**Dental Clinic Services System**

Progress 2

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 Longani

**Test Plan**

**Revision Table**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Document Name** | **Detail** | **Status** | **Date** | **Viewable** | **Reviewer& Responsible** |
| DCSS - Test plan - ver 0.1 | Add all document | Draft | 10/18/2014 | Adviser | Kanokwan&Worapun |
| DCSS - Test plan - ver 0.2 | Edit all document | Draft | 10/19/2014 | Adviser | Kanokwan&Worapun |
| DCSS - Test plan - ver 0.3 | Correct numbering and add more Test cases | Draft | 10/20/2014 | Adviser | Kanokwan&Worapun |
| DCSS - Test plan - ver 0.4 | Correct numbering and add more Test cases | Draft | 10/20/2014 | Adviser | Kanokwan&Worapun |
| DCSS - Test plan - ver 0.5 | Double check | Draft | 10/21/2014 | Adviser | Kanokwan&Worapun |

Table of Contents

[Chapter One: Introduction 5](#_Toc401695394)

[1. Introduction 5](#_Toc401695395)

[1.1 Objectives 5](#_Toc401695396)

[1.2 Purpose 5](#_Toc401695397)

[1.2 Project Scope 5](#_Toc401695398)

[1.3 User Classes and Characteristics 6](#_Toc401695399)

[1.4 Definition and Acronyms 7](#_Toc401695400)

[1.5 Acronyms and Definitions Acronyms 8](#_Toc401695401)

[Chapter Two: Test Plan and Test Procedure 9](#_Toc401695402)

[2. Test Plan and Test Procedure 9](#_Toc401695403)

[2.1 Test Objectives 9](#_Toc401695404)

[2.2 Scope 9](#_Toc401695405)

[2.3 Test Duration 9](#_Toc401695406)

[2.4 Test Responsibility 9](#_Toc401695407)

[2.5 Test Strategy 10](#_Toc401695408)

[2.6 Result of Testing 10](#_Toc401695409)

[2.7 Test Environment 10](#_Toc401695410)

[Chapter Three: Unit Testing 12](#_Toc401695429)

[3. Unit Testing of Web Application 12](#_Toc401695430)

[Unit Test Case 25 (UTC-25): testp\_getAllTreatment() 12](#_Toc401695431)

[Unit Test Case 26 (UTC-26): testp\_calculate() 12](#_Toc401695432)

[Unit Test Case 27 (UTC-27): testp\_get\_information() 14](#_Toc401695433)

[Unit Test Case 28 (UTC-28): testp\_viewFollowUpByQid() 14](#_Toc401695434)

[Unit Test Case 29 (UTC-29): testp\_answer\_data() 16](#_Toc401695435)

[Unit Test Case 30 (UTC-30): testp\_view\_follow\_up() 16](#_Toc401695436)

[Unit Test Case 31 (UTC-31): testp\_view\_detail() 18](#_Toc401695437)

[Unit Test Case 32 (UTC-32): testd\_get\_p\_db() 19](#_Toc401695438)

[Unit Test Case 33 (UTC-33): testd\_follow\_up() 20](#_Toc401695439)

[Unit Test Case 34 (UTC-34): testd\_follow\_up\_data() 21](#_Toc401695440)

[Unit Test Case 35 (UTC-35): testd\_getAllReply() 21](#_Toc401695441)

[Unit Test Case 36 (UTC-36): testo\_addDentalTreatment() 22](#_Toc401695442)

[Unit Test Case 37 (UTC-37): testo\_editDentalTreatment() 23](#_Toc401695443)

[Unit Test Case 38 (UTC-38): testo\_editDentalTreatment\_data() 24](#_Toc401695444)

[Unit Test Case 39 (UTC-39): testo\_deleteDentalTreatment() 25](#_Toc401695445)

[Unit Test Case 40 (UTC-40): testo\_get\_treatment() 26](#_Toc401695446)

[Unit Test Case 41 (UTC-41): testo\_add\_information() 26](#_Toc401695447)

[Unit Test Case 42 (UTC-42): testo\_edit\_info() 27](#_Toc401695448)

[Unit Test Case 43 (UTC-43): testo\_edit\_info\_data() 29](#_Toc401695449)

[Unit Test Case 44 (UTC-44): testo\_delete\_info() 29](#_Toc401695450)

[Unit Test Case 45 (UTC-45): testo\_get\_information() 30](#_Toc401695451)

[Unit Test Case 46 (UTC-46): testo\_check\_time() 31](#_Toc401695452)

[Unit Test Case 47 (UTC-47): testo\_add\_queue() 32](#_Toc401695453)

[Unit Test Case 48 (UTC-48): testo\_reset\_queue() 33](#_Toc401695454)

[Unit Testing of Mobile Application 34](#_Toc401695455)

