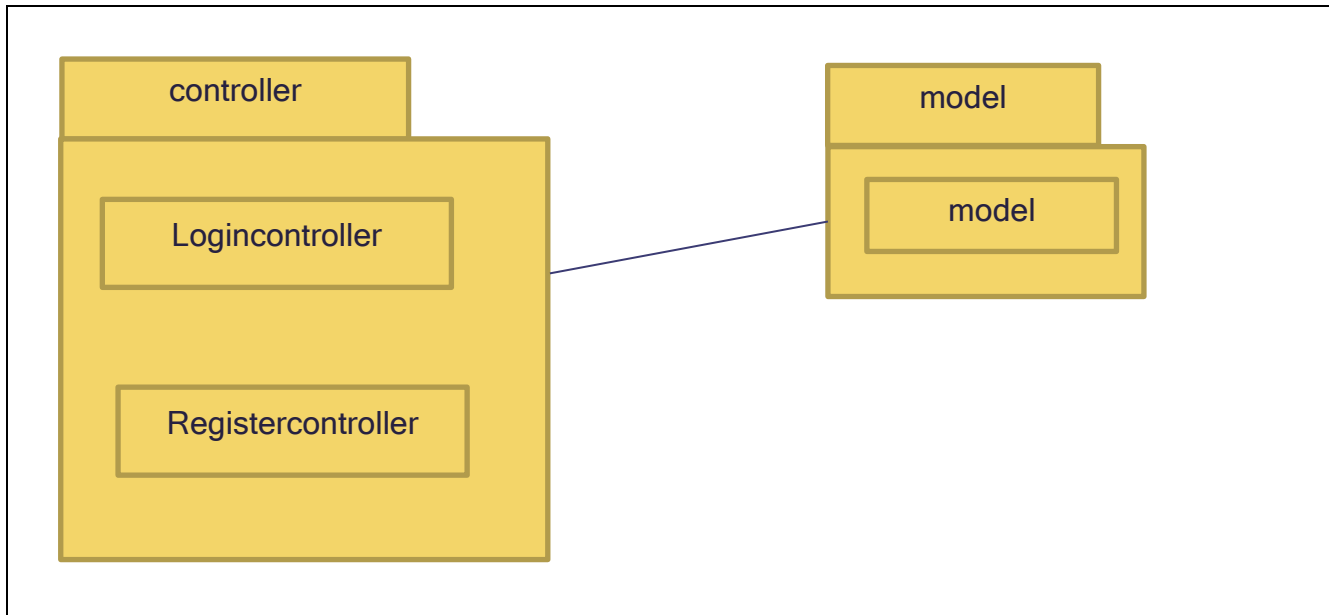
4.19 LoginController [SDD-REQ-1700] [CHOONG ZHEN YUAN CB18059]



4.19.1 LoginController[SDD-REQ-1701] [CHOONG ZHEN YUAN CB18059]

Class Type	Controller	
Responsibility	Manage all function of Login	
Attributes	Attributes Name	Attributes Type
	username	Varchar(50)
	password	Varchar(255)
	usertype	Varchar(255)
Methods	Method Name	Description
	Login()	A function to allow customer, service provider and runner login into the system
Algorithm	START Received requested function Send requested function to model Received Result Send Result to requestor END	

4.20 RegisterController [SDD-REQ-1800] [CHOONG ZHEN YUAN CB18059]



4.20.1 RegisterController[SDD-REQ-1801] [CHOONG ZHEN YUAN CB18059]

Class Type	Controller	
Responsibility	Manage all function of Register	
Attributes	Attributes Name	Attributes Type
	username	Varchar(50)
	password	Varchar(255)
	usertype	Varchar(255)
Methods	Method Name	Description
	Cregister()	A function to allow customer register to system
	SPregister()	A function to allow service provider to register the system
	Rregister()	To allow runner to register to the system
Algorithm	START Validate username If username is count, error message "This username already taken" else username is valid	

