**Dental Clinic Services System**

Software Design

By

**Kanokwan Maneerat 542115002**

**Worapun Wongkium 542115055**

Department of Software Engineering

College of Arts, Media, and Technology

Chiang Mai University

Project Adviser

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

Mrs. Yun Rim Park

**Software Design**

**Revision Table**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Document Name** | **Detail** | **Status** | **Date** | **Viewable** | **Reviewer& Responsible** |
| DCSS – SDD – v 0.1 | Add all document | Draft | 04/06/2014 | Adviser | Kanokwan & Worapun |
| DCSS – SDD – v 0.2 | Change class diagram and class diagram description | Draft | 10/06/2014 | Adviser | Kanokwan & Worapun |
| DCSS – SDD – v 0.3 | Edit class diagram and class diagram description | Draft | 27/06/2014 | Adviser | Kanokwan & Worapun |
| DCSS – SDD – v 0.4 | Edit class diagram, class diagram description, Sequence diagram | Draft | 07/06/2014 | Adviser | Kanokwan & Worapun |
| DCSS – SDD – v 0.5 | Edit class diagram, class diagram description, Sequence diagram | Draft | 07/06/2014 | Adviser | Kanokwan & Worapun |
| DCSS – SDD – v 0.6 | Double check all document | Draft | 07/06/2014 | Adviser | Kanokwan & Worapun |
| DCSS – SDD – v 0.7 | Edit sequence diagram | Draft | 07/15/2014 | Adviser | Kanokwan & Worapun |
| DCSS – SDD – v 0.8 | Edit class diagram and database description | Draft | 07/20/2014 | Adviser | Kanokwan & Worapun |
| DCSS – SDD – v 0.9 | Edit Sequence diagram | Draft | 07/23/2014 | Adviser | Kanokwan & Worapun |

Table of Contents

[Chapter One: Introduction 4](#_Toc393897425)

[1. Introduction 4](#_Toc393897426)

[1.1 Objectives 4](#_Toc393897427)

[1.2 Project Scope 4](#_Toc393897428)

[1.3 User Classes and Characteristics 5](#_Toc393897429)

[1.4 Operation Environment 5](#_Toc393897430)

[1.5 Acronyms and Definitions Acronyms 6](#_Toc393897431)

[Chapter Two: Overall Description 7](#_Toc393897432)

[2. Overall Description 7](#_Toc393897433)

[2.1 Product Perspective 7](#_Toc393897434)

[2.2 Product Features 7](#_Toc393897435)

[2.3 Design and Implementation Constraints 7](#_Toc393897436)

[Chapter Three: System Architecture 8](#_Toc393897437)

[3. System Architecture 8](#_Toc393897438)

[3.1 Technologies used 9](#_Toc393897439)

[3.1.1 Adobe® PhoneGap™ Build 9](#_Toc393897440)

[3.1.2 HTML5 9](#_Toc393897441)

[3.1.3 MySQL Server 9](#_Toc393897442)

[3.1.4 Apache web server 2.5.10 9](#_Toc393897443)

[3.1.5 CodeIgniter 2.1.4 9](#_Toc393897444)

[Chapter Four: Detail Design 10](#_Toc393897445)

[4. Detail Design 10](#_Toc393897446)

[4.1 Class diagram 11](#_Toc393897447)

[4.1.1 DCSS website class diagram 11](#_Toc393897448)

[4.1.2 DCSS mobile application class diagram 12](#_Toc393897449)

[4.2Class Diagram Description 13](#_Toc393897450)

[4.2.1 Class Diagram Description (DCSS website) 13](#_Toc393897451)

[Chapter Five: Sequence Diagram 25](#_Toc393897969)

[5. Sequence Diagram 25](#_Toc393897970)

[5.1 DCSS website Sequence Diagram 25](#_Toc393897971)

[5.2 DCSS mobile application Sequence Diagram 44](#_Toc393897972)

[Chapter Six: Database Design 49](#_Toc393897973)

[6. Database Design 49](#_Toc393897974)

[6.1 Database design 49](#_Toc393897975)

[6.2 Database Description 50](#_Toc393897976)

[Chapter Seven: User Interface Design 54](#_Toc393898021)

[7. User Interface Design 54](#_Toc393898022)

[7.1 Web design 54](#_Toc393898023)

# Chapter One: Introduction

## Introduction

## 1.1 Objectives

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

This 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.

### 

## 1.3 User Classes and Characteristics

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

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

* Patient can receive a pateintID and password from the dental clinic.
* Patient can login to the web application and the mobile application.
* Patient can logout from the web application and the mobile application.
* Patient can view his/her appointment in the patient schedule.
* Patient can view all appointments in the dental clinic schedule.
* **Officer**

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

* Officer can login to the web application using userID and password.
* Officer can logout from the web application.
* Officer can register for new patients and dentists.
* Officer can add an appointment into the dental clinic schedule.
* Officer can edit an appointment in the dental clinic schedule.
* Officer can delete an appointment in the dental clinic schedule.

**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 userID and password to login to the system to use the services. Dentists are able to:

* Dentist can login to the web application by using userID and password
* Dentist can logout from the web application
* Dentist can view his/her appointments in the dentist schedule
* Dentist can view all appointments in the dental clinic schedule

## 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 |
| 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

## 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: User registration and authentication
* Feature 3: Appointment management
* 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 1, 2, and 3

## 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

## 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

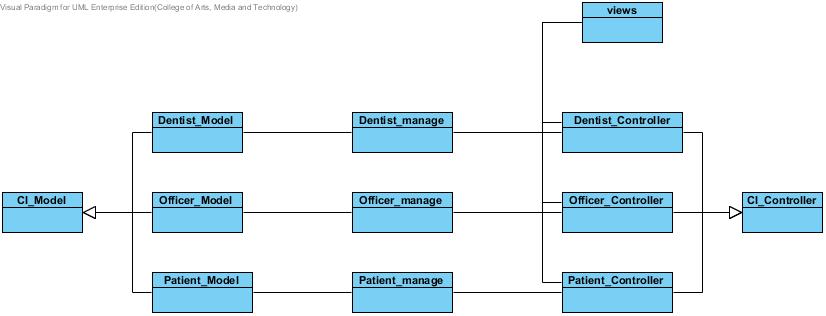


Figure 3: Shows class diagram of Dental Clinic Services System

### 4.1.2 DCSS mobile application class diagram

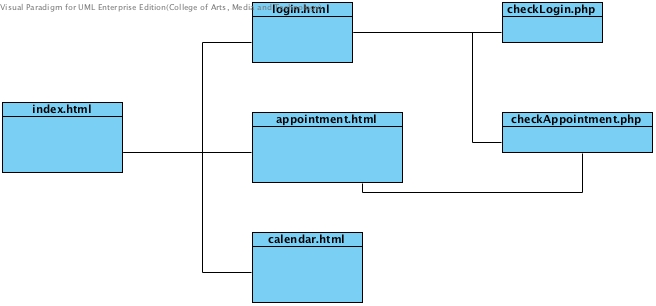
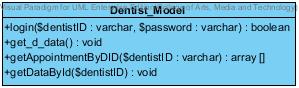
****

