# Test Plan

Diabetes Care

By

# Jirayu Chinpongsuwan 552115011

Department of Software Engineering College of Arts, Media and Technology Chiang Mai University

# **Project Advisor**

Dr. Prompong Sugunnasil

Document Name	Diabetes Care -Test Plan-V.0.2.docx	Owner	Jirayu	Page	1 / 83
Document Type	Test Plan	Release Date	July30,2016	Print Date	July30,2016

# **Document History**

<b>Document Name</b>	Version	Status	Date	Viewa ble	Reviewe r	Responsibl e
Document						
diabetes –Test plan- V.0.1.docx	Add: Chapter One   Introduction Add: Chapter Two   Test Procedure Add: Chapter Three   Software Testing	Draft	Febru ary 27, 2016	JC, PS	JC, PS	JC, PS
diabetes –Test plan- V.0.1.docx	Update: Chapter Three   Software Testing Add: SystemTest	Draft	Febru ary 29, 2016	JC, PS	JC, PS	JC, PS
diabetes –Test plan- V.0.1.docx	Update: System test Add: Unit test	Draft	July 10,20 16	JC, PS	JC, PS	JC, PS
diabetes –Test plan- V.0.1.docx	Update: System test Update: Unit test	Draft	July 23,20 16	JC, PS	JC, PS	JC, PS
diabetes –Test plan- V.0.1.docx	Update: Unit test	Draft	July 25,20 16	JC, PS	JC, PS	JC, PS
diabetes –Test plan- V.0.1.docx	Update: Unit test	Release	July 30,20 16	JC, PS	JC, PS	JC, PS

\*PS: Dr. Prompong Sugunnasil \*JC: Mr. Jirayu Chinpongsuwan

Document Name	Diabetes Care -Test Plan-V.0.2.docx	Owner	Jirayu	Page	2 / 83
Document Type	Test Plan	Release Date	July30,2016	Print Date	July30,2016

# **Table Content**

Document Type

Test Plan

Chapter One	Introduction					4
1.1 Docume	ent purpose					4
1.2 Scope						4
1.3 Definiti	ons, Acronyms, and A	Abbreviation				4
1.3.1 Ke	y Definitions				•••••	4
1.3.2 Ke	y Acronyms and Abbi	reviation			•••••	5
Chapter Two	Test Procedure				•••••	6
2.1 Test Str	ategy				•••••	6
2.2 Pass/Fa	il Criteria				•••••	6
2.3 Test En	vironment		•••••		•••••	6
Chapter Three	Software Testing					8
3.1 Objectiv	ve of Testing				•••••	8
3.2 Scope of	of Testing		•••••		•••••	9
3.3 System	Test				1	1
3.3.1 Fea	ature 1		•••••		1	12
3.3.2 Fea	ature 2				1	15
3.3.3 Fea	ature 3				2	29
3.3.4 Fea	ature 4		•••••		3	35
3.3.5 Fea	ature 5				4	11
3.3.6 Fea	ature 6				4	13
3.3.7 Fea	ature 7				4	15
3.3.8 Fea	ature 8				5	50
3.4 Unit Te	st				5	52
3.4.1 Cla	ass AccountController				5	53
Document Name	Diabetes Care -Test Plan-V.0.2.docx	Owner	Jirayu	Page	3 / 83	

July30,2016

Release Date

Print Date

July30,2016

3.4.2 Class FoodController	72
3.4.3 Class ExerciseController	76
3.4.4 Class ManageController	81

Document Name	Diabetes Care -Test Plan-V.0.2.docx	Owner	Jirayu	Page	4 / 83
Document Type	Test Plan	Release Date	July30,2016	Print Date	July30,2016

# **Chapter One | Introduction**

A Software Test Plan is a document describing the testing scope and activities. It is the basis for formally testing any software or product in a project. For this Test Plan Document is provide for testing the software product cover the features in phase of implementation the progress report II.

#### 1.1 Document purpose

Test plan document is a document used to determine and explain testing process of Diabetes care web application. The process of testing is purpose to find the software error that may occur during development process to make sure that the error has been reduce. Therefore, this can guarantee the quality of the software before delivery that result in the application will be the most efficiency. The test case will design base on Software Requirement Specification document and Software Design Document.

#### 1.2 Scope

The Test Plan document covers recording and result of testing for each function in Diabetes web application to find the error or failure of the system. Also the test plan is made to measure the user requirements and system requirement. For the content of this document is consist of part of testing schedule, testing process, and data of testing to estimate an expect result of test before can get the actual result in the test record.

### 1.3 Definitions, Acronyms, and Abbreviation

#### 1.3.1 Key Definitions

Diabetes

"Name of project"

"Computer programs, procedure and associated documentation and data pertain to the operation of computer system" [IEEE90]

Document Name	Diabetes Care -Test Plan-V.0.2.docx	Owner	Jirayu	Page	5 / 83
Document Type	Test Plan	Release Date	July30,2016	Print Date	July30,2016

**Requirement** "The period of time in the software life cycle

during which the requirement for a software

product defines and document" [IEEE90]

**Test** An activity in which a system or

component is executed under specified conditions, the results are observed or

recorded, and an evaluation is made of some

aspect of the system or component.

[IEEE90]

**Unit testing** Testing of an individual hardware or

software units or groups of related units.

[IEEE90]

### 1.3.2 Key Acronyms and Abbreviation

TC Test case

**ID** Identify number

UT Unit test ST System test

Document Name	Diabetes Care -Test Plan-V.0.2.docx	Owner	Jirayu	Page	6 / 83
Document Type	Test Plan	Release Date	July30,2016	Print Date	July30,2016

# **Chapter Two | Test Procedure**

### 2.1 Test Strategy

Tester test Integration and Developer test unit level Step of Integration Test following:

- 1. Design test case
- 2. Prepare testing data
- 3. Determine expected result
- 4. Perform testing
- 5. Result of testing is to be record
- 6. Result and test files will be place in the project library

#### 2.2 Pass/Fail Criteria

Test record of this Software Testing Document will device to two parts

**1. Actual output** Actual output is a real output of the

application from testing.

**2. Result of Testing** Result of Testing is a result of the testing

process there are two kinds is:

Pass is a result that actual output has data as

same as expect output. When the result of

testing is pass tester will record data.

**Fail** Fail is a result that data of actual output

difference from expect output. When the result of testing is fail tester will comment error and send record to developer for edit or repair that error then send new application to

tester for retested.

#### 2.3 Test Environment

# Hardware Dell inspiron 5459

T 4 1 '4

- Intel core i5
- RAM 4GB 64-bit, Operating System, HDD 1 TB

Document Name	Diabetes Care -Test Plan-V.0.2.docx	Owner	Jirayu	Page	7 / 83
Document Type	Test Plan	Release Date	July30,2016	Print Date	July30,2016

# **Chapter Three | Software Testing**

# 3.1 Objective of Testing

This testing has objective for make all functions are consistency and reliability with a highest quality by check correctness of Diabetes web application. To make sure that the result of testing has record are correctly, every test have to follow test from this part of Test Plan document.

Document Name	Diabetes Care -Test Plan-V.0.2.docx	Owner	Jirayu	Page	8 / 83
Document Type	Test Plan	Release Date	July30,2016	Print Date	July30,2016

# **3.2 Scope of Testing**

### **3.2.1 Unit Test**

Use Case ID	Use case name
UT01	AccountController: AddTodayFood()
UT02	AccountController: addTodayfood()
UT03	AccountController: RemoveTodayFood()
UT04	AccountController: HealthPlan()
UT05	AccountController: AddGlycemic()
UT06	AccountController: addUserGlycemic()
UT07	AccountController: AddFood()
UT08	AccountController: addUSerFood()
UT09	AccountController: DeleteFood()
UT10	AccountController: AddMedicine()
UT11	AccountController: addMedicineData()
UT12	AccountController: DeleteMedicine()
UT13	AccountController: EditUserProfile()
UT14	AccountController: Register()
UT15	AccountController: Login()
UT16	AccountController:LogOff()
UT17	FoodController:Create()
UT18	FoodController:Edit()
UT19	FoodController:Delete()
UT20	ExercisesController:Create()
UT21	ExercisesController: Edit()
UT22	ExercisesController:Delete()
UT23	ManageController ChangePassword()

**Table 1:** Black Box testing technique (UT01-UT23)

Document Name	Diabetes Care -Test Plan-V.0.2.docx	Owner	Jirayu	Page	9 / 83
Document Type	Test Plan	Release Date	July30,2016	Print Date	July30,2016

### 3.2.2 System Test

<b>Use Case ID</b>	Use case name	
ST01	Login to the system.	
ST02	Logout to the system.	
ST03	Nurse can registration patient to Web Application.	
ST04	Patient can view information of his on profile page.	
ST05	Patient can update info on his profile page.	
ST06	Patient can change password.	
ST07	Admin can add user to system.	
ST08	Admin can edit information of user.	
ST09	Admin can delete user from system	
ST10	Nutritionists can add food to the database.	
ST11	Nutritionists can edit food on the database.	
ST12	Nutritionists can delete food from the database	
ST13	Nurse/Doctor can add food to the database.	
ST14	Nurse/Doctor can edit food on the database.	
ST15	Nurse/Doctor can delete food from the database	
ST16	Patient can select the foods from database.	
ST17	Patient can select the activity from database.	
ST18	Patient can view body mass value that calculated.	
ST19	Patient can record blood sugar.	
ST20	Patient can view statistic graph of blood sugar.	
ST21	Patient can get recommend about blood sugar from system.	
ST22	Patient can select food.	
ST23	Patient can submit and view about analyze result.	