validate password

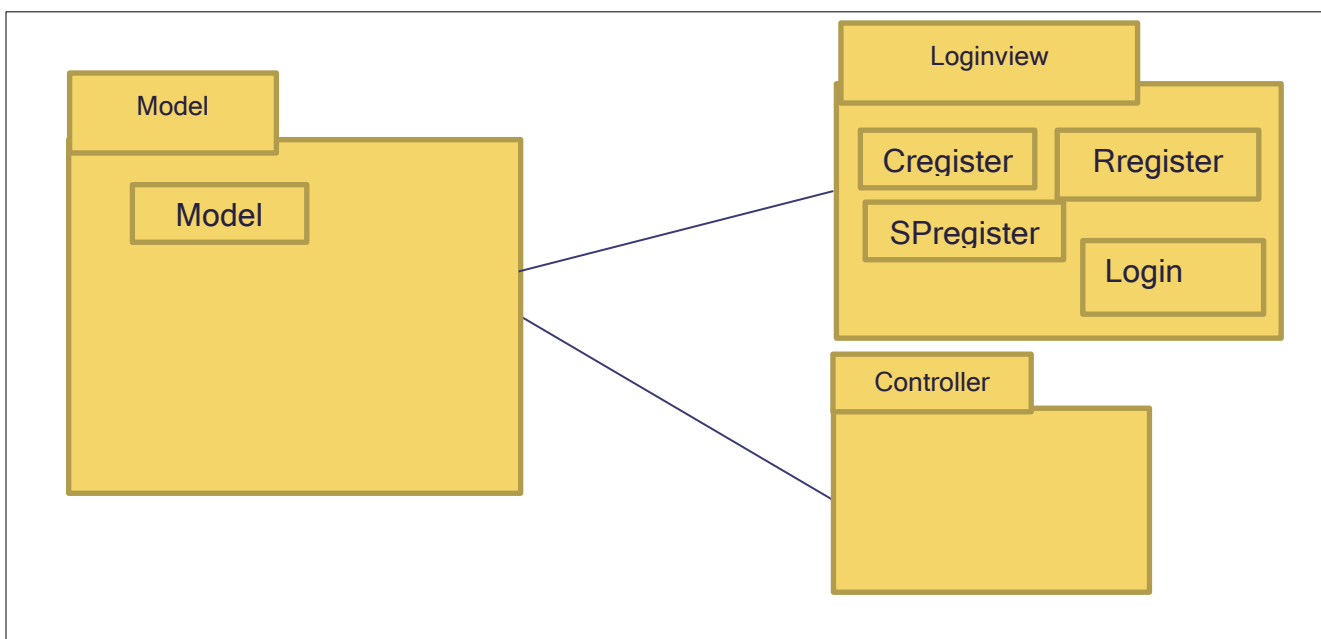
IF password character less than 6, error message "Password must have atleast 6 characters."

Else password validate completed

Validate confirm password

END

4.21 model [SDD-REQ-1900] [CHOONG ZHEN YUAN CB18059]



4.21 model [SDD-REQ-1901] [CHOONG ZHEN YUAN CB18059]

Class Type	Model	
Responsibility	To store the information of user into database	
Attributes	Attributes Name	Attributes Type
	username	Varchar(50)
	password	Varchar(255)
	usertype	Varchar(255)
	created_at	datetime
Methods	Method Name	Description
	registerCustomer()	To allow system store customer info
	registerServiceProvider()	To allow system store service provider info
	registerRunner()	To allow system store Runner info
	addUser()	To allow system store all users info
Algorithm	validateUser()	To allow system validate username and password from system.
	START	
	registerCustomer()	
	begin	
	connect to database;	
Algorithm	retrieve the data information;	
	send the data information into system;	
	end	
	registerServiceProvider()	
	begin	
Algorithm	connect to database;	
	retrieve the data information;	
	send the data information into system;	
	end	
	registerRunner()	

	<pre> begin connect to database; retrieve the data information; send the data information into system; end addUser() begin connect to database; retrieve the data information; send the data information into system; validateUser() begin connect to database; retrieve the data information; send the data information or error message for user view; END </pre>	
zClass Type	Model	
Responsibility	To store the information of user into database	
Attributes	Attributes Name	Attributes Type
	username	Varchar(50)
	password	Varchar(255)
	usertype	Varchar(255)
	created_at	datetime
Methods	Method Name	Description
	registerCustomer()	To allow system store customer info
	registerServiceProvider()	To allow system store service provider info
	registerRunner()	To allow system store Runner info
	addUser()	To allow system store all users info
	validateUser()	To allow system validate username and password from system.
Algorithm	START	

```
registerCustomer()  
    begin  
        connect to database;  
        retrieve the data information;  
        send the data information into system;  
    end  
registerServiceProvider()  
    begin  
        connect to database;  
        retrieve the data information;  
        send the data information into system;  
    end  
registerRunner()  
    begin  
        connect to database;  
        retrieve the data information;  
        send the data information into system;  
    end  
addUser()  
    begin  
        connect to database;  
        retrieve the data information;  
        send the data information into system;  
    end  
validateUser()  
    begin  
        connect to database;  
        retrieve the data information;  
        send the data information or error message for user view;  
    end  
END
```

