**Dental Clinic Services System**

Progress 3

Software Design

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

**Software Design**

**Revision Table**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Document Name** | **Detail** | **Status** | **Date** | **Viewable** | **Reviewer& Responsible** |
| DCSS – SDD – v 0.1 | Add all document | Draft |  | Adviser | Kanokwan & Worapun |
| DCSS – SDD – v 0.2 | Change class diagram and class diagram description | Draft |  | Adviser | Kanokwan & Worapun |
| DCSS – SDD – v 0.3 | Edit class diagram and class diagram description | Draft |  | Adviser | Kanokwan & Worapun |
| DCSS – SDD – v 0.4 | Edit class diagram, class diagram description, Sequence diagram | Draft |  | Adviser | Kanokwan & Worapun |
| DCSS – SDD – v 0.5 | Edit class diagram description, Sequence diagram | Draft |  | Adviser | Kanokwan & Worapun |
| DCSS – SDD – v 0.6 | Edit class diagram | Draft |  | Adviser | Kanokwan & Worapun |
| DCSS – SDD – v 0.7 | Edit class diagram and Sequence diagram | Draft |  | Adviser | Kanokwan & Worapun |
| DCSS – SDD – v 0.8 | Edit class diagram and Sequence diagram | Draft |  | Adviser | Kanokwan & Worapun |
| DCSS – SDD – v 1.0 | Double check | Release |  | Adviser | Kanokwan & Worapun |
| DCSS – SDD – v 1.1 | Correct all documents | Release |  | Adviser | Kanokwan & Worapun |

Table of Contents

Chapter One: Introduction 4

1. Introduction 4

1.1 Objectives 4

1.2 Project Scope 4

1.3 User Classes and Characteristics 5

1.4 Operation Environment 6

1.5 Acronyms and Definitions Acronyms 6

Chapter Two: Overall Description 7

2. Overall Description 7

2.1 Product Perspective 7

2.2 Product Features 7

2.3 Design and Implementation Constraints 7

Chapter Three: System Architecture 8

3. System Architecture 8

3.1 Technologies used 9

3.1.1 Adobe® PhoneGap™ Build 9

3.1.2 HTML5 9

3.1.3 MySQL Server 9

3.1.4 Apache web server 2.5.10 9

3.1.5 CodeIgniter 2.1.4 9

Chapter Four: Detail Design 10

4. Detail Design 10

4.1 Class diagram 11

4.1.1 DCSS website class diagram 11

4.1.2 DCSS mobile application class diagram 12

4.2 Class Diagram Description 13

4.2.1 Class Diagram Description (DCSS website) 13

Chapter Five: Sequence Diagram 27

5. Sequence Diagram 28

5.1 DCSS website Sequence Diagram 28

5.2 DCSS mobile application Sequence Diagram 44

Chapter Six: Database Design 50

6. Database Design 50

6.1 Database design 50

6.2 Database Description 51

Chapter Seven: User Interface Design 58

7. User Interface Design 58

7.1 Web design 58

7.2 Mobile application design 64

# 

# 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 Software Design has described in more detail of each design of system and each function flow that the system should do. The document includes the Scope of project, Product function, User characteristic, System architecture, Detail design, Sequence diagram, and User interface design.

## 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 Operation Environment

Dental clinic services system uses web service technology and iOS technology. All the characters must have the Internet connection. Patients who want to use the mobile application must use a smartphone which supports iOS7.

## 1.5 Acronyms and Definitions Acronyms

|  |  |
| --- | --- |
| **Acronyms** | **Stands for** |
| Registered User | Patient, Dentist, and Officer |
| Non-registered User | Visitor |
| PSR | Project Status Report |
| 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: Overall Description

## 2. Overall Description

## 2.1 Product Perspective

Dental clinic services system consists a mobile application on iOS and a web application. The system helps patients save their time to contact with the dental clinic by providing features such as making an appointment, viewing upcoming appointment, and getting cost estimation. It also helps dentists to follow up their patients and view/manage their appointment schedule. The system reduces work load of officers by providing them with features such as using QR code to identify patients at the clinic and managing appointments for patients and dentists. Dental clinic services system integrates the services that are frequently sought and commonly used 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.

## 2.2 Product Features

The dental clinic services system offers seven features.

* Feature 1: Schedule management
* Feature 2: Appointment management
* Feature 3: User registration and authentication
* Feature 4: Patient identification using QR code
* Feature 5: Dental care consulting and following up
* Feature 6: A dental clinic information and promotion
* Feature 7: Cost estimation of dental treatments

Each of features will create a step-by-step suit by priority. In this progress we will develop Feature 4, 5, 6, and 7

## 2.3 Design and Implementation Constraints

**List of project constraints:**

* The application works only on iOS version7.
* The application requires the Internet connection.
* The application is available for the users who are already registered to the system and also people who are not. Some features are offered to the registered users only.

# Chapter Three: System Architecture

## 3. System Architecture

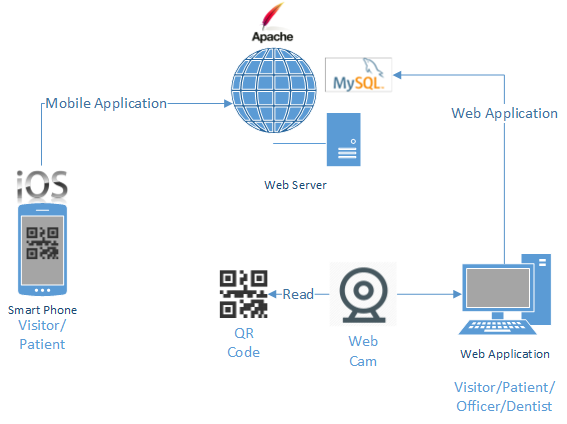


Figure 1: Shows the system architecture of Dental Clinic Services System

**Mobile Application for Smart Phone:** This part uses html language to create a hybrid mobile application from PhoneGap and PhoneGap Build. The application supports smart phones running on iOS only. User who does not login to the system are considered a ‘*Visitor*’. Visitor can view all appointments of the clinic, get cost estimation, view promotion and dental clinic information, and gain basic knowledge of dental treatment. User who uses a patientID and a password to login to the system is called ‘*Patient*’. Patient can view his/her appointment schedule, receive a QR Code that is used for user identification at the clinic, consult with dentists, and use all services provided for visitors.

