**Dental Clinic Services System**

Final Progress

Test Plan

By

**Kanokwan Maneerat 542115002**

**Worapun Wongkium 542115055**

Department of Software Engineering

College of Arts, Media, and Technology

Chiang Mai University

Project Adviser

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Miss Pattama Longanee

**Test Plan**

**Revision Table**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Document Name** | **Detail** | **Status** | **Date** | **Viewable** | **Reviewer& Responsible** |
| DCSS - Test plan - ver 0.1 | Add all document | Draft |  | Adviser | Kanokwan&Worapun |
| DCSS - Test plan - ver 0.2 | Edit the unit testing and system testing | Draft |  | Adviser | Kanokwan&Worapun |
| DCSS - Test plan - ver 0.3 | Edit the unit testing and system testing | Draft |  | Adviser | Kanokwan&Worapun |
| DCSS - Test plan - ver 0.4 | Edit the unit testing and system testing | Draft |  | Adviser | Kanokwan&Worapun |
| DCSS - Test plan - ver 0.5 | Edit the unit testing and system testing | Draft |  | Adviser | Kanokwan&Worapun |
| DCSS - Test plan - ver 0.6 | Edit the unit testing and system testing | Draft |  | Adviser | Kanokwan&Worapun |
| DCSS - Test plan - ver 0.7 | Double check all document | Draft |  | Adviser | Kanokwan&Worapun |
| DCSS - Test plan - ver 0.8 | Edit the unit testing and system testing | Draft |  | Adviser | Kanokwan&Worapun |
| DCSS - Test plan - ver 0.9 | Edit unit testing and appendix | Draft |  | Adviser | Kanokwan&Worapun |
| DCSS - Test plan - ver 1.0 | Add more test cases and double check | Release |  | Adviser | Kanokwan&Worapun |
| DCSS - Test plan - ver 1.1 | Edit some information | Release |  | Adviser | Kanokwan&Worapun |

Table of Contents

Chapter One: Introduction 6

1. Introduction 6

1.1 Objectives 6

1.2 Purpose 6

1.3 Project Scope 7

1.4 User Classes and Characteristics 7

1.5 Definition and Acronyms 8

1.6 Acronyms and Definitions Acronyms 9

Chapter Two: Test Plan and Test Procedure 10

2. Test Plan and Test Procedure 10

2.1 Test Objectives 10

2.2 Scope 10

2.3 Test Duration 10

2.4 Test Responsibility 11

2.5 Test Strategy 11

2.6 Result of Testing 11

2.7 Test Environment 12

Chapter Three: Unit Testing 14

3. Unit Testing of Web Application 14

Unit Test Case 01 (UTC-01): testp\_login() 14

Unit Test Case 02 (UTC-02): testp\_ownCalendar() 15

Unit Test Case 03 (UTC-03): testd\_login() 16

Unit Test Case 04 (UTC-04): testd\_ownCalendar() 17

Unit Test Case 05 (UTC-05): testo\_login() 18

Unit Test Case 06 (UTC-06): testget\_p\_db() 19

Unit Test Case 07 (UTC-07): testget\_db() 20

Unit Test Case 08 (UTC-08): testp\_register() 21

Unit Test Case 09 (UTC-09): testp\_edit() 22

Unit Test Case 10 (UTC-10): testp\_delete() 23

Unit Test Case 11 (UTC-11): testd\_register() 23

Unit Test Case 12 (UTC-12): testd\_edit() 24

Unit Test Case 13 (UTC-13): testd\_delete() 25

Unit Test Case 14 (UTC-14): testview\_appointment() 26

Unit Test Case 15 (UTC-15): testmake\_appointment() 27

Unit Test Case 16 (UTC-16): testapp\_edit() 28

Unit Test Case 17 (UTC-17): testapp\_delete() 29

Unit Test Case 18 (UTC-18): testPatientCalendar() 30

Unit Test Case 19 (UTC-19): testDentistCalendar() 31

Unit Test Case 20 (UTC-20): testp\_edit\_data() 32

Unit Test Case 21 (UTC-21): testd\_edit\_data() 33

Unit Test Case 22 (UTC-22): testd\_edit\_data() 34

Class: PatientMobile\_manage 35

Unit Test Case 23 (UTC-23): testMobileLogin() 35

Unit Test Case 24 (UTC-24): testOwnAppointmentMobile() 36

Unit Test Case 25 (UTC-25): testp\_getAllTreatment() 37

Unit Test Case 26 (UTC-26): testp\_calculate() 37

Unit Test Case 27 (UTC-27): testp\_get\_information() 38

Unit Test Case 28 (UTC-28): testp\_viewFollowUpByQid() 40

Unit Test Case 29 (UTC-29): testp\_answer\_data() 40

Unit Test Case 30 (UTC-30): testp\_view\_follow\_up() 41

Unit Test Case 31 (UTC-31): testp\_view\_detail() 42

Unit Test Case 32 (UTC-32): testd\_get\_p\_db() 44

Unit Test Case 33 (UTC-33): testd\_follow\_up() 45

Unit Test Case 34 (UTC-34): testd\_follow\_up\_data() 46

Unit Test Case 35 (UTC-35): testd\_getAllReply() 46

Unit Test Case 36 (UTC-36): testo\_addDentalTreatment() 47

Unit Test Case 37 (UTC-37): testo\_editDentalTreatment() 48

Unit Test Case 38 (UTC-38): testo\_editDentalTreatment\_data() 49

Unit Test Case 39 (UTC-39): testo\_deleteDentalTreatment() 49

Unit Test Case 40 (UTC-40): testo\_get\_treatment() 50

Unit Test Case 41 (UTC-41): testo\_add\_information() 51

Unit Test Case 42 (UTC-42): testo\_edit\_info() 52

Unit Test Case 43 (UTC-43): testo\_edit\_info\_data() 53

Unit Test Case 44 (UTC-44): testo\_delete\_info() 54

Unit Test Case 45 (UTC-45): testo\_get\_information() 54

Unit Test Case 46 (UTC-46): testo\_check\_time() 55

Unit Test Case 47 (UTC-47): testo\_add\_queue() 56

Unit Test Case 48 (UTC-48): testo\_reset\_queue() 57

Unit Testing of Mobile Application 58

Unit Test Case 49 (UTC-49): testpMobile\_getAllTreatment() 58

Unit Test Case 50 (UTC-50): testpMobile\_calculate() 58

Unit Test Case 51 (UTC-51): testpMobile\_get\_information() 59

Unit Test Case 52 (UTC-52): testpMobile\_viewFollowUpByQid() 60

Unit Test Case 53 (UTC-53): testpMobile\_answer\_data() 61

Unit Test Case 54 (UTC-54): testpMobile\_view\_follow\_up() 62

Unit Test Case 55 (UTC-55): testpMobile\_view\_detail() 63

Chapter Four: System Testing 65

4. System testing 65

System Test Case 01(STC-01): Patient can view own appointment in mobile application 65

System Test Case 02(STC-02): Patient can view all appointments from Google calendar in mobile application 66

System Test Case 03(STC-03): Patient can view own appointment in website 67

System Test Case 04(STC-04): Dentist can view own appointment in website 68

System Test Case 05(STC-05): Visitor can view all appointment from Google calendar in website 69

System Test Case 06(STC-06): Officer can view all appointments as a list in website 70

System Test Case 07(STC-07): Officer can view patients’ appointment as a list in website 71

System Test Case 08(STC-08): Officer can view dentists’ appointment as a list in website 72

System Test Case 09(STC-09): Patient can view all appointments from Google calendar in the website 73

System Test Case 10(STC-10): Officer can create new appointment 74

System Test Case 11(STC-11): Officer can edit the appointment 77

System Test Case 12(STC-12): Officer can delete the appointment 79

System Test Case 13(STC-13): Officer can save the appointment to Google calendar 80

System Test Case 14(STC-14): Patient can login to the mobile application 81

System Test Case 15(STC-15): Patient can login to the website 82

System Test Case 16(STC-16): Patient can logout from the mobile application 83

System Test Case 17(STC-017): Patient can logout from the website 83

System Test Case 18(STC-18): Officer can login into the websites 84

System Test Case 19(STC-19): Officer can logout from website 85

System Test Case 20(STC-20): Officer can create an account for new patient 86

System Test Case 21(STC-21): Officer can edit patients’ information 88

System Test Case 22(STC-22): Officer can delete patients’ account 90

System Test Case 23(STC-23): Officer can create dentist account 91

System Test Case 24(STC-24): Officer can edit dentists’ information 93

System Test Case 25(STC-25): Officer can delete dentists’ account 95

System Test Case 26(STC-26): Officer can view all patients’ account as a list 96

System Test Case 27(STC-27): Officer can view all dentists’ account as a list 97

System Test Case 28(STC-28): Officer can view all appointments from Google calendar in website 98

System Test Case 29(STC-29): Dentist login to the website 99

System Test Case 30(STC-30): Dentist logout from the website 100

System Test Case 31(STC-31): Dentist can view all appointments from Google calendar in website 101

System Test Case 32(STC-32): Patient can view their QR code from mobile application 102

System Test Case 33(STC-33): Patient can view their QR code from website 102

System Test Case 34(STC-34): Officer can add patient to a queue 103

System Test Case 35(STC-35): Reset queue 104

System Test Case 36(STC-36): Dentist can view his/her patient follow up message as a list 105

System Test Case 37(STC-37): Dentist can post question to follow up his/her patient 106

System Test Case 38(STC-38): Patient can view follow up massage that his/her dentist asked in the website as a list 107

System Test Case 39(STC-39): Patient can answer follow up massage that his/her dentist asked in the website as a list 108

System Test Case 40(STC-40): Patient can view follow up massage that his/her dentist asked in mobile application as a list 109

System Test Case 41(STC-41): Patient can answer view follow up massage that his/her dentist asked in mobile application as a list 110

System Test Case 42(STC-42): Patient can view the dental clinic information and promotion in the website 111

System Test Case 43(STC-43): Patient can view the dental clinic information and promotion in mobile application 112

System Test Case 44(STC-44): Officer can add the information to the database 113

System Test Case 45(STC-45): Officer can edit the dental clinic information in the database 114

System Test Case 46(STC-46): Officer can delete the dental clinic information from the database 115

System Test Case 47(STC-47): Visitor can view the dental clinic information and promotion 116

System Test Case 48(STC-48): Patient can select the dental treatment to estimate the cost by him/herself 117

System Test Case 49(STC-49): Patient can select the dental treatment to estimate the cost by him/herself in the mobile application 118

System Test Case 50(STC-50): Visitor can select the dental treatment to estimate the cost by him/herself in the website 119

System Test Case 51(STC-51): Officer can add dental treatment information including the detail and cost into the database 120

System Test Case 52(STC-52): Officer can edit dental treatment information including the detail and cost in the database 121

System Test Case 53(STC-53): Officer can delete dental treatment information including the detail and cost in the database 122

System Test Case 54(STC-54): Officer can view dental treatment information as a list 123

Chapter Five: Appendix 124

# Chapter One: Introduction

## 1. Introduction

### 1.1 Objectives

Dental clinic service system consists of a web application and a mobile application which supports iOS7. The system supports English language for people who are patients, officers, or dentists of dental clinics. Furthermore, it offers some features for visitors who would like to gain information about the clinic. This system will reduce officers’ work and help patients save their time. Patients can find information about the dental clinic and personal information using the mobile application as well as the web application. For the visitor who wants to have a dental treatment from the clinic, and the dentist can use the system for checking their appointment schedule. Dental clinic services system integrates the services that are provided by most dental clinics into the web application and the mobile application. By combining all the services into one, the system provides more convenience to dental clinic users as well as staff.

This Test Plan has described in more detail of each test plan and each function that needs to test. The document includes Test Plan, Test Procedures, Unit Test, and System Test.

### 1.2 Purpose

The purpose of the Test Plan of Dental clinic services system is to establish a plan that will be used to detect and fix any bugs or other defects that may occur in the web application or the mobile application. The test plan is created to guarantee that all the defects are found before the system is released so that the software is faultless. The unit testing is to cover all of the implemented methods of Dental clinic services system and the system testing verifies all the user requirements.

## 1.3 Project Scope

The objective of this software design is to specify the requirements that the system should fulfill, which are as follows:

* Dental clinic services system provides an application that is accessible via web browser on a computer and also on a mobile phone.
* Dental clinic services system on mobile phone support iOS.
* Dental clinic services system supports English language only.
* Dental clinic services system provides features for dental clinics’ registered user whose are patients, officers and dentists.
* Dental clinic services system assists managing appointments of patients and dentists using the web application and the mobile application.
* Dental clinic services system provides a user authentication for each patient and dentist through dental clinic officers.
* Dental clinic services system provide features for generate QR code to patients from both web application and mobile application
* Dental clinic services system provide features for managing the queue of dental clinic
* Dental clinic services system assists following up treatment between dentist and patient
* Dental clinic services system provide the feature for promoting the clinic also managing those promotion and information
* Dental clinic services system help visitor and patient to predict the cost of dental care

### 1.4 User Classes and Characteristics

The system divides users into 4 groups. Information and characteristics of each group are listed below.

* **Visitor**

This group of user no needs to register to the system to use the services on the website or to use their mobile phone. Visitor is able to:

* Visitor can view calendar from Google Calendar.
* Visitor can view dental clinic information and promotion
* Visitor can select dental treatment to estimate the cost
* **Patient**

This group of user needs to be registered to the system at the dental clinic. They will get patientID and password from the clinic to use the services on the website or to use their mobile phone. Patients use their patientID to login. Patients are able to:

* Patient can receive a pateintID and password from the dental clinic.
* Patient can login to the web application and the mobile application.
* Patient can logout from the web application and the mobile application.
* Patient can view his/her appointment in the patient schedule.
* Patient can view all appointments in the dental clinic schedule.
* Patient can view their QR code.
* Patient can view follow up question.
* Patient can answer to follow up question.
* Patient can view dental clinic information and promotion.
* Patient can select dental treatment to estimate the cost.
* Patient can identify him/herself by using the QR code
* **Officer**

This group of users has the highest privilege in the system. They are given access to management functions for maintaining the system. Officers are able to:

* Officer can login to the web application using officerID and password.
* Officer can logout from the web application.
* Officer can register for new patients and dentists.
* Officer can add an appointment into the dental clinic schedule.
* Officer can edit an appointment in the dental clinic schedule.
* Officer can delete an appointment in the dental clinic schedule.
* Officer can add patient to a queue and reset queue.
* Officer can add information and promotion into the website.

### 1.5 Definition and Acronyms

**Definitions**

Feature Transformation of input parameters to output parameters based on a specified algorithm. It describes the functionality of a produce in the language of the product. Used for requirements analysis, design, coding, testing or maintenance. [IEEE90]

Design The period of time in the software life cycle during which the designs for architecture, software component, interfaces and data are created, documented, and verified to satisfy requirements. [IEEE90]

IEEE Institute for Electrical and Electronics Engineers. Biggest global interest group for engineers of different branches and for computer scientists. [IEEE90]

Requirement (1) A condition or capability needed by a user to solve a problem or achieve an objective. (2) A condition or capability that must be met or processed by system or system component to satisfy a contract, standard, specification, or other formally imposed document. (3) A documented representation of a condition or capability as in definition (1) or (2). [IEEE90]

Specification Precise description of an activity or work product which serves as basis or input for further activities or work product. A specification can comprise requirements to a product and how they will be solved. Different parts of a specification (e.g. what is to be done, how it will be done) must not be mixed. [IEEE90]

White box testing Testing process that focus on internal structure. The tester should know the code inside the program and test it through the code and determines the appropriate outputs.