5. REQUIREMENT TRACEABILITY

5.1 Manage Food module [SRS-REQ-100] [Kam Jian Wei CB18137]

Requirement ID	Description	Design ID
SRS-REQ-401	This requirement will allow customer to click the food picture to choose the food	SDD-REQ-101 SDD-REQ-104 SDD-REQ-1501 SDD-REQ-1601
SRS-REQ-402	This requirement is allow the customer can add more food into the cart	SDD-REQ-101 SDD-REQ-1501 SDD-REQ-1601
SRS-REQ-403	This requirement is allow the customer can delete food into the cart	SDD-REQ-301 SDD-REQ-1501 SDD-REQ-1601
SRS-REQ-404	This requirement is for Service Provider to accept the food order by customer.	SDD-REQ-801 SDD-REQ-1501 SDD-REQ-1601
SRS-REQ-405	This requirement is allow for Service Provider to add the new food product.	SDD-REQ-702 SDD-REQ-703 SDD-REQ-1501 SDD-REQ-1601
SRS-REQ-406	This requirement is allow for Service Provider to delete the food product.	SDD-REQ-706 SDD-REQ-1501 SDD-REQ-1601
SRS-REQ-407	This requirement is allow for Service Provider to edit the food product.	SDD-REQ-706 SDD-REQ-1501 SDD-REQ-1601
SRS-REQ-408	This requirement is allow for Service Provider to view the food product.	SDD-REQ-709 SDD-REQ-712 SDD-REQ-1501 SDD-REQ-1601

SRS-REQ-409	This system will allow Runner to accept the request by the food order	SDD-REQ-501 SDD-REQ-1201 SDD-REQ-1301
SRS-REQ-410	This system will allow Runner to reject the request by the food order	SDD-REQ-501 SDD-REQ-1201 SDD-REQ-1301

5.2 Manage Good Module [SRS-REQ-200] Yamunnawahthi A/P Somasundharam [CB18042]

Requirement ID	Description	Design ID
SRS-REQ-201	The customer can select the good and add the item to the cart.	SDD-REQ-102 SDD-REQ-1001 SDD-REQ-1101
SRS-REQ-202	The customer can remove the item in the cart.	SDD-REQ-301 SDD-REQ-1001 SDD-REQ-1101
SRS-REQ-203	Runner can view the order delivery details.	SDD-REQ-502 SDD-REQ-1201 SDD-REQ-1301
SRS-REQ-204	Runner can update the order status to completed.	SDD-REQ-501 SDD-REQ-1201 SDD-REQ-1301
SRS-REQ-205	The Service Provider can add another product by full fill the required form such as brand,model,name,quantity and brief description and images.	SDD-REQ-702 SDD-REQ-704 SDD-REQ-1501 SDD-REQ-1601
SRS-REQ-206	The Service Provider can edit the item which has new stock in Product by click to edit button.	SDD-REQ-707 SDD-REQ-1501 SDD-REQ-1601

Requirement ID	Description	Design ID
SRS-REQ-207	Service Provider must fill up all the required in the form include the image, system would not accept the addition if there is no image.	SDD-REQ-707 SDD-REQ-1501 SDD-REQ-1601
SRS-REQ-208	Service Provider can view the ordered status by customer	SDD-REQ-701 SDD-REQ-1501 SDD-REQ-1601
SRS-REQ-209	Service Provider can accept the order.	SDD-REQ-801 SDD-REQ-1501 SDD-REQ-1601

5.3 Manage Medical module [SRS-REQ-300] [Safinatunnajah Binti Sahrawi CB18002]

Requirement ID	Description	Design ID
SRS-REQ-301	The service provider can view orders made by customers.	SDD-REQ-801 SDD-REQ-1501 SDD-REQ-1601
SRS-REQ-302	The runner able to view delivery order information issued by service provider.	SDD-REQ-502 SDD-REQ-12201 SDD-REQ-1301
SRS-REQ-303	The customer can add medical products to cart and choose the quantity of products.	SDD-REQ-103 SDD-REQ-1001 SDD-REQ-1101
SRS-REQ-304	The customer can delete medical products in cart if they do not want to buy.	SDD-REQ-301 SDD-REQ-1001 SDD-REQ-1101
SRS-REQ-305	The service provider can view and update account information.	SDD-REQ-601 SDD-REQ-602 SDD-REQ-1501 SDD-REQ-1601

SRS-REQ-306	The service provider can confirm order by customers and pass delivery information to runner.	SDD-REQ-801 SDD-REQ-1501 SDD-REQ-1601
SRS-REQ-307	The service provider can add medical product listings on their page.	SDD-REQ-702 SDD-REQ-705 SDD-REQ-1501 SDD-REQ-1601
SRS-REQ-308	The service provider can edit products information such as information or quantity.	SDD-REQ-708 SDD-REQ-1501 SDD-REQ-1601
SRS-REQ-309	The service provider can delete product from listings.	SDD-REQ-708 SDD-REQ-1501 SDD-REQ-1601
SRS-REQ-310	The customer can edit product quantity from cart.	SDD-REQ-301 SDD-REQ-1501 SDD-REQ-1601

5.4 Payment module [SRS-REQ-400] [Choong Zhen Yuan CB18059]

Requirement ID	Description	Design ID
SRS-REQ-500	This requirement is to allow the Customer to view and choose the products by categories in the system.	SDD-REQ-201
SRS-REQ-501	This requirement will allow customer to choose the product and add it into the cart.	SDD-REQ-202
SRS-REQ-502	This requirement is for Customer to edit or delete the ordered product. Customer will check their order before the payment.	SDD-REQ-203