**Web Application**: Web application can be used by all types of users (Patient, Officer, and Dentist). Patient can use all services provided by both the web application and the mobile application. Officer can use the web application to manage appointments, dentist accounts, patient accounts, and also manage all the data in web application and the mobile application. For example, officer can change the price in cost estimation function and change information of the dental clinic and dentists.

**Web Server:** Web server is the server that supports the services provided for the mobile application as well as the web application. It stores and provides the system data that is accessed via both the web application and the mobile application.

**QR Code and Web Cam:** QR Code is used to identify the patients at the clinic. Each QR Code contains a patientID for the web cam to read. Once the system retrieves the patient information from the QR Code, the information is used to look up the patient’s appointments and manage the waiting queue at the clinic. Web cam is used to read QR Codes.

## 3.1 Technologies used

### 3.1.1 Adobe® PhoneGap™ Build

Adobe® PhoneGap™ Build is an open-source frame work that allow us to upload html, css and javascript to build as a hybrid application for mobile phone. Adobe PhoneGap support many operating system such as iOS, Android, Windows Phone, webOS, Blackbery and more.

### 3.1.2 HTML5

HTML5 is markup language used for creating websites. Evolved from the HTML standard, HTML5 has more features such as working with MAP system, and creating a graphic without flash. It works with CSS and JavaScript and can be used not only to develop for the World Wide Web also but also to create mobile applications.

### 3.1.3 MySQL Server

MySQL Server is a Rational Database Management System(RDBMS) that stores data in separate tables. MySQL Server is an open source database that can be used in many different platforms and is very fast, reliable, scalable, and easy to use. It also provides a high performance, multi-threading, and multiuser rational database management.

### 3.1.4 Apache web server 2.5.10

Apache is a [public-domain](http://www.webopedia.com/TERM/P/public_domain_software.html) [open source](http://www.webopedia.com/TERM/O/open_source.html) [Web server](http://www.webopedia.com/TERM/W/Web_server.html) which provide a web page through the HTTP protocol. It is support in every operating system, including [Unix](http://en.wikipedia.org/wiki/Unix), [FreeBSD](http://en.wikipedia.org/wiki/FreeBSD), [Linux](http://en.wikipedia.org/wiki/Linux), [Solaris](http://en.wikipedia.org/wiki/Solaris_%28operating_system%29), [Novell NetWare](http://en.wikipedia.org/wiki/Novell_NetWare), [OS X](http://en.wikipedia.org/wiki/OS_X), [Microsoft Windows](http://en.wikipedia.org/wiki/Microsoft_Windows), [OS/2](http://en.wikipedia.org/wiki/OS/2), [TPF](http://en.wikipedia.org/wiki/Transaction_Processing_Facility), [OpenVMS](http://en.wikipedia.org/wiki/OpenVMS) and [eComStation](http://en.wikipedia.org/wiki/EComStation).

### 3.1.5 CodeIgniter 2.1.4

CaodeIgniter is a free and open source web application framework created in PHP language. CodeIgniter is designed based on Model-View-Controller(MVC) It works with PHP version 4 and 5, and makes it easier to use CRUD for any databases such as MySQL, Postgress, and SqlServer.

# Chapter Four: Detail Design

## 4. Detail Design

MVC(Model-View-Controller) is one of design pattern. MVC divides the structure of the system into three parts, which are model, view, and controller.

* Model is the form that is used to work with database
* View is the form that brings data from the models and display it in the user interface
* Controller is the form that collects user input and processes it. It also coordinates the view and the model to work together.



Figure 2: Shows the MVC pattern by using CodeIgniter

## 4.1 Class diagram

### 4.1.1 DCSS website class diagram

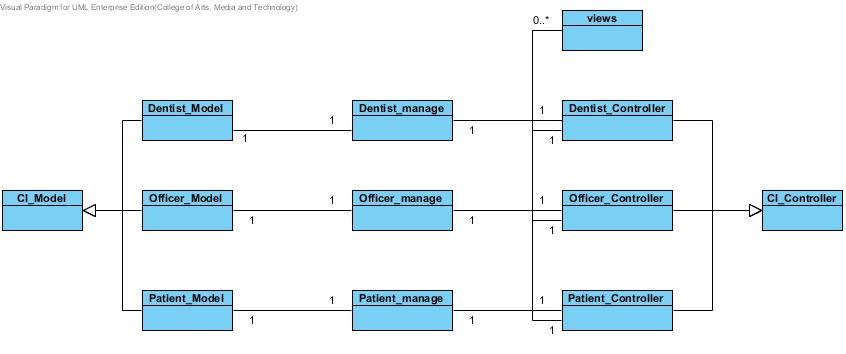


Figure 3: Shows class diagram of Dental Clinic Services System

### 4.1.2 DCSS mobile application class diagram

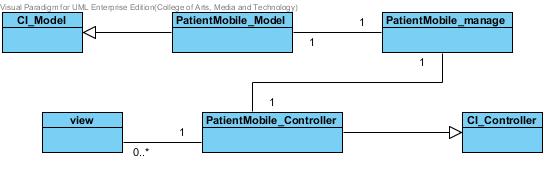
****

Figure 4: Shows class diagram of Dental Clinic Services System in mobile application

## **4.2 Class Diagram Description**

### **4.2.1 Class Diagram Description (DCSS website)**

CD-01, Class name: Dentist\_Model

**Dentist Model.jpg**

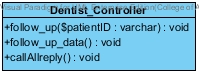
Property

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Name | Description | Remark |
| - | - | - | - |

Method

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Name | Description | Parameters | Return |
| 1 | add\_follow\_up | Method uses to add the message for following up the patient | $data : array[] | boolean |
| 2 | getFollowUpReplyByDid | Method uses to get all follow up data as a list by dentistID | $dentistID : varchar | array[] |

**CD-02, Class name : Dentist\_Controller**



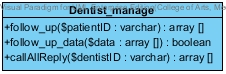
Property

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Name | Description | Remark |
| - | - | - | - |

Method

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Name | Description | Parameters | Return |
| 1 | follow\_up | Method uses to select patient to follow up by patientID | $patientID : varchar | void |
| 2 | follow\_up\_data | Method uses to get data from follow up | - | void |
| 3 | callAllreply | Method uses to get all reply follow up message from patient | - | void |