Black box testing Process, device or system that focus on input, output and transfer characteristics without knowledge about it internal structure.

Unit Testing A level of the software testing process where individual units/components of a software/system are tested. The purpose is to validate that each unit of the performs as designed.

System testing A level of the software testing process where a complete, integrated system/software is tested. The purpose of this test is to evaluate the system's compliance with the specified requirements.

### 1.6 Acronyms and Definitions Acronyms

**Acronyms**

|  |  |
| --- | --- |
| **Acronyms** | **Stands for** |
| Registered User | Patient, Dentist, and Officer |
| Non-registered User | Visitor |
| PM | Project Management |
| PMP | Project Management Plan |
| URS | User Requirement Specification |
| SRS | Software Requirement Specification |
| VSE | Very Small Entity |
| QR code | Quick Response Code |
| UC | Use Case |
| AD | Activity Diagram |
| UI | User Interface |
| UTC | Unit Test Case |
| STC | System Test Case |
| CD | Class Diagram |
| SD | Sequence Diagram |

# Chapter Two: Test Plan and Test Procedure

## 2. Test Plan and Test Procedure

### 2.1 Test Objectives

The objectives of testing Dental clinic services system of the web application and the mobile application project are as follows:

1. All bugs or defects are detected.
2. Those bugs or defects are fixed.
3. The system meets user's requirements.
4. All functionalities and features must are provided.

### 2.2 Scope

Dental clinic services system will be tested using white-box testing and black-box testing techniques, which are unit testing and system testing respectively. And the test results will be documented in the Test Record.

### 2.3 Test Duration

|  |  |
| --- | --- |
| **Progress** | **Date and Duration** |
| **Progress 1 report** | Perform date: 1st July 2014 – 3rd July 2014 (Unit test, System test) Duration: 3 days |
| **Progress 2 report** | Perform date: 28th August 2014 – 2nd September 2014 (Unit test, System test) Duration: 5 days |
| **Final Progress report** | Perform date: 5th December 2014 – 9th December 2014 (Unit test, System test) Duration: 4 days |

### 2.4 Test Responsibility

|  |  |
| --- | --- |
| **Item** | **Responsibility** |
| Unit test of web application | Kanokwan & Worapun |
| Unit test of iOS mobile application | Kanokwan & Worapun |
| Record unit test of web application | Kanokwan & Worapun |
| Record unit test of iOS mobile application | Kanokwan & Worapun |
| System test of web application | Kanokwan & Worapun |
| System test of iOS mobile application | Kanokwan & Worapun |
| Record system test of web application | Kanokwan & Worapun |
| Record system test of iOS mobile application | Kanokwan & Worapun |

### 2.5 Test Strategy

Dental clinic services system testing will be performed in the following steps:

1. Design test cases for each feature.
2. Prepare test data for each feature.
3. Determine expected results.
4. Perform testing on individual feature.
5. Record the result of the testing.
6. Store all test files in the project repository.

### 2.6 Result of Testing

In the Test Record, the test result will be separated into two parts, which are:

1. Actual output: The actual outputs that are produced by each test case.

2. Pass/Fail criteria:

2.1 Pass: the actual result of the test is same as the expected result.

2.2 Fail: the actual result of the test is not same as the expected result.

### 2.7 Test Environment

**2.7.1 Hardware**

* **Computer**
* **Dell Inspiron N4110**

**Processor:** Intel(R) Core(TM) i3-2310M CPU@ 2.10GHz, 2.10 GHz

**Memory:** 6.00 GB

**Operating system:** Window 7 Ultimate

* **Sony VAIO SVE14118FHB**

**Processor:** Intel(R) Core(TM) i7-3612QM CPU@ 2.10GHz, 2.10 GHz

**Memory:** 8.00 GB

**Operating system:** Windows 7 Home Premium

* **MacBook Pro**

**Processor:** 2.5 GHz Intel Core i5

**Memory:** 4 GB 1600 MHz DDR38.00 GB

**Operating system:** W OS X 10.9.2 (13C1021)

* **iPhone**
* **iPhone 5s**

**Processor:** Dual core, 1300 MHz

**Memory:** 32 GB internal storage, 1 GB RAM DDR3

**Operating** System: iOS7

**2.7.2 Software**

* Google Chrome Version 19.0 or later
* HTML version 5
* PHP editor Version 5
* Xcode version 5.1.1
* Codeigniter framework version 2.1.4
* Phonegap version 3.0
* Ios operating system version ios7
* Adobe Dreamweaver CS6
* Ciqrcode for qr code generator

# 

# Chapter Three: Unit Testing

## 3. Unit Testing of Web Application

**Class:** Patient\_manage

### Unit Test Case 01 (UTC-01): testp\_login()

**Description:**

Test the login function of patient in class Patient\_manage by input patientID and password : public function login($patientID, $password)

**Test Data: Refer to Appendix at the end of this Test Plan for test data**

|  |  |  |
| --- | --- | --- |
| **Parameter name** | **Parameter type** | **Value** |
| $in\_user\_f | Varchar | xxxx |
| $in\_pass\_f | Varchar | xxxx |
| $in\_user\_t | Varchar | P001 |
| $in\_pass\_t | Varchar | 1234 |

**Test Case:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | | **Remark** | **Expected** |
| **patientID** | **password** |
| **1** | Test login in false case with no registered patientID and password | $in\_user\_f | $in\_pass\_f | Both patientID and password are not in database | false |
| **2** | Test login in true case with registered patientID and password | $in\_user\_t | $in\_pass\_t | Both patientID and password are in database | true |
| **3** | Test login in false case with patientID and incorrect password | $in\_user\_t | $in\_pass\_f | - | false |
| **4** | Test login in false case with incorrect patientID and correct password | $in\_user\_f | $in\_pass\_t | - | false |

### Unit Test Case 02 (UTC-02): testp\_ownCalendar()

**Description:**

Test the function of patient which can display their own calendar and get patients’ schedule from patientID : public function getCalendarByPID($patientID)

**Test Data: Refer to Refer to Appendix at the end of this Test Plan for test data**

|  |  |  |
| --- | --- | --- |
| **Parameter name** | **Parameter type** | **Value** |
| $in\_id\_f | Varchar | xxxx |
| $in\_id\_t | Varchar | P001 |
| $expResult1 | Array | null |
| $expResult2 | Array | [appointmentID] => 1 [patientID] => P001 [dentistID] => D001 [aDate] => 2014-08-02 [startTime] => 09:30:00 [endTime] => 10:00:00 [treatment] => Tooth braces [description] => 100 baht [submit] => Submit |

**Test Case:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Remark** | **Expected** |
| **patientID** |
| **1** | Test own calendar in false case with no registered patientID | $in\_id\_f | - | $expResult1 |
| **2** | Test own calendar in true case with registered patientID | $in\_id\_t | Data in $expResult2 is already in database | $expResult2 |

**Class:** Dentist\_manage

### Unit Test Case 03 (UTC-03): testd\_login()

**Description:**

Test the login function of dentist in class Dentist\_manage by input dentistID and password : public function login($dentistID, $password)

**Test Data: Refer to Appendix at the end of this Test Plan for test data**

|  |  |  |
| --- | --- | --- |
| **Parameter name** | **Parameter type** | **Value** |
| $in\_user\_f | Varchar | xxxx |
| $in\_pass\_f | Varchar | xxxx |
| $in\_user\_t | Varchar | D001 |
| $in\_pass\_t | Varchar | 1234 |

**Test Case:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | | **Remark** | **Expected** |
| **patientID** | **password** |
| **1** | Test login in false case with no registered dentistID and password | $in\_user\_f | $in\_pass\_f | Both dentistID and password are not in database | false |
| **2** | Test login in true case with registered dentistID and password | $in\_user\_t | $in\_pass\_t | Both dentistID and password are in database | true |
| **3** | Test login in false case with dentistID and incorrect password | $in\_user\_t | $in\_pass\_f | - | false |
| **4** | Test login in false case with incorrect dentistID and correct password | $in\_user\_f | $in\_pass\_t | - | false |

### Unit Test Case 04 (UTC-04): testd\_ownCalendar()

**Description:**

Test the function of patient which can display their own calendar and get dentists’ schedule from dentistID : public function getCalendarByDID($dentistID)

**Test Data: Refer to Appendix at the end of this Test Plan for test data**

|  |  |  |
| --- | --- | --- |
| **Parameter name** | **Parameter type** | **Value** |
| $in\_id\_f | Varchar | xxxx |
| $in\_id\_t | Varchar | D002 |
| $expResult1 | Array | null |
| $expResult2 | Array | [appointmentID] => 2 [patientID] => P002 [dentistID] => D002 [aDate] => 2014-08-04 [startTime] => 11:00:00 [endTime] => 11:30:00 [treatment] => Tooth braces [description] => 1000 baht [submit] => Submit |

**Test Case:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Remark** | **Expected** |
| **dentistID** |
| **1** | Test own calendar in false case with no registered dentistID | $in\_id\_f | - | $expResult1 |
| **2** | Test own calendar in true case with registered dentistID | $in\_id\_t | Data in $expResult2 is already in database | $expResult2 |

**Class:** Officer\_manage

### Unit Test Case 05 (UTC-05): testo\_login()

**Description:**

Test the login function of dentist in class Officer\_manage by input officerID and password : public function login($officerID, $password)

**Test Data: Refer to Appendix at the end of this Test Plan for test data**

|  |  |  |
| --- | --- | --- |
| **Parameter name** | **Parameter type** | **Value** |
| $in\_user\_f | Varchar | xxxx |
| $in\_pass\_f | Varchar | xxxx |
| $in\_user\_t | Varchar | OF001 |
| $in\_pass\_t | Varchar | 1234 |

**Test Case:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | | **Remark** | **Expected** |
| **patientID** | **password** |
| **1** | Test login in false case with no registered dentistID and password | $in\_user\_f | $in\_pass\_f | Both dentistID and password are not in database | false |
| **2** | Test login in true case with registered dentistID and password | $in\_user\_t | $in\_pass\_t | Both dentistID and password are in database | true |
| **3** | Test login in false case with OfficerID and incorrect password | $in\_user\_t | $in\_pass\_f | - | false |
| **4** | Test login in false case with incoreect OfficerID and correct password | $in\_user\_f | $in\_pass\_t | - | false |

### Unit Test Case 06 (UTC-06): testget\_p\_db()

**Description:**

Test the function of get patients’ account data from database to view as a list : public function get\_p\_db()

**Test Data: Refer to Appendix at the end of this Test Plan for test data**

|  |  |  |
| --- | --- | --- |
| **Parameter name** | **Parameter type** | **Value** |
| $expResult2 | Array[0] | [patientID] => P001 [password] => 1234 [f\_name] => Mickey [l\_name] => Mouse [age] => 80 [gender] => 1 [treatment] => Tooth braces,tooth zeplin [address] => Chaingmai Thailand [tel] => 053111111 [email]=> M.ky@hotmail.com [submit] => Submit |
| Array[1] | [patientID] => P002 [password] => 1111 [f\_name] => Minnie [l\_name] => Mouse [age] => 80 [gender] => 2 [treatment] => Tooth braces,tooth zeplin [address] => New York,USA [tel] => 0120000000 [email]=> M.Nie@hotmail.com [submit] => Submit |
| Array[2] | [patientID] => P003 [password] => 5555 [f\_name] => Goofy [l\_name] => Goof [age] => 4 [gender] => 1 [treatment] => EF line [address] => California,USA [tel] => 0230000000 [email]=> GG@yahoo.com [submit] => Submit |

**Test Case:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Remark** | **Expected** |
| **1** | Test view all patient list | - | $expResult is already in database | $expResult |

### Unit Test Case 07 (UTC-07): testget\_db()

**Description:**

Test the function of get dentists’ account data from database to view as a list : public function get\_db()

**Test Data: Refer to Appendix at the end of this Test Plan for test data**

|  |  |  |
| --- | --- | --- |
| **Parameter name** | **Parameter type** | **Value** |
| $expResult | Array[0] | [dentistID] => D001 [password] => 1234 [f\_name] => Donald [l\_name] => Duck [address] => Chaingmai Thailand [email] => DD@gmail.com [tel] => 0810000000 [submit] => Submit |
| Array[1] | [dentistID] => D002 [password] => 6789 [f\_name] => Daisy [l\_name] => Duck [address] => New York,USA [email] => D\_lady@hotmail.com [tel] => 0810000001 [submit] => submit |

**Test Case:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Remark** | **Expected** |
| **1** | Test view all clinic user list | - | $expResult is already in database | $expResult |

### Unit Test Case 08 (UTC-08): testp\_register()

**Description:**

Test register for patient function by officer and input all required data : public function p\_register($data)

**Test Data: Refer to Appendix at the end of this Test Plan for test data**

|  |  |  |
| --- | --- | --- |
| **Parameter name** | **Parameter type** | **Value** |
| $data | Array | [patientID] => P004 [password] => 1234 [f\_name] => Paula [l\_name] => Abdul [age] => 44 [gender] => 2 [treatment] => Whitening [address] => 199 sun st., Honolulu, Hawaii, USA [tel] => 111111111 [email]=> paula.a@aloha.com [submit] => Submit |
| $data2 | Array | [patientID] => P001 [password] => 1234 [f\_name] => Lawrence [l\_name] =>Devera [age] => 26 [gender] => 1 [treatment] => Cleaning [address] => California, USA [tel] => 111111111 [email]=> law.d@gmail.com [submit] => Submit |

**Test Case:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Remark** | **Expected** |
| **1** | Test register for patient true case | $data | - | true |
| **2** | Test register for patient false case | $data2 | patientID in $data2 is already exist | false |

### Unit Test Case 09 (UTC-09): testp\_edit()

**Description:**

Test patient edit function to see that system can retrieve data from database correctly or not : public function p\_edit($patientID)

**Test Data: Refer to Appendix at the end of this Test Plan for test data**

|  |  |  |
| --- | --- | --- |
| **Parameter name** | **Parameter type** | **Value** |
| $in\_id\_f | Varchar | P000 |
| $in\_id\_t | Varchar | P001 |
| $expResult1 | Array | Null |
| $expResult2 | Array | [patientID] => P001 [password] => 1234 [f\_name] => Mickey [l\_name] => Mouse [age] => 80 [gender] => 1 [treatment] => Tooth braces,tooth zeplin [address] => Chaingmai Thailand [tel] => 053111111 [email]=> M.ky@hotmail.com [submit] => Submit |

**Test Case:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Remark** | **Expected** |
| **1** | Test call patient edit page with not exist patientID | $in\_id\_f | - | $expResult1 |
| **2** | Test call patient edit page with exist patientID | $in\_id\_t | $expResult2 is already in database | $expResult2 |