**Table 2:** Black Box testing technique (ST01-ST23)

Document Name	Diabetes Care -Test Plan-V.0.2.docx	Owner	Jirayu	Page	10 / 83
Document Type	Test Plan	Release Date	July30,2016	Print Date	July30,2016

# 3.3 System Test

### Black Box technique (ST01-ST23)

Use Case ID	Use case name	
ST01	Login to the system.	
ST02	Logout to the system.	
ST03	Nurse can registration patient to Web Application.	
ST04	Patient can view information of his on profile page.	
ST05	Patient can update info on his profile page.	
ST06	Patient can change password.	
ST07	Admin can add user to system.	
ST08	Admin can edit information of user.	
ST09	Admin can delete user from system	
ST10	Nutritionists can add food to the database.	
ST11	Nutritionists can edit food on the database.	
ST12	Nutritionists can delete food from the database	
ST13	Nurse/Doctor can add food to the database.	
ST14	Nurse/Doctor can edit food on the database.	
ST15	Nurse/Doctor can delete food from the database	
ST16	Patient can select the activity from database.	
ST17	Patient can select the foods from database.	
ST18	Patient can view body mass value that calculated.	
ST19	Patient can record blood sugar.	
ST20	Patient can view statistic graph of blood sugar.	
ST21	Patient can get recommend about blood sugar from system.	
ST22	Patient can select food.	
ST23	Patient can submit and view about analyze result.	

**Table 3:** Black Box testing technique (ST01-ST23)

Document Name	Diabetes Care -Test Plan-V.0.2.docx	Owner	Jirayu	Page	11 / 83
Document Type	Test Plan	Release Date	July30,2016	Print Date	July30,2016

# 3.3.1 Feature#1 Authentication system.

**ST01:** Login to the system.

Description	Test for <b>UC01</b> : Login to the system.				
Precondition	Test data must be prepared.	Test data must be prepared.			
<b>Prerequisites or Test input</b>	Table: AspUser				
	Username atomsuperza				
	Password 12345678				
Test Script	1. User go to login page.				
	2. User inputs information on login form.				
	3. User clicks "Login" button.				

Case	Description	Input	Expect Output		
1	Test login with all	Username = "atomsuperza"	The system redirect to page "Account/Index".		
1	correct information.	Password = 12345678"			
	Test login when	Username="atom12"	The system display error message " Invalid login attempt"		
	input invalid email	Password = "12345678"			
	or password.		Invalid login attempt.		
2			UserName atom		
			_		
			Password		

Document Name	Diabetes Care -Test Plan-V.0.2.docx	Owner	Jirayu	Page	12 / 83
Document Type	Test Plan	Release Date	1 March 2016	Print Date	1 March 2016

	Test login with	Username = ""	The system display error message "The UserName field is required.",			
	blank information.	Password = ""	"The Password field is required."			
			UserName			
3			The UserName field is required.			
			Password			
				The Password field is required.		

Table 4: Test Case of ST02: Login to the system.

Document Name	Diabetes Care -Test Plan-V.0.2.docx	Owner	Jirayu	Page	13 / 83
Document Type	Test Plan	Release Date	1 March 2016	Print Date	1 March 2016

**ST02:** Logout the system.

<b>Description</b> Test for <b>UC02</b> : Logout the system.	
<b>Precondition</b> Test data must be prepared.	
<b>Prerequisites or Test input</b>	-
Test Script	1. User clicks "Logout" button.

Case	Description	Input	Expect Output
	Test for logout the	Click "Logout" button	The system redirect to login page.
1	system.	on navigation.	

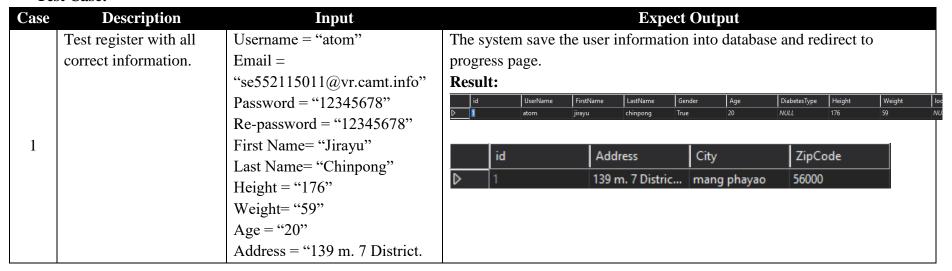
 Table 5: Test Case of ST03: Logout the system.

Document Name	Diabetes Care -Test Plan-V.0.2.docx	Owner	Jirayu	Page	14 / 83
Document Type	Test Plan	Release Date	1 March 2016	Print Date	1 March 2016

### 3.3.2 Feature#2: Account management.

**ST03:** Register to the system.

Description	Test for <b>UC03</b> : Nurse can registration patient to Web Application.			
Precondition	est data must be prepared.			
<b>Prerequisites or Test input</b>	-			
Test Script	1. User go to register page.			
	2. User inputs information on register form.			
	3. User clicks "Submit" button.			



Document Name	Diabetes Care -Test Plan-V.0.2.docx	Owner	Jirayu	Page	15 / 83	
Document Type	Test Plan	Release Date	1 March 2016	Print Date	1 March 2016	

		tom"	
		City= "Phayao"	
		Zip Code = "56000"	
2	Test register when input	Email = " <u>aaaaaa</u> "	The system display error message "The Email field is not a valid e-mail
2	existed email.		address."
	Test register when	Password = "12345678"	The system display error message "The password and confirmation
3	password and re-	Re-password = "12345677"	password do not match"
	password is not match.		
	Test register with blank	Username = ""	The system display error message "field is required."
	information.	Email = ""	
		Password = ""	
		Re-password = ""	
		First Name= ""	
4		Last Name= ""	
4		Height = ""	
		Weight= ""	
		Age = ""	
		Address = " "	
		City= " "	
		Zip Code = " "	

 Table 6: Test Case of ST01: Register to the system.

Document Name	Diabetes Care -Test Plan-V.0.2.docx	Owner	Jirayu	Page	16 / 83	
Document Type	Test Plan	Release Date	1 March 2016	Print Date	1 March 2016	

ST04: Patient can view information of his on profile page.

Description	Test for <b>UC04</b> : Patient can view infe	est for <b>UC04</b> : Patient can view information of his on profile page.							
Precondition	Test data must be prepared.	est data must be prepared.							
<b>Prerequisites or Test input</b>	Table: AspNet	Table: AspNet							
	First name	Jirayu	qwe						
	Last name	Chinpong	wqr						
	Height	176	123						
	Weight	56	213						
	Age	20	123						
	Address	139 m. 7 District. tom	139 m. 7 District. tom						
	City	phayao	phayao						
	Zipcode	2500	2500						
Test Script	1. Patient go to a profile page								

Document Name	Diabetes Care -Test Plan-V.0.2.docx	Owner	Jirayu	Page	17 / 83	
Document Type	Test Plan	Release Date	1 March 2016	Print Date	1 March 2016	

Case	Description	Input	Expect Output
1	Test thesystem shall display user profile.	First Name= "Jirayu" Last Name= "Chinpong" Height = "176" Weight= "59" Age = "20" Address = "139 m. 7 District. tom" City= "Phayao" Zip Code = "56000"	The system shall redirect to "Manage" page and display pertinent information.  Diabeles Care About Patient Nutritionis Nutricionis Nutrici

Table 7: Test Case of ST05: Custom notification.

Document Name	Diabetes Care -Test Plan-V.0.2.docx	Owner	Jirayu	Page	18 / 83	
Document Type	Test Plan	Release Date	1 March 2016	Print Date	1 March 2016	

# **ST05:** Patient can update info on his profile page.

Description	Test for <b>UC05</b> : Patient can update info on his profile page.					
Precondition	Test data must be prepared.					
<b>Prerequisites or Test input</b>	The test data are username, email, password, re-password, first name, last name, height, weight, age, gender,					
	ldress, diabetes type.					
Test Script	Patient click username that show on layout page.					
	2. Patient click update info.					
	3. Patient input new info.					
	4. Patient click "submit".					

Document Name	Diabetes Care -Test Plan-V.0.2.docx	Owner	Jirayu	Page	19 / 83	
Document Type	Test Plan	Release Date	1 March 2016	Print Date	1 March 2016	

Case	Description	Input					Expect	t Output	t			
	Test for	First Name= "Jirayuu"	The system save the user information into database.									
	successfully	Last Name=										
	edit personal	"Chinpongsuwan"	id	UserName	FirstName	LastName	Gender	Age	DiabetesType	Height	Weight	locallmage
	information.	Height = "177"	<u> </u>	atom	jirayu	chinpong	True	20	NULL	176	59	NULL
		Weight="60"	<u> </u>	atom	jiiayu	chinpong	IIUC	20	NOLL	110	J7	NOLL
1		Age = "20"										
	Test register	Username = ""	The system	m displa	ay error	message	e "field i	s require	d."			
	with blank	Email = ""		1		Č		•				
2	information.	Password = ""										
		Re-password = ""										
	Test input	Email = "a"	The system	m displa	ay error	message	e "The I	Email fie	ld is not	a valid	e-mail	address."
3	email in											
	wrong											
	format.											

 Table 8: Test Case of ST05: Patient can update info on his profile page.

Document Name	Diabetes Care -Test Plan-V.0.2.docx	Owner	Jirayu	Page	20 / 83	
Document Type	cument Type Test Plan		1 March 2016	Print Date	1 March 2016	

# **ST06:** Patient can change password.

Description	Test for <b>UC06</b> : Patient can change password.			
Precondition	Test data must be prepared.	Test data must be prepared.		
<b>Prerequisites or Test input</b>	Table: AspUser	Table: AspUser		
	password	12345678		
Test Script	<ol> <li>Patient go profile page</li> </ol>			
	2. Patient click "change password"			
	3. Patient input new password.			
	4. Patient clicks "submit"	'button.		