**CD-03, Class name : Dentist\_manage**



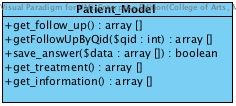
Property

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Name | Description | Remark |
| - | - | - | - |

Method

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Name | Description | Parameters | Return |
| 1 | follow\_up | Method uses to select patient to follow up by patientID | $patientID : varchar | array[] |
| 2 | follow\_up\_data | Method uses to get data from follow up | $data : array[] | boolean |
| 3 | callAllReply | Method uses to get all reply follow up message from Dentist\_Controller | dentistID : varchar | array[] |

CD-04, Class name: Patient\_Model

****

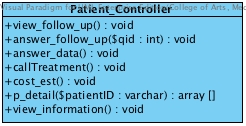
Property

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Name | Description | Remark |
| - | - | - | - |

Method

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Name | Description | Parameters | Return |
| 1 | get\_follow\_up | Method uses to get follow up message as a list from database | - | array[] |
| 2 | getFollowUpByQid | Method use to get the follow up question by question ID | $qid : varchar | array[] |
| 3 | save\_answer | Method uses to save the reply answer | $data : array[] | boolean |
| 4 | get\_tratement | Method uses to get the dental treatment data as a list from the database | - | array[] |
| 5 | get\_information | Method uses to get the dental clinic information and promotion data the database | - | array[] |

CD-05, Class name: Patient\_Controller

****

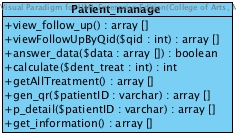
Property

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Name | Description | Remark |
| - | - | - | - |

Method

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Name | Description | Parameters | Return |
| 1 | view\_follow\_up | Method uses to display the follow up data from the database | - | void |
| 2 | answer\_follow\_up($qid) | Method use to select the question to answer by questioned | $qid : int | void |
| 3 | answer\_data | Method uses to get the input data from patient | - | void |
| 4 | callTratement | Method uses to get the dental treatment data from the database | - | void |
| 5 | cost\_est | Method uses to estimate the cost of dental treatment | - | void |
| 6 | p\_datail | Method uses to get the personal information of patient including the QR code | - | void |
| 7 | view\_information | Method uses to display the dental clinic information and promotion | - | void |

CD-06, Class name: Patient\_manage

****

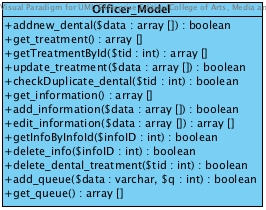
Property

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Name | Description | Remark |
| - | - | - | - |

Method

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Name | Description | Parameters | Return |
| 1 | view\_follow\_up | Method uses to display the follow up data from the database | - | void |
| 2 | viewFollowUpByQid | Method use to display the question that patient selected by questionID | $qid : int | array[] |
| 3 | answer\_data | Method uses to get the input data from Patient\_Controller | $data : array[] | array[] |
| 4 | calculate | Method uses to calculate the dental treatment cost | $dent\_treat : int | int |
| 5 | gen\_qr | Method uses to generate the patient’s information to be the QR code | - | array[] |
| 6 | p\_datail | Method uses to get the personal information of patient including the QR code | - | void |
| 7 | get\_information | Method uses to get the dental clinic information and promotion data the database | - | array[] |

CD-07, Class name: Officer\_Model

****

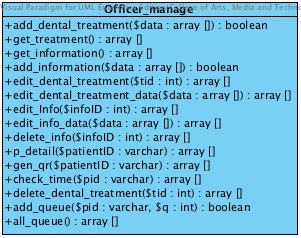
Property

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Name | Description | Remark |
| - | - | - | - |

Method

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Name | Description | Parameters | Return |
| 1 | addnew\_dental | Method uses to add new dental treatment to the database | - | boolean |
| 2 | get\_treatment | Method uses to get all the follow up data from database | - | array[] |
| 3 | getTreatmentById | Method uses to get dental treatment data by treatmentID | $tid : int | array[] |
| 4 | update\_treatment | Method uses to update the dental treatment data | $data : array | boolean |
| 5 | checkDuplicate\_dental | Method uses to check the duplication of the dental clinic data by checking the id of each dental treatment | $tid : int | boolean |
| 6 | get\_infromation | Method uses to get all dental clinic information into the database | - | array[] |
| 7 | add\_information | Method uses to add new dental clinic information or promotion into the database | $data : array[] | boolean |
| 8 | edit\_information | Method uses to edit the dental clinic information or promotion | $data : array[] | array[] |
| 9 | get\_InfoByInfoId | Method uses to get the dental clinic information or promotion data by informationID | $InfoID : int | boolean |
| 10 | delete\_info | Method uses to delete dental clinic information or promotion | $infoID : int | boolean |
| 11 | delete\_dental\_treatment | Method uses to delete dental treatment by tid | $tid : int | boolean |
| 12 | add\_queue | Method uses to add the patient ‘appointment into the queue by using patientID | $pid : varchar  $q : int | boolean |
| 13 | get\_queue | Method uses to get the patient ‘appointment queue from the database | - | array[] |

CD-08, Class name: Officer\_manage

****

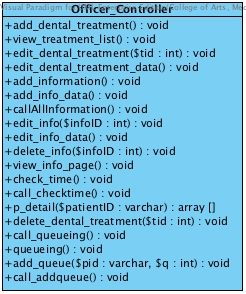
Property

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Name | Description | Remark |
| - | - | - | - |