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

## **4.2Class Diagram Description**

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

CD-01, Class name: Dentist\_Model



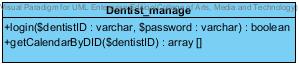
Property

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

Method

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Name | Description | Parameters | Return |
| 1 | login | Method uses to login for dentist | $dentistID:varchar,  $password:varchar | boolean |
| 2 | get\_d\_data | Method uses to get the dentists’ data from database | - | void |
| 3 | getAppointmentByDID | Method uses to get appointments data of dentist from database by using dentistID | $dentistID:varchar | Array[] |
| 4 | getDataById | Method uses to get the dentists’ data from database by using dentistID | $dentistID:varchar | void |

CD-02, Class name: Dentist\_Manage



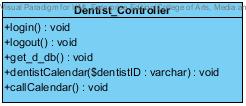
Property

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

Method

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Name | Description | Parameters | Return |
| 1 | login | Method uses to login for dentist | $dentistID:varchar,$password:varchar | boolean |
| 2 | getCalendarByDID | Method uses to get appointments data  of dentist from database by using dentistID | $dentistID:varchar | Array[] |

CD-03, Class name: Dentist\_Controller



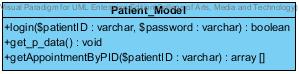
Property

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

Method

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Name | Description | Parameters | Return |
| 1 | login() | Method uses to login for dentist. | - | void |
| 2 | logout() | Method uses to logout for dentist. | - | void |
| 3 | get\_d\_db() | Method uses to get dentist data from database | - | void |
| 4 | dentistCalendar | Method uses to get dentist calendar by dentistID | $userID:varchar | Array[] |
| 5 | callCalendar | Method uses to call appointment calendar page | - | void |

CD-04, Class name: Patient\_Model



Property

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

Method

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Name | Description | Parameters | Return |
| 1 | login | Method uses to login for patient. | $patientID:varchar,$password:varchar | boolean |
| 2 | get\_p\_data() | Method use to get the patients’ data from database | - | void |
| 3 | getAppointmentByID | Method uses to get the appointments’ data from database | $patientID:varchar | Array[] |

CD-05, Class name: Patient\_Manage

D:\from desktop\senior project\progress1\usecase\cd.jpg\patient_manage.jpg

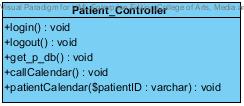
Property

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

Method

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Name | Description | Parameters | Return |
| 1 | login | Method uses to login for patient. | $patientID:varchar,$password:varchar | boolean |
| 2 | getCalendarByPID | Method uses to get patients’ calendar by using patientID | $patientID:varchar | Array[] |

CD-06, Class name: Patient\_Controller



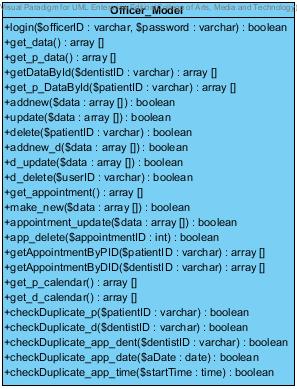
Property

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

Method

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Name | Description | Parameters | Return |
| 1 | login | Method uses to login for patient. | - | void |
| 2 | logout | Method uses to logout for patient. | - | void |
| 3 | get\_p\_db | Method uses to get patients’ data from database | - | void |
| 4 | callCalendar | Method uses to call calendar page | - | void |
| 5 | patientCalendar | Method uses to call their own calendar | $patientID:varchar | Array[] |

CD-07, Class name: Officer\_Model



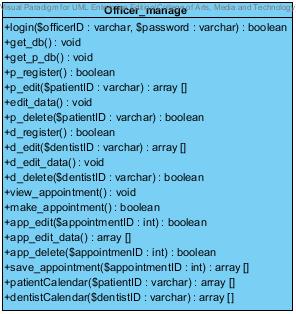
Property

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

Method

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Name | Description | Parameters | Return |
| 1 | login | Method uses to login for patient. | $dentistID:varchar | void |
| 2 | get\_db | Method uses to get all dentists’ data and officers’ data from database | - | Array[] |
| 3 | get\_p\_data | Method uses to get all patients’ data from database | - | Array[] |
| 4 | getDataByID | Method uses to get data of dentist by using dentistID | $dentistID:varchar | Array[] |
| 4 | get\_p\_DataByID | Method uses to get data of patient by using patientID | $pateintID:varchar | Array[] |
| 5 | addnew | Method uses to add new patient to the database | $data:array[] | boolean |
| 6 | update | Method uses to update patients’ data to database | $data:array[] | boolean |
| 8 | delete | Method uses to delete patient from database by using of patientID | $patientID:varchar | boolean |
| 9 | addnew\_d | Method uses to add new dentist to the database | $data:array[] | boolean |
| 10 | d\_update | Method uses to update dentists’ data to database | $data:array[] | boolean |
| 11 | get\_appointment | Method uses to get data of appointments from database | - | Array[] |
| 12 | make\_new | Method uses to add new appointment to database | $data:varchar | boolean |
| 13 | appointment\_update | Method uses to update appointments’ data to database | $data:varchar | boolean |
| 14 | app\_delete | Method uses to delete appointments’ data from database | $appointmentID:int | boolean |
| 15 | getAppointmentByPID | Method uses to get appointments’ data from database by using patientID | $patientID:varchar | Array[] |
| 16 | getAppointmentByDID | Method uses to get appointments’ data from database by using dentistID | $userID:varchar | Array[] |
| 17 | get\_p\_calendar | Method uses to get all appointments’ data from database | - | Array[] |
| 18 | get\_d\_calendar | Method uses to get all appointments’ data from database | - | Array[] |
| 19 | checkDuplicate\_p | Method uses to check that the patientID is duplicate or not | $patientID:varchar | boolean |
| 20 | checkDuplicate\_d | Method uses to check that the dentistID is duplicate or not | $dentistID:varchar | boolean |
| 21 | checkDuplicate\_app\_dent | Method uses to check that the dentistID is duplicate in appointment table or not | $dentistID:varchar | boolean |
| 22 | checkDuplicate\_app\_date | Method uses to check that there are duplicate date in appointment table or not | $aDate:date | boolean |
| 23 | checkDuplicate\_app\_time | Method uses to check that there are duplicate time in appointment table or not | $startTime:time | boolean |