### Unit Test Case 10 (UTC-10): testp\_delete()

**Description:**

Test function for delete patient from database by input a patientID : public function p\_delete($patientID)

**Test Data: Refer to Appendix at the end of this Test Plan for test data**

|  |  |  |
| --- | --- | --- |
| **Parameter name** | **Parameter type** | **Value** |
| $in\_id\_f | Varchar | 0 |
| $in\_id\_t | Varchar | P004 |

**Test Case:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Remark** | **Expected** |
| **1** | Test delete patient from database with no patientID | $in\_id\_f | $in\_id\_f is not in database | false |
| **2** | Test delete patient from database with patientID | $in\_id\_t | $in\_id\_t is already in database | true |

### Unit Test Case 11 (UTC-11): testd\_register()

**Description:**

Test register for dentist function by officer and input all required data : public function d\_register($data)

**Test Data: Refer to Appendix at the end of this Test Plan for test data**

|  |  |  |
| --- | --- | --- |
| **Parameter name** | **Parameter type** | **Value** |
| $data | Array | [dentistID] => D003 [password] => 1234 [f\_name] => dent [l\_name] => tist [address] => clinic [tel] => 0000000000 [email] => dentist.t@clinic.com [submit] => Submit |
| $data2 | Array | [dentistID] => D001 [password] => 1234 [f\_name] => John [l\_name] => Doe [address] => Munich, Germany [tel] => 0000000000 [email] => j.doe@clinic.com [submit] => Submit |

**Test Case:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Remark** | **Expected** |
| **1** | Test register for dentist true case | $data | - | true |
| **2** | Test register for dentist false case | $data2 | dentistID in $data2 is already exist | false |

### Unit Test Case 12 (UTC-12): testd\_edit()

**Description:**

Test dentist edit function to see that system can retrieve data from database correctly or not : public function d\_edit($dentistID)

**Test Data: Refer to Appendix at the end of this Test Plan for test data**

|  |  |  |
| --- | --- | --- |
| **Parameter name** | **Parameter type** | **Value** |
| $in\_id\_f | Varchar | D000 |
| $in\_id\_t | Varchar | D003 |
| $expResult1 | Array | Null |
| $expResult2 | Array | [dentistID] => D003 [password] => 1234 [f\_name] => dent [l\_name] => tist [address] => clinic [tel] => 0000000000 [email] => dentist.t@clinic.com [submit] => Submit |

**Test Case:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Remark** | **Expected** |
| **1** | Test call dentist edit page with not exist dentistID | $in\_id\_1 | - | $expResult1 |
| **2** | Test call dentist edit page with exist dentistID | $in\_id\_2 | $in\_id\_t is already in database | $expResult2 |

### Unit Test Case 13 (UTC-13): testd\_delete()

**Description:**

Test function for delete dentist from database by input dentistID : public function d\_delete($dentistID)

**Test Data: Refer to Appendix at the end of this Test Plan for test data**

|  |  |  |
| --- | --- | --- |
| **Parameter name** | **Parameter type** | **Value** |
| $in\_id\_f | Varchar | 0 |
| $in\_id\_t | Varchar | D003 |

**Test Case:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Remark** | **Expected** |
| **1** | Test delete dentist from database with no dentistID | $in\_id\_f | $in\_id\_f is not in database | false |
| **2** | Test delete dentist from database with dentistID | $in\_id\_t | $in\_id\_t is already in database | true |

### Unit Test Case 14 (UTC-14): testview\_appointment()

**Description:**

Test the function of get all appointment data from database to view as a list : public function view\_appointment()

**Test Data: Refer to Appendix at the end of this Test Plan for test data**

|  |  |  |
| --- | --- | --- |
| **Parameter name** | **Parameter type** | **Value** |
| $expResult | Array[0] | [appointmentID] => 1 [patientID] => P001 [dentistID] => D001 [aDate] => 2014-08-01 [startTime] => 09:00:00 [endTime] => 09:30:00 [treatment] => Tooth braces [description] =>100 baht [submit] => Submit |
| Array[1] | [appointmentID] => 2 [patientID] => P002 [dentistID] => D002 [aDate] => 2014-08-05 [startTime] => 13:00:00 [endTime] => 14:00:00 [treatment] => Whitening [description] => 1000 baht [submit]=>null [submit] => Submit |

**Test Case:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Remark** | **Expected** |
| **1** | Test view all appointment list | - | $expResult is already in database | $expResult |

### Unit Test Case 15 (UTC-15): testmake\_appointment()

**Description:**

Test make appointment function by officer and input all required data : public function make\_appointment($data)

**Test Data: Refer to Appendix at the end of this Test Plan for test data**

|  |  |  |
| --- | --- | --- |
| **Parameter name** | **Parameter type** | **Value** |
| $data | Array | [patientID] => P001 [dentistID] => D002 [aDate] => 2014-12-05 [startTime] => 10:00:00 [endTime] => 10:30:00 [treatment] => Whitening [description] => 4500 THB [submit] => Submit |
| $data2 | Array | [patientID] => P002 [dentistID] => D001 [aDate] => 2014-12-05 [startTime] => 10:30:00 [endTime] => 11:00:00 [treatment] => Whitening [description] => 4500 THB [submit] => Submit |
| $data3 | Array | [patientID] => P003 [dentistID] => D002 [aDate] => 2014-12-05 [startTime] => 10:00:00 [endTime] => 10:30:00 [treatment] => Whitening [description] => 4500 THB [submit] => Submit |
| $data4 | Array | [patientID] => P004 [dentistID] => D001 [aDate] => 2014-12-05 [startTime] => 10:00:00 [endTime] => 10:30:00 [treatment] => Whitening [description] => 4500 THB [submit] => Submit |
| $data5 | Array | [patientID] => P001 [dentistID] => D003 [aDate] => 2014-12-05 [startTime] => 11:00:00 [endTime] => 11:30:00 [treatment] => Whitening [description] => 4500 THB [submit] => Submit |
| $data6 | Array | [patientID] => P002 [dentistID] => D003 [aDate] => 2014-12-05 [startTime] => 10:30:00 [endTime] => 11:00:00 [treatment] => Whitening [description] => 4500 THB [submit] => Submit |
| $data7 | Array | [patientID] => P001 [dentistID] => D001 [aDate] => 2014-12-06 [startTime] => 19:00:00 [endTime] => 19:30:00 [treatment] => Whitening [description] => 4500 THB [submit] => Submit |
| $data8 | Array | [patientID] => P002 [dentistID] => D001 [aDate] => 2014-12-06 [startTime] => 10:00:00 [endTime] => 10:30:00 [treatment] => Whitening [description] => 4500 THB [submit] => Submit |
| $data9 | Array | [patientID] => P003 [dentistID] => D001 [aDate] => 2014-12-05 [startTime] => 19:00:00 [endTime] => 19:30:00 [treatment] => Whitening [description] => 4500 THB [submit] => Submit |
| $data10 | Array | [patientID] => P004 [dentistID] => D001 [aDate] => 2014-12-05 [startTime] => 10:00:00 [endTime] => 10:30:00 [treatment] => Whitening [description] => 4500 THB [submit] => Submit |
| $data11 | Array | [patientID] => P001 [dentistID] => D001 [aDate] => 2014-12-06 [startTime] => 13:00:00 [endTime] => 13:30:00 [treatment] => Whitening [description] => 4500 THB [submit] => Submit |
| $data12 | Array | [patientID] => P002 [dentistID] => D002 [aDate] => 2014-12-06 [startTime] => 13:30:00 [endTime] => 13:00:00 [treatment] => Whitening [description] => 4500 THB [submit] => Submit |
| $data13 | Array | [patientID] => P020 [dentistID] => D001 [aDate] => 2014-12-07 [startTime] => 14:00:00 [endTime] => 14:30:00 [treatment] => Whitening [description] => 4500 THB [submit] => Submit |
| $data14 | Array | [patientID] => P001 [dentistID] => D030 [aDate] => 2014-12-07 [startTime] => 14:00:00 [endTime] => 14:30:00 [treatment] => Whitening [description] => 4500 THB [submit] => Submit |
| $data15 | Array | [patientID] => P0020 [dentistID] => D030 [aDate] => 2014-12-07 [startTime] => 14:00:00 [endTime] => 14:30:00 [treatment] => Whitening [description] => 4500 THB [submit] => Submit |

**Test Case:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Remark** | **Expected** |
| **1** | Test make an appointment true case | $data | - | true |
| **2** | Test make an appointment false case | $data2 | date, start time, and dentist are already added to the database | false |
| **3** | Test make an appointment true case | $data3 | Same dentist and date but different time | true |
| **4** | Test make an appointment truecase | $data4 | Same dentist but different date | true |

### Unit Test Case 16 (UTC-16): testapp\_edit()

**Description:**

Test appointment edit function to see that system can retrieve data from database correctly or not : public function app\_edit($appointmentID)

**Test Data: Refer to Appendix(p.68) for test data**

|  |  |  |
| --- | --- | --- |
| **Parameter name** | **Parameter type** | **Value** |
| $in\_id\_f | Varchar | 3 |
| $in\_id\_t | Varchar | 2 |
| $expResult1 | Array | Null |
| $expResult2 | Array | [appointmentID] => 2 [patientID] => P002 [dentistID] => D002 [aDate] => 2014-08-05 [startTime] => 13:00:00 [endTime] => 14:00:00 [treatment] => Whitening [description] => 1000 baht [submit] => Submit |

**Test Case:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Remark** | **Expected** |
| **1** | Test call appointment edit page with not exist dentistID | $in\_id\_f | $in\_id\_f is not in database | $expResult1 |
| **2** | Test call appointment edit page with exist dentistID | $in\_id\_t | $in\_id\_t is already in database | $expResult2 |

### Unit Test Case 17 (UTC-17): testapp\_delete()

**Description:**

Test function for delete appointment from database by input appointmentID : public function app\_delete($appointmentID)

**Test Data: Refer to Appendix at the end of this Test Plan for test data**

|  |  |  |
| --- | --- | --- |
| **Parameter name** | **Parameter type** | **Value** |
| $in\_id\_f | Varchar | 0 |
| $in\_id\_t | Varchar | 3 |

**Test Case:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Remark** | **Expected** |
| **1** | Test delete appointment from database with no appointment ID | $in\_id\_f | $in\_id\_f is not in database | false |
| **2** | Test delete appointment from database with appointment ID | $in\_id\_t | $in\_id\_t is already in database | true |

### Unit Test Case 18 (UTC-18): testPatientCalendar()

**Description:**

Test view patients’ appointment function with specific patientID to see that system can retrieve data from appointment table in database correctly or not : public function patientCalendar($patientID)

**Test Data: Refer to Appendix at the end of this Test Plan for test data**

|  |  |  |
| --- | --- | --- |
| **Parameter name** | **Parameter type** | **Value** |
| $in\_id\_f | Varchar | P0002 |
| $in\_id\_t | Varchar | P001 |
| $expResult1 | Array | null |
| $expResult2 | Array | [appointmentID] => 1 [patientID] => P001 [dentistID] => D001 [aDate] => 2014-08-01 [startTime] => 09:00:00 [endTime] => 09:30:00 [treatment] => Tooth braces [description] => 100 baht [submit] => Submit |

**Test Case:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Remark** | **Expected** |
| **1** | Test call patients’ appointment edit page with not exist patientID | $in\_id\_f | - | $expResult1 |
| **2** | Test call patients’ appointment edit page with exist patientID | $in\_id\_t | $expResult2 is already in database | $expResult2 |

### Unit Test Case 19 (UTC-19): testDentistCalendar()

**Description:**

Test view dentists’ appointment function with specific dentistID to see that system can retrieve data from appointment table in database correctly or not : public function dentistCalendar($dentistID)

**Test Data: Refer to Appendix at the end of this Test Plan for test data**

|  |  |  |
| --- | --- | --- |
| **Parameter name** | **Parameter type** | **Value** |
| $in\_id\_f | Varchar | dent001 |
| $in\_id\_t | Varchar | D002 |
| $expResult1 | Array | Null |
| $expResult2 | Array | [appointmentID] => 2 [patientID] => P002 [dentistID] => D002 [aDate] => 2014-08-05 [startTime] => 13:00:00 [endTime] => 14:00:00 [treatment] => Whitening [description] => 31000 baht [submit] => Submit |

**Test Case:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Remark** | **Expected** |
| **1** | Test call dentists’ appointment edit page with not exist dentistID | $in\_id\_f | - | $expResult1 |
| **2** | Test call dentists’ appointment edit page with exist dentistID | $in\_id\_t | $expResult2 is already in database | $expResult2 |

### Unit Test Case 20 (UTC-20): testp\_edit\_data()

**Description:**

Test edit patients’ account function by officer and input all required data : public function edit\_data($data)

**Test Data: Refer to Appendix at the end of this Test Plan for test data**

|  |  |  |
| --- | --- | --- |
| **Parameter name** | **Parameter type** | **Value** |
| $data | Array | [patientID] => P001 [password] => 1234 [f\_name] => Lawrence [l\_name] => Mouse [age] => 80 [gender] => 1 [treatment] => Tooth braces,tooth zeplin [address] => Chaingmai Thailand [tel] => 053111111 [email]=> law.d@gmail.com |
| $expResult | Array | [patientID] => P001 [password] => 1234 [f\_name] => Lawrence [l\_name] => Mouse [age] => 80 [gender] => 1 [treatment] => Tooth braces,tooth zeplin [address] => Chaingmai Thailand [tel] => 053111111 [email]=> law.d@gmail.com |

**Test Case:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Remark** | **Expected** |
| **1** | Test edit for patients’ account true case | $data | - | $expResult |

### Unit Test Case 21 (UTC-21): testd\_edit\_data()

**Description:**

Test edit dentists’ account function by officer and input all required data : public function d\_edit\_data($data)

**Test Data: Refer to Appendix at the end of this Test Plan for test data**

|  |  |  |
| --- | --- | --- |
| **Parameter name** | **Parameter type** | **Value** |
| $data | Array | [dentistID] => D001 [password] => 1234 [f\_name] => Dumbo [l\_name] => Duck [address] => Chaingmai Thailand [email] => DD@gmail.com [tel] => 0810000000 |
| $expResult | Array | [dentistID] => D001 [password] => 1234 [f\_name] => Dumbo [l\_name] => Duck [address] => Chaingmai Thailand [email] => DD@gmail.com [tel] => 0810000000 |

**Test Case:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Remark** | **Expected** |
| **1** | Test edit for dentists’ account true case | $data | - | $expResult |

### Unit Test Case 22 (UTC-22): testd\_edit\_data()

**Description:**

Test edit appointments’ data function by officer and input all required data : public function app\_edit\_data($data)

**Test Data: Refer to Appendix at the end of this Test Plan for test data**

|  |  |  |
| --- | --- | --- |
| **Parameter name** | **Parameter type** | **Value** |
| $data | Array | [appointmentID] => 2 [patientID] => P002 [dentistID] => D002 [aDate] => 2014-08-05 [startTime] => 13:00:00 [endTime] => 14:00:00 [treatment] => Whitening [description] => 1000 baht |
| $expResult | Array | [appointmentID] => 2 [patientID] => P002 [dentistID] => D002 [aDate] => 2014-08-05 [startTime] => 13:00:00 [endTime] => 14:00:00 [treatment] => Whitening [description] => 1000 baht |

