**Dental Clinic Services System**

Test Plan

By

**Kanokwan Maneerat 542115002**

**Worapun Wongkium 542115055**

Department of Software Engineering

College of Arts, Media, and Technology

Chiang Mai University

Project Adviser

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

Mrs. Yun Rim Park

**Test Plan**

**Revision Table**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Document Name** | **Detail** | **Status** | **Date** | **Viewable** | **Reviewer& Responsible** |
| DCSS - Test plan - ver 0.1 | Add all document | Draft | 04/26/2014 | Adviser | Kanokwan&Worapun |
| DCSS - Test plan - ver 0.2 | Edit the unit testing and system testing | Draft | 06/10/2014 | Adviser | Kanokwan&Worapun |
| DCSS - Test plan - ver 0.3 | Edit the unit testing and system testing | Draft | 07/01/2014 | Adviser | Kanokwan&Worapun |
| DCSS - Test plan - ver 0.4 | Edit the unit testing and system testing | Draft | 07/02/2014 | Adviser | Kanokwan&Worapun |
| DCSS - Test plan - ver 0.5 | Edit the unit testing and system testing | Draft | 07/03/2014 | Adviser | Kanokwan&Worapun |
| DCSS - Test plan - ver 0.6 | Edit the unit testing and system testing | Draft | 07/03/2014 | Adviser | Kanokwan&Worapun |
| DCSS - Test plan - ver 0.7 | Double check all document | Draft | 07/07/2014 | Adviser | Kanokwan&Worapun |
| DCSS - Test plan - ver 0.8 | Edit the unit testing and system testing | Draft | 07/19/2014 | Adviser | Kanokwan&Worapun |
| DCSS - Test plan - ver 0.9 | Edit unit testing and appendix | Draft | 07/26/2014 | Adviser | Kanokwan&Worapun |
| DCSS - Test plan - ver 1.0 | Add more test cases and double check | Release | 07/30/2014 | Adviser | Kanokwan&Worapun |

Table of Contents

[Chapter One: Introduction 5](#_Toc394490840)

[1. Introduction 5](#_Toc394490841)

[1.1 Objectives 5](#_Toc394490842)

[1.2 Purpose 6](#_Toc394490843)

[1.3 Project Scope 6](#_Toc394490844)

[1.4 Definition and Acronyms 6](#_Toc394490845)

[1.5 Acronyms and Definitions Acronyms 7](#_Toc394490846)

[Chapter Two: Test Plan and Test Procedure 8](#_Toc394490847)

[2. Test Plan and Test Procedure 8](#_Toc394490848)

[2.1 Test Objectives 8](#_Toc394490849)

[2.2 Scope 8](#_Toc394490850)

[2.3 Test Duration 8](#_Toc394490851)

[2.4 Test Responsibility 9](#_Toc394490852)

[2.5 Test Strategy 9](#_Toc394490853)

[2.6 Result of Testing 9](#_Toc394490854)

[2.7 Test Environment 10](#_Toc394490855)

[Chapter Three: Unit Testing 12](#_Toc394490874)

[3. Unit Testing of Web Application 12](#_Toc394490875)

[Unit Test Case 01 (UTC-01): testp\_login() 12](#_Toc394490876)

[Unit Test Case 02 (UTC-02): testp\_ownCalendar() 13](#_Toc394490877)

[Unit Test Case 03 (UTC-03): testd\_login() 14](#_Toc394490878)

[Unit Test Case 04 (UTC-04): testd\_ownCalendar() 15](#_Toc394490879)

[Unit Test Case 05 (UTC-05): testo\_login() 16](#_Toc394490880)

[Unit Test Case 06 (UTC-06): testget\_p\_db() 17](#_Toc394490881)

[Unit Test Case 07 (UTC-07): testget\_db() 18](#_Toc394490882)

[Unit Test Case 08 (UTC-08): testp\_register() 19](#_Toc394490883)

[Unit Test Case 09 (UTC-09): testp\_edit() 20](#_Toc394490884)

[Unit Test Case 10 (UTC-10): testp\_delete() 21](#_Toc394490885)

[Unit Test Case 11 (UTC-11): testd\_register() 21](#_Toc394490886)

[Unit Test Case 12 (UTC-12): testd\_edit() 22](#_Toc394490887)

[Unit Test Case 13 (UTC-13): testd\_delete() 23](#_Toc394490888)

[Unit Test Case 14 (UTC-14): testview\_appointment() 24](#_Toc394490889)

[Unit Test Case 15 (UTC-15): testmake\_appointment() 25](#_Toc394490890)

[Unit Test Case 16 (UTC-16): testapp\_edit() 26](#_Toc394490891)

[Unit Test Case 17 (UTC-17): testapp\_delete() 27](#_Toc394490892)

[Unit Test Case 18 (UTC-18): testPatientCalendar() 28](#_Toc394490893)

[Unit Test Case 19 (UTC-19): testDentistCalendar() 29](#_Toc394490894)

[Unit Test Case 20 (UTC-20): testp\_edit\_data() 30](#_Toc394490895)

[Unit Test Case 21 (UTC-21): testd\_edit\_data() 31](#_Toc394490896)

[Unit Test Case 22 (UTC-22): testapp\_edit\_data() 32](#_Toc394490897)

[3.2 Unit Testing of Mobile Application 33](#_Toc394490898)

[Class: PatientMobile\_manage 33](#_Toc394490899)

[Unit Test Case 23 (UTC-23): testMobileLogin() 33](#_Toc394490900)