Method

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Name | Description | Parameters | Return |
| 1 | add\_dental\_treatment | Method uses to add new dental treatment to the database | $data : varchar | boolean |
| 2 | get\_treatment | Method uses to get all dental treatment data from the database | - | array[] |
| 3 | get\_information | Method uses to get all information and promotion as a list | - | array[] |
| 4 | add\_information | Method uses to add new dental clinic information or promotion into the database | - | array[] |
| 5 | edit\_dental\_treatment | Method uses to select dental treatment for editing by tid | $tid : int | array[] |
| 6 | edit\_dental\_treatment\_data | Method uses to edit some dental treatment detail | $data : array[] | array[] |
| 7 | edit\_info | Method uses to select the dental clinic information or promotion by infoID | $infoID : int | array[] |
| 8 | edit\_info\_data | Method uses to edit the dental clinic information or promotion | $data : array[] | array[] |
| 9 | delete\_info | Method uses to delete the dental clinic information or promotion by infoID | $infoID : int | array[] |
| 10 | p\_detail | Method uses to display the patient ‘s detail by patientID | $patientID : varchar | array[] |
| 11 | gen\_qr | Method uses to generate patient’s information to be the QR code | $patientID : varchar | array[] |
| 12 | check\_time | Method uses to check patient appointment time by patientID | $pid : varchar | array[] |
| 13 | delete\_dental\_treatment | Method uses to delete dental treatment by tid | $tid : int | array[] |
| 14 | add\_queue | Method uses to add the patient ‘appointment into the queue by using patientID | $pid : varchar  $q : int | boolean |
| 15 | all\_queue | Method uses to get the patient ‘appointment queue from the database | - | array[] |

CD-09, Class name: Officer\_Controller

****

Property

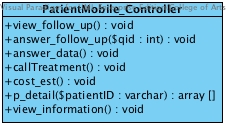
|  |  |  |  |
| --- | --- | --- | --- |
| ID | Name | Description | Remark |
| - | - | - | - |

Method

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Name | Description | Parameters | Return |
| 1 | add\_dental\_treatment | Method uses to add new dental treatment to the database | - | void |
| 2 | view\_treatment\_list | Method uses to display all dental treatment data as a list in the all treatment page in the website | - | void |
| 3 | edit\_dental\_treatement | Method uses to select the dental treatment for editing by tid | $tid : int | void |
| 4 | edit\_dental\_treatment\_data | Method uses to edit dental treatment data | - | void |
| 5 | add\_information | Method uses to add new dental clinic information or promotion to the database | - | void |
| 6 | add\_info\_data | Method uses to add new dental clinic information or promotion to the database | - | void |
| 7 | callAllInformation | Method uses to display all information and promotion as a list | - | void |
| 8 | edit\_info | Method uses to selects the information to editing by InfoID | $infoID : int | void |
| 9 | edit\_info\_data | Method uses to edit some detail of dental clinic information or promotion | - | void |
| 10 | delete\_info | Method uses to delete the dental clinic information or promotion | $infoID : int | void |
| 11 | view\_info\_page | Method uses to display dental clinic information and promotion | - | void |
| 12 | check\_time | Method uses to check patient appointment time | - | void |
| 13 | call\_checkTime | Method uses to get the list of patient appointment after checking the time | - | void |
| 14 | p\_detail | Method uses to display the patient ‘s detail by patientID | $patientID : varchar | array[] |
| 15 | delete\_dental\_treatment | Method uses to delete dental treatment by tid | $tid : int | void |
| 16 | call\_queueing | Method uses to get all the patient’s appointment queue | - | void |
| 17 | queueing | Method uses to check the patient’s appointment queue | - | void |
| 18 | add\_queue | Method uses to add patient’s appointment in to the queue | $pid:varcahr,  $q : int | void |
| 19 | call\_addqueue | Method uses to get the patient’s appointment queue after add the queue | - | void |

4.2.2 Class Diagram Description (DCSS mobile application)

CD-10, Class name: PatientMobile\_Controller

****

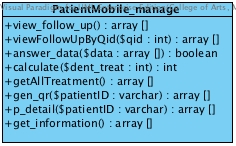
Property

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Name | Description | Remark |
| - | - | - | - |

Method

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Name | Description | Parameters | Return |
| 1 | view\_follow\_up | Method uses to display the follow up data from the database | - | void |
| 2 | answer\_follow\_up($qid) | Method use to select the question to answer by questioned | $qid : int | void |
| 3 | answer\_data | Method uses to get the input data from patient | - | void |
| 4 | call\_treatment | Method uses to get the dental treatment data from the database | - | void |
| 5 | cost\_est | Method uses to gestimate the dental treatment cost | - | void |
| 6 | p\_detail | Method uses to display the patient ‘s detail by patientID | $patientID : varchar | array[] |
| 7 | view\_information | Method uses to display the dental treatment data from the database | - | void |

CD-11, Class name: PatientMobile\_manage

****

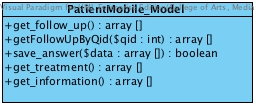
Property

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Name | Description | Remark |
| - | - | - | - |

Method

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Name | Description | Remark | ID |
| 1 | view\_follow\_up | Method uses to display the follow up data from the database | - | void |
| 2 | viewFollowUpByQid | Method use to display the question that patient selected by questionID | $qid : int | array[] |
| 3 | answer\_data | Method uses to get the input data from Patient\_Controller | $data : array[] | array[] |
| 4 | calculate | Method uses to calculate the dental treatment cost | $dent\_treat : int | int |
| 5 | getAllTreatment | Method uses to get all dental treatment form the database | - | array[] |
| 6 | gen\_qr | Method uses to generate the patient’s information to be the QR code | $patientID : varchar | array[] |
| 7 | p\_datail | Method uses to get the personal information of patient including the QR code | $patientID : varchar | void |
| 8 | get\_information | Method uses to get the dental treatment data from the database | - | void |

CD-12, Class name: PatientMobile\_Model

****

Property

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Name | Description | Remark |
| - | - | - | - |

Method

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Name | Description | Parameters | Return |
| 1 | view\_follow\_up | Method uses to display the follow up data from the database | - | void |
| 2 | answer\_follow\_up($qid) | Method use to select the question to answer by questioned | $qid : int | void |
| 3 | answer\_data | Method uses to get the input data from patient | - | void |
| 4 | get\_treatment | Method uses to get the dental treatment data from the database | - | void |
| 5 | get\_information | Method uses to get the dental treatment data from the database | - | void |

# Chapter Five: Sequence Diagram

# Page guideline

|  |  |  |
| --- | --- | --- |
| **SD** | **URS** | **Page** |
| SD-22 | URS-26 | Page 44 |
| SD-37 | URS-27 | Page 41 |
| SD-38 | URS-28 | Page 42 |
| SD-39 | URS-29 | Page 43 |
| SD-40 | URS-48 | Page 49 |

\*\* DCSS website Sequence Diagram SD:

\*\* DCSS mobile application Sequence Diagram

**SD-37: Patient can view their QR code from website**

* **URS-27: Patient can view their QR code from website**

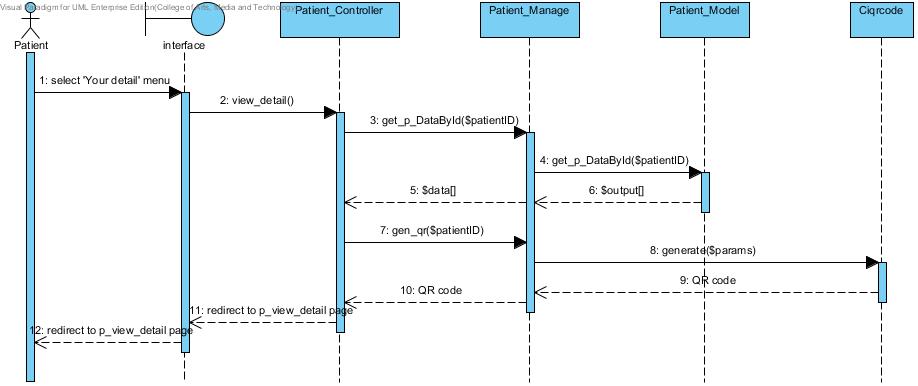
****

Figure 19: View their QR code from website

**SD-38: Officer can add patient to a queue**

* **URS-28: Officer can add patient to a queue**

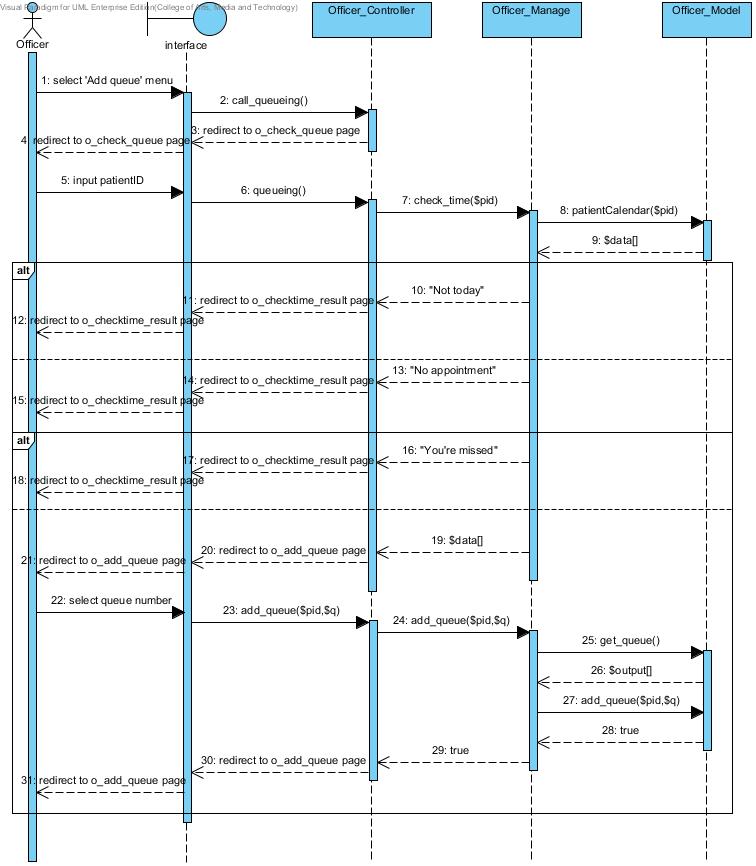
****

Figure 20: Officer can add patient to a queue

**SD-39: Officer can reset queue**

* **URS-29: Officer can reset queue**

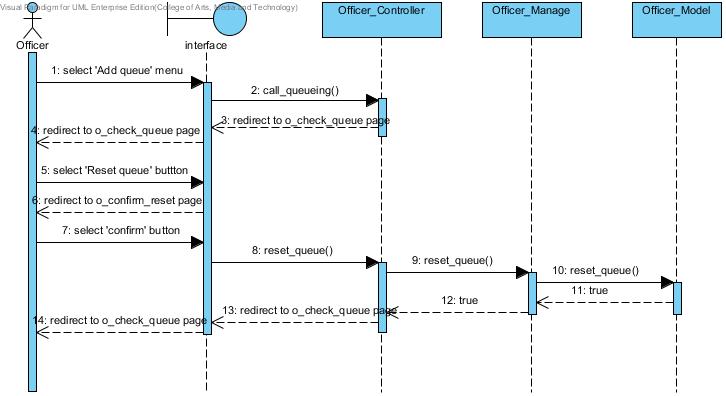


Figure 21: Officer can reset queue

## 5.2 DCSS mobile application Sequence Diagram

**SD-22: Use QR code to identify him/herself**

* **URS-26: Patient can view their QR code from mobile application**

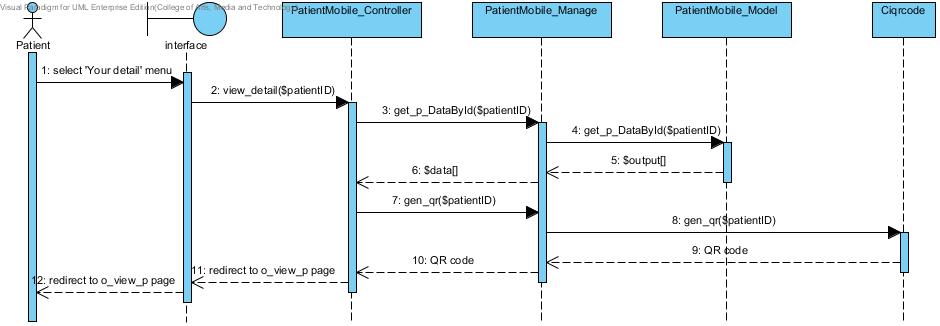
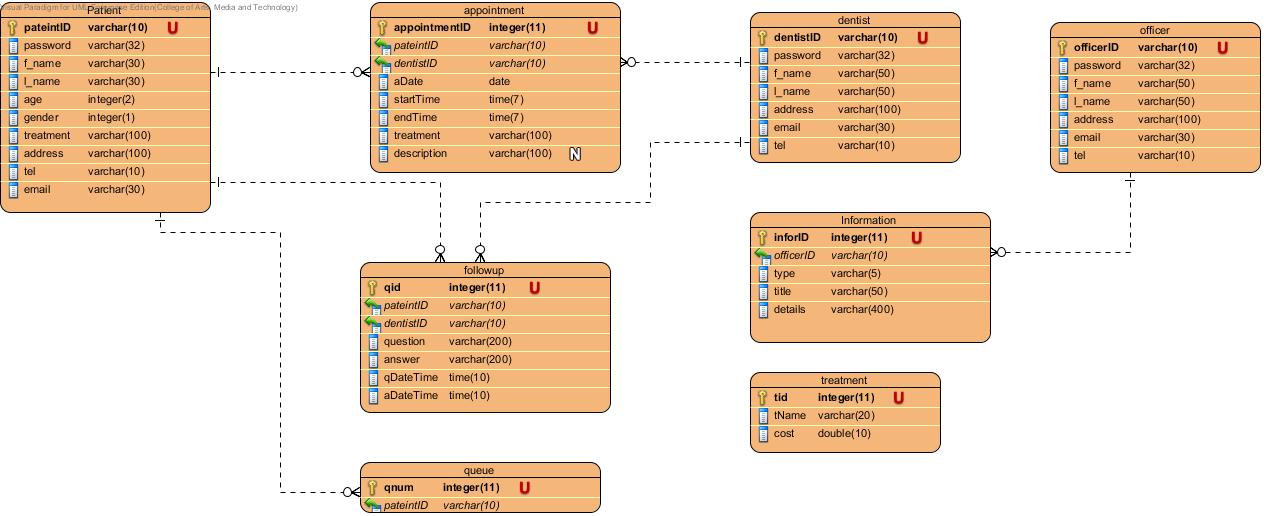
****

Figure 22 : patient view the QR code the mobile applicatio

**Chapter Six: Database Design**

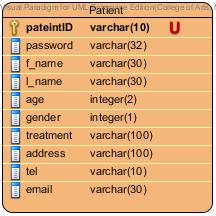
## 6. Database Design

## 6.1 Database design



## **6.2 Database Description**

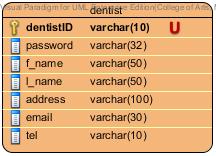
DB-01: patient

****

Description

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Field Name** | **Description** | **Type** | **Length** | **Primary Key** | **Foreign Key** | **Nullable** |
| patientID | patientID | varchar | 10 | PK | - | Not Null |
| password | Password of patient | varchar | 30 | - | - | Not Null |
| f\_name | First name of patient in English | varchar | 50 | - | - | Not Null |
| l\_name | Last name of patient in English | varchar | 50 | - | - | Not Null |
| age | Ages of patient | int | 2 | - | - | Not Null |
| gender | Gender of patient | int | 1 | - | - | Not Null |
| treatment | Treatment that patient have | varchar | 100 | - | - | Not Null |
| address | Address of patient in English | varchar | 100 | - | - | Not Null |
| tel | Telephone number of patient | varchar | 10 | - | - | Not Null |
| email | Email of patient | varchar | 30 | - | - | Not Null |

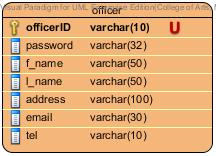
DB-02: dentist



Description

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Field Name** | **Description** | **Type** | **Length** | **Primary Key** | **Foreign Key** | **Nullable** |
| dentistID | dentistID | varchar | 10 | PK | - | Not Null |
| password | Password of dentist | varchar | 30 | - | - | Not Null |
| f\_name | First name of dentist in English | varchar | 50 | - | - | Not Null |
| l\_name | Last name of dentist in English | varchar | 50 | - | - | Not Null |