**Test Case:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Remark** | **Expected** |
| **1** | Test edit for appointments’ data true case | $data | - | $expResult |

### Class: PatientMobile\_manage

### Unit Test Case 23 (UTC-23): testMobileLogin()

**Description:**

Test the login function in mobile application of patient in class PatientMobile\_manage by input patientID and password : public function loginMobile($patientID, $password)

**Test Data:**

|  |  |  |
| --- | --- | --- |
| **Parameter name** | **Parameter type** | **Value** |
| $in\_user\_f | Varchar | xxxx |
| $in\_pass\_f | Varchar | xxxx |
| $in\_user\_t | Varchar | P001 |
| $in\_pass\_t | Varchar | 1234 |

**Test Case:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | | **Remark** | **Expected** |
| **patientID** | **password** |
| **1** | Test login in false case with no registered patientID and password | $in\_user\_f | $in\_pass\_f | Both patientID and password are not in database | false |
| **2** | Test login in true case with registered patientID and password | $in\_user\_t | $in\_pass\_t | Both patientID and password are in database | true |
| **3** | Test login in false case with patientID and incorrect password | $in\_user\_t | $in\_pass\_f | - | false |

### Unit Test Case 24 (UTC-24): testOwnAppointmentMobile()

**Description:**

Test view his/her appointment function of patient in mobile application which can display his/her appointment and get patients’ schedule from patientID : public function getCalendarByPID($patientID)

**Test Data:**

|  |  |  |
| --- | --- | --- |
| **Parameter name** | **Parameter type** | **Value** |
| $in\_id\_f | Varchar | xxxx |
| $in\_id\_t | Varchar | P001 |
| $expResult1 | Array | null |
| $expResult2 | Array | [appointmentID] => 1 [patientID] => P001 [dentistID] => D001 [aDate] => 2014-08-02 [startTime] => 09:30:00 [endTime] => 10:00:00 [treatment] => Tooth braces [description] => 100 baht |

**Test Case:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Remark** | **Expected** |
| **patientID** |
| **1** | Test own appointment in false case with no registered patientID | $in\_id\_f | - | $expResult1 |
| **2** | Test own appointment in true case with registered patientID | $in\_id\_t | Data in $expResult2 is already in database | $expResult2 |

**Class:** Patient\_manage

### Unit Test Case 25 (UTC-25): testp\_getAllTreatment()

**Description:**

Test the function of get all data of table treatment from database: public function getAllTreatment()

**Test Data: Refer to Appendix at the end of this Test Plan for test data**

|  |  |  |
| --- | --- | --- |
| **Parameter name** | **Parameter type** | **Value** |
| $expResult | Array[0] | [tid]=>1, [tName]=>Full mouth checkup, [cost]=>0 |
| Array[1] | [tid]=>2, [tName]=>Composite filling, [cost]=>1000 |
| Array[2] | [tid]=>3, [tName]=>Fluoride application, [cost]=>400 |
| Array[3] | [tid]=>4, [tName]=>Consultation, [cost]=>0 |

**Test Case:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Remark** | **Expected** |
| **1** | Test gat all dental treatment from database | - | $expResult is already in database | $expResult |

### Unit Test Case 26 (UTC-26): testp\_calculate()

**Description:**

Test the cost estimation by using calculate function to calculate the cost of dental treatment: public function calculate($dent\_treat)

**Test Data: Refer to Appendix at the end of this Test Plan for test data**

|  |  |  |
| --- | --- | --- |
| **Parameter name** | **Parameter type** | **Value** |
| $in\_treat1 | Array[0] | [tid]=>1, [tName]=>Full mouth checkup, [cost]=>0 |
| $in\_treat2 | Array[0] | [tid]=>2, [tName]=>Composite filling, [cost]=>1000 |
| Array[1] | [tid]=>3, [tName]=>Fluoride application, [cost]=>400 |
| $in\_treat3 | Array[0] | null |
| $result1 | Double | 0 |
| $result2 | Double | 1400 |

**Test Case:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Remark** | **Expected** |
| **treatment** |
| **1** | Test calculate function that use to estimate the cost of dental treatment by input one dental treatment | $in\_treat1 | - | $result1 |
| **2** | Test calculate function that use to estimate the cost of dental treatment by input two dental treatment | $in\_treat2 | - | $result2 |
| **3** | Test calculate function that use to estimate the cost of dental treatment by not input any dental treatment | $in\_treat3 | - | $result1 |

### Unit Test Case 27 (UTC-27): testp\_get\_information()

**Description:**

Test to get information and promotion function of dental clinic: public function get\_informaion()

**Test Data: Refer to Appendix at the end of this Test Plan for test data**

|  |  |  |
| --- | --- | --- |
| **Parameter name** | **Parameter type** | **Value** |
| $expResult | Array[0] | [infoID]=>1, [type]=>promo, [title]=>Free checkup, [details]=>Full mouth checkup for free at the clinic, [officerID]=>OF001 |
| Array[1] | [infoID]=>2, [type]=>information, [title]=>EF Line, [details]=>EF Line is tooth braces for children between 4-15 years old, [officerID]=>OF001 |
| Array[2] | [infoID]=>3, [type]=>promo, [title]=>Whitening 30% off, [details]=>End of the year sale, whitening treatment is noe 30% off, [officerID]=>OF002 |

**Test Case:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Remark** | **Expected** |
| **1** | Test gat all information and promotion from database | - | $expResult is already in database | $expResult |

### Unit Test Case 28 (UTC-28): testp\_viewFollowUpByQid()

**Description:**

Test the function of view follow up by question ID for patient to answer each specific question: public function viewFollowUpByQid($qid)

**Test Data: Refer to Appendix at the end of this Test Plan for test data**

|  |  |  |
| --- | --- | --- |
| **Parameter name** | **Parameter type** | **Value** |
| $in\_qid1 | Int | 14 |
| $in\_qid2 | Int | 15 |
| $expResult1 | Array[0] | [qid]=>14, [patientID]=>P001, [dentistID]=>D001, [question]=>How are your composite filling? Is that fine?, [answer]=>null, [qDateTime]=>2014-10-19 11:04:35, [aDateTime]=>null |
| $expResult2 | Array | null |

**Test Case:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Remark** | **Expected** |
| **qid** |
| **1** | Test get follow up question by qid by input already exist qid | $in\_qid1 | $expResult1 is already in database | $expResult1 |
| **2** | Test get follow up question by qid by input not exist qid | $in\_qid2 | - | $expResult2 |

### Unit Test Case 29 (UTC-29): testp\_answer\_data()

**Description:**

Test the function of answer a follow up question: public function answer\_data($data)

**Test Data: Refer to Appendix at the end of this Test Plan for test data**

|  |  |  |
| --- | --- | --- |
| **Parameter name** | **Parameter type** | **Value** |
| $in\_data1 | Array | [qid]=>14, [patientID]=>P001, [dentistID]=>D001, [question]=>How are your composite filling? Is that fine?, [answer]=>Nothing wrong, still fine, [qDateTime]=>2014-10-19 11:04:35, [aDateTime]=>2014-10-19 19:39.00 |
| $in\_data2 | Array | [qid]=>null |
| $expResult1 | Boolean | true |
| $expResult2 | Boolean | false |

**Test Case:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Remark** | **Expected** |
| **data** |
| **1** | Test answer a follow up question at not answered question | $in\_data1 | - | $expResult1 |
| **2** | Test answer a follow up question at answered question | $in\_data2 | - | $expResult2 |

### Unit Test Case 30 (UTC-30): testp\_view\_follow\_up()

**Description:**

Test the function of view all follow up of each patient: public function view\_follow\_up()

**Test Data: Refer to Appendix at the end of this Test Plan for test data**

|  |  |  |
| --- | --- | --- |
| **Parameter name** | **Parameter type** | **Value** |
| $in\_pid | Varchar | P002 |
| $expResult | Array[0] | [qid]=>13, [patientID]=>P002, [dentistID]=>D002, [question]=>Your retainer is ready in next 3 days (2014-10-15) please come an pick it up at the clinic anytime [answer]=>Ok, thank you very much, [qDateTime]=>2014-10-12 09:15:22, [aDateTime]=>2014-10-13 08:00.28 |

**Test Case:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Remark** | **Expected** |
| **patientID** |
| **1** | Test gat all follow up from database | $in\_pid | $expResult is already in database | $expResult |

### Unit Test Case 31 (UTC-31): testp\_view\_detail()

**Description:**

Test the function of view patients’ own information in website: public function own\_detail($patientID)

**Test Data: Refer to Appendix at the end of this Test Plan for test data**

|  |  |  |
| --- | --- | --- |
| **Parameter name** | **Parameter type** | **Value** |
| $in\_pid | Varchar | P001 |
| $expResult | Array[0] | [patientID] => P001 [password] => 1234 [f\_name] => Mickey [l\_name] => Mouse [age] => 80 [gender] => 1 [treatment] => Tooth braces,tooth zeplin [address] => Chaingmai Thailand [tel] => 053111111 [email]=> M.ky@hotmail.com [submit] => Submit |

**Test Case:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Remark** | **Expected** |
| **patientID** |
| **1** | Test gat patients’ own information by using patientID | $in\_pid | $expResult is already in database | $expResult |

**Class:** Dentist\_manage

### Unit Test Case 32 (UTC-32): testd\_get\_p\_db()

**Description:**

Test the function of get all patient list to select the patient to follow up: public function get\_p\_db()

**Test Data: Refer to Appendix at the end of this Test Plan for test data**

|  |  |  |
| --- | --- | --- |
| **Parameter name** | **Parameter type** | **Value** |
| $expResult | Array[0] | [patientID] => P001 [password] => 1234 [f\_name] => Mickey [l\_name] => Mouse [age] => 80 [gender] => 1 [treatment] => Tooth braces,tooth zeplin [address] => Chaingmai Thailand [tel] => 053111111 [email]=> M.ky@hotmail.com [submit] => Submit |
| Array[1] | [patientID] => P002 [password] => 1111 [f\_name] => Minnie [l\_name] => Mouse [age] => 80 [gender] => 2 [treatment] => Tooth braces,tooth zeplin [address] => New York,USA [tel] => 0120000000 [email]=> M.Nie@hotmail.com [submit] => Submit |
| Array[2] | [patientID] => P003 [password] => 5555 [f\_name] => Goofy [l\_name] => Goof [age] => 4 [gender] => 1 [treatment] => EF line [address] => California,USA [tel] => 0230000000 [email]=> GG@yahoo.com [submit] => Submit |

**Test Case:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Remark** | **Expected** |
| **1** | Test view all patient list | - | $expResult is already in database | $expResult |

### Unit Test Case 33 (UTC-33): testd\_follow\_up()

**Description:**

To test the function of follow up by input specific patientID to ask question: public function follow\_up($patientID)

**Test Data: Refer to Appendix at the end of this Test Plan for test data**

|  |  |  |
| --- | --- | --- |
| **Parameter name** | **Parameter type** | **Value** |
| $in\_pid | Varchar | P001 |
| $expResult | Varchar | P001 |

**Test Case:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Remark** | **Expected** |
| **patientID** |
| **1** | Test get patientID for asking follow up question | $in\_pid | - | $expResult |

### 

### Unit Test Case 34 (UTC-34): testd\_follow\_up\_data()

**Description:**

To test the function of adding follow up question to specific patientID: public function follow\_up\_data($data)

**Test Data: Refer to Appendix at the end of this Test Plan for test data**

|  |  |  |
| --- | --- | --- |
| **Parameter name** | **Parameter type** | **Value** |
| $in\_data | Array | [qid]=>14, [patientID]=>P001, [dentistID]=>D001, [question]=>How are your composite filling? Is that fine?, [answer]=>null, [qDateTime]=>2014-10-19 11:04:35, [aDateTime]=>null |
| $expResult1 | Boolean | true |

**Test Case:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Remark** | **Expected** |
| **data** |
| **1** | Test to add a follow up question and save to the database | $in\_data | - | $expResult1 |

### Unit Test Case 35 (UTC-35): testd\_getAllReply()

**Description:**

To test the function of getting all reply from a patient to a dentist. Dentist can get only their follow up of their patient: public function callAllReply($dentistID)

**Test Data: Refer to Appendix at the end of this Test Plan for test data**

|  |  |  |
| --- | --- | --- |
| **Parameter name** | **Parameter type** | **Value** |
| $in\_did1 | Varchar | D002 |
| $in\_did2 | Varchar | D001 |
| $expResult1 | Array[0] | [qid]=>13, [patientID]=>P002, [dentistID]=>D002, [question]=>Your retainer is ready in next 3 days (2014-10-15) please come an pick it up at the clinic anytime [answer]=>Ok, thank you very much, [qDateTime]=>2014-10-12 09:15:22, [aDateTime]=>2014-10-13 08:00.28 |
| $expResult2 | Array | null |

**Test Case:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Remark** | **Expected** |
| **data** |
| **1** | Test to get all follow up reply as a list with dentistID that are match to the logged-in dentistID | $in\_did1 | $expResult1 is already in database | $expResult1 |
| **2** | Test to get all follow up reply as a list with dentistID that are not match to the logged-in dentistID | $in\_did2 | - | null |

**Class:** Officer\_manage

### Unit Test Case 36 (UTC-36): testo\_addDentalTreatment()

**Description:**

Test the function of adding a dental treatment to the database: public function add\_dental\_treatment($data)

**Test Data: Refer to Appendix at the end of this Test Plan for test data**

|  |  |  |
| --- | --- | --- |
| **Parameter name** | **Parameter type** | **Value** |
| $in\_data1 | Array | [tid]=>5, [tName]=>Extraction, [cost]=>800 |
| $in\_data2 | Array | [tid]=>4, [tName]=>Extraction, [cost]=>800 |
| $expResult1 | Boolean | true |
| $expResult2 | Boolean | false |

**Test Case:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Remark** | **Expected** |
| **data** |
| **1** | Test to add a dental treatment with do not exist tid | $in\_data1 | - | $expResult1 |
| **2** | Test to add a dental treatment with do exist tid | $in\_data2 | - | $expResult2 |

### Unit Test Case 37 (UTC-37): testo\_editDentalTreatment()

**Description:**

Test the function of calling dental treatment by tid for officer to edit : public function edit\_dental\_treatment($tid)

**Test Data: Refer to Appendix at the end of this Test Plan for test data**

|  |  |  |
| --- | --- | --- |
| **Parameter name** | **Parameter type** | **Value** |
| $in\_tid1 | Integer | 5 |
| $in\_tid2 | Integer | 6 |
| $expResult1 | Array | [tid]=>5, [tName]=>Extraction, [cost]=>800 |
| $expResult2 | Array | null |

**Test Case:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Remark** | **Expected** |
| **tid** |
| **1** | Test to call a dental treatment with existed tid | $in\_tid1 | $expResult1 is already in database | $expResult1 |
| **2** | Test to call a dental treatment with not existed tid | $in\_tid2 | - | $expResult2 |

### Unit Test Case 38 (UTC-38): testo\_editDentalTreatment\_data()

**Description:**

Test the function of save an edited a dental treatment to the database: public function edit\_dental\_treatment\_data($data)