CD-08, Class name: Officer\_Manage

****

Property

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

Method

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Name | Description | Parameters | Return |
| 1 | login | Method uses to login for officer. | $patientID:varchar | boolean |
| 2 | get\_db | Method uses to get all dentists’ data and officers’ data from database | - | void |
| 3 | get\_p\_data | Method uses to get all patients’ data from database | - | void |
| 4 | p\_register | Method uses to register new patient to the database | - | boolean |
| 5 | p\_edit | Method uses to get patient’s data for edit | $patientID:varchar | Array[] |
| 6 | edit\_data | Method uses to collect new data of patient for edit | - | void |
| 7 | p\_delete | Method uses to delete patient from database by using of patientID | $patientID:varchar | boolean |
| 8 | d\_register | Method uses to add new dentist to the database | - | boolean |
| 9 | d\_edit | Method uses to get dentist’s data for edit | $daentistID:varchar | Array[] |
| 10 | d\_edit\_data | Method uses to collect new data of dentist for edit | - | void |
| 11 | d\_delete | Method uses to delete dentist from database by using of dentistID | $dentistID:varchar | Array[] |
| 12 | view\_appointment | Method uses to get data of appointments from database | - | void |
| 13 | make\_appointment | Method uses to add new appointment to database | - | boolean |
| 14 | app\_edit | Method uses to get appointment’s data for edit | $appointmentID:int | boolean |
| 15 | app\_edit\_data | Method uses to collect new data of appointment for edit | - | Array[] |
| 16 | app\_delete | Method uses to delete appointments’ data from database | $appointmentID:int | boolean |
| 17 | save\_appointment | Method uses to call appointments’ data from database to save to Google calendar | $appointmentID:int | Array[] |
| 18 | patientCalendar | Method uses to get all appointments’ data from database by using patientID | $patientID:varchar | Array[] |
| 19 | dentistCalendar | Method uses to get all appointments’ data from database by using dentistID | $dentistID:varchar | Array[] |

CD-09, Class name: Officer\_Controller

****

Property

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

Method

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Name | Description | Parameters | Return |
| 1 | login | Method uses to login for patient. | - | void |
| 2 | logout | Method uses to logout for patient. | - | void |
| 3 | get\_db | Method uses to get all dentists’ data and officers’ data from database | - | void |
| 4 | get\_p\_db | Method uses to get all patients’ data from database. | - | void |
| 5 | p\_register | Method uses to register new patient to the system | - | void |
| 6 | p\_edit | Method uses to call patient edit page with appointment data | $patientID:varchar | void |
| 7 | edit\_data | Method uses to save the edit data of patient to the database | - | void |
| 8 | p\_delete | Method uses to delete any patient from the system and database | $patientID:varchar | void |
| 9 | callCalendar | Method uses to call calendar from google calendar | - | void |
| 10 | d\_register | Method uses to register new dentist to the system | - | void |
| 11 | d\_edit | Method uses to call dentist edit page with dentist data | $dentistID:varchar | void |
| 12 | d\_edit\_data | Method uses to save the edit data of dentist to the database | - | void |
| 13 | d\_delete | Method uses to delete any dentist from the system and database | $dentistID:varchar | void |
| 14 | view\_appointment | Method uses to get all appointment data to view | - | void |
| 15 | make\_appointment | Method uses to make new appointment | - | void |
| 16 | app\_edit | Method uses to call appointment edit page with appointment data | $appointmentID:varchar | void |
| 17 | app\_edit\_data | Method uses to save the edit data of appointment to the database | - | void |
| 18 | app\_delete | Method uses to delete any appointment from the system and database | $appointmentID:varchar | void |
| 19 | save\_appointment | Method uses to save appointment to google calendar | $appointmentID:varchar | void |
| 20 | patientCalendar | Method uses to get patients’ calendar by using patientID | $patientID:varchar | void |
| 21 | dentistCalendar | Method uses to get dentists’ calendar by using dentistID | $dentistID:varchar | void |