| address | Address of dentist | varchar | 100 | - | - | Not Null |
| tel | Telephone number of dentist | varchar | 30 | - | - | Not Null |
| email | Email of dentist | varchar | 10 | - | - | Not Null |

DB-03: officer

****

Description

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Field Name** | **Description** | **Type** | **Length** | **Primary Key** | **Foreign Key** | **Nullable** |
| officerID | officerID | varchar | 10 | PK | - | Not Null |
| password | Password of officer | varchar | 30 | - | - | Not Null |
| f\_name | First name of officer in English | varchar | 50 | - | - | Not Null |
| l\_name | Last name of officer in English | varchar | 50 | - | - | Not Null |
| address | Address of officer | varchar | 100 | - | - | Not Null |
| tel | Telephone number of officer | varchar | 30 | - | - | Not Null |
| email | Email of officer | varchar | 10 | - | - | Not Null |

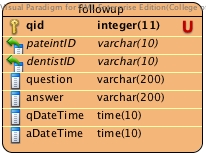
DB-04: appointment



Description

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Field Name** | **Description** | **Type** | **Length** | **Primary Key** | **Foreign Key** | **Nullable** |
| appointmentID | appointmentID | varchar | 10 | PK | - | Not Null |
| patientID | ID of patient who is in each appointment | varchar | 30 | - | FK | Not Null |
| dentistID | ID of dentist who is in each appointment | varchar | 50 | - | FK | Not Null |
| aDate | Date of an appointment | varchar | 50 | - | - | Not Null |
| startTime | Start time of an appointment | varchar | 100 | - | - | Not Null |
| endTime | End time of an appointment | varchar | 30 | - | - | Not Null |
| treatment | Treatment that going to have | varchar | 10 | - | - | Not Null |
| description | More description of appointment | varchar | 100 | - | - | - |

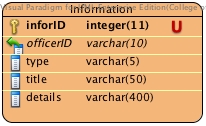
DB-05: follow up

****

Description

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Field Name** | **Description** | **Type** | **Length** | **Primary Key** | **Foreign Key** | **Nullable** |
| qid | qid | varchar | 11 | PK | - | Not Null |
| patientID | ID of patient who is in follow up | varchar | 10 | - | FK | Not Null |
| dentistID | ID of dentist who is in each follow up | varchar | 10 | - | FK | Not Null |
| question | Question of the follow up | varchar | 200 | - | - | Not Null |
| answer | Answer of the follow up | varchar | 200 | - | - | Not Null |
| qDateTime | The date of the question in follow up | varchar | 10 | - | - | Not Null |
| aDateTime | The date of the answer in follow up | varchar | 10 | - | - | Not Null |