Document Name	Diabetes Care -Test Plan-V.0.2.docx	Owner	Jirayu	Page	21 / 83
Document Type	Test Plan	Release Date	1 March 2016	Print Date	1 March 2016

Case	Description	Input	Expect Output
	Test for successfully	Current password = "12345678"	The system get new password and save change in the
1	information.	New password = "1234567789"	database
1		Confirm password=	
		"123456789"	
	Test change password when	Current password = "12345555"	The system display error message " current password
2	input invalid current password.		invalid."
	Test change password when	New password = "12345678"	The system display error message "confirm password not
	input New password and	Confirm password= "22345679"	match"
3	Confirm password not the same.	1	
	· ·		
4	Test create plan with blank	Current password = " "	The system display error message "field is required."
4	information.	New password = " "	

 Table 9: Test Case of ST06: Patient can change password.

Document Name	Diabetes Care -Test Plan-V.0.2.docx	Owner	Jirayu	Page	22 / 83
Document Type	Test Plan	Release Date	1 March 2016	Print Date	1 March 2016

**ST07:** Admin can add user to system.

Description	Test for <b>UC07</b> : Admin can	Test for UC07: Admin can add nutritionists to system.			
Precondition	Test data must be prepared	Test data must be prepared.			
	Username= "Nutritionists"				
	Email "Nutritionists@n.co	<u>m</u> "			
	Password= "nutritionists"				
	Roles "Nutritionists"				
<b>Prerequisites or Test input</b>	Table: AspUser				
	Username	Nutritionists	Doctor	]	
	Email	Nutritionists@n.com	Doctor@d.com		
	Password	nutritionists	12345678		
	Role	Nutritionists	Doctor		
Test Script	1. Admin login to syste	em by username and passy	word for admin		
	2. Admin click Admin	istration on layout.			
	3. Admin click "Account management".				
	4. Admin click "Create New user account".				
	5. Admin select Roles	"Nutrtionists".			

Document Name	Diabetes Care -Test Plan-V.0.2.docx	Owner	Jirayu	Page	23 / 83
Document Type	Test Plan	Release Date	1 March 2016	Print Date	1 March 2016

Case	Description	Input	Expect Output
	Test for successfully	Username=	The system shall record the data on the database.
	create plan information.	"Nutritionists"	Result: db964c7a-79ed Nutritionists@n False AARTh77GW8h f5cd9ff3-7098-4 NULL
1		Email  "Nutritionists@n.com"  Password=  "nutritionists"  Roles "Nutritionists"	False False NULL True 0 nutritionists
2	Test create plan with blank information.	Username= "" Email ="" Password= "" Roles ""	The system display error message "field is required."
3	Test input password 5 digits.	Password " nutri"	The system shall display error message "Failed to create the user.".
4	Test input email wrong format.	Email = "Nurtiton.com"	The system shall display error message "Email wrong format".

 Table 10: Test Case of ST07: Admin can add nutritionists to system.

Document Name	Diabetes Care -Test Plan-V.0.2.docx	Owner	Jirayu	Page	24 / 83
Document Type	Test Plan	Release Date	1 March 2016	Print Date	1 March 2016

**ST08:** Admin can edit information of user.

Description	Test for <b>UC07</b> : Admin can	Test for <b>UC07:</b> Admin can edit information of nutritionists.			
Precondition	Test data must be prepared				
	Username= "Nutritionists2	,,,			
	Role= "nutrition"				
<b>Prerequisites or Test input</b>	Table: AspUser				
	Username	Nutritionists	Doctor		
	Email	Nutritionists@n.com	Doctor@d.com		
	Password	nutritionists	12345678		
	Role	Nutritionists	Doctor		
Test Script	1. Admin go to manag	e page			
	2. Admin click nutrition	onists.			
	3. Admin click "edit".				
	4. Admin input new username, password, name				
	5. Admin click "submi	•			

Document Name	Diabetes Care -Test Plan-V.0.2.docx	Owner	Jirayu	Page	25 / 83
Document Type	Test Plan	Release Date	1 March 2016	Print Date	1 March 2016

Case	Description	Input	Expect Output
	Test for the admin can	Username= "Nutritionists2"	The system shall record the data on the database.
1	edit the user roles.	Role="doctor"	Result:
1			Username= "Nutritionists2"
			Role="doctor"
	When the admin dose	Username " "	The system display error message "field is required."
2	not input new info of	Password " "	
	user account.	Role=""	

Table 11: Test Case of ST08: Admin can edit information of nutritionists.

Document Name	Diabetes Care -Test Plan-V.0.2.docx	Owner	Jirayu	Page	26 / 83
Document Type	Test Plan	Release Date	1 March 2016	Print Date	1 March 2016

**ST09:** Admin can delete user from system

Description	Test for <b>UC07</b> : Admin can delete nutritionists from system						
Precondition	Test data must be prepared	Test data must be prepared. { Username = nutritionists}					
<b>Prerequisites or Test input</b>	Table: AspUser						
	Username	Nutritionists	Doctor				
	Email	Nutritionists@n.com	Doctor@d.com				
	Password	nutritionists	12345678				
	Role	Nutritionists	Doctor				
Test Script	1. Admin go to manag	e page					
	2. Admin click nutrition	onists.					
	3. Admin select nutriti	3. Admin select nutritionists.					
	4. Admin click "Delete".						

Document Name	Diabetes Care -Test Plan-V.0.2.docx	Owner	Jirayu	Page	27 / 83
Document Type	Test Plan	Release Date	1 March 2016	Print Date	1 March 2016

Case	Description	Input	Expect Output		
	Test for the system	Username =	The system shall delete the nutritionists from	n the database.	
	shall delete user	nutritionists	atomsuperza	Edit   <mark>Delete</mark>	
	account from the		doctor	Edit   <mark>Delete</mark>	
	system		nutritionists	Edit   <mark>Delete</mark>	
1			After delete		
			atomsuperza	Edit   Delete	
			doctor	Edit   Delete	

Table 12: Test Case of ST09: Admin can delete nutritionists from system

Document Name	Diabetes Care -Test Plan-V.0.2.docx	Owner	Jirayu	Page	28 / 83
Document Type	Test Plan	Release Date	1 March 2016	Print Date	1 March 2016

# **3.3.3** Feature#3: Nutrition management.

**ST10:** Nutritionists can add food to the database.

Description	Test for <b>UC08</b> : Nutritionists can	Test for <b>UC08</b> : Nutritionists can add food to the database.				
Precondition	Test data must be prepared.					
	Food{ food name = "banana2", g	glycemic index =110	,calorie = 110}			
<b>Prerequisites or Test input</b>	Table: Food					
	food name	banana1	banana2			
	glycemic	100	110			
	calorie	100	110			
Test Script	1 Nutritionists go to food man	agement page				
	2 Nutritionists click "create ne	ew".				
	3 Nutritionists input food_nam	ne, glycemic index, ca	alorie			
	4 Nutritionists click "submit"					

Document Name	Diabetes Care -Test Plan-V.0.2.docx	Owner	Jirayu	Page	29 / 83
Document Type	Test Plan	Release Date	1 March 2016	Print Date	1 March 2016

Case	Description	Input			<b>Expect Outp</b>	ut		
	Test for the	Food_name= "banana"	The system sh	all record th	ne data on the database	2.		
	system shall add	food_Calories = "100"	Result:					
	the information	food_GlycemicIndex =	food name	food Calories	food GlycemicIndex			
1	of food. When	"20"	_	-	- /			
	the nutritionists		banana1	100	100	Edit   Details   Delete		
	add new food.		banana2	110	110	Edit   Detaiis   Delete		
	Test when the	Food_name= ""	The system display error message "The food_Calories field is required.", "The					
	nutritionists dose	food_Calories = ""	food_Glycemi	food_GlycemicIndex field is required."				
2	not input	food_GlycemicIndex = ""	food	_name bar	nana2			
	glycemic inde		food_Ca	The f	food_Calories field is required.			
	and calorie		food_Glycemic		ood_GlycemicIndex field is requ	iired.		
	Test input	food_Calories = "abc"	The system sh	all display	error message "The fie	eld food_Calories must be a		
	character in	food_GlycemicIndex =	number." And	"The field	food_GlycemicIndex	must be a number."		
	"food_Calories"	"abc"	food_	Calories	aaa			
3	and				The field food_Calories i	must be a number.		
	"food_GlycemicI		food_Glycen	nicIndex	aaa			
	ndex".				The field food_Glycemic	Index must be a number.		

 Table 13: Test Case of ST10: Nutritionists can add food to the database.