**Test Data: Refer to Appendix at the end of this Test Plan for test data**

|  |  |  |
| --- | --- | --- |
| **Parameter name** | **Parameter type** | **Value** |
| $in\_data | Array | [tid]=>5, [tName]=>Extraction, [cost]=>900 |
| $expResult | Boolean | true |

**Test Case:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Remark** | **Expected** |
| **tid** |
| **1** | Test to save the edited dental treatment to the database | $in\_data | - | $expResult |

### Unit Test Case 39 (UTC-39): testo\_deleteDentalTreatment()

**Description:**

Test the function of deleting a dental treatment from database: public function delete\_dental\_treatment($tid)

**Test Data: Refer to Appendix at the end of this Test Plan for test data**

|  |  |  |
| --- | --- | --- |
| **Parameter name** | **Parameter type** | **Value** |
| $in\_tid1 | Integer | 5 |
| $in\_tid2 | Integer | 6 |
| $expResult1 | Boolean | true |
| $expResult2 | Boolean | false |

**Test Case:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Remark** | **Expected** |
| **tid** |
| **1** | Test to delete a dental treatment by existed tid | $in\_tid1 | - | $expResult1 |
| **2** | Test to delete a dental treatment by existed tid | $in\_tid2 | - | $expResult2 |

### Unit Test Case 40 (UTC-40): testo\_get\_treatment()

**Description:**

Test the function of get all treatment and treatment details from database: public function get\_treatment()

**Test Data: Refer to Appendix at the end of this Test Plan for test data**

|  |  |  |
| --- | --- | --- |
| **Parameter name** | **Parameter type** | **Value** |
| $expResult | Array[0] | [tid]=>1, [tName]=>Full mouth checkup, [cost]=>0 |
| Array[1] | [tid]=>2, [tName]=>Composite filling, [cost]=>1000 |
| Array[2] | [tid]=>3, [tName]=>Fluoride application, [cost]=>400 |
| Array[3] | [tid]=>4, [tName]=>Consultation, [cost]=>0 |

**Test Case:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Remark** | **Expected** |
| **1** | Test to get all dental treatment from database | - | $expResult is already in database | $expResult |

### 

### Unit Test Case 41 (UTC-41): testo\_add\_information()

**Description:**

Test the function of adding information and promotion to the database: public function add\_information($data)

**Test Data: Refer to Appendix at the end of this Test Plan for test data**

|  |  |  |
| --- | --- | --- |
| **Parameter name** | **Parameter type** | **Value** |
| $in\_data1 | Array | [infoID]=>4, [type]=>promo, [title]=>Buy one get one free, [details]=>Buy one fluoride application course now get one more course for free, [officerID]=>OF001 |
| $in\_data2 | Array | [infoID]=>5, [type]=>info, [title]=>Why full mouth checkup is important?[details]=>To checkup full mouth is help you to reduce the bad helth of your mouth, [officerID]=>OF001 |
| $expResult | Boolean | true |

**Test Case:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Remark** | **Expected** |
| **data** |
| **1** | Test to add a promotion to the database | $in\_data1 | - | $expResult |
| **2** | Test to add an information to the database | $in\_data2 | - | $expResult |

### Unit Test Case 42 (UTC-42): testo\_edit\_info()

**Description:**

Test the function of call information and promotion to edit from the database: public function edit\_info($infoID)

**Test Data: Refer to Appendix at the end of this Test Plan for test data**

|  |  |  |
| --- | --- | --- |
| **Parameter name** | **Parameter type** | **Value** |
| $in\_infoid1 | Integer | 4 |
| $in\_infoid2 | Integer | 5 |
| $expResult1 | Array | [infoID]=>4, [type]=>promo, [title]=>Buy one get one free, [details]=>Buy one fluoride application course now get one more course for free, [officerID]=>OF001 |
| $expResult2 | Array | [infoID]=>5, [type]=>info, [title]=>Why full mouth checkup is important?[details]=>To checkup full mouth is help you to reduce the bad helth of your mouth, [officerID]=>OF001 |

**Test Case:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Remark** | **Expected** |
| **infoID** |
| **1** | Test to call a promotion from specific infoID | $in\_infoid1 | $expResult1 is already in database | $expResult1 |
| **2** | Test to call an information from specific infoID | $in\_infoid2 | $expResult2 is already in database | $expResult2 |

### Unit Test Case 43 (UTC-43): testo\_edit\_info\_data()

**Description:**

Test the function of save edited information and promotion to the database: public function edit\_info\_data($data)

**Test Data: Refer to Appendix at the end of this Test Plan for test data**

|  |  |  |
| --- | --- | --- |
| **Parameter name** | **Parameter type** | **Value** |
| $in\_data1 | Array | [infoID]=>4, [type]=>promo, [title]=>Buy one get one free, [details]=>Buy one fluoride application course now get one more course for free available until the end of the year, [officerID]=>OF001 |
| $in\_data2 | Array | [infoID]=>5, [type]=>info, [title]=>Why full mouth checkup is important?[details]=>To checkup full mouth can remind you how health of your mouth, [officerID]=>OF001 |
| $expResult | Boolean | true |

**Test Case:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Remark** | **Expected** |
| **data** |
| **1** | Test to save edited data of a promotion to the database | $in\_data1 | - | $expResult |
| **2** | Test to save edited data of an information to the database | $in\_data2 | - | $expResult |

### Unit Test Case 44 (UTC-44): testo\_delete\_info()

**Description:**

Test the function of delete information and promotion from the database: public function delete\_info($infoID)

**Test Data: Refer to Appendix at the end of this Test Plan for test data**

|  |  |  |
| --- | --- | --- |
| **Parameter name** | **Parameter type** | **Value** |
| $in\_infoid1 | Integer | 4 |
| $in\_infoid2 | Integer | 5 |
| $in\_infoid3 | Integer | 6 |
| $expResult1 | Boolean | true |
| $expResult2 | Boolean | false |

**Test Case:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Remark** | **Expected** |
| **infoID** |
| **1** | Test to delete a promotion at infoID from database | $in\_infoid1 | - | $expResult1 |
| **2** | Test to delete an information at infoID from database | $in\_infoid2 | - | $expResult1 |
| **3** | Test to delete an information at infoID that not exist from database | $in\_infoid3 | - | $expResult2 |

### Unit Test Case 45 (UTC-45): testo\_get\_information()

**Description:**

Test the function of get all information and promotion from the database: public function get\_information()

**Test Data: Refer to Appendix at the end of this Test Plan for test data**

|  |  |  |
| --- | --- | --- |
| **Parameter name** | **Parameter type** | **Value** |
| $expResult | Array[0] | [infoID]=>1, [type]=>promo, [title]=>Free checkup, [details]=>Full mouth checkup for free at the clinic, [officerID]=>OF001 |
| Array[1] | [infoID]=>2, [type]=>information, [title]=>EF Line, [details]=>EF Line is tooth braces for children between 4-15 years old, [officerID]=>OF001 |
| Array[2] | [infoID]=>3, [type]=>promo, [title]=>Whitening 30% off, [details]=>End of the year sale, whitening treatment is noe 30% off, [officerID]=>OF002 |

**Test Case:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Remark** | **Expected** |
| **1** | Test gat all information and promotion from database | - | $expResult is already in database | $expResult |

### Unit Test Case 46 (UTC-46): testo\_check\_time()

**Description:**

Test the function of check the appointment time before add patient to a queue: public function check\_time()

**Test Data: Refer to Appendix at the end of this Test Plan for test data**

|  |  |  |
| --- | --- | --- |
| **Parameter name** | **Parameter type** | **Value** |
| $in\_pid1 | Varchar | P001 |
| $in\_pid2 | Varchar | P002 |
| $in\_pid3 | Varchar | P003 |
| $in\_pid4 | Varchar | P004 |
| $expResult1 | Array | “Not today” |
| $expResult2 | Array | “No appointment” |
| $expResult3 | Array | “You missed” |
| $expResult4 | Array | 10:00:00 |

**Test Case:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Remark** | **Expected** |
| **infoID** |
| **1** | Test to check the appointment time with correct day and before appointment time | $in\_pid1 | $expResult4 is already in database | $expResult4 |
| **2** | Test to check the appointment time with wrong day | $in\_pid2 | $expResult1 is already in database | $expResult1 |
| **3** | Test to check the appointment time with no appointment | $in\_pid3 | $expResult2 is already in database | $expResult2 |
| **4** | Test to check the appointment time with correct day but at late appointment time | $in\_pid4 | $expResult3 is already in database | $expResult3 |

### Unit Test Case 47 (UTC-47): testo\_add\_queue()

**Description:**

Test the function of add a paitent to a queue: public function add\_queue($pid,$q)

**Test Data: Refer to Appendix at the end of this Test Plan for test data**

|  |  |  |
| --- | --- | --- |
| **Parameter name** | **Parameter type** | **Value** |
| $in\_pid1 | Varchar | P001 |
| $in\_pid2 | Varchar | P002 |
| $expResult | Boolean | true |

**Test Case:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Remark** | **Expected** |
| **infoID** |
| **1** | Test to add a patient to an empty queue with pid(patientID) | $in\_pid1 | - | $expResult |
| **2** | Test to add a patient to a not empty queue with pid(patientID) | $in\_pid2 | - | $expResult |

### Unit Test Case 48 (UTC-48): testo\_reset\_queue()

**Description:**

Test the function of reset a queue to be empty: public function reset\_queue()

**Test Data: Refer to Appendix at the end of this Test Plan for test data**

|  |  |  |
| --- | --- | --- |
| **Parameter name** | **Parameter type** | **Value** |
| $expResult | Boolean | true |

**Test Case:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Remark** | **Expected** |
| **1** | Test to reset a queue to be empty | - | - | $expResult |

## Unit Testing of Mobile Application

**Class:** PatientMobile\_manage

### Unit Test Case 49 (UTC-49): testpMobile\_getAllTreatment()

**Description:**

Test the function of get all data of table treatment from database: public function getAllTreatment()

**Test Data: Refer to Appendix at the end of this Test Plan for test data**

|  |  |  |
| --- | --- | --- |
| **Parameter name** | **Parameter type** | **Value** |
| $expResult | Array[0] | [tid]=>1, [tName]=>Full mouth checkup, [cost]=>0 |
| Array[1] | [tid]=>2, [tName]=>Composite filling, [cost]=>1000 |
| Array[2] | [tid]=>3, [tName]=>Fluoride application, [cost]=>400 |
| Array[3] | [tid]=>4, [tName]=>Consultation, [cost]=>0 |

**Test Case:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Remark** | **Expected** |
| **1** | Test gat all dental treatment from database | - | $expResult is already in database | $expResult |

### Unit Test Case 50 (UTC-50): testpMobile\_calculate()

**Description:**

Test the cost estimation by using calculate function to calculate the cost of dental treatment: public function calculate($dent\_treat)

**Test Data: Refer to Appendix at the end of this Test Plan for test data**

|  |  |  |
| --- | --- | --- |
| **Parameter name** | **Parameter type** | **Value** |
| $in\_treat1 | Array[0] | [tid]=>1, [tName]=>Full mouth checkup, [cost]=>0 |
| $in\_treat2 | Array[0] | [tid]=>2, [tName]=>Composite filling, [cost]=>1000 |
| Array[1] | [tid]=>3, [tName]=>Fluoride application, [cost]=>400 |
| $result1 | Double | 0 |
| $result2 | Double | 1400 |

**Test Case:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Remark** | **Expected** |
| **treatment** |
| **1** | Test calculate function that use to estimate the cost of dental treatment by input one dental treatment | $in\_treat1 | - | $result1 |
| **2** | Test calculate function that use to estimate the cost of dental treatment by input two dental treatment | $in\_treat2 | - | $result2 |
| **3** | Test calculate function that use to estimate the cost of dental treatment by not input any dental treatment | - | - | $result1 |

### Unit Test Case 51 (UTC-51): testpMobile\_get\_information()

**Description:**

Test to get information and promotion function of dental clinic: public function get\_informaion()

**Test Data: Refer to Appendix at the end of this Test Plan for test data**

|  |  |  |
| --- | --- | --- |
| **Parameter name** | **Parameter type** | **Value** |
| $expResult | Array[0] | [infoID]=>1, [type]=>promo, [title]=>Free checkup, [details]=>Full mouth checkup for free at the clinic, [officerID]=>OF001 |
| Array[1] | [infoID]=>2, [type]=>information, [title]=>EF Line, [details]=>EF Line is tooth braces for children between 4-15 years old, [officerID]=>OF001 |
| Array[2] | [infoID]=>3, [type]=>promo, [title]=>Whitening 30% off, [details]=>End of the year sale, whitening treatment is noe 30% off, [officerID]=>OF002 |

**Test Case:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Remark** | **Expected** |
| **1** | Test gat all information and promotion from database | - | $expResult is already in database | $expResult |

### Unit Test Case 52 (UTC-52): testpMobile\_viewFollowUpByQid()

**Description:**

Test the function of view follow up by question ID for patient to answer each specific question: public function viewFollowUpByQid($qid)

**Test Data: Refer to Appendix at the end of this Test Plan for test data**

|  |  |  |
| --- | --- | --- |
| **Parameter name** | **Parameter type** | **Value** |
| $in\_qid1 | Int | 14 |
| $in\_qid2 | Int | 15 |
| $expResult1 | Array[0] | [qid]=>14, [patientID]=>P001, [dentistID]=>D001, [question]=>How are your composite filling? Is that fine?, [answer]=>null, [qDateTime]=>2014-10-19 11:04:35, [aDateTime]=>null |
| $expResult2 | Array | null |

**Test Case:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Remark** | **Expected** |
| **qid** |
| **1** | Test get follow up question by qid by input already exist qid | $in\_qid1 | $expResult1 is already in database | $expResult1 |
| **2** | Test get follow up question by qid by input not exist qid | $in\_qid2 | - | $expResult2 |

### Unit Test Case 53 (UTC-53): testpMobile\_answer\_data()

**Description:**

Test the function of answer a follow up question: public function answer\_data($data)

**Test Data: Refer to Appendix at the end of this Test Plan for test data**

|  |  |  |
| --- | --- | --- |
| **Parameter name** | **Parameter type** | **Value** |
| $in\_data | Array | [qid]=>14, [patientID]=>P001, [dentistID]=>D001, [question]=>How are your composite filling? Is that fine?, [answer]=>Nothing wrong, still fine, [qDateTime]=>2014-10-19 11:04:35, [aDateTime]=>2014-10-19 19:39.00 |
| $expResult1 | Boolean | true |
| $expResult2 | Boolean | false |

**Test Case:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Remark** | **Expected** |
| **data** |
| **1** | Test answer a follow up question at not answered question | $in\_data | - | $expResult1 |
| **2** | Test answer a follow up question at answered question | $in\_data | - | $expResult2 |

### Unit Test Case 54 (UTC-54): testpMobile\_view\_follow\_up()

**Description:**

Test the function of view all follow up of each patient: public function view\_follow\_up()