[Unit Test Case 49 (UTC-49): testpMobile\_getAllTreatment() 34](#_Toc401695456)

[Unit Test Case 50 (UTC-50): testpMobile\_calculate() 34](#_Toc401695457)

[Unit Test Case 51 (UTC-51): testpMobile\_get\_information() 35](#_Toc401695458)

[Unit Test Case 52 (UTC-52): testpMobile\_viewFollowUpByQid() 36](#_Toc401695459)

[Unit Test Case 53 (UTC-53): testpMobile\_answer\_data() 37](#_Toc401695460)

[Unit Test Case 54 (UTC-54): testpMobile\_view\_follow\_up() 38](#_Toc401695461)

[Unit Test Case 55 (UTC-55): testpMobile\_view\_detail() 39](#_Toc401695462)

[Chapter Four: System Testing 40](#_Toc401695463)

[4. System testing 40](#_Toc401695464)

# 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.

## 1.2 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

## 1.3 User Classes and Characteristics

* The system divides users into three 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 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 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.
* **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 add patient to a queue and reset queue.
* Officer can add information and promotion into the website.
* Officer can edit information and promotion from the website.
* Officer can delete information and promotion from the website.
* Officer can add dental treatment and cost into the website.
* Officer can edit dental treatment and cost from the website.
* Officer can delete dental treatment and cost from the website.
* Officer can view all dental treatment.

**Dentist**

This group of user needs to be registered to the system at the dental clinic. They will register their account at the clinic. Dentist uses their dentistID and password to login to the system to use the services. Dentists are able to:

* Dentist can view their follow up question.
* Dentist can post question to follow up their patient

### 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: 17th October 2014 – 22nd October 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 CS6
* Ciqrcode for qr code generator

# 

# Chapter Three: Unit Testing

## 3. Unit Testing of Web Application

**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 |
| $result1 | Double | 0 |
| $result1 | 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 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\_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 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** |
| $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 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 | 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 |
| $expResult2 | Array | null |

**Test Case:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Remark** | **Expected** |
| **tid** |
| **1** | Test to delete a dental treatment by existed tid | $in\_tid1 | $expResult1 is already in database | $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 |
| $result1 | 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 32(STC-32): URS-26: 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): URS-27: 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): URS-28: 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 Acount,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 display QR code with duplicate patientID | P001 | System shall not add P001 in the available time after checking appointment date and time |

System Test Case 35(STC-35): URS-29: 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): URS-30: Dentist can view his/her patient follow up message as a list in follow up page**

**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): URS-31: Dentist can ask his/her patient follow up message in follow up page**

**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** |
| **form** |
| 1 | Test the dentist ‘s question is saved in the database | xxxxx | 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 | xxxxx | System shall display the dentist’s question and show the follow up status that show number “0” |

**System Test Case 38(STC-38): URS-32: Patient can view his/her dentist follow up message as a list in follow up page in the website**

**Description:**

Test the patient can view all dentist 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, Follow up Account

**Test Cases:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | | **Expected** |
| **detistID** | **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): URS-33: Patient can answer his/her dentist follow up message as a list in follow up page in the website**

**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 dentist ‘s question is saved in the database | xxxxx | 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 | xxxxx | System shall display the dentist’s question and show the follow up status that show number “0” |

**System Test Case 40(STC-40): URS-34: Patient can view his/her dentist follow up message as a list in follow up page in the mobile application**

**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** |
| **detistID** | **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): URS-35: Patient can answer his/her dentist follows up message in follow up page in mobile application**

**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 question is saved in the database | xxxxx | System shall display the patient’s question and show the follow up status that show number “1” |
| 2 | Test the patient ‘s question is not saved in the database | xxxxx | System shall display the patient’s question and show the follow up status that show number “0” |

**System Test Case 42(STC-42): URS-36: 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): URS-37: 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 website | 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 website | System shall not display the dental clinic information or promotion in index page |

**System Test Case 44(STC-44): URS-38: 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 | xxxxx | 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 | xxxxx | 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): URS-39: 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 “xxxxx” to “yyyy” | 1 | yyyyy | information | System shall redirect to all information and promotion page. Display the dental clinic information of infoID “1” the column description should be “yyyyy” instead “xxxxx” |
| 2 | Test officer input dental clinic information in description column from “xxxxx” to “yyyy” | 2 | yyyyy | promotion | System shall redirect to all information and promotion page. Display the dental clinic promotion of infoID “2” the column description should be “yyyyy” instead “xxxxx” |
| 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): URS-40: 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): URS-41: 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): URS-42: 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): URS-43: 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): URS-44: 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): URS-45: 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 | xxxxx | 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): URS-46: 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 | xxxxx | 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 53(STC-53): URS-47: 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):** URS-48: 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 |