# Chapter Five: Sequence Diagram

## 5. Sequence Diagram

## 5.1 DCSS website Sequence Diagram

**SD-01: Login**

* **URS-15**

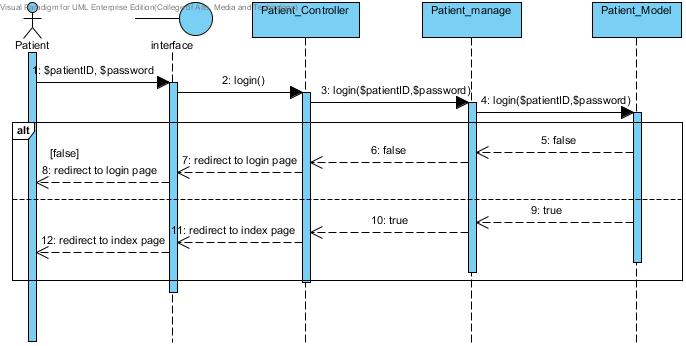


Figure 5: Patient Login

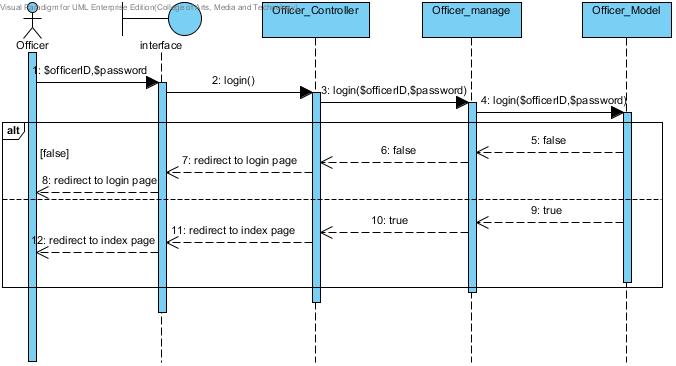


Figure 6: Officer Login

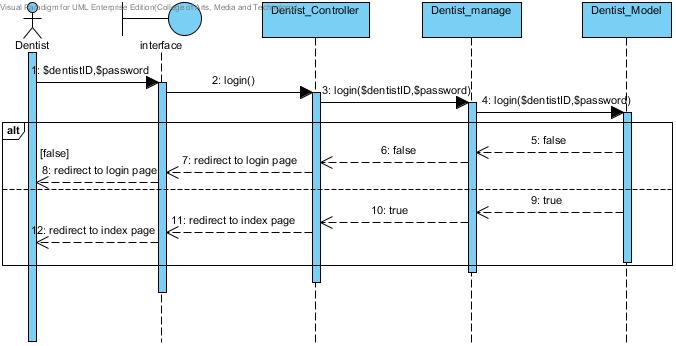


Figure 7: Dentist login

**SD-02: Logout**

* **URS-17**

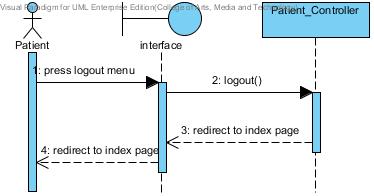
****

Figure 8: Patient logout

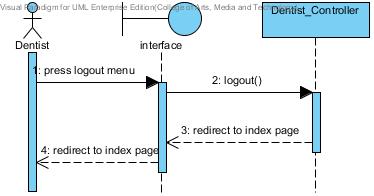
****

Figure 9: Officer logout

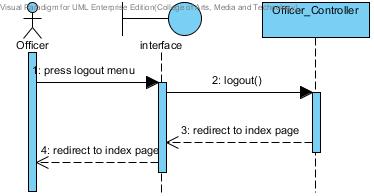
****

Figure 10: Dentist Logout

**SD-03: View dental clinic calendar**

* **URS-09**

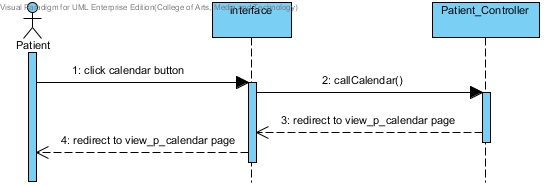
****

Figure 11: Patient view dental clinic calendar

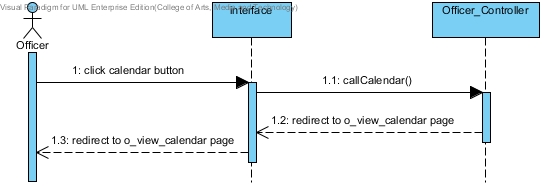
****

Figure 12: Officer view dental clinic calendar

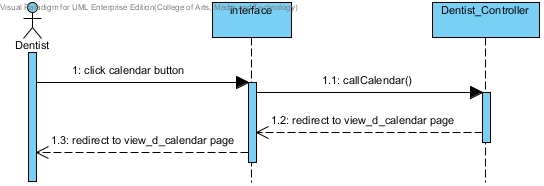
****

Figure 13: Dentist view dental clinic calendar

**SD-04: View his/her own appointment**

* **URS-03**

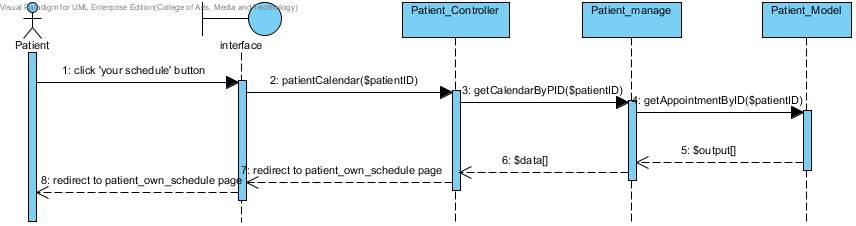
****

Figure 14: Patient can view his/her own appointment

* **URS-04**

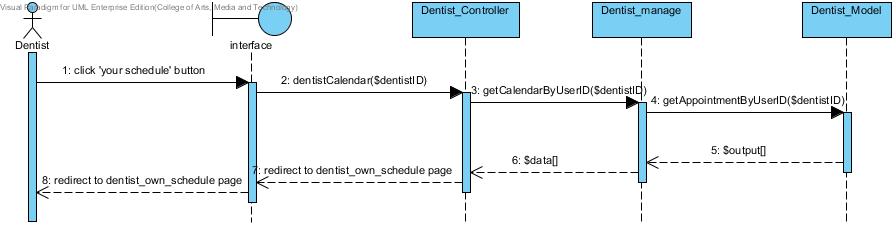


Figure 15: Dentist can view his/her own appointment

**SD-05: View all appointment in dental clinic**

* **URS-06**

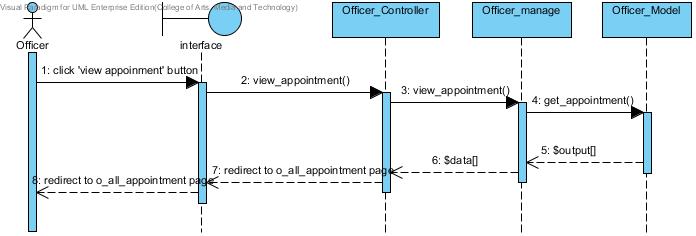
****

Figure 16:Officer can view all appointment in dental clinic

**SD-06:Create patient account**

* **URS-18**

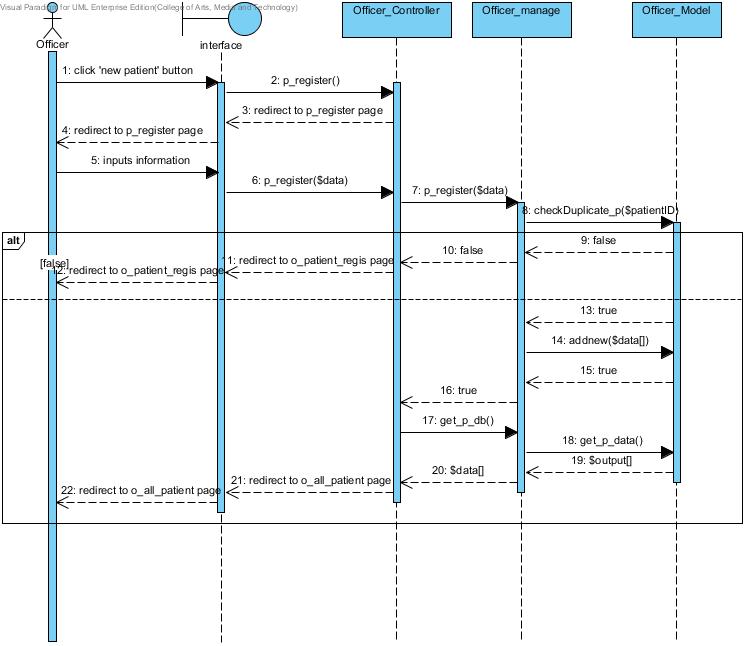
****

Figure 17: Officer can create new patient account and the list of all patients show after register.