Document Name	Diabetes Care -Test Plan-V.0.2.docx	Owner	Jirayu	Page	30 / 83
Document Type	Test Plan	Release Date	1 March 2016	Print Date	1 March 2016

**ST11:** Nutritionists can edit food on the database.

Description	Test for <b>UC08</b> : Nutritionists can e	dit food on the databas	e.				
Precondition	Test data must be prepared.	Test data must be prepared.					
	Food {food name = "banana2"						
	glycemic index =110						
	calorie = 110}						
<b>Prerequisites or Test input</b>	Table: Food						
	food name	banana1	banana2				
	glycemic	100	110				
	calorie	100	110				
Test Script	1 Nutritionists go to food mana	gement page					
	3 Nutritionists click "edit".						
	4 Nutritionists input new food_	4 Nutritionists input new food_name, glycemic index, calorie					
	5 Nutritionists click "submit"						

Document Name	Diabetes Care -Test Plan-V.0.2.docx	Owner	Jirayu	Page	31 / 83
Document Type	Test Plan	Release Date	1 March 2016	Print Date	1 March 2016

Case	Description	Input			<b>Expect Outpu</b>	ıt	
	Test for the system	food name = "banana2"	The systen	n shall record	the data on the data	abase.	
	shall edit the	glycemic index =111	Result: be	fore edit			
	information of food.	calorie = 111	food_name	food_Calories	food_GlycemicIndex		
	When the nutritionists		banana1	100	100	Edit   Details   Delete	
1	edit the information of		banana2	110	110	Edit   Details   Delet	
	food.		After				
			food_name	food_Calories	food_GlycemicIndex		
			banana1	100	100	Edit   Details   Delete	
			banana2	111	111	Edit   Details   Delete	
2	Test when the nutritionists dose not input glycemic inde and calorie	food name = "banana2" glycemic index = "" calorie = ""	The system display error message "The food_Calories field is required.", "The food_GlycemicIndex field is required."    food_name				
	Test input character in	glycemic index =110 calorie =	The systen	n shall display	y error message "Tł	ne field food_Calories must	
	"food_Calories" and	110	be a numb	er." And "The	e field food_Glycer	nicIndex must be a number."	
3	"food_GlycemicIndex".		foo	od_Calories	aaa The field food_Cald	ories must be a number.	
			food_Gly	cemicIndex	aaal The field food_Glyd	cemicIndex must be a number.	

Table 14: Test Case of ST11: Nutritionists can edit food on the database.

Document Name	Diabetes Care -Test Plan-V.0.2.docx	Owner	Jirayu	Page	32 / 83
Document Type	Test Plan	Release Date	1 March 2016	Print Date	1 March 2016

**ST12:** Nutritionists can delete food from the database.

Description	Test for <b>UC08:</b> Nutritionists can delete food from the database.						
Precondition	Test data must be prepared.						
	Food {food name = {"banana1", "1	oanana2"},					
	glycemic index = $\{100,110\}$ ,						
	calorie = {100,110}}						
<b>Prerequisites or Test input</b>	Table: Food						
	food name banana1 banana2						
	glycemic	100	110				
	calorie 100 110						
Test Script	1 Nutritionists go to food management page						
	2 Nutritionists select food.						
	3 Nutritionists click "Delete".						

Document Name	Diabetes Care -Test Plan-V.0.2.docx	Owner	Jirayu	Page	33 / 83
Document Type	Test Plan	Release Date	1 March 2016	Print Date	1 March 2016

Description	Input	Expect Output							
Test for the system	food name = "banana2"	The system	The system shall delete the food from the database.						
shall delete	glycemic index =110	food_name	food_Calories	food_GlycemicIndex					
exercises from list.	calorie = 110	banana1	100	100	Edit   Details   Delete				
When the doctor or		banana2	110	-110	Edit   Details   Delete				
nurse click delete.		When delete							
		food_name	food_Calories	food_GlycemicIndex					
		banana1	100	100	Edit   Details   Delete				
	Test for the system shall delete exercises from list. When the doctor or	Test for the system shall delete system exercises from list.  When the doctor or shall food name = "banana2" glycemic index =110 calorie = 110	Test for the system shall delete system exercises from list. When the doctor or nurse click delete.  food name = "banana2" glycemic index =110 calorie = 110  when dele	Test for the system shall delete the sh	Test for the system shall delete the food from the datab glycemic index =110 calorie = 110  When the doctor or nurse click delete.  The system shall delete the food from the datab food_Calories food_GlycemicIndex $ \frac{100}{banana2} = \frac{110}{110} $ When delete $ \frac{100}{banana2} = \frac{110}{110} $ When delete $ \frac{100}{banana2} = \frac{110}{110} $	Test for the system shall delete the food from the database.  shall delete exercises from list. When the doctor or nurse click delete.  The system shall delete the food from the database.  food_name = "banana2"  glycemic index =110  calorie = 110  The system shall delete the food from the database.  food_calories food_GlycemicIndex  banana1 100 100 Edit   Details   Delete  banana2 110  When delete  food_name food_Calories food_GlycemicIndex			

Table 15: Test Case of ST12: Nutritionists can delete food from the database.

Document Name	Diabetes Care -Test Plan-V.0.2.docx	Owner	Jirayu	Page	34 / 83
Document Type	Test Plan	Release Date	1 March 2016	Print Date	1 March 2016

# 3.3.4 Feature#4: Activity management.

**ST13:** Nurse/Doctor can add activity to the database.

Description	Test for <b>UC09</b> : Nurse/Doctor can a	add activity to the database.						
Precondition	Test data must be prepared.							
	Symptom = {" symptom1"}							
	suggestion= {"suggestion1"}							
	ProperExercise ={ "ProperExercise	21"}						
	ExerciseInappropriate = {"Exercise	Inappropriate1"}						
	StepExercise = {StepExercise1"}							
<b>Prerequisites or Test input</b>	Table: Exercise							
	Symptom	Symptom	Symptom1					
	suggestion suggestion Suggestion1							
	ProperExercise ProperExercise ProperExercise1							
	ExerciseInappropriate ExerciseInappropriate1 ExerciseInappropriate1							
	StepExercise StepExercise StepExercise1							
Test Script	1 Nurse/Doctor go to activity ma	anagement page						
	2 Nurse/Doctor click "create new".							
	3 Nurse/Doctor input symptom,	suggestion, ProperExercise, Exercise	eInappropriate, StepExercise.					
	4 Nurse/Doctor click "submit"							

Document Name	Diabetes Care -Test Plan-V.0.2.docx	Owner	Jirayu	Page	35 / 83
Document Type	Test Plan	Release Date	1 March 2016	Print Date	1 March 2016

Case	Description	Input				<b>Expect Output</b>	t			
	Test for the	Symptom = {"	The syst	The system shall record the data on the database.						
	system shall add	symptom1"}	Result:							
	the exercises to	suggestion=	aumptom	ouggostion	DronorEvaraina	Evereieelnennrenriete	StepExercise			
	the database.	{"suggestion1"}	symptom	suggestion	ProperExercise	ExerciseInappropriate	StepExercise			
1	When the doctor	ProperExercise ={	symptom	suggestion	ProperExercise	ExerciseInappropriate	StepExercise	Edit   Details   Delete		
1	or nurse add new	"ProperExercise1"}	aumntam4	aumostion4	ProperExercise1	Everele el mararren rieto 1	StepExercise1	Edit I Dataila I Dalata		
	exercises.	ExerciseInappropriate =	symptom1	suggestion1	ProperExercise I	ExerciseInappropriate1	StepExercise i	Edit   Details   Delete		
		{"ExerciseInappropriate1"}								
		StepExercise =								
		{StepExercise1"}								

**Table 16**: Test Case of **ST13**: Nurse/Doctor can add activity to the database.

Document Name	Diabetes Care -Test Plan-V.0.2.docx	Owner	Jirayu	Page	36 / 83
Document Type	Test Plan	Release Date	1 March 2016	Print Date	1 March 2016

**ST14:** Nurse/Doctor can edit activity on the database.

Description	Test for <b>UC09</b> : Nurse/Doctor can e	edit activity on the database.				
Precondition	Test data must be prepared.					
	Symptom = {" symptom1"}					
	<pre>suggestion= {"suggestion1"}</pre>					
	ProperExercise = { "ProperExercise	21"}				
	ExerciseInappropriate = {"Exercise	Inappropriate1"}				
	StepExercise = {StepExercise1"}					
<b>Prerequisites or Test input</b>	Table: Exercise					
	Symptom	Symptom	Symptom1			
	suggestion	suggestion	Suggestion1			
	ProperExercise	ProperExercise	ProperExercise1			
	ExerciseInappropriate	ExerciseInappropriate	ExerciseInappropriate1			
	StepExercise	StepExercise	StepExercise1			
Test Script	1 Nurse/Doctor go to activity m	anagement page				
	2 Nurse/Doctor click "edit".					
	3 Nurse/Doctor input symptom,	suggestion, ProperExercise, Exerci	seInappropriate, StepExercise.			
	4 Nurse/Doctor click "submit"					

Document Name	Diabetes Care -Test Plan-V.0.2.docx	Owner	Jirayu	Page	37 / 83
Document Type	Test Plan	Release Date	1 March 2016	Print Date	1 March 2016

Case	Description	Input				<b>Expect Outpu</b>	t	
	Test for the system	Symptom = {" symptom2"}	The system shall record the data on the database.					
	shall edit the	<pre>suggestion= {"suggestion2"}</pre>	Result	: Before	edit			
	information of	ProperExercise ={	symptom	suggestion	ProperExercise	ExerciseInappropriate	StepExercise	
	exercises. When the	"ProperExercise2"}	symptom	suggestion	ProperExercise	ExerciseInappropriate	StepExercise	Edit   Details   Delete
	doctor or nurse edit	ExerciseInappropriate =	symptom1	suggestion1	ProperExercise1	ExerciseInappropriate1	StepExercise1	Edit   Details   Delete
1	the information of	{"ExerciseInappropriate1"}						
1	exercises.	StepExercise = {StepExercise1"}	After					
		(StepExereise)	symptom	suggestion	ProperExercise	ExerciseInappropriate	StepExercise	
			symptom	suggestion	ProperExercise	ExerciseInappropriate	StepExercise	Edit   Details   Delete
			symptom2	suggestion2	ProperExercise2	ExerciseInappropriate1	StepExercise1	Edit   Details   Delete

 Table 17: Test Case of ST14: Nurse/Doctor can edit activity on the database.