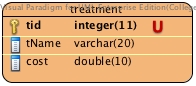
DB-06: information

****

Description

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Field Name** | **Description** | **Type** | **Length** | **Primary Key** | **Foreign Key** | **Nullable** |
| infoID | qid | integer | 11 | PK | - | Not Null |
| officerID | ID of officer who is add the information | varchar | 10 | - | FK | Not Null |
| type | Type of the information should be information or promotion | varchar | 5 | - | - | Not Null |
| title | Title of the information | varchar | 50 | - | - | Not Null |
| details | Details of the information | varchar | 400 | - | - | Not Null |

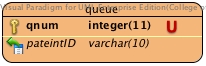
DB-07: treatment

****

Description

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Field Name** | **Description** | **Type** | **Length** | **Primary Key** | **Foreign Key** | **Nullable** |
| tid | tid | integer | 11 | PK | - | Not Null |
| tName | tName is the name of the treatment | varchar | 20 | - | - | Not Null |
| cost | Cost of the treatment | varchar | 10 | - | - | Not Null |

DB-08: queue

****

Description

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Field Name** | **Description** | **Type** | **Length** | **Primary Key** | **Foreign Key** | **Nullable** |
| qnum | qnum | integer | 11 | PK | - | Not Null |
| patientID | patientID who is in the queue | varchar | 10 | - | FK | Not Null |

# Chapter Seven: User Interface Design

## 7. User Interface Design

## 7.1 Web design

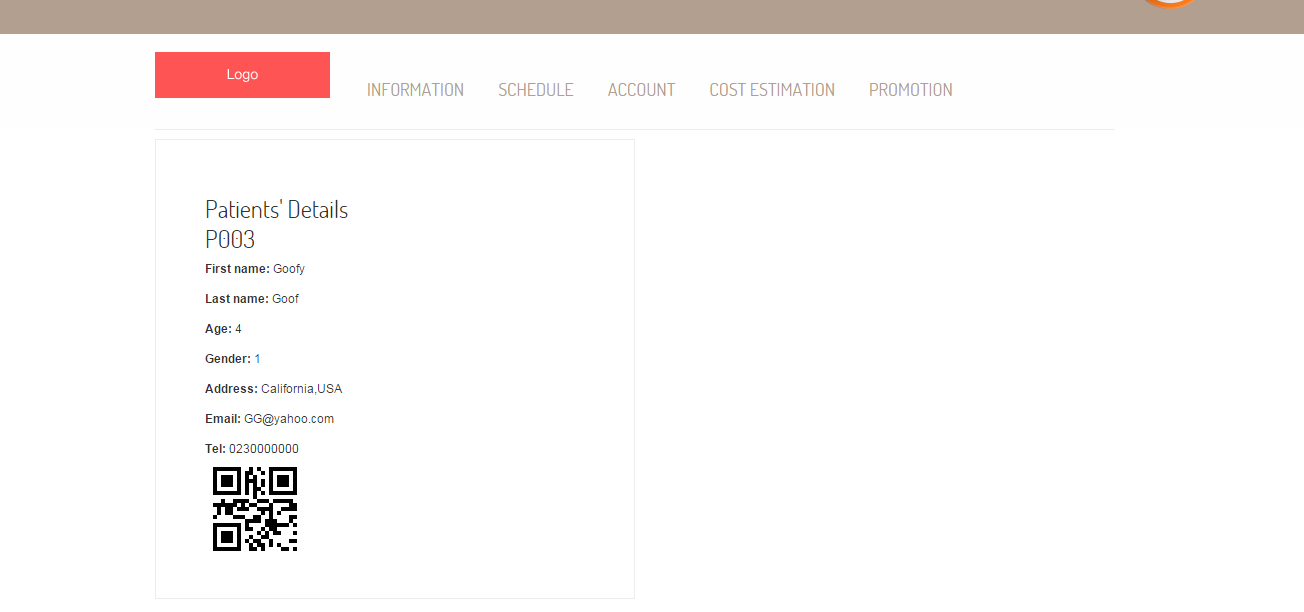
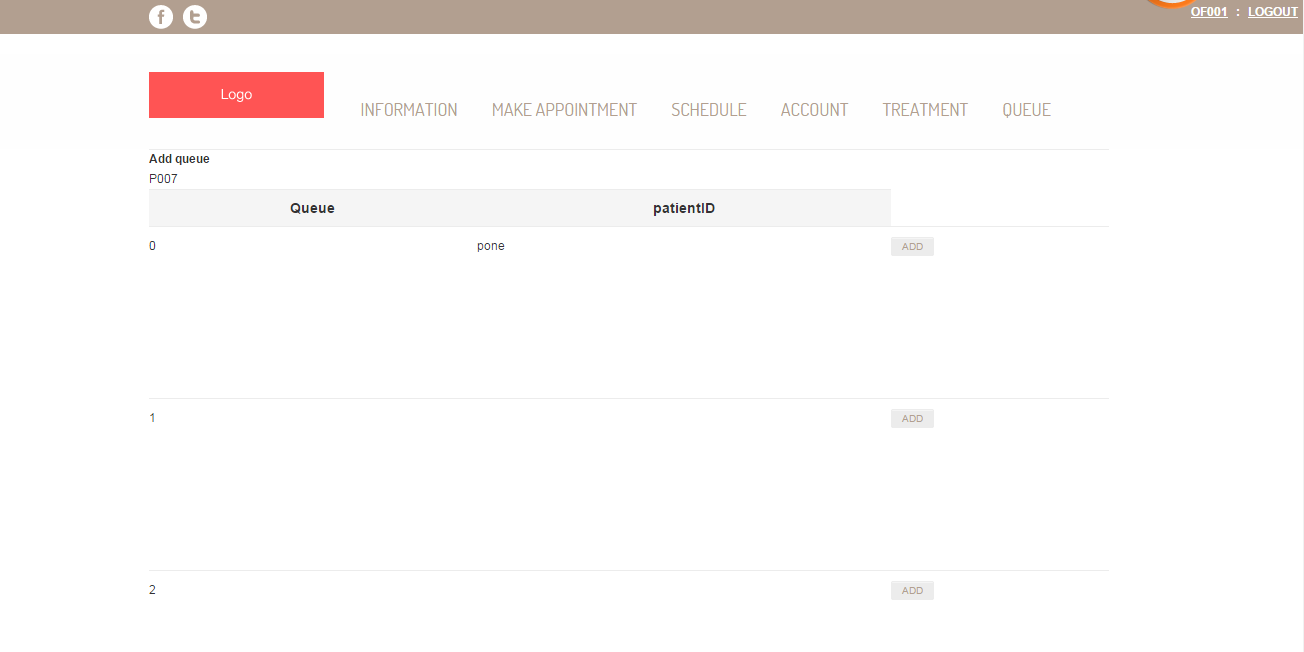