**SD-07: Create dentist account**

* **URS-21**

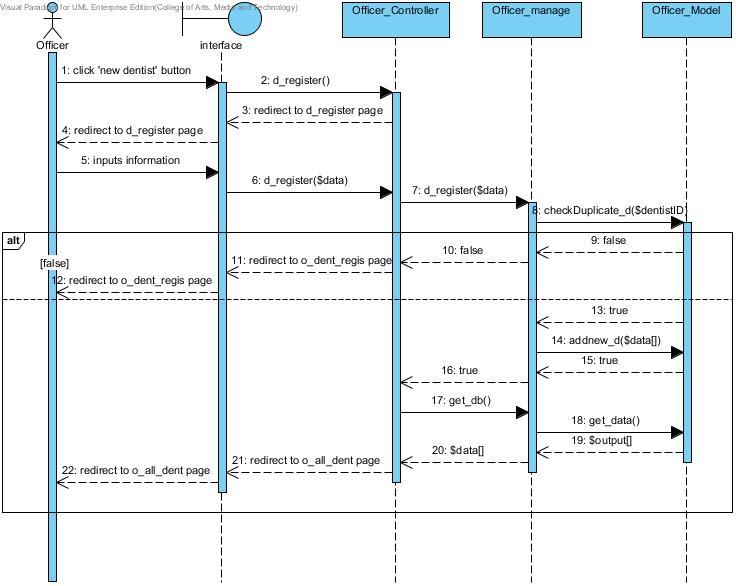
****

Figure 18: Officer can create new dentist account and the list of all clinic user shows after register

**SD-08: View all patients as a list**

* **URS-24**

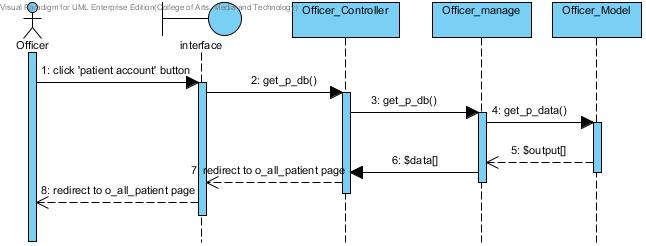
****

Figure 19: Officer can view all patients as a list

**SD-09: View all dentist as a list**

* **URS-25**

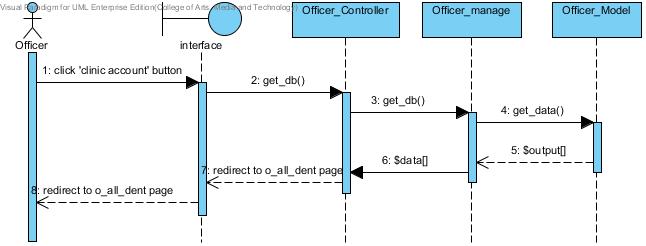
****

Figure 20: Officer can view all dentists in the list

**SD-10: Make an appointment**

* **URS-10**

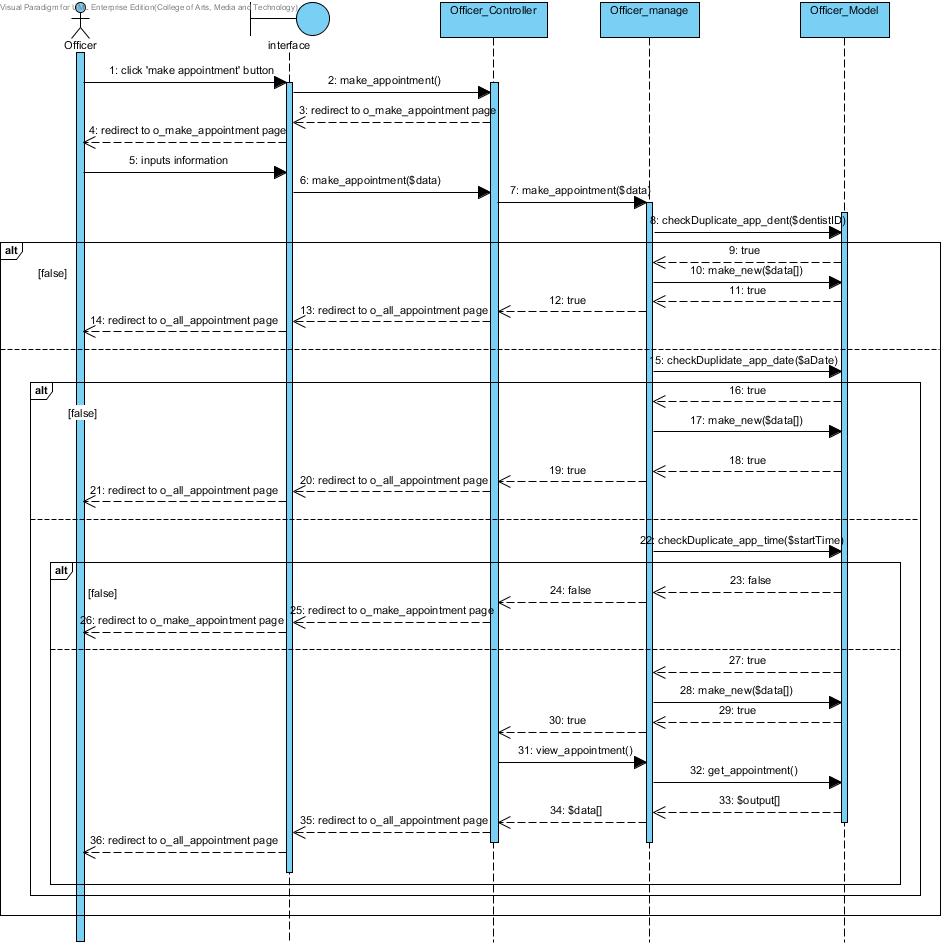
****

Figure 21: Officer can add new appointment into the database and it will update in the dental clinic appointment schedule