**Test Data: Refer to Appendix at the end of this Test Plan for test data**

|  |  |  |
| --- | --- | --- |
| **Parameter name** | **Parameter type** | **Value** |
| $in\_pid | Varchar | P002 |
| $expResult | Array[0] | [qid]=>13, [patientID]=>P002, [dentistID]=>D002, [question]=>Your retainer is ready in next 3 days (2014-10-15) please come an pick it up at the clinic anytime [answer]=>Ok, thank you very much, [qDateTime]=>2014-10-12 09:15:22, [aDateTime]=>2014-10-13 08:00.28 |

**Test Case:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Remark** | **Expected** |
| **patientID** |
| **1** | Test gat all follow up from database | $in\_pid | $expResult is already in database | $expResult |

### Unit Test Case 55 (UTC-55): testpMobile\_view\_detail()

**Description:**

Test the function of view patients’ own information in mobile application: public function own\_detail($patientID)

**Test Data: Refer to Appendix at the end of this Test Plan for test data**

|  |  |  |
| --- | --- | --- |
| **Parameter name** | **Parameter type** | **Value** |
| $in\_pid | Varchar | P001 |
| $expResult | Array[0] | [patientID] => P001 [password] => 1234 [f\_name] => Mickey [l\_name] => Mouse [age] => 80 [gender] => 1 [treatment] => Tooth braces,tooth zeplin [address] => Chaingmai Thailand [tel] => 053111111 [email]=> M.ky@hotmail.com [submit] => Submit |

**Test Case:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Remark** | **Expected** |
| **patientID** |
| **1** | Test gat patients’ own information by using patientID | $in\_pid | $expResult is already in database | $expResult |

# Chapter Four: System Testing

## 4. System testing

### System Test Case 01(STC-01): Patient can view own appointment in mobile application

**Description:**

To test the view appointment function of patient to view own appointment schedule in the appointment page and login is required

**Test script:**

1. Patient login to the mobile application by using patientID and password
2. Patient selects appointment menu

**Data needs:** Patient account, Appointment account

**Test Cases:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | | **Expected** |
| **patientID** | **password** |
| 1 | Test view his/her appointment schedule with correct patientID and password | P001 | 1234 | System shall provide appointment schedule to patient whose patientID is “P001” |
| 2 | Test view his/her appointment schedule with wrong patientID and password | xxxx | xxxx | System does not provide appointment schedule to patient whose patientID is “xxxx” |
| 3 | Test view his/her appointment schedule with empty patientID and password fields | - | - | System does not provide appointment schedule to patient |
| 4 | Test view his/her appointment schedule without login | - | - | System does not provide appointment schedule to patient |

### System Test Case 02(STC-02): Patient can view all appointments from Google calendar in mobile application

**Description:**

To test the view calendar function in mobile application of patient to view all appointment from Google calendar in the calendar page without login

**Test script:**

1. Patient selects calendar menu

**Data needs:** Appointment account

**Test Cases:**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Expected** |
| **Path** |
| 1 | Test view calendar appointment | Click view calendar appointment function | System shall provide calendar appointment in appointment page |
| 2 | Test view calendar appointment with login | Input patientID “P001” and password “1234” | System shall provide calendar with all appointment in appointment page |

### System Test Case 03(STC-03): Patient can view own appointment in website

**Description:**

To test the view patient appointment function of patient to view own appointment schedule in the patient appointment page and login is required

**Test script:**

1. Officer login to the website by using patientID and password
2. Officer selects appointment menu

**Data needs:** Patient account, Appointment account

**Test Cases:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | | **Expected** |
| **patientID** | **password** |
| 1 | Test view his/her appointment schedule with correct patientID and password | P001 | 1234 | System shall provide appointment schedule to patient whose patientID is “P001” |
| 2 | Test view his/her appointment schedule with wrong patientID and password | xxxx | xxxx | System does not provide appointment schedule to patient whose patientID is “xxxx” |
| 3 | Test view his/her appointment schedule with empty patientID and password fields | - | - | System does not provide appointment schedule to patient |
| 4 | Test view his/her appointment schedule without login | - | - | System does not provide appointment schedule to patient |

### System Test Case 04(STC-04): Dentist can view own appointment in website

**Description:**

To test the view dentist appointment function of dentist to view own appointment schedule in the dentist appointment page and login is required

**Test script:**

1. Dentist login to the website by using dentistID and password
2. Dentist selects appointment menu

**Data needs:** Dentist account, Appointment account

**Test Cases:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | | **Expected** |
| **dentistID** | **password** |
| 1 | Test view his/her appointment schedule with correct dentistID and password | D001 | 1234 | System shall provide appointment schedule to dentist whose dentistID is “D001” |
| 2 | Test view his/her appointment schedule with wrong dentistID and password | xxxx | xxxx | System does not provide appointment schedule to dentist whose dentistID is “xxxx” |
| 3 | Test view his/her appointment schedule with empty dentistID and password fields | - | - | System does not provide appointment schedule to patient |
| 4 | Test view his/her appointment schedule without login | - | - | System does not provide appointment schedule to dentist |

### System Test Case 05(STC-05): Visitor can view all appointment from Google calendar in website

**Description:**

To test the view calendar appointment function of visitor to view all appointment from Google calendar in the website without login

**Test script:**

1. Visitor select calendar appointment menu

**Data needs:** Appointment account

**Test Cases:**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Expected** |
| **Path** |
| 1 | Test view calendar appointment | localhost/CI/index.php/Patient\_Controller/callCalendar | System shall provide calendar with all appointment in appointment page |
| 2 | Test view calendar appointment | Click view calendar appointment function | System shall provide calendar with all appointment in calendar appointment page |

### System Test Case 06(STC-06): Officer can view all appointments as a list in website

**Description:**

To test the view all appointment as a list function of officer to display all appointment lists in all appointment page and login is required

**Test script:**

1. Officer login to the website by using officerID and password
2. Officer select appointment menu

**Data needs:** Appointment account, Officer account

**Test Cases:**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Expected** |
| **Path** |
| 1 | Test view all appointment as a list in website | localhost/CI/index.php/Officer\_Controller/callAppointment | System shall provide calendar with all appointment list in schedule in all appointment page |

### System Test Case 07(STC-07): Officer can view patients’ appointment as a list in website

**Description:**

To test the view patients’ own appointment as a list to display patients’ account that is officer select patientID from patient list page

**Test script:**

1. Officer login to the website by using officerID and password
2. Officer selects patient list menu
3. Officer selects one patientID form patient list page

**Data needs:** Officer account, Patient account, Appointment account

**Test Cases:**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Expected** |
| **Path** |
| 1 | Test view patients’ own appointment list | Click patientID “P001” | System shall provide patients’ appointment information |
| 2 | Test view patients’ own appointment list | localhost/CI/index.php/Officer\_Controller/patientCalendar/patient | System shall provide patients’ appointment information |

### System Test Case 08(STC-08): Officer can view dentists’ appointment as a list in website

**Description:**

To test the view dentists’ own appointment list to display dentists’ account that is officer select dentistID from dentist list page

**Test script:**

1. Officer login to the website by officer ID and password
2. Officer selects dentist list menu
3. Officer select one dentistID from dentist list page

**Data needs:** Officer account, Dentist account, Appointment account

**Test Cases:**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Expected** |
| **Path** |
| 1 | Test view dentists’ own appointment list | Click dentistID “D001” | System shall provide dentists’ appointment information |
| 2 | Test view dentists’ own appointment list | localhost/CI/index.php/Dentist\_Controller/dentistCalendar/D001 | System shall provide dentists’ appointment information |

### System Test Case 09(STC-09): Patient can view all appointments from Google calendar in the website

**Description:**

To test the view calendar appointment function of patient to view all appointment in the website without login

**Test script:**

1. Patient select calendar appointment menu

**Data needs:** Appointment account

**Test Cases:**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Expected** |
| **Path** |
| 1 | Test view calendar appointment | localhost/CI/index.php/Patient\_Controller/callCalendar | System shall provide calendar with all appointment in calendar appointment page |
| 2 | Test view calendar appointment | Click view calendar appointment menu | System shall provide calendar with all appointment in calendar appointment page |
| 3 | Test view calendar appointment with login | Input patientID “P001” and password “1234” | System shall provide calendar with all appointment in calendar appointment page |

### System Test Case 10(STC-10): Officer can create new appointment

**Description:**

Test creates new appointment function officer.

**Test script:**

1. Officer login to the website by using officerID and password
2. Officer selects create an appointment function
3. Officer input appointment information includes patientID,dentistID,starttime,endtime,date,description
4. Officer press save button

**Data needs:** Officer Account, Appointment account

**Test Cases:**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | | | | | | | **Expected** |
| **patientID** | **dentistID** | **date** | **startTime** | **endTime** | **Treatment** | **Description** |  |
| 1 | Test create new appointment with correct form | P001 | D001 | 2014-08-01 | 09.30 AM | 10.30 AM | Change braces | 100 baht/once | System shall redirect to appointment schedule page and display this appointment in the list |
| 2 | Test create new appointment with duplicate time/date/ and dentist | P001 | D001 | 2014-08-01 | 09.30 AM | 10.30 AM | Change braces | 100 baht/once | System does not redirect to appointment schedule page |
| 3 | Test create new appointment with duplicate date/ and dentist but different time | P001 | D001 | 2014-08-01 | 09.00 AM | 09.30 AM | Change braces | 100 baht/once | System shall redirect to appointment schedule page and display this appointment in the list |
| 4 | Test create new appointment with duplicate dentist but different date | P001 | D001 | 2014-08-02 | 09.30 AM | 10.30 AM | Change braces | 100 baht/once | System shall redirect to appointment schedule page and display this appointment in the list |
| 5 | Test create new appointment with duplicate date and time but different dentist | P001 | D002 | 2014-08-01 | 09.30 AM | 10.30 AM | Change braces | 100 baht/once | System shall redirect to appointment schedule page and display this appointment in the list |
| 3 | Test create new appointment with empty fields | - | - | - | - | - | - | - | System shall provide dialog blog “Please fill out this field” |

### System Test Case 11(STC-11): Officer can edit the appointment

**Description:**

Test edits appointment function of officer.

**Test script:**

1. Officer login to the website by using officerID and password
2. Officer selects all appointment page
3. Officer selects edit function
4. Officer edit appointment information in the edit form
5. Officer press save button

**Data needs:** Officer account, Appointment account

**Test Cases:**

**Test Cases:**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | | | | | | | **Expected** |
| **patientID** | **dentistID** | **date** | **startTime** | **endTime** | **Treatment** | **Description** |  |
| 1 | Test edit an appointment with correct form | P001 | D002 | 2014-08-04 | 09.30 AM | 10.30 AM | Change brances | 100 baht/once | System shall redirect to all appointment page and display this appointment in the list |
| 2 | Test edit appointment with duplicate time/date/ and dentist | P001 | D002 | 2014-08-05 | 09.30 AM | 10.30 AM | Change brances | 100 baht/once | System does not redirect to all appointment page |
| 3 | Test edit appointment with empty fields | - | - | - | - | - | - | - | System shall provide dialog blog “Please fill out this field” |

### System Test Case 12(STC-12): Officer can delete the appointment

**Description:**

Test delete appointment function of officer.

**Test script:**

1. Officer login to the website by using officerID and password
2. Officer selects all appointment page
3. Officer selects delete function
4. Officer press save button

**Data needs:** Officer account, Appointment account

**Test Cases:**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Expected** |
| **Path** |
| 1 | Test delete an appointment | localhost/CI/index.php/Officer\_Controller/app\_delete/1 | System shall provide appointment schedule in all appointments page without the appointment which is appointmentID 1 as selected |
| 2 | Test delete an appointment | Click delete function at the appointment that want to delete | System shall provide appointment schedule in all appointments page without the appointment appointmentID number 1 as selected |
| 3 | Test delete an appointment | - | System does not provide appointment schedule in the all appointments page |

### System Test Case 13(STC-13): Officer can save the appointment to Google calendar

**Description:**

To test the save appointment to Google calendar function with correct appointments’ information and display in Google calendar which is provide in the website

**Data needs:** Officer account, Appointment account, Google emails’ account of dental clinic,

**Test Cases:**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Expected** |
| **Path** |
| 1 | Test save appointment to Google calendar | Click save to Google calendar function | System shall provide Google calendar and click save for saving  The saving appointment will display in the calendar appointment page in DCSS website |

### System Test Case 14(STC-14): Patient can login to the mobile application

**Description:**

To test the login function of patient that can login to the mobile application with correct patientID and password and cannot login to the mobile application if there are wrong patientID or password

**Test script:**

1. Patient enters to the mobile application select login function.
2. Patient inputs patientID and password in login form.
3. Patient selects login button.

**Data needs:** Patient account

**Test Cases:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | | **Expected** |
| **patientID** | **password** |
| 1 | Test login with correct patientID and password | P001 | 1234 | System shall provide dialog blog “You’re login” |
| 2 | Test login with wrong patientID and password | xxxx | xxxx | System shall provide dialog blog “Your login fail” |
| 3 | Test login with empty fields | - | - | System shall provide dialog blog “You must enter patientID and password” |

### System Test Case 15(STC-15): Patient can login to the website

**Description:**

To test the login function of patient that can login to the website with correct patientID and password and cannot login to the website if there are wrong patientID or password

**Test script:**

1. Patient login to the website select login function.
2. Patient input patientID and password in login form.
3. Patient selects login button

**Data needs:** Patient account

**Test Cases:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | | **Expected** |
| **patientID** | **password** |
| 1 | Test login with correct patientID and password | P001 | 1234 | System shall provide redirect to index page |
| 2 | Test login with wrong patientID and password | xxxx | xxxx | System shall provide dialog blog “Invalid patientID or password” |
| 3 | Test login with empty fields | - | - | System shall provide dialog blog “Please fill out this field ” |

### System Test Case 16(STC-16): Patient can logout from the mobile application

**Description:**

To test the logout from mobile application

**Test script:**

1. Patient selects logout menu

**Data needs:**

**Test Cases:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | | **Expected** |
| **patientID** | **password** |
| 1 | Test logout with patientID and password | P001 | 1234 | System shall redirect to the index page and does not show the patientID in any page after logout |

### System Test Case 17(STC-017): Patient can logout from the website

**Description:**

To test the logout from the website

**Test script:**

1. Patient selects logout function
2. The system shall provide index page without the patientID on the index page

**Data needs:**

**Test Cases:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | | **Expected** |
| **patientID** | **password** |
| 1 | Test logout with patientID and password | P001 | 1234 | System shall redirect to the index page and does not show the patientID in any page after logout |

### System Test Case 18(STC-18): Officer can login into the websites

**Description:**

To test the login function of officer that can login to the website with correct officerID and password and cannot login to the website if there are wrong officerID or password

**Test script:**

1. Officer enters to the website select login function.
2. Officer inputs officerID and password in login form.
3. Officer selects login button.

**Data needs:** Officer account

**Test Cases:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | | **Expected** |
| **officerID** | **password** |
| 1 | Test login with correct officerID and password | OF001 | 1234 | System shall provide redirect to index page |
| 2 | Test login with wrong officerID and password | xxxx | xxxx | System shall provide dialog blog “Invalid officerID or password” |
| 3 | Test login with empty fields | - | - | System shall provide dialog blog “Please fill out this field ” |

### System Test Case 19(STC-19): Officer can logout from website

**Description:**

To test the logout from the website

**Test script:**

1. Officer selects logout function

**Data needs:**

**Test Cases:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | | **Expected** |
| **officerID** | **password** |
| 1 | Test logout with officerID and password | OF001 | 1234 | System shall redirect to the index page and does not show the officerID in any page after logout |

### System Test Case 20(STC-20): Officer can create an account for new patient

**Description:**

Test create new patient account function of officer that can register for new patient.