Figure 28 : View the QR code and patient’s information in the website [UI-21]

****Figure 33: Officer add patient appointment queue [UI-26]

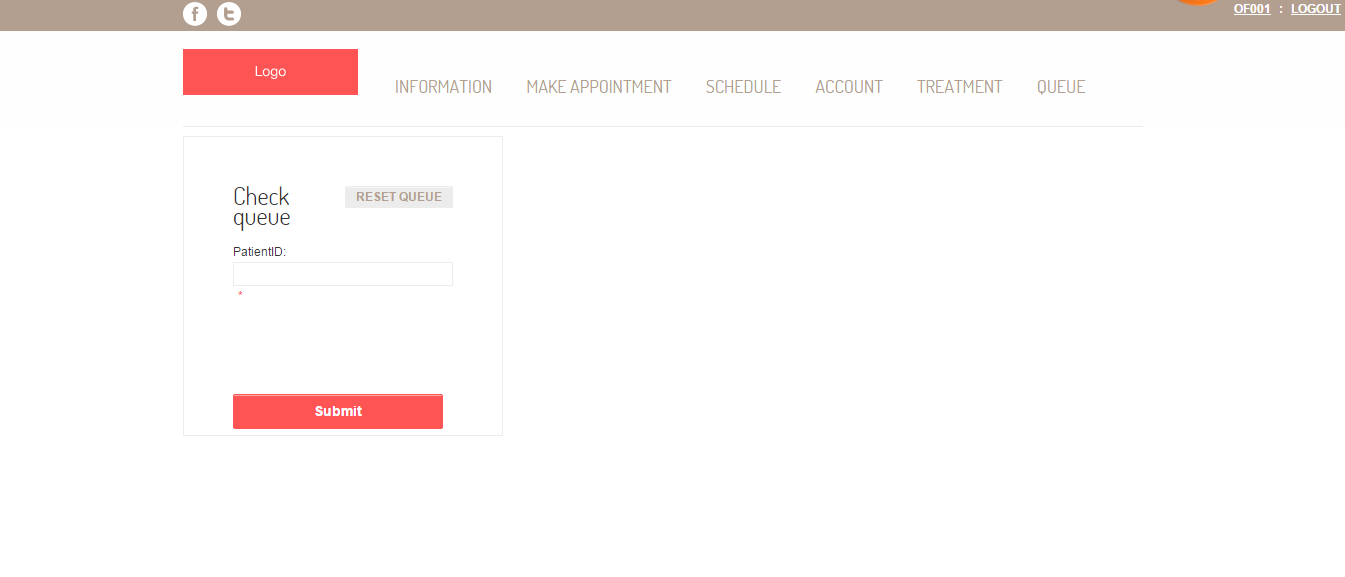


Figure 34: Check patient appointment queue [UI-27]

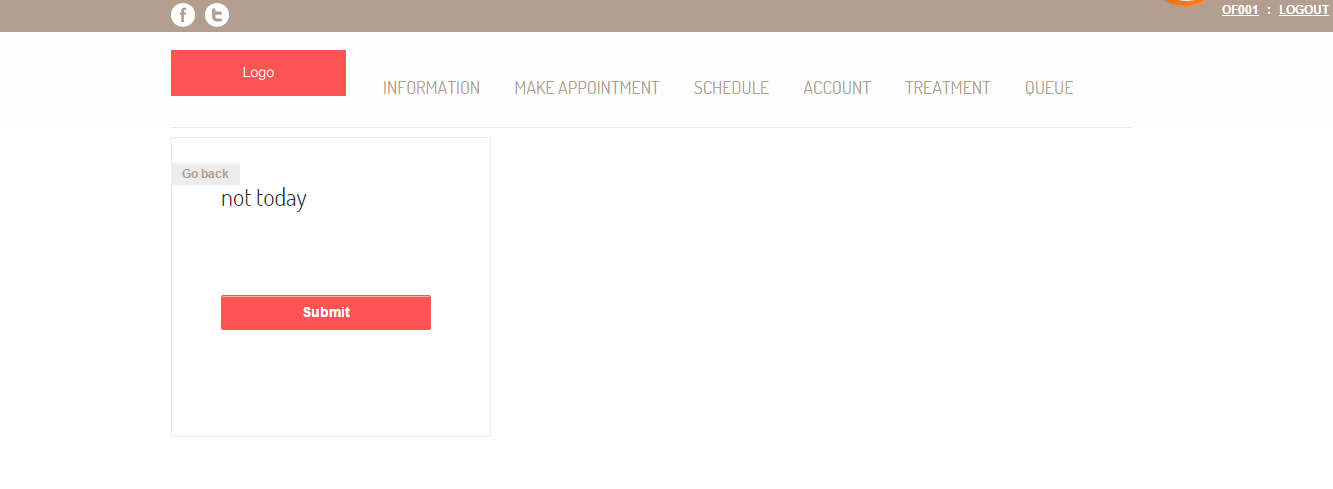


Figure 35: Display patient appointment queue [UI-28]

## Figure 36: Reset patient appointment queue[UI-29]

## 

## 7.2 Mobile application design

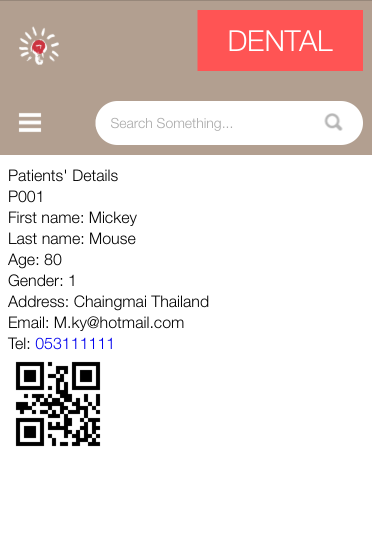


Figure 45: View the QR code and patient’s information in mobile application [UI-38]