**SD-11: Edit an appointment**

* **URS-11**

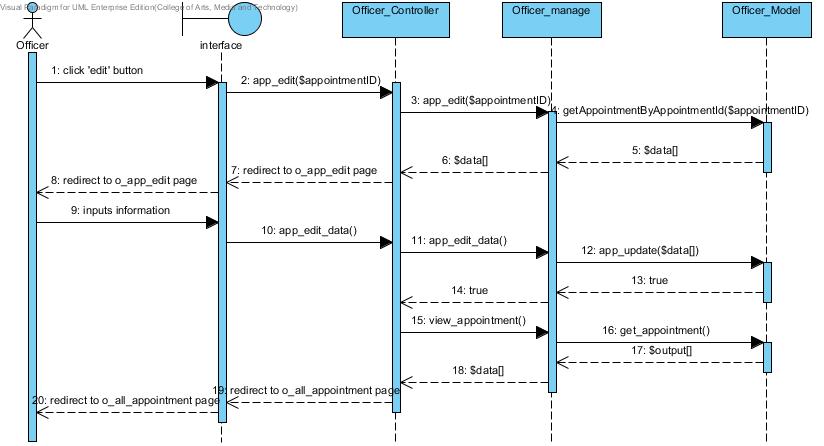
****

Figure 22: Officer can edit an appointment into the database and it will update in the dental clinic appointment schedule

**SD-12: Delete an appointment**

* **URS-12**

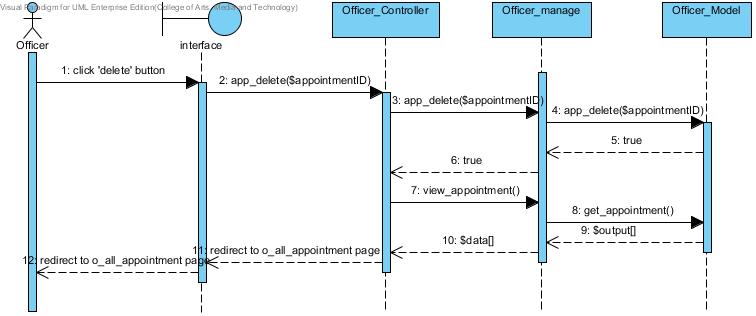
****

Figure 23: Officer can delete an appointment into the database and it will update in the dental clinic appointment schedule

**SD-13: Edit patients’ account**

* **URS-19**

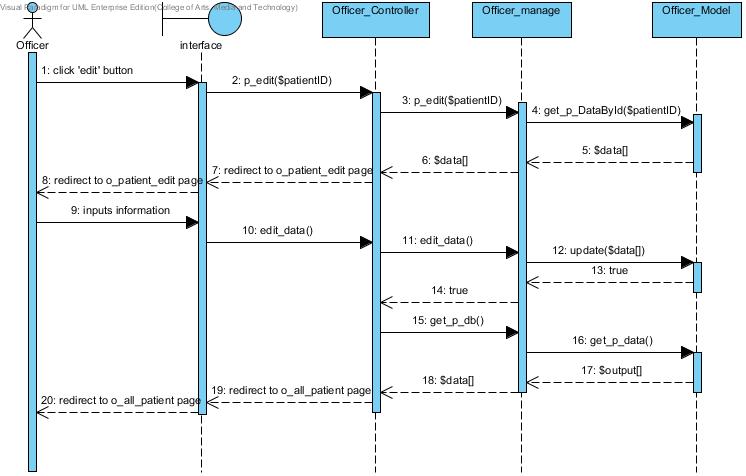


Figure 24: Officer can edit patients’ account and the list of all patients shows after edit

**SD-14: Edit dentists’ account**

* **URS-22**

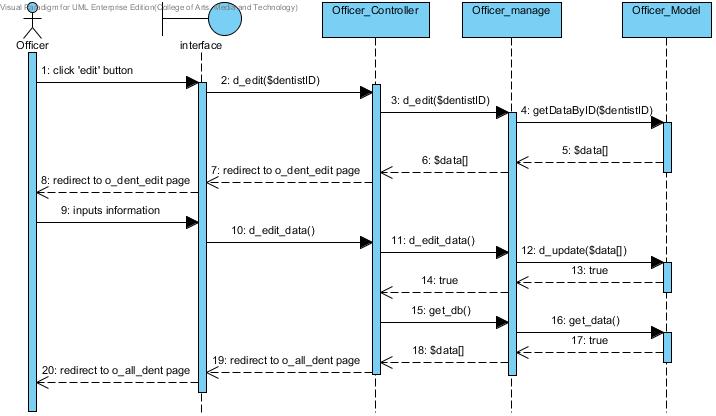
****

Figure 25: Officer can edit dentists’ account and the list of all clinic users shows after edit

**SD-15: Delete patients’ account**

* **URS-20**

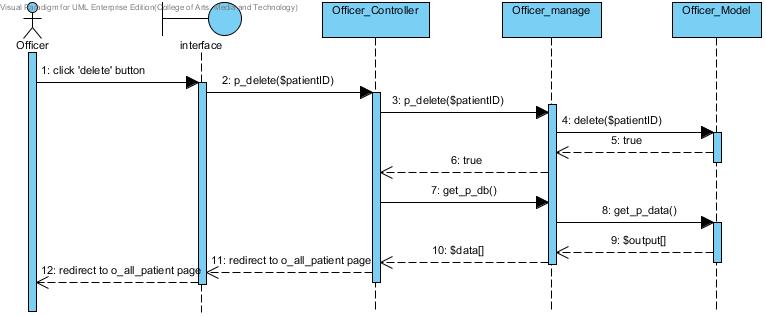


Figure 26: Officer can delete patients’ account and the list of all patients shows after edit

**SD-16: Delete dentists’ account**

* **URS-23**

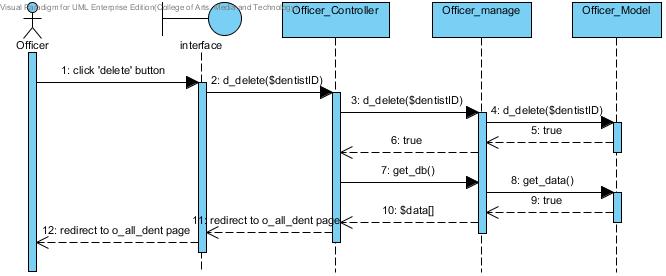


Figure 27: Officer can delete dentists’ account and the list of all dentists shows after edit