**Test script:**

1. Officer login to the website by using officerID and password
2. Officer selects create patient account function
3. Officer inputs patients’ information that includes patientID, password, firstname, surname, age, gender, address,tel., email, treatment
4. Officer click submit button for saving

**Data needs:** Officer account, Patient account

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | | | | | | | | | **Expected** |
| **patientID** | **password** | **f\_name** | **l\_name** | **age** | **treatment** | **address** | **tel** | **email** |
| 1 | Test create patient account with correct form | P001 | 1234 | Mickey | Mouse | 80 | Tooth braces | Chaingmai,Thailand | 053111111 | M.ky@hotmail.com | System shall redirect to the patient list page and show account information whose patientID is “patient” |
| 2 | Test create account with duplicate patientID | P001 | xxxx | Minnie | Mouse | 80 | Tooth braces | Chaingmai,Thailand | 0911234568 | Minnie.M@hotmail.com | System does not redirect to patient list page |
| 3 | Test create account with empty fields | - | - | - | - | - | - | - | - | - | System shall provide dialog blog “Please fill out this field” |

**Test Cases:**

### System Test Case 21(STC-21): Officer can edit patients’ information

**Description:**

Tests edit patient function that can change/edit patients’ information.

**Test script:**

1. Officer login to the website by using officerID and password
2. Officer selects patient list page
3. Officer selects edit function
4. Officer input some information in the edit form for changing
5. Officer clicks submit button

**Data needs:** Officer account, Patient account

**Test Case:**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | | | | | | | | | **Expected** |
| **patientID** | **password** | **f\_name** | **l\_name** | **age** | **treatment** | **address** | **tel** | **email** |
| 1 | Test edit patient account with correct form | P001 | 1234 | Mickey | Mouse | 80 | Tooth braces, ,tooth zeplin | Chaingmai,Thailand | 053111111 | M.ky@hotmail.com | System shall redirect to the patient list page and show new account information that has edited already, whose patientID is “patient” |
| 2 | Test edit account with duplicate patientID | P001 | xxxx | Minnie | Mouse | 80 | Tooth braces | Chaingmai,Thailand | 0911234568 | Minnie.M@hotmail.com | System does not redirect to patient list page |
| 3 | Test edit account with empty fields | - | - | - | - | - | - | - | - | - | System shall provide dialog blog “Please fill out this field” |

### System Test Case 22(STC-22): Officer can delete patients’ account

**Description:**

Test delete patient function in patient list page that will remove patients’ account from the database.

**Test script:**

1. Officer login to the website by using officerID and password
2. Officer selects patient list page
3. Officer selects delete function

**Data needs:** Officer account, Patient account

**Test Cases:**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Expected** |
| **Path** |
| 1 | Test delete patients’ account | localhost/CI/index.php/Officer\_Controller/p\_delete/ patient | System shall provide patients’ schedule in patient list page without patient whose patientID is “patient” |
| 2 | Test delete patients’ account | Click view delete function at patientID that want to delete | System shall provide patients’ schedule in patient list page without patient whose patientID is “patient” |
| 3 | Test delete patients’ account | - | System does not provide patients’ schedule in the patient list page |

### System Test Case 23(STC-23): Officer can create dentist account

**Description:**

Test create new dentist account function of officer that can register for new dentist.

**Test script:**

1. Officer login to the website by using officerID and password
2. Officer selects create dentist account function
3. Officer input dentists’ information that includes dentisttID,password,firstname,surname,age,gender,address,tel.,email
4. Officer clicks submit button for saving

**Data needs:** Officer account, Dentist account

**Test Cases:**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | | | | | | | | **Expected** |
| **dentistID** | **role** | **password** | **f\_name** | **l\_name** | **address** | **email** | **tel** |
| 1 | Test create dentist account with correct form | D001 | Dentist | 1234 | Donald | Duck | Chaingmai,Thailand | DD@hotmail.com | 0810000000 | System shall redirect to the dentist list page and show account information whose dentistID is “D001” |
| 2 | Test create account with duplicate dentistID | D001 | Dentist | 1234 | Daisy | Duck | New York-USA | D\_lady@hotmail.com | 0810000000 | System does not redirect to dentist list page |
| 3 | Test create account with empty fields | - | - | - | - | - | - | - |  | System shall provide dialog blog “Please fill out this field” |

### System Test Case 24(STC-24): Officer can edit dentists’ information

**Description:**

Test the edit dentist function that can change/edit dentists’ information.

**Test script:**

1. Officer login to the website by using officerID and password
2. Officer selects dentist list page
3. Officer selects edit function
4. Officer input some information in the edit form for changing
5. Officer clicks submit button

**Data needs:** Officer account, Dentist account

**Test Cases:**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | | | | | | | | **Expected** |
| **dentistID** | **role** | **password** | **f\_name** | **l\_name** | **address** | **email** | **tel** |
| 1 | Test edit dentists’ account | D001 | Dentist | 1234 | Donald | Duck | NewYork,USA | DD@hotmail.com | 0810000000 | System shall redirect to the dentist list page and show new account information that has edited already, whose dentistID is “D001” |
| 2 | Test create account with duplicate dentistID | D001 | Dentist | 1234 | Daisy | Duck | NewYork,USA | D\_lady@hotmail.com | 0810000000 | System does not redirect to dentist list page |
| 3 | Test create account with empty fields | - | - | - | - | - | - | - |  | System shall provide dialog blog “Please fill out this field” |

### System Test Case 25(STC-25): Officer can delete dentists’ account

Description:

Test delete function in dentist list that will remove information from the database.

**Test script:**

1. Officer login to the website by using officerID and password
2. Officer selects dentist list page
3. Officer selects delete function

**Data needs:** Officer account, Dentist account

**Test Cases:**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Expected** |
| **Path** |
| 1 | Test delete dentists’ account | localhost/CI/index.php/Officer\_Controller/get\_db/d\_delete/D001 | System shall provide dentists’ schedule in dentist list page without dentist whose dentistID is “D001” |
| 2 | Test delete dentists’ account | Click view delete function at patientID that want to delete | System shall provide dentists’ schedule in dentist list page without dentist whose dentistID is “D001” |
| 3 | Test delete dentists’ account | - | System does not provide dentists’ schedule in the dentist list page |

### System Test Case 26(STC-26): Officer can view all patients’ account as a list

**Description:**

Test officer can view patient list in the patient list page.

**Test script:**

1. Officer login to the website by using officerID and password
2. Officer selects patient list page

**Data needs:** Officer account , Patient account

**Test Cases:**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Expected** |
| **Path** |
| 1 | Test view patient list in the patients’ schedule | localhost/CI/index.php/Officer\_Controller/get\_p\_db | System shall provide patients’ schedule in the patient list page |
| 2 | Test view patient list in the patients’ schedule | Click view patient list function | System shall provide patients’ schedule in the patient list page |
| 3 | Test view patient list in the patients’ schedule without login | - | System does not provide patients’ schedule in the patient list page |

### System Test Case 27(STC-27): Officer can view all dentists’ account as a list

**Description:**

Test officer can view dentist list in the dentist list page.

**Test script:**

1. Officer login to the website by using officerID and password
2. Officer select dentist list page

**Data needs:** Officer account , Dentist account

**Test Cases:**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Expected** |
| **Path** |
| 1 | Test view dentist list in the patients’ schedule | localhost/CI/index.php/Officer\_Controller/get\_d\_db | System shall provide dentists’ schedule in the dentist list page |
| 2 | Test view dentist list in the dentists’ schedule | Click view dentist list function | System shall provide dentists’ schedule in the dentist list page |
| 3 | Test view patient list in the patients’ schedule without login | - | System does not provide dentists’ schedule in the dentist list page |

### System Test Case 28(STC-28): Officer can view all appointments from Google calendar in website

**Description:**

To test the view all appointment function of user to view the appointment schedule in the appointment page without login

**Test script:**

1. Officer selects calendar appointment menu

**Data needs:** Appointment account

**Test Cases:**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Expected** |
| **Path** |
| 1 | Test view all appointment as a calendar | localhost/CI/index.php/Officert\_Controller/callCalendar | System shall provide calendar with all appointment in calendar appointment page |
| 2 | Test view all appointment as a calendar | Click view calendar appointment menu | System shall provide calendar with all appointment in calendar appointment page |
| 3 | Test view all appointment as a calendar | Input officerID “OF001” and password “1234” | System shall provide calendar with all appointment in calendar appointment page |

### System Test Case 29(STC-29): Dentist login to the website

**Description:**

To test the login function of dentist that can login to the website with correct dentistID and password and cannot login to the website if there are wrong dentistID or password

**Test script:**

1. Dentist selects login function
2. Dentist inputs dentistID and password in login form.
3. Dentist selects login button.

**Data needs:** Dentist Account

**Test Cases:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | | **Expected** |
| **dentistID** | **password** |
| 1 | Test login with correct dentistID and password | D001 | 1234 | System shall provide redirect to index page |
| 2 | Test login with wrong dentistID and password | xxxx | xxxx | System shall provide dialog blog “Invalid dentistID or password” |
| 3 | Test login with empty fields | - | - | System shall provide dialog blog “Please fill out this field ” |

### System Test Case 30(STC-30): Dentist logout from the website

**Description:**

Test the logout function of dentist.

1. Dentist selects logout function

**Data needs:**

**Test Cases:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | | **Expected** |
| **detistID** | **password** |
| 1 | Test logout with dentistID and password | D001 | 1234 | System shall redirect to the index page and does not show the dentistID in any page after logout |

### System Test Case 31(STC-31): Dentist can view all appointments from Google calendar in website

**Description:**

Test the view all appointment schedule function of dentist.

**Test script:**

1. Dentist selects calendar appointment menu

**Data needs:** Appointment account, Dentist account

**Test Cases:**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Expected** |
| **Path** |
| 1 | Test view all appointment as a calendar | localhost/CI/index.php/Dentist\_Controller/callCalendar | System shall provide calendar with all appointment in calendar appointment page |
| 2 | Test view all appointment as a calendar | Click view calendar appointment menu | System shall provide calendar with all appointment in calendar appointment page |
| 3 | Test view all appointment as a calendar | Input dentistID “D001” and password “1234” | System shall provide calendar with all appointment in calendar appointment page |

### System Test Case 32(STC-32): Patient can view their QR code from mobile application

**Description:**

Test the patient can view their QR code in mobile application

**Test script:**

1. Patient login to the mobile application by using patientID and password
2. Patient select QR code function in the mobile application

**Data needs:** Patient Account

**Test Cases:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | | **Expected** |
| **patientID** | **password** |
| 1 | Test display QR code with patientID | P001 | 1234 | System shall display the patient detail including the QR code of patientID “P001” |
| 2 | Test display QR code with patientID but incorrect password | P001 | xxxx | System shall not display the patient detail including the QR code of patientID “P001” |

### System Test Case 33(STC-33): Patient can view their QR code from website

**Description:**

Test the patient can view their QR code from website

**Test script:**

1. Patient login to the website by using patientID and password
2. Patient select QR code function in the mobile application

**Data needs:** Patient Account

**Test Cases:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | | **Expected** |
| **patientID** | **password** |
| 1 | Test display QR code with patientID | P001 | 1234 | System shall display the patient detail including the QR code of patientID “P001” |
| 2 | Test display QR code with patientID but incorrect password | P001 | xxxx | System shall not display the patient detail including the QR code of patientID “P001” |

### System Test Case 34(STC-34): Officer can add patient to a queue

**Description:**

Test the officer can add patient to a queue

**Test script:**

1. Officer login to the website by using officerID and password
2. Officer selects add queue menu

**Data needs:** Officer Account, Patient Account

**Test Cases:**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Expected** |
| **patientID** |
| 1 | Test add queue by using patientID | P001 | System shall add P001 in the available time after checking appointment date and time |
| 2 | Test add queue by using patientID in unavailable time | P001 | System shall not add P001 in the available time after checking appointment date and time |

### 

### System Test Case 35(STC-35): Reset queue

**Description:**

Test the officer can reset a queue every end of the day to clear the queue for the next day

**Test script:**

1. Officer login to the website by using officerID and password
2. Officer select the reset queue button in the add queue page

**Data needs:** Officer Account, Patient Account, queue

**Test Cases:**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Expected** |
| **path** |
| 1 | Test reset queue | Select reset queue button | System shall reset the list of patient queue to be empty |

### System Test Case 36(STC-36): Dentist can view his/her patient follow up message as a list

**Description:**

Test the dentist can view all patient follow up message as list in follow up page

**Test script:**

1. Dentist login to the website by using dentistID and password
2. Dentist selects follow up menu

**Data needs:** Patient Account

**Test Cases:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | | **Expected** |
| **detistID** | **password** |
| 1 | Test display patient follow up message by using dentistID | D001 | 1234 | System shall display the patient follow up message of dentistID “D001” |
| 2 | Test display patient follow up message by using dentistID but incorrect password | D001 | xxxx | System shall not display the patient follow up message of dentistID “D001” |
| 3 | Test display patient follow up message without login | - | - | System shall not display the patient follow up message |

### System Test Case 37(STC-37): Dentist can post question to follow up his/her patient

**Description:**

Test the dentist can input the question to follow up his/her patient

**Test script:**

1. Dentist login to the website by using dentistID and password
2. Dentist selects follow up menu
3. Dentist selects patientID to asking question for following up
4. Dentist input the question in the form and press submit

**Data needs:** Dentist Account, Follow up Account

**Test Cases:**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Expected** |
| **question** |
| 1 | Test the dentist ‘s question is saved in the database | How are you? | System shall display the dentist’s question and show the follow up status that show number “1” |
| 2 | Test the dentist ‘s question is not saved in the database | How are you? | System shall display the dentist’s question and show the follow up status that show number “0” |
| 3 | Test the dentist ‘s question is not saved in the database if the field is empty | - | System shall alert the message “Please fill out this field” |

### System Test Case 38(STC-38): Patient can view follow up massage that his/her dentist asked in the website as a list

**Description:**

Test the patient can view all dentist follow up message as list in follow up page

**Test script:**

1. Patient login to the website by using patientID and password
2. Patient selects follow up menu

**Data needs:** Patient Account, Follow up Account

**Test Cases:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | | **Expected** |
| **patientID** | **password** |
| 1 | Test display patient follow up message by using patientID | P001 | 1234 | System shall display the dentist follow up message of patientID “P001” |
| 2 | Test display patient follow up message by using patientID but incorrect password | P001 | xxxx | System shall not display the dentist follow up message of patientID “P001” |
| 3 | Test display patient follow up message without login | - | - | System shall not display the follow up message |

### System Test Case 39(STC-39): Patient can answer follow up massage that his/her dentist asked in the website as a list

**Description:**

Test the patient can view all dentist follow up message as list in follow up page