Document Name	Diabetes Care -Test Plan-V.0.2.docx	Owner	Jirayu	Page	38 / 83
Document Type	Test Plan	Release Date	1 March 2016	Print Date	1 March 2016

**ST15:** Nurse/Doctor can delete activity from the database.

Description	Test for UC09: Nurse/Doctor can de	elete activity from the database.						
Precondition	est data must be prepared.							
	Symptom = {"symptom"," sympton	Symptom = {"symptom"," symptom1"}						
	suggestion= {"suggestion", "sugges	tion1"}						
	ProperExercise = { "ProperExercise"	', "ProperExercise1"}						
	ExerciseInappropriate = {"ExerciseI	nappropriate", "ExerciseInappropri	ate1"}					
	StepExercise = {"StepExercise", "St	epExercise1"}						
<b>Prerequisites or Test input</b>	Table: Exercise							
	Symptom	Symptom	Symptom1					
	suggestion	suggestion	Suggestion1					
	ProperExercise	ProperExercise	ProperExercise1					
	ExerciseInappropriate	ExerciseInappropriate	ExerciseInappropriate1					
	StepExercise	StepExercise	StepExercise1					
Test Script	1 Nurse/Doctor go to activity management page							
	2 Nurse/Doctor select activity.							
	3 Nurse/Doctor clicks "Delete"	·						

Document Name	Diabetes Care -Test Plan-V.0.2.docx	Owner	Jirayu	Page	39 / 83
Document Type	Test Plan	Release Date	1 March 2016	Print Date	1 March 2016

Case	Description	Input				Expect (	Output	
	Test for the system	Symptom = symptom1	The s	system s	hall delete	the activity f	rom the da	itabase.
	shall delete	suggestion= suggestion1			D	F	Char Faranda	
	exercises from list.	ProperExercise =	symptom	suggestion	ProperExercise	Exerciselnappropriate	StepExercise	
		"ProperExercise1"	symptom	suggestion	ProperExercise	ExerciseInappropriate	StepExercise	Edit   Details   Delete
1	nurse click delete.	ExerciseInappropriate = "ExerciseInappropriate1" StepExercise = "StepExercise1"	symptom1	suggestion1	ProperExercise1	ExerciseInappropriat1	StepExercise1	Edit   Details   Delete
_			When	ı delete.				
			symptom	suggestion	ProperExercise	Exerciselnappropriate	StepExercise	
			symptom	suggestion	ProperExercise	ExerciseInappropriate	StepExercise	Edit   Details   Delete

 Table 18: Test Case of ST15: Nurse/Doctor can delete activity from the database.

Document Name	Diabetes Care -Test Plan-V.0.2.docx	Owner	Jirayu	Page	40 / 83
Document Type	Test Plan	Release Date	1 March 2016	Print Date	1 March 2016

# 3.3.5 Feature#5: Activity plan and recommend system.

**ST16:** Patient can select the activity from database.

Description	Test for <b>UC10</b> : Patient can select th	e activity from database.					
Precondition	Test data must be prepared.						
	Symptom = {"symptom"," symptom	Symptom = {"symptom"," symptom1"}					
	suggestion= {"suggestion", "sugges	stion1"}					
	ProperExercise = { "ProperExercise"	", "ProperExercise1"}					
	ExerciseInappropriate = {"ExerciseI	nappropriate", "ExerciseInappropr	iate1"}				
	StepExercise = {"StepExercise", "St	tepExercise1"}					
<b>Prerequisites or Test input</b>	Table: Exercise						
	Symptom	Symptom	Symptom1				
	suggestion	suggestion	Suggestion1				
	ProperExercise	ProperExercise	ProperExercise1				
	ExerciseInappropriate	ExerciseInappropriate	ExerciseInappropriate1				
	StepExercise	StepExercise	StepExercise1				
Test Script	1 Patients go to health plan pag	ge.					
	2 Patient click "Exercises suggestion" link button.						
	3 Patient select symptom.						
	4 Patient must click "select"						
	5 The system re view food nan	ne to patient.					

Document Name	Diabetes Care -Test Plan-V.0.2.docx	Owner	Jirayu	Page	41 / 83
Document Type	Test Plan	Release Date	1 March 2016	Print Date	1 March 2016

Case	Description	Input	Expect Output
	Test for the	Symptom = "symptom",	The system shall provide list of symptom to patient.
	system shall get	"symptom1"	symptom ▼
	list of exercises		
	from table		symptom1 ▼
1	"Exercises" on		
	the database. And		
	display list of		
	exercises to the		
	patient.		
	Test for the	Symptom = "symptom",	The system shall provide suggestion about precautions exercise.
	system shall get		symptom
	exercises		
	suggestion from		suggestion
2	table "Exercises"		ProperExercise
2	on the database.		ExerciseInappropriate
	And display		StepExercise
	exercises		OtopExoroido
	suggestion to the		
	patient.		

 Table 19: Test Case of ST16: Patient can select the activity from database.

Document Name	Diabetes Care -Test Plan-V.0.2.docx	Owner	Jirayu	Page	42 / 83
Document Type	Test Plan	Release Date	1 March 2016	Print Date	1 March 2016

# 3.3.6 Feature#6: Nutrition plan system.

**ST17:** Patient can select the foods from database.

Description	Test for UC12: Patient can select the foods from database.				
Precondition	Test data must be prepared. Food {	Test data must be prepared. Food {"banana1", "banana2"}.			
<b>Prerequisites or Test input</b>	Table: Food				
	food name	banana1	banana2		
	glycemic	100	110		
	calorie	100	110		
Test Script	1 Patients go to health plan page.				
	2 Patient select foods name.				
	3 Patient must click "add"				
	4 The system re view food name to patient.				

Document Name	Diabetes Care -Test Plan-V.0.2.docx	Owner	Jirayu	Page	43 / 83
Document Type	Test Plan	Release Date	1 March 2016	Print Date	1 March 2016

Case	Description	Input	Expect Output	
1	Test for the system shall provide list of food from the database table "Food" to the patient	Food = "banana1", "banana2"	The system shall provide list of food name to patient.  banana1  banana2  T	
2	Test for the system shall add food to table "TodayFood" on the database. And display to the patient.	Food = "banana1"	The system shall display names of food to patient.  banana2  Delete  banana1  Delete	

**Table 20**: Test Case of **ST17**: Patient can select the foods from database.

Docume Name	ent	Diabetes Care -Test Plan-V.0.2.docx	Owner	Jirayu	Page	44 / 83
Docum	ent Type	Test Plan	Release Date	1 March 2016	Print Date	1 March 2016

# 3.3.7 Feature#7: Health monitors system.

**ST18:** Patient can view body mass value that calculated.

Description	Test for <b>UC16</b> : Patient	Test for <b>UC16</b> : Patient can view body mass value that calculated.		
Precondition	Test data must be prepa	red. Height{123}, Weight{2	13}, Gender{"male"}	
<b>Prerequisites or Test input</b>	Table: AspUser	Table: AspUser		
	First name	qwe		
	Last name	wqr		
	Gender	Male		
	Age	123		
	Height	123		
	Weight	213		
Test Script	1. Patient go to a he	ealth plan page		

Document Name	Diabetes Care -Test Plan-V.0.2.docx	Owner	Jirayu	Page	45 / 83
Document Type	Test Plan	Release Date	1 March 2016	Print Date	1 March 2016

Case	Description	Input		<b>Expect Output</b>
	Test for the system	Height = 123	The system redired	ct to health plan page and display BMI.
	shall get the data of	Weight =213	BMR:	2762.7
	height, weight and	Gender = male	- Simu	273211
	gender to calculate		BMI:	140.789212770176
1	BMR and BMI.			
	And the system			
	shall display to the			
	patient.			

 Table 21: Test Case of ST18: Patient can view body mass value that calculated.

Document Name	Diabetes Care -Test Plan-V.0.2.docx	Owner	Jirayu	Page	46 / 83
Document Type	Test Plan	Release Date	1 March 2016	Print Date	1 March 2016

ST19: Patient can record blood sugar.

Description	Test for UC15: Patie	Test for <b>UC15</b> : Patient can record blood sugar.		
Precondition	Test data must be pre	Test data must be prepared. Blood sugar value {120, 110, -100}		
<b>Prerequisites or Test input</b>	Table: UserGlycemic			
	UserId	1		
	Value	120, 110		
Test Script	1. Patient go health page			
	2. Patient input "blood sugar"			
	3. Patient clicks '	"add" button.		

Case	Description	Input	Expect Output
	Test for the system	Blood sugar = 120, 110	The system record value of blood sugar to the database
1	shall record the value		
1	of blood sugar to the		
	database		
	Test when input blood	Blood sugar="-100"	The system display error message "value must not minus"
3	sugar value is equal		
	minus		

 Table 22: Test Case of ST19: Patient can record blood sugar.

Document Name	Diabetes Care -Test Plan-V.0.2.docx	Owner	Jirayu	Page	47 / 83
Document Type	Test Plan	Release Date	1 March 2016	Print Date	1 March 2016

ST20: Patient can view statistic graph of blood sugar.

Description	Test for UC17: Patie	Test for UC17: Patient can view statistic graph of blood sugar.		
Precondition	Test data must be pr	epared. Blood sugar value {120, 110}		
<b>Prerequisites or Test input</b>	Table: UserGlycemic			
	UserId 1			
	Value	120, 110		
Test Script	1. Patient go hea	alth page		

Case	Description	Input	Expect Output
1	Test for the system shall get value of blood sugar to analyze	Blood sugar value = 120 Blood sugar value = 110	The system shall redirect to health plan page and dis play graph to user
	and display graph to the patient.		all clause extremes

Table 23: Test Case of ST20: Patient can view statistic graph of blood sugar.