**SD-17: Save appointment to Google Calendar**

* **URS-13**

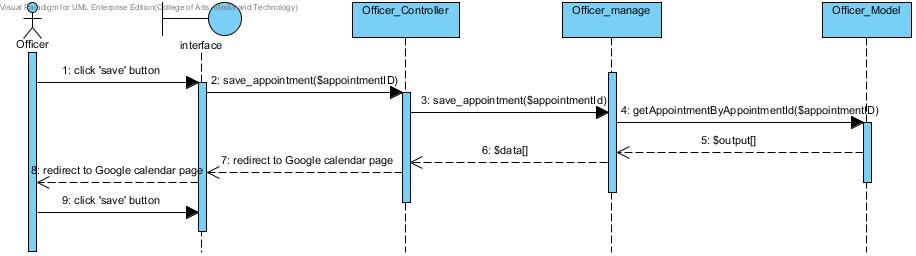
****

Figure 28: Officer can save an appointment to Google Calendar by using dental clinic services system Google account

## 5.2 DCSS mobile application Sequence Diagram

**SD-18: Login to the mobile application**

* **URS-14**

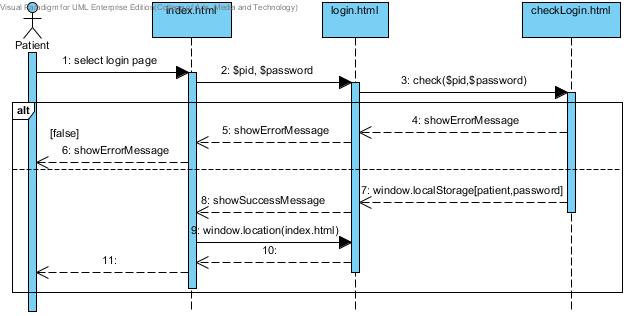
****

Figure 29: Patient can login to the mobile application

**SD-19: View his/her own appointment**

* **URS-01**

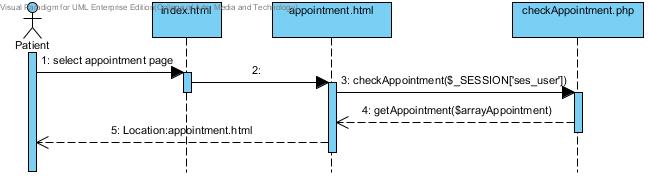
****

Figure 30: Patient can view his/her appointment after login if user does not login the appointment will not diaplay