**Test script:**

1. Patient login to the website by using patientID and password
2. Patient selects follow up menu
3. Patient selects questionID to answer question for following up
4. Patient input the answer in the form and press submit

**Data needs:** Patient Account, Follow up Account

**Test Cases:**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Expected** |
| **answer** |
| 1 | Test the patient ‘s answers saved in the database | “I am fine” | System shall display the patient’s answer if the answer is saved the status answer should be “Answered” |
| 2 | Test the patient ‘s answers is not saved in the database | “I am fine” | System shall not display the patient’s answer if the answer is not saved the status answer should be “Answer” |
| 3 | Test the patient’s answer is not saved in the database if the field is empty | - | System shall alert the message “Please fill out this field” |

### System Test Case 40(STC-40): Patient can view follow up massage that his/her dentist asked in mobile application as a list

**Description:**

Test the patient can view all dentist follow up message as list in follow up page

**Test script:**

1. Patient login to the mobile application by using patienttID and password
2. Patient selects follow up menu

**Data needs:** Patient Account, Follow up Account

**Test Cases:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | | **Expected** |
| **patientID** | **password** |
| 1 | Test display patient follow up message by using patientID | P001 | 1234 | System shall display the dentist follow up message of patientID “P001” |
| 2 | Test display patient follow up message by using patientID but incorrect password | P001 | xxxx | System shall not display the dentist follow up message of patientID “P001” |
| 3 | Test display patient follow up message without login | - | - | System shall not display the follow up message |

### System Test Case 41(STC-41): Patient can answer view follow up massage that his/her dentist asked in mobile application as a list

**Description:**

Test the patient can view all dentist follow up message as list in follow up page

**Test script:**

1. Patient login to the website by using patientID and password
2. Patient selects follow up menu
3. Patient selects questionID to answer question for following up
4. Patient input the answer in the form and press submit

**Data needs:** Patient Account, Follow up Account

**Test Cases:**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Expected** |
| **form** |
| 1 | Test the patient ‘s answers is saved in the database | “I am fine” | System shall display the patient’s answer if the answer is saved the status answer should be “Answered” |
| 2 | Test the patient ‘s answers is not saved in the database | “I am fine” | System shall not display the patient’s answer if the answer is not saved the status answer should be “Answer” |
|  | Test the patient’s answer is not saved in the database if the field is empty | - | System shall alert the message “Please fill out this field” |

### System Test Case 42(STC-42): Patient can view the dental clinic information and promotion in the website

**Description:**

Test the patient can view all dental clinic information and promotion in the website

**Test script:**

1. Patient enter to the website
2. Patient selects dental clinic information and promotion

**Data needs:** Information and promotion account

**Test Cases:**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Expected** |
| **path** |
| 1 | Test the patient can view the detail of dental clinic information or promotion | Enter to website | System shall display the dental clinic information or promotion in index page |
| 2 | Test the patient can view the detail of dental clinic information or promotion without internet | Enter to website | System shall not display the dental clinic information or promotion in index page |

### System Test Case 43(STC-43): Patient can view the dental clinic information and promotion in mobile application

**Description:**

Test the patient can view all dental clinic information and promotion in the mobile application

**Test script:**

1. Patient enter to the mobile application
2. Patient selects dental clinic information and promotion

**Data needs:** Information and promotion account

**Test Cases:**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Expected** |
| **path** |
| 1 | Test the patient can view the detail of dental clinic information or promotion | Enter to index page in mobile application | System shall display the dental clinic information or promotion in index page |
| 2 | Test the patient view the detail of dental clinic information or promotion without internet | Enter to index page in mobile application | System shall not display the dental clinic information or promotion in index page |

### System Test Case 44(STC-44): Officer can add the information to the database

**Description:**

Test the officer can add information or promotion into the database

**Test script:**

1. Officer login to the website by using officerID and password
2. Officer selects create information and promotion menu
3. Officer input the detail of dental clinic information or promotion and press submit

**Data needs:** Information and promotion account, Officer Account

**Test Cases:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | | | **Expected** |
| **infoID** | **description** | **type** |  |
| 1 | Test officer input dental clinic information | 1 | “Address 221/3 Chiang Mai 50100” | information | System shall redirect to all information and promotion page. Display the dental clinic information of infoID “1” |
| 2 | Test officer input dental clinic promotion | 2 | “Promotion on December 2014” | promotion | System shall redirect to all information and promotion page. Display the dental clinic promotion of infoID “2” |
| 3 | Test officer input dental clinic information with empty field | 3 | - | information | System shall provide dialog blog “Please fill out this field” |
| 4 | Test officer input dental clinic promotion with empty field | 4 | - | promotion | System shall provide dialog blog “Please fill out this field” |

### System Test Case 45(STC-45): Officer can edit the dental clinic information in the database

**Description:**

Test the officer can edit dental clinic information or promotion in the database

**Test script:**

1. Officer login to the website by using officerID and password
2. Officer selects all information and promotion page
3. Officer selects edit function
4. Officer input the detail of dental clinic information or promotion for editing and press submit

**Data needs:** Information and promotion account, Officer Account

**Test Cases:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | | | **Expected** |
| **infoID** | **description** | **type** |  |
| 1 | Test officer input dental clinic information in description column from “Address 221/3” to “Address 1/4” | 1 | Address 1/4 | information | System shall redirect to all information and promotion page. Display the dental clinic information of infoID “1” the column description should be “Address 1/4”instead “Address 221/3” |
| 2 | Test officer input dental clinic information in description column from “December 2014” to “January 2015” | 2 | January 2015 | promotion | System shall redirect to all information and promotion page. Display the dental clinic promotion of infoID “2” the column description should be “January 2015”instead “December 2014” |
| 3 | Test officer input dental clinic information with empty field | 3 | - | information | System shall provide dialog blog “Please fill out this field” |
| 4 | Test officer input dental clinic promotion with empty field | 4 | - | promotion | System shall provide dialog blog “Please fill out this field” |

### System Test Case 46(STC-46): Officer can delete the dental clinic information from the database

**Description:**

Test the officer can delete dental clinic information or promotion in the database

**Test script:**

1. Officer login to the website by using officerID and password
2. Officer selects all information and promotion page
3. Officer selects delete function
4. Officer input the detail of dental clinic information or promotion for editing and press submit

**Data needs:** Information and promotion account, Officer Account

**Test Cases:**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Expected** |
| **path** |  |
| 1 | Test delete dental clinic information data | localhost/CI/index.php/Officer\_Controller/get\_info/delete\_info/1 | System shall provide the list of dental clinic information and promotion all information and promotion page without the infoID is “1” |
| 2 | Test delete dental clinic promotion data | localhost/CI/index.php/Officer\_Controller/get\_info/delete\_info/2 | System shall provide the list of dental clinic information and promotion all information and promotion page without the infoID is “2” |
| 3 | Test delete dental clinic information data | Click delete function at infoID “1” | System shall provide the list of dental clinic information and promotion all information and promotion page without the infoID is “1” |
| 4 | Test delete dental clinic promotion data | Click delete function at infoID “2” | System shall provide the list of dental clinic information and promotion all information and promotion page without the infoID is “2” |

### System Test Case 47(STC-47): Visitor can view the dental clinic information and promotion

**Description:**

Test the visitor can view all dental clinic information and promotion in the mobile application

**Test script:**

1. Visitor enters to the mobile application
2. Visitor selects dental clinic information and promotion

**Data needs:** Information and promotion account

**Test Cases:**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Expected** |
| **path** |
| 1 | Test the visitor can view the detail of dental clinic information or promotion | Enter to website | System shall display the dental clinic information or promotion in index page |
| 2 | Test the visitor view the detail of dental clinic information or promotion without internet | Enter to website | System shall not display the dental clinic information or promotion in index page |

### 

### System Test Case 48(STC-48): Patient can select the dental treatment to estimate the cost by him/herself

**Description:**

Test the patient can select the dental treatment to estimate the cost of dental treatment

**Test script:**

1. Patient selects the cost estimation page
2. Patient selects the dental treatment and press submit
3. Patient views the total price of the dental treatment

**Data needs:** Dental treatment account

**Test Cases:**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Expected** |
| **path** |
| 1 | Test the patient can select the dental treatment to estimate the cost | Selects dental treatment which are fluoride , price is “2000” and whitening , price is “2000” | System shall display the total price that patient selected and the total price should be “4000” |
| 2 | Test the patient can select the dental treatment to estimate the cost with login of patientID “P001” | Selects dental treatment which are fluoride , price is “2000” and whitening , price is “2000” | System shall display the total price that patient selected and the total price should be “4000” |
| 3 | Test the patient select the dental treatment to estimate the cost without internet | Selects dental treatment which are fluoride , price is “2000” and whitening , price is “2000” | System shall not display the total price that patient selected |

### System Test Case 49(STC-49): Patient can select the dental treatment to estimate the cost by him/herself in the mobile application

**Description:**

Test the patient can select the dental treatment to estimate the cost of dental treatmentin the mobile application

**Test script:**

1. Patient selects the cost estimation page in mobile application
2. Patient selects the dental treatment and press submit
3. Patient views the total price of the dental treatment

**Data needs:** Dental treatment account

**Test Cases:**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Expected** |
| **path** |
| 1 | Test the patient can select the dental treatment to estimate the cost | Selects dental treatment which are fluoride , price is “2000” and whitening , price is “2000” | System shall display the total price that patient selected and the total price should be “4000” |
| 2 | Test the patient can select the dental treatment to estimate the cost with login of patientID “P001” | Selects dental treatment which are fluoride , price is “2000” and whitening , price is “2000” | System shall display the total price that patient selected and the total price should be “4000” |
| 3 | Test the patient select the dental treatment to estimate the cost without internet | Selects dental treatment which are fluoride , price is “2000” and whitening , price is “2000” | System shall not display the total price that patient selected |

### System Test Case 50(STC-50): Visitor can select the dental treatment to estimate the cost by him/herself in the website

**Description:**

Test the visitor can select the dental treatment to estimate the cost of dental treatment

**Test script:**

1. Visitor selects the cost estimation page
2. Visitor selects the dental treatment and press submit
3. Visitor views the total price of the dental treatment

**Data needs:** Dental treatment account

**Test Cases:**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Expected** |
| **path** |
| 1 | Test the visitor can select the dental treatment to estimate the cost | Selects dental treatment which are fluoride , price is “2000” and whitening , price is “2000” | System shall display the total price that visitor selected and the total price should be “4000” |
| 2 | Test the visitor select the dental treatment to estimate the cost without internet | Selects dental treatment which are fluoride , price is “2000” and whitening , price is “2000” | System shall not display the total price that patient selected |

### System Test Case 51(STC-51): Officer can add dental treatment information including the detail and cost into the database

**Description:**

Test the officer can add dental clinic treatment into the database

**Test script:**

1. Officer login to the website by using officerID and password
2. Officer selects create dental treatment menu
3. Officer input the detail of dental treatment and press submit

**Data needs:** Dental treatment Account, Officer Account

**Test Cases:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | | | **Expected** |
| **tid** | **tName** | **cost** |  |
| 1 | Test officer input dental treatment detail | 1 | Whitening | 1000 | System shall redirect to all dental treatment page. Display the dental treatment of tid “1” in the list. |
| 2 | Test officer input dental treatment with empty field | 2 | - | 1000 | System shall provide dialog blog “Please fill out this field” |

### System Test Case 52(STC-52): Officer can edit dental treatment information including the detail and cost in the database

**Description:**

Test the officer can edit dental clinic treatment into the database

**Test script:**

1. Officer login to the website by using officerID and password
2. Officer selects all dental treatment page
3. Officer selects edit function
4. Officer input the detail of dental treatment for editing and press submit

**Data needs:** Dental treatment Account, Officer Account

**Test Cases:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | | | **Expected** |
| **tid** | **tName** | **cost** |  |
| 1 | Test officer input dental treatment detail | 1 | Whitening | 1500 | System shall redirect to all dental treatment page. Display the dental treatment of tid “1” in the list. |
| 2 | Test officer input dental treatment with empty field | 2 | - | 1000 | System shall provide dialog blog “Please fill out this field” |

### System Test Case 53(STC-53): Officer can delete dental treatment information including the detail and cost in the database

**Description:**

Test the officer can delete dental clinic treatment into the database

**Test script:**

1. Officer login to the website by using officerID and password
2. Officer selects all dental treatment page
3. Officer selects delete function

**Data needs:** Dental treatment Account, Officer Account

**Test Cases:**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Expected** |
| **path** |  |
| 1 | Test delete dental treatment data | localhost/CI/index.php/Officer\_Controller/get\_treatment/delete\_dental\_treatment/1 | System shall provide the list of dental treatment in all dental treatment page without the tid is “1” |
| 3 | Test delete dental clinic information data | Click delete function at tid “1” | System shall provide the list of dental treatment in all dental treatment page without the tid is “1” |

### System Test Case 54(STC-54): Officer can view dental treatment information as a list

**Description:**

Test the officer can view dental treatment information as a list

**Test script:**

1. Officer login to the website by using officerID and password
2. Officer selects all dental treatment page

**Data needs:** Dental treatment Account, Officer Account

**Test Cases:**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Expected** |
| **path** |  |
| 1 | Test view dental treatment data as a list | localhost/CI/index.php/Officer\_Controller/get\_treatment/ | System shall provide the list of dental treatment in all dental treatment page |
| 3 | Test view dental treatment data as a list | Click all dental treatment page | System shall provide the list of dental treatment in all dental treatment page |

# Chapter Five: Appendix

**5.1 Patient account :**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **patientID** | **password** | **f\_name** | **l\_name** | **age** | **gender** | **treatment** | **address** | **tel** | **email** |
| P001 | 1234 | Mickey | Mouse | 80 | Male | Tooth braces,tooth zeplin | Chaingmai Thailand | 053111111 | M.ky@hotmail.com |
| P002 | 1111 | Minnie | Mouse | 80 | Female | Tooth braces,tooth zeplin | New York,USA | 0120000000 | M.Nie@hotmail.com |
| P003 | 5555 | Goofy | Goof | 4 | Male | EF line | California,USA | 0230000000 | GG@yahoo.com |

**5.2 Dentist Account**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **dentistID** | **password** | **f\_name** | **l\_name** | **address** | **email** | **tel** |
| D001 | 1234 | Donald | Duck | Chaingmai Thailand | DD@gmail.com | 0810000000 |
| D002 | 6789 | Daisy | Duck | New York,USA | D\_lady@hotmail.com | 0810000001 |

**5.3 Officer Account**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **officerID** | **password** | **f\_name** | **l\_name** | **address** | **email** | **tel** |
| OF001 | 1234 | Bug | Bunny | Chaingmai Thailand | BB@gmail.com | 0811111111 |
| OF002 | 0000 | Duffy | Duck | New York,USA | Duffy@hotmail.com | 0811111112 |

**5.4 Appointment account**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **appointmentID** | **patientID** | **dentistID** | **aDate** | **startTime** | **endTime** | **treatment** | **description** |
| 1 | P001 | D001 | 2014-08-01 | 09:00:00 | 09:30:00 | Tooth braces | 100 baht |
| 2 | P002 | D002 | 2014-08-05 | 13:00:00 | 14:00:00 | Whitening | 1000 baht |