Document Name	Diabetes Care -Test Plan-V.0.2.docx	Owner	Jirayu	Page	48 / 83
Document Type	Test Plan	Release Date	1 March 2016	Print Date	1 March 2016

**ST21:** Patient can get interpretation about blood sugar from system.

Description	Test for UC18: Pati	Γest for UC18: Patient can get interpretation about blood sugar from system.			
Precondition	Test data must be pr	Γest data must be prepared. Blood sugar value {120, 110}			
<b>Prerequisites or Test input</b>	Table: UserGlycemic				
	UserId	1			
	Value	120, 110			
Test Script	1. Patient go he	alth page			

Case	Description	Input	Expect Output
	Test for the system	Blood sugar value = 120	The system shall redirect to health plan page. And display
	shall get value of	Blood sugar value = 110	interpretation about blood sugar to the patient.
	blood sugar to analyze		
1	and display		
	interpretation to the		
	patient.		

Table 24: Test Case of ST21: Patient can get interpretation about blood sugar from system.

Document Name	Diabetes Care -Test Plan-V.0.2.docx	Owner	Jirayu	Page	49 / 83
Document Type	Test Plan	Release Date	1 March 2016	Print Date	1 March 2016

# **3.3.8** Feature#8: Behavior monitors system.

**ST22:** Patient can select food.

Description	Test for <b>UC19</b> : Patient can select f	Test for <b>UC19</b> : Patient can select food.			
Precondition	Test data must be prepared. Food {"banana1", "banana2"}.				
<b>Prerequisites or Test input</b>	Table: Food				
	food name banana1 banana2				
	glycemic 100 110				
	calorie	100	110		
Test Script	1 Patients go to behavior page.				
	2 Patient select foods name.				
	3 Patient must click "select"				
	4 The system review food name to patient.				
		<del>-</del>			

Document Name	Diabetes Care -Test Plan-V.0.2.docx	Owner	Jirayu	Page	50 / 83
Document Type	Test Plan	Release Date	1 March 2016	Print Date	1 March 2016

**Test Case: Edit** 

Case	Description	Input	Expect Output
1	Test for the system shall provide list of food from the database table "Food" to the patient	Food = "banana1", "banana2"	The system shall provide list of food name to patient.  Eat Today  All Food  banana1
2	Test for the system shall add food to table "TodayFood" on the database. And display to the patient.	Food = "banana1"	The system shall display names of food to patient.  All Food  — Select — Add  Name Calories  banana1 100 Remove  banana1 100 Remove

Table 25: Test Case of ST22: Patient can select food.

Document Name	Diabetes Care -Test Plan-V.0.2.docx	Owner	Jirayu	Page	51 / 83
Document Type	Test Plan	Release Date	1 March 2016	Print Date	1 March 2016

**ST23:** Patient can submit and view about analyze result.

Description	Test for UC21: Patient can submit	Test for UC21: Patient can submit and view about analyze result.			
Precondition	Test data must be prepared. Food ("banana1", "banana2"), Medicine ("para1", "para2")				
<b>Prerequisites or Test input</b>	Table: Food				
	food name banana1 banana2				
	glycemic 100 110				
	calorie 100 110				
	Table: Medicine				
	name	para	para2		
Test Script	1 Patients go to behavior page	2.			
	2 Patient select data.				
	3 Patient must click submit.				

**Test Case: Edit** 

Cas	e Description	Input	Expect Output
1	Test for the system shall display analyze result to the patient. After the patient submit the data.	Food = banana1 Food = banana2 Medicine = para Medicine = para2	The system shall provide analyze result to patient.  Status Today

Table 26: Test Case of ST23: Patient can submit and view about analyze result.

Document Name	Diabetes Care -Test Plan-V.0.2.docx	Owner	Jirayu	Page	52 / 83
Document Type	Test Plan	Release Date	1 March 2016	Print Date	1 March 2016

# 3.4 Unit Test

Black Box technique (UT01-UT11)

<b>Use Case ID</b>	Use case name
UT01	AccountController: AddTodayFood()
UT02	AccountController: addTodayfood()
UT03	AccountController: RemoveTodayFood()
UT04	AccountController: HealthPlan()
UT05	AccountController: AddGlycemic()
UT06	AccountController: addUserGlycemic()
UT07	AccountController: AddFood()
UT08	AccountController: addUSerFood()
UT09	AccountController: DeleteFood()
UT10	AccountController: AddMedicine()
UT11	AccountController: addMedicineData()
UT12	AccountController: DeleteMedicine()
UT13	AccountController: EditUserProfile()
UT14	AccountController: Register()
UT15	AccountController: Login()
UT16	AccountController:LogOff()
UT17	FoodController:Create()
UT18	FoodController:Edit()
UT19	FoodController:Delete()
UT20	ExercisesController:Create()
UT21	ExercisesController: Edit()
UT22	ExercisesController:Delete()
UT23	ManageController ChangePassword()

Table 26: Black Box technique (UT01-UT11)

Document Name	Diabetes Care -Test Plan-V.0.2.docx	Owner	Jirayu	Page	53 / 83
Document Type	Test Plan	Release Date	1 March 2016	Print Date	1 March 2016

## 3.4.1 Class: AccountController

**UT01:** public ActionResult AddTodayFood(int value = 0)

**Description:** Test for the method AddTodayFood has redirected is correct after process success.

**Test Case:** AddTodayFood()

Case	Description	Input	Expect Output
1	Tests method AddTodayFood, test return type.	value = 1	Object of class ActionResult
2	Tests method redirects result	value = 1	Redirect to Account

Table 27: Test Case of UT01

Document Name	Diabetes care -Test Plan-V.0.1.docx	Owner	Jirayu	Page	54 / 83
Document Type	Test Plan	Release Date	4 March 2016	Print Date	4 March 2016

**UT02**: public virtual TodayFood addTodayfood(UserProfileInfo profile, int value = 0)

**Description:** Test for insert the data of food to table TodayFood on the database. This AddTodayFood function uses to add food to analyze select food by food id

•

**Test Case:** addValueTodayfood()

Case	Description	Input	<b>Expected Output</b>
	Test for insert user_id and	UserProfileInfo {UserId = 1}, value = 1	TodayFood {User Id = 1,
1	food_id to method		Food $id = 1$ }
1	addTodayfood.		
	(Assert.IsNotNull)		

Table 28: Test Case of UT02

**Test Case:** addNullValueTodayfood()

Case	Description	Input	<b>Expect Output</b>
	Test for insert user_id and	null	null
	food_id to method		
1	addTodayfood. When value		
	equal null.		
	(Assert.IsNull)		

Table 29: Test Case of UT02

Document Name	Diabetes care -Test Plan-V.0.1.docx	Owner	Jirayu	Page	55 / 83
Document Type	Test Plan	Release Date	4 March 2016	Print Date	4 March 2016

**UT03:** public ActionResult RemoveTodayFood(int value = 0)

**Description:** Test for the method RemoveTodayFood has redirected is correct after delete process success.

**Test Case:** RemoveTodayFoodTest()

Case	Description	Input	<b>Expect Output</b>
	Tests method	value = 1	Object of class ActionResult
1	RemoveTodayFood, test return		
	type.		
2	Tests method redirects result	value =1	Redirect to Account

Table 30: Test Case of UT03

Document Name	Diabetes care -Test Plan-V.0.1.docx	Owner	Jirayu	Page	56 / 83
Document Type	Test Plan	Release Date	4 March 2016	Print Date	4 March 2016

## **UT04:** public ActionResult HealthPlan()

**Description :** Test for calculate the BMI, BMR, Food calorie, Food glycemic value This HealthPlan function uses to calculate by height, weight, age, food calorie, food glycemic,

**Test Case:** HealthPlanTest()

Case	Description	Input		<b>Expected Output</b>			
		Height	Weight	Age	Food calorie	Food glycemic	
	Test to calculate BMI	1	1	-	-	-	BMI = 10000
1	by input user height,						
	user weight						
	Test to calculate BMR	1	1	1	-	-	BMRM =77.9
2	for male by input user						
	height, user weight, and						
	user age.						
	Test to calculate BMR	1	1	1	-	-	BMRF =671.7
3	for female by input user						
	height, user weight, and						
	user age.						

Document Name	Diabetes care -Test Plan-V.0.1.docx	Owner	Jirayu	Page	57 / 83
Document Type	Test Plan	Release Date	4 March 2016	Print Date	4 March 2016

		Test to calculate calorie	-	-	-	100,100,200	-	Calorie = 400
4	1	of food						
5	<del></del>	Test to calculate	-	-	-	-	100,100,200	Glycemic = 400
٦	)	glycemic index of food						

Table 31: Test Case of UT04

Document Name	Diabetes care -Test Plan-V.0.1.docx	Owner	Jirayu	Page	58 / 83
Document Type	Test Plan	Release Date	4 March 2016	Print Date	4 March 2016

**UT05:** public ActionResult AddGlycemic(int value = 0)

**Description:** Test for the method AddGlycemic has redirected is correct after process success.

**Test Case:** AddGlycemic()

Case	Description	Input	<b>Expect Output</b>
1	Test returns type of the method AddGlycemic().	value = 0	Object of class ActionResult
3	Tests method redirects result.	value = 0	Redirect to HealthPlan

Table 32: Test Case of UT05

Document Name	Diabetes care -Test Plan-V.0.1.docx	Owner	Jirayu	Page	59 / 83
Document Type	Test Plan	Release Date	4 March 2016	Print Date	4 March 2016

**UT06:** public virtual UserGlycemic addUserGlycemic (UserProfileInfo profile, int value = 0)

**Description:** Test for insert the value of blood sugar to table UserGlycemic on the database. This AddGlycemic function uses to add blood sugar value to system include data user id, blood value, and date.