[Unit Test Case 24 (UTC-24): testOwnAppointmentMobile() 34](#_Toc394490901)

[Chapter Four: System Testing 35](#_Toc394490902)

[4. System testing 35](#_Toc394490903)

[System Test Case 01(STC-01): Patient can view own appointment in mobile application 35](#_Toc394490904)

[System Test Case 02(STC-02): Patient can view all appointments from Google calendar in mobile application 36](#_Toc394490905)

[System Test Case 03(STC-03): Patient can view own appointment in website 37](#_Toc394490906)

[System Test Case 04(STC-04): Dentist can view own appointment in website 38](#_Toc394490907)

[System Test Case 05(STC-05): Visitor can view all appointment from Google calendar in website 39](#_Toc394490908)

[System Test Case 06(STC-06): Officer can view all appointments as a list in website 40](#_Toc394490909)

[System Test Case 07(STC-07): Officer can view patients’ appointment as a list in website 41](#_Toc394490910)

[System Test Case 08(STC-08): Officer can view dentists’ appointment as a list in website 42](#_Toc394490911)

[System Test Case 09(STC-09): Patient can view all appointments from Google calendar in the website 43](#_Toc394490912)

[System Test Case 10(STC-10): Officer can create new appointment 44](#_Toc394490913)

[System Test Case 11(STC-11): Officer can edit the appointment 47](#_Toc394490914)

[System Test Case 12(STC-12): Officer can delete the appointment 49](#_Toc394490915)

[System Test Case 13(STC-13): Officer can save the appointment to Google calendar 50](#_Toc394490916)

[System Test Case 14(STC-14): Patient can login to the mobile application 51](#_Toc394490917)

[System Test Case 15(STC-15): Patient can login to the website 52](#_Toc394490918)

[System Test Case 16(STC-16): Patient can logout from the mobile application 53](#_Toc394490919)

[System Test Case 17(STC-017): Patient can logout from the website 53](#_Toc394490920)

[System Test Case 18(STC-18): Officer can login into the websites 54](#_Toc394490921)

[System Test Case 19(STC-19): Officer can logout from website 55](#_Toc394490922)

[System Test Case 20(STC-20): Officer can create an account for new patient 56](#_Toc394490923)

[System Test Case 21(STC-21): Officer can edit patients’ information 58](#_Toc394490924)

[System Test Case 22(STC-22): Officer can delete patients’ account 60](#_Toc394490925)

[System Test Case 23(STC-23): Officer can create dentist account 61](#_Toc394490926)

[System Test Case 24(STC-24): Officer can edit dentists’ information 63](#_Toc394490927)

[System Test Case 25(STC-25): Officer can delete dentists’ account 65](#_Toc394490928)

[System Test Case 26(STC-26): Officer can view all patients’ account as a list 66](#_Toc394490929)

[System Test Case 27(STC-27): Officer can view all dentists’ account as a list 67](#_Toc394490930)

[System Test Case 28(STC-28): Officer can view all appointments from Google calendar in website 68](#_Toc394490931)

[System Test Case 29(STC-29): Dentist login to the website 69](#_Toc394490932)

[System Test Case 30(STC-30): Dentist logout from the website 70](#_Toc394490933)

[System Test Case 31(STC-31): Dentist can view all appointments from Google calendar in website 71](#_Toc394490934)

[5. Appendix 72](#_Toc394490935)

# 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, he/she can make an appointment with a dentist through the application, and the dentist can use the system for checking patients’ 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.

### 1.4 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.

### 1.5 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: 14th November 2014 – 17th November 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 CC
* Zxing qr code generater

# 

# 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(p.67) 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 Appendix(p.67) 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(p.67) 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(p.67) 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(p.68) 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(p.67) 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(p.68) 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 dentist 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(p.67) 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(p.67) 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(p.67) 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(p.68) 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(p.68) 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(p.68) 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(p.68) 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] => 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(p.68) for test data**

|  |  |  |
| --- | --- | --- |
| **Parameter name** | **Parameter type** | **Value** |
| $data | Array | [patientID] => P003 [dentistID] => D003 [aDate] => 2014-08-04 [startTime] => 13:00:00 [endTime] => 13:30:00 [treatment] => Whitening [description] => 4500 THB [submit] => Submit |
| $data2 | Array | [patientID] => P003 [dentistID] => D001 [aDate] => 2014-08-01 [startTime] => 09:00:00 [endTime] => 09:30:00 [treatment] => Whitening [description] => 4500 THB [submit] => Submit |
| $data3 | Array | [patientID] => P003 [dentistID] => D001 [aDate] => 2014-08-01 [startTime] => 09:30:00 [endTime] => 10:00:00 [treatment] => Whitening [description] => 4500 THB [submit] => Submit |
| $data4 | Array | [patientID] => P003 [dentistID] => D001 [aDate] => 2014-08-02 [startTime] => 09:30:00 [endTime] => 10:00:00 [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(p.68) 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 appointmentID to see that system can retrieve data from database correctly or not : public function patientCalendar($patientID)

**Test Data: Refer to Appendix(p.68) 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 appointmentID to see that system can retrieve data from database correctly or not : public function dentistCalendar($dentistID)

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

|  |  |  |
| --- | --- | --- |
| **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:**

|  |  |  |
| --- | --- | --- |
| **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:**

|  |  |  |
| --- | --- | --- |
| **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 |

## 3.2 Unit Testing of Mobile Application

### 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.

**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

**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 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 |

# 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 |

# 5. Appendix

**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 |

**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 |

**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 |

**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 |