**Test Case:** addValueAddGlycemic()

Case	Description	Input	Expect Output
	Test for insert user_id and	UserProfileInfo{id=1}, value = 100	UserGlycemic{User id =1,
1	value to method		Value =100}
1	addUserGlycemic.		
	(Assert.IsNotNull)		

Table 33: Test Case of UT06

**Test Case:** addNullValueAddGlycemic()

Case	Description	Input	Expect Output
	Test for insert user_id and	null	null
	value to method		
1	addUserGlycemic.		
	When value equal null.		
	(Assert.IsNull)		

Table 34: Test Case of UT06

Document Name	Diabetes care -Test Plan-V.0.1.docx	Owner	Jirayu	Page	60 / 83
Document Type	Test Plan	Release Date	4 March 2016	Print Date	4 March 2016

**UT07:** public ActionResult AddFood(int id = 0)

**Description:** Test for the method AddFood has redirected is correct after process success.

**Test Case:** AddFood()

Case	Description	Input	<b>Expect Output</b>
1	Test returns type of the	id = 0	Object of class ActionResult
1	method.		
	Tests method redirects result,	Id = 0	Redirect to HealthPlan
2	when redirects correct url		
	"/Account/HealthPlan"		

Table 35: Test Case of UT07

Document Name	Diabetes care -Test Plan-V.0.1.docx	Owner	Jirayu	Page	61 / 83
Document Type	Test Plan	Release Date	4 March 2016	Print Date	4 March 2016

**UT08:** public virtual UserFood addUSerFood(UserProfileInfo profile, int id = 0)

**Description:** Test for method inserts the data of food to table UserFood on the database. This addUSerFood method uses to select food from database Food to add in database UserFood table.

**Test Case:** addValue AddFood()

Case	Description	Input	Expect Output
	Test for insert user_id and	UserProfileInfo{id=1},id = 1	UserFood{User id = $1$ ,
1	food_id to method		Food Id =1}
1	addUSerFood.		
	(Assert.IsNotNull)		

Table 36: Test Case of UT08

**Test Case:** addNullValue AddFood()

Case	Description	Input	Expect Output
	Test for insert user_id and	null	null
	food_id to method		
1	addUSerFood.		
	When value equal null.		
	(Assert.IsNull)		

Table 37: Test Case of UT08

Document Name	Diabetes care -Test Plan-V.0.1.docx	Owner	Jirayu	Page	62 / 83
Document Type	Test Plan	Release Date	4 March 2016	Print Date	4 March 2016

**UT09:** public ActionResult DeleteFood(int id = 0)

**Description:** Test for the method DeleteFood has redirected is correct after delete process is success.

**Test Case:** DeleteFoodTest()

Case	Description	Input	Expect Output
1	Test returns type of the	id = 0	Object of class ActionResult
1	method.		
2	Tests method redirects result,	id = 0	Redirect to HealthPlan
	when redirects correct url.		

Table 38: Test Case of UT09

Document Name	Diabetes care -Test Plan-V.0.1.docx	Owner	Jirayu	Page	63 / 83
Document Type	Test Plan	Release Date	4 March 2016	Print Date	4 March 2016

**UT10:** public ActionResult AddMedicine(string name = null)

**Description:** Test for the method AddMedicine has redirected is correct after process success.

Test Case: AddMedicine()

Case	Description	Input	<b>Expect Output</b>
1	Test returns type of the	name = "para"	Object of class ActionResult
1	method.		
2	Tests method redirects result,	name = "para"	Redirect to DisplayProfile
2	when redirects correct url.		

Table 39: Test Case of UT10

Document Name	Diabetes care -Test Plan-V.0.1.docx	Owner	Jirayu	Page	64 / 83
Document Type	Test Plan	Release Date	4 March 2016	Print Date	4 March 2016

**UT11:** public virtual Medicine addMedicineData (UserProfileInfo profile , string name = null)

**Description**: Test for method inserts the data of medicine to table Medicine on the database.

**Test Case:** addValueAddMedicine()

Case	Description	Input	Expect Output
1	Test for insert user_id and name to method addMedicineData.  (Assert.IsNotNull)	UserProfileInfo{id=1}, name= "para"	Medicine {User id =1, MedicineName = "para"}

Table 40: Test Case of UT11

Document Name	Diabetes care -Test Plan-V.0.1.docx	Owner	Jirayu	Page	65 / 83
Document Type	Test Plan	Release Date	4 March 2016	Print Date	4 March 2016

**Test Case:** addNullValueAddMedicine()

Case	Description	Input	<b>Expect Output</b>
	Test for insert user_id and	null	null
	name to method		
1	addMedicineData.		
	When value equal null.		
	(Assert.IsNull)		
	Test for insert user_id and	UserProfileInfo{id=1}, name= null	null
	name to method		
2	addMedicineData.		
	When name equal null.		
	(Assert.IsNull)		

Table 41: Test Case of UT11

Document Name	Diabetes care -Test Plan-V.0.1.docx	Owner	Jirayu	Page	66 / 83
Document Type	Test Plan	Release Date	4 March 2016	Print Date	4 March 2016

## **UT12:** public ActionResult DeleteMedicine(int id = 0)

**Description:** Test for the method DeleteMedicine has redirected is correct after delete process is success.

Test Case: DeleteMedicineTest()

Case	Description	Input	<b>Expect Output</b>
1	Test returns type of the	id = 0	Object of class ActionResult
1	method.		
2	Tests method redirects result,	id = 0	Redirect to DisplayProfile
2	when redirects correct url.		

Table 42: Test Case of UT12

Document Name	Diabetes care -Test Plan-V.0.1.docx	Owner	Jirayu	Page	67 / 83
Document Type	Test Plan	Release Date	4 March 2016	Print Date	4 March 2016

## **UT13:** public ActionResult EditUserProfile(UserProfileInfo userprofile)

**Description:** Test for method EditUserProfile has redirected is correct after delete process method edit/update the information of user in the table AspNetUser on the database is success.

**Test Case:** EditUserProfileTest(UserProfileInfo userprofile)

Case	Description	Input	<b>Expect Output</b>
1	Test returns type of the	$UserProfileInfo\{id = 1\}$	Object of class ActionResult
1	method.		
2	Tests method redirects result,	UserProfileInfo{id = 1}	Redirect to Index
2	when redirects correct url.		

Table 43: Test Case of UT13

Document Name	Diabetes care -Test Plan-V.0.1.docx	Owner	Jirayu	Page	68 / 83
Document Type	Test Plan	Release Date	4 March 2016	Print Date	4 March 2016

## **UT14:** public async Task<ActionResult> Register(RegisterViewModel model)

**Description:** Test for method register user account to insert data to the database. This Register method uses to register user account to the system for using the system.

**Test Case:** RegisterTestValid()

Case	Description	Input	<b>Expect Output</b>
	Test for insert username, email, first	RegisterViewModel{Username = "atom",	Register {Username = "atom"
	name, last name gender, age, height,	Email = "at@om.com", FirstName = "Jira",	Email = "at@om.com"
	weight, address, city, zipcode to	LastName = "yu", Gender= "male", Age = 20	FirstName = "Jira"
	method Register.	, Height = 170 , Weight = 60 , Address =	LastName = "yu"
	(Assert.IsNotNull)	"139 m.7", City = "Phayao", Zipcode =	Gender= "male"
1		56000;}	Age = 20
1			Height = 170
			Weight = 60
			Address = "139 m.7"
			City = "Phayao"
			Zipcode = 56000}

Table 44: Test Case of UT14

Document Name	Diabetes care -Test Plan-V.0.1.docx	Owner	Jirayu	Page	69 / 83
Document Type	Test Plan	Release Date	4 March 2016	Print Date	4 March 2016

**Test Case:** RegisterTestInValid()

Case	Description	Input	Expect Output
	Test for insert username, email, first	null	Null
	name, last name gender, age, height,		
1	weight, address, city, zipcode to method		
	Register. When value equal null.		
	(Assert.IsNull)		
	Test for insert username, email, first	RegisterViewModel{Username = null,	Null
	name, last name gender, age, height,	Email = "at@om.com", FirstName = null,	
2	weight, address, city, zipcode to method	LastName = null, Gender= "male", Age =	
2	Register. When some value equal null.	20, Height = $170$ , Weight = $60$ , Address	
	(Assert.IsNull)	= "139 m.7", City = "Phayao", Zipcode	
		= 56000;}	

Table 45: Test Case of UT14

Document Name	Diabetes care -Test Plan-V.0.1.docx	Owner	Jirayu	Page	70 / 83
Document Type	Test Plan	Release Date	4 March 2016	Print Date	4 March 2016

**UT15:** public async Task<ActionResult> Login(LoginViewModel model, string returnUrl)

**Description**: Test for method login system. This Login method uses to login to using the system.

**Test Case:** LoginValid()

Case	Description	Input	<b>Expect Output</b>
	Test for insert username,	LoginViewModel {Username = "Admin@Admin.com",	Login {Username =
1	password method Login.	Password = "Password#1"}	"Admin@Admin.com"
	(Assert.IsNotNull)		Password = "Password#1"}

Table 46: Test Case of UT15

**Test Case:** LoginInValid()

Case	Description	Input	Expect Output
	Tests method Login, when	null	null
1	input value is null		
	(Assert.IsNull)		

Table 47: Test Case of UT15

Document Name	Diabetes care -Test Plan-V.0.1.docx	Owner	Jirayu	Page	71 / 83
Document Type	Test Plan	Release Date	4 March 2016	Print Date	4 March 2016

## **UT016:** public ActionResult LogOff()

**Description:** Test for the method Logoff has redirected is correct after user logoff from the system.

Test Case: LogOffTest()

Case	Description	Input	Expect Output
1	Tests method redirects result,	دد »،	Redirect to Home/Index
1	when redirects correct url.		

Table 48: Test Case of UT16

Document Name	Diabetes care -Test Plan-V.0.1.docx	Owner	Jirayu	Page	72 / 83
Document Type	Test Plan	Release Date	4 March 2016	Print Date	4 March 2016

#### 3.4.2 Class: FoodController

**UT017:** public ActionResult Create([Bind(Include = "id\_food,food\_name,food\_Calories,food\_GlycemicIndex")] Food food)

**Description:** Test for insert the foods to table Food on the database. This Create function uses to add foods info to the system include data id, name, calories, glycemic index.

**Test Case:** CreateValid()

Case	Description		Input	<b>Expect Output</b>
	Test for insert food_id, food_name,	Bind{	id_food =1,	Create {id_food =1,
1	food_calories and food_Glycemic to		food_name = "banana",	food_name = "banana",
1	method Create.		food _calories = 100,	food _calories = 100,
	(Assert.IsNotNull)		food _Glucemic =100}	food _Glucemic =100}

Table 49: Test Case of UT17

Document Name	Diabetes care -Test Plan-V.0.1.docx	Owner	Jirayu	Page	73 / 83
Document Type	Test Plan	Release Date	4 March 2016	Print Date	4 March 2016

**Test Case:** CreateInValid()

Case	Description		Input	Expect Output
1	Tests input value equal null. (Assert.IsNull)		null	null
2	Tests input some value equal null. (Assert.IsNull)	Bind{	id_food =1, food_name = null food_calories = 100, food_Glucemic =100}	null

Table 50: Test Case of UT17

Document Name	Diabetes care -Test Plan-V.0.1.docx	Owner	Jirayu	Page	74 / 83
Document Type	Test Plan	Release Date	4 March 2016	Print Date	4 March 2016

**UT018:** public ActionResult Edit([Bind(Include = "id\_food,food\_name,food\_Calories,food\_GlycemicIndex")] Food food)

**Description:** Test for edit the foods in table Food on the database. This Edit function uses to edit foods info to the system include data id, name, calories, glycemic index.

**Test Case:** EditfoodValid ()

Case	Description		Input	<b>Expect Output</b>
	Test for insert food_id,	Bind{	id_food =1,	Edit{id_food =1,
	food_name, food_calories		food_name = "banana2",	food_name = "banana2",
1	and food_Glycemic to		food _calories = 101,	food _calories = 101,
	method Create.		food _Glucemic =101}	food _Glucemic =101}
	(Assert.IsNotNull)			

Table 51: Test Case of UT18

Test Case: EditfoodInValid ()

Case	Description	Input	<b>Expect Output</b>
1	Tests input value equal null.	null	null
1	(Assert.IsNull)		

Table 52: Test Case of UT18

Document Name	Diabetes care -Test Plan-V.0.1.docx	Owner	Jirayu	Page	75 / 83
Document Type	Test Plan	Release Date	4 March 2016	Print Date	4 March 2016

## **UT019:** public ActionResult DeleteConfirmed(int id)

**Description:** Test for the method DeleteConfirmed has redirected is correct after delete process is success.

Test Case: DeleteTest()

Case	Description	Input	Expect Output
1	Test returns type of the	id = 1	Object of class ActionResult
1	method.		
2	Tests method redirects result,	id = 1	Redirect to Index
2	when redirects correct url.		

Table 53: Test Case of UT19

Document Name	Diabetes care -Test Plan-V.0.1.docx	Owner	Jirayu	Page	76 / 83
Document Type	Test Plan	Release Date	4 March 2016	Print Date	4 March 2016

#### 3.4.3 Class: ExerciseController

**UT020:** public ActionResult Create([Bind(Include =

 $"id, symptom, suggestion, Proper Exercise, Exercise In appropriate, Step Exercise")]\ Exercises\ exercises)$ 

**Description:** Test for insert the exercises to table Exercises on the database. This Create function uses to add exercises info to the system include data id, symptom, suggestion, proper exercise, exercise inappropriate, step exercise.

**Test Case:** CreateExercisesValid()

Case	Description	Input	<b>Expect Output</b>
	Test for insert id,	Bind {id =1, symptom = "knee pain", suggestion=	Create {id =1
	symptom,suggestion,ProperExercise,ExerciseIna	"should not take him", ProperExercise, = "exercise",	Symptom = "knee pain"
	ppropriate,StepExercise to method Create.	ExerciseInappropriate = "high impact events",	suggestion= "should not
	(Assert.IsNotNull)	StepExercise = "1. Warm 2. Exercise";}	take him"
1			ProperExercise, =
1			"exercise"
			ExerciseInappropriate =
			"high impact events"
			StepExercise = "1. Warm
			2. Exercise"}

Table 54: Test Case of UT20

Document Name	Diabetes care -Test Plan-V.0.1.docx	Owner	Jirayu	Page	77 / 83
Document Type	Test Plan	Release Date	4 March 2016	Print Date	4 March 2016

**Test Case:** CreateExercisesInValid()

Case	Description	Input	<b>Expect Output</b>
1	Tests input value equal null.	null	null
1	(Assert.IsNull)		
	Tests input some value equal null.	Bind {id =1, symptom = "knee pain",	null
2	(Assert.IsNull)	suggestion= "should not take him",	
2		ProperExercise, = null, ExerciseInappropriate =	
		<pre>null, StepExercise = null;}</pre>	

 Table 55: Test Case of UT20

Document Name	Diabetes care -Test Plan-V.0.1.docx	Owner	Jirayu	Page	78 / 83
Document Type	Test Plan	Release Date	4 March 2016	Print Date	4 March 2016

#### **UT021:** public ActionResult Edit([Bind(Include =

"id,symptom,suggestion,ProperExercise,ExerciseInappropriate,StepExercise")] Exercises exercises)

**Description:** Test for edit the exercises in table Exercises on the database. This Edit function uses to edit exercises info to the system include data id, symptom, suggestion, proper exercise, exercise inappropriate, step exercise.

Test Case: EditExercisesValid ()

Case	Description	Input	<b>Expect Output</b>
	Test for insert id,	Bind {id =1, symptom = "knee pain", suggestion=	Edit {id = 1, symptom =
	symptom, suggestion, ProperExercise, Exercis	"should not take him", ProperExercise, = "exercise",	"knee pain"
	eInappropriate,StepExercise to method Edit.	ExerciseInappropriate = "high impact events",	suggestion="should not
	(Assert.IsNotNull)	StepExercise = "1. Warm 2. Exercise";}	take him"
1			ProperExercise, =
1			"exercise"
			ExerciseInappropriate =
			"high impact events"
			StepExercise = "1. Warm
			2. Exercise"}

Table 56: Test Case of UT21

Document Name	Diabetes care -Test Plan-V.0.1.docx	Owner	Jirayu	Page	79 / 83
Document Type	Test Plan	Release Date	4 March 2016	Print Date	4 March 2016

**Test Case:** EditExercisesInValid ()

Case	Description	Input	<b>Expect Output</b>
2	Tests input value equal null.	null	null
2	(Assert.IsNull)		

Table 57: Test Case of UT21

Document Name	Diabetes care -Test Plan-V.0.1.docx	Owner	Jirayu	Page	80 / 83
Document Type	Test Plan	Release Date	4 March 2016	Print Date	4 March 2016

## **UT022:** public ActionResult DeleteConfirmed(int id)

**Description:** Test for the method DeleteConfirmed has redirected is correct after delete process is success.

**Test Case:** DeleteExercisesTest()

Case	Description	Input	Expect Output
1	Tests method redirects result,	id =1	Redirect to Index
1	when redirects correct url.		

Table 58: Test Case of UT22

Document Name	Diabetes care -Test Plan-V.0.1.docx	Owner	Jirayu	Page	81 / 83
Document Type	Test Plan	Release Date	4 March 2016	Print Date	4 March 2016

# 3.4.4 Class: ManageController

UT023: public ActionResult ChangePassword(ChangePasswordViewModel model)

**Description:** Test for change password of user. This ChangePassword method uses to change password for login to the system include new password, old password, confirm password.

**Test Case:** ChangePasswordTest()

Case	Description	Input	<b>Expect Output</b>
	Tests method redirects result,	ChangePasswordViewModel {NewPassword =	Redirect to Index
1	when redirects correct url.	"testpassword", OldPassword = "testpass",	
		ConfirmPassword = "testpassword"}	
	Test returns type of the	ChangePasswordViewModel {NewPassword =	Object of class ActionResult
2	method.	"testpassword", OldPassword = "testpass",	
		ConfirmPassword = "testpassword"}	

Table 59: Test Case of UT23

Document Name	Diabetes care -Test Plan-V.0.1.docx	Owner	Jirayu	Page	82 / 83
Document Type	Test Plan	Release Date	4 March 2016	Print Date	4 March 2016

**Test Case:** ChangePasswordValid ()

Case	Description	Input	<b>Expect Output</b>
	Test for insert new	ChangePasswordViewModel {NewPassword =	ChangePassword {New password =
	password, old password and	"testpassword", OldPassword = "testpass",	"testpassword"
1	confirm password to method	ConfirmPassword = "testpassword"}	Old password = "testpass"
	ChangePassword		Confirm password = "testpassword"}
	(Assert.IsNotNull)		

Table 60: Test Case of UT23

**Test Case:** ChangePasswordInValid ()

C	ase	Description	Input	<b>Expect Output</b>
	1	Tests input value equal null.	null	null
	1	(Assert.IsNull)		

Table 61: Test Case of UT23

Document Name	Diabetes care -Test Plan-V.0.1.docx	Owner	Jirayu	Page	83 / 83
Document Type	Test Plan	Release Date	4 March 2016	Print Date	4 March 2016