**SD-20: View appointment in Google calendar**

* **URS-02**

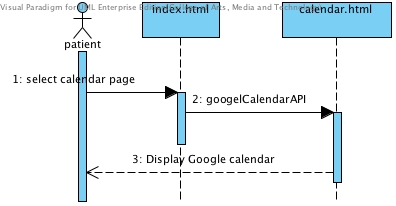
****

Figure 31: Patient can view all appointment in google calendar

**SD-21: Logout from mobile application**

* **URS-16**

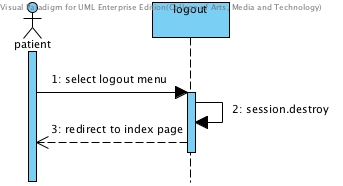
****

Figure 32: Patient logout from mobile application

**SD-21: Visitor can view all appointment from Google calendar**

* **URS-05**

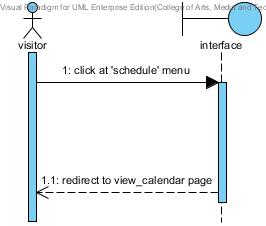


Figure 33: Visitor can view all appointment from Google calendar

**SD-23: Officer view dentists’ appointment**

* **URS-08**

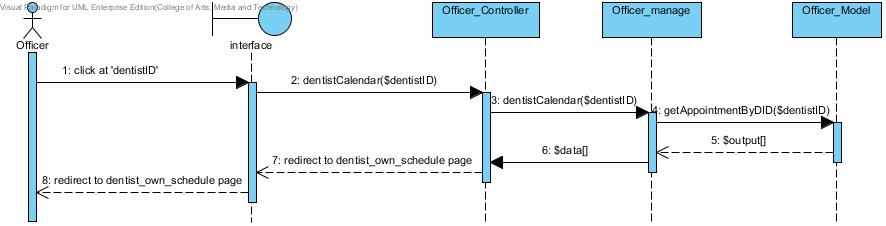


Figure 34: Officer views dentists’ appointment

**SD-24: Officer view patients’ appointment**

* **URS-07**

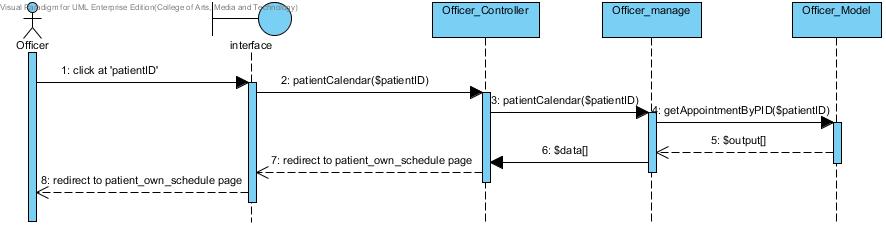


Figure 35: Officer views patients’ appointment

# Chapter Six: Database Design

## 6. Database Design

## 6.1 Database design

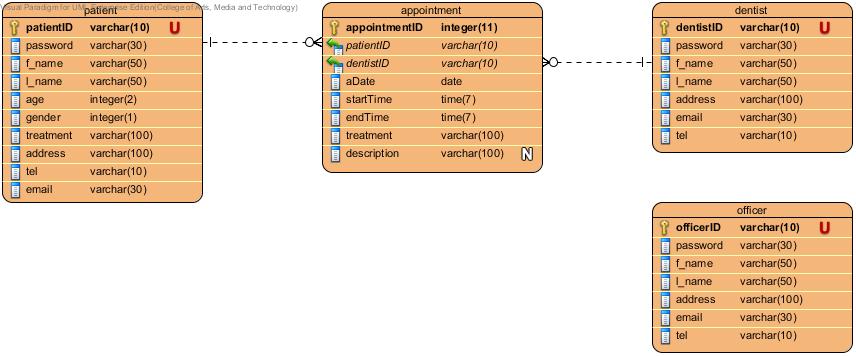


Figure 36: Database design of Dental Clinic Services System

## **6.2 Database Description**

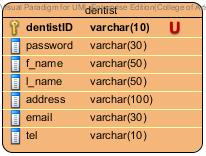
DB-01: patient

****

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 | - | - | - |

# Chapter Seven: User Interface Design

## 7. User Interface Design

## 7.1 Web design

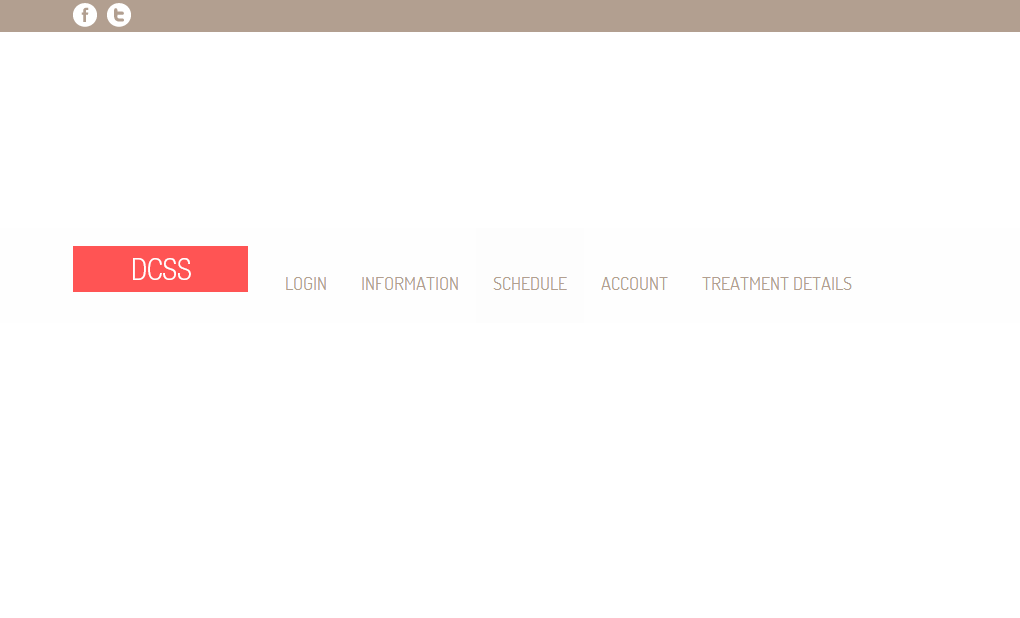


Figure 37: Shows the index page of website[UI-01]

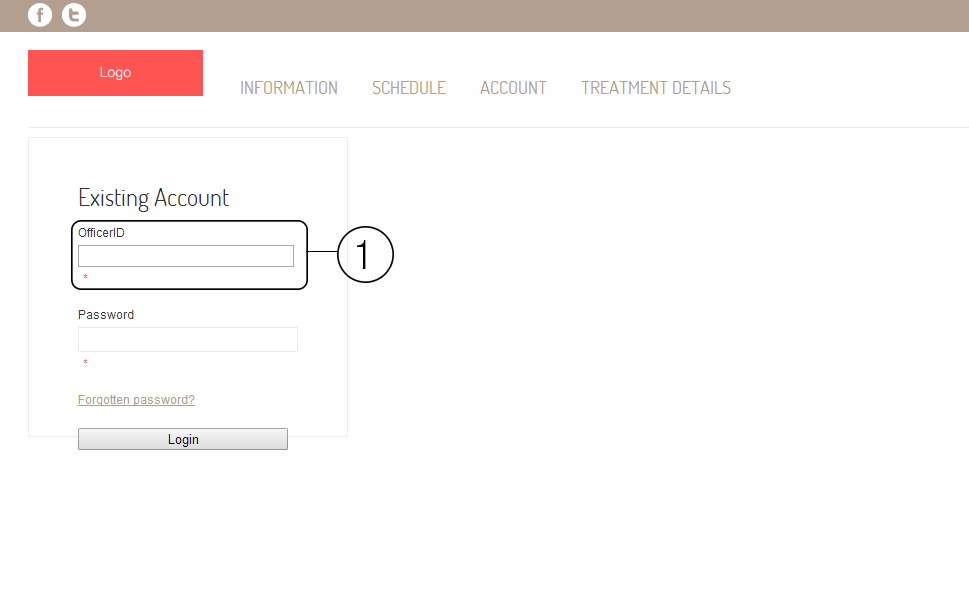


Figure 38: Shows the login page of the website. The area number 1 is the field of input userID and patientID. In officer login page will show as OfficerID, in patient login page will show as PatientID, and in dentist login page will show as DentistID.[UI-02]

**Patient**

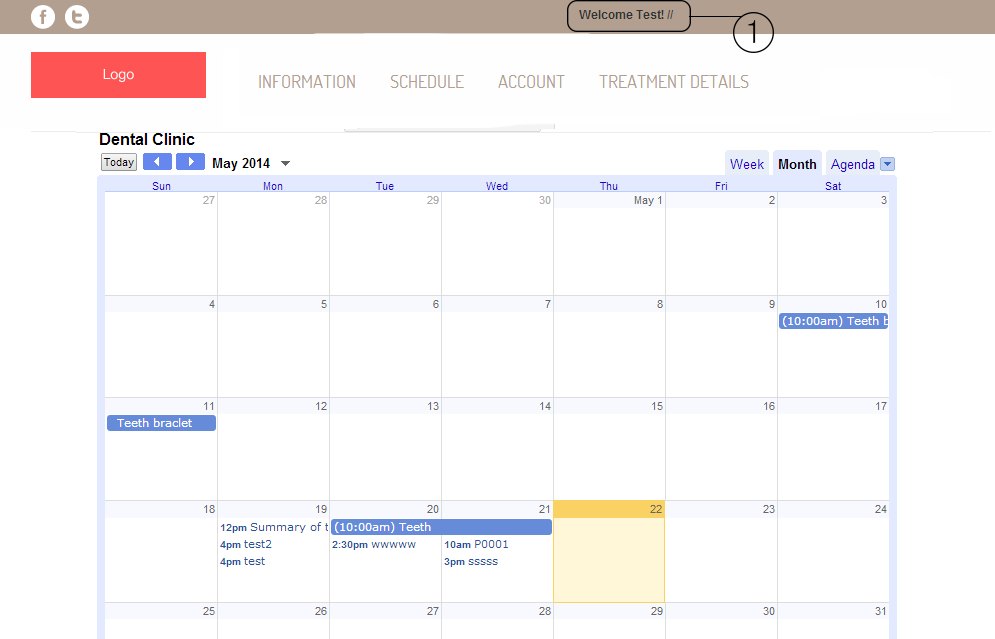


Figure 39: Shows the schedule of dental clinic from google calendar. The area number 1 will shows the name of patient.[UI-03]

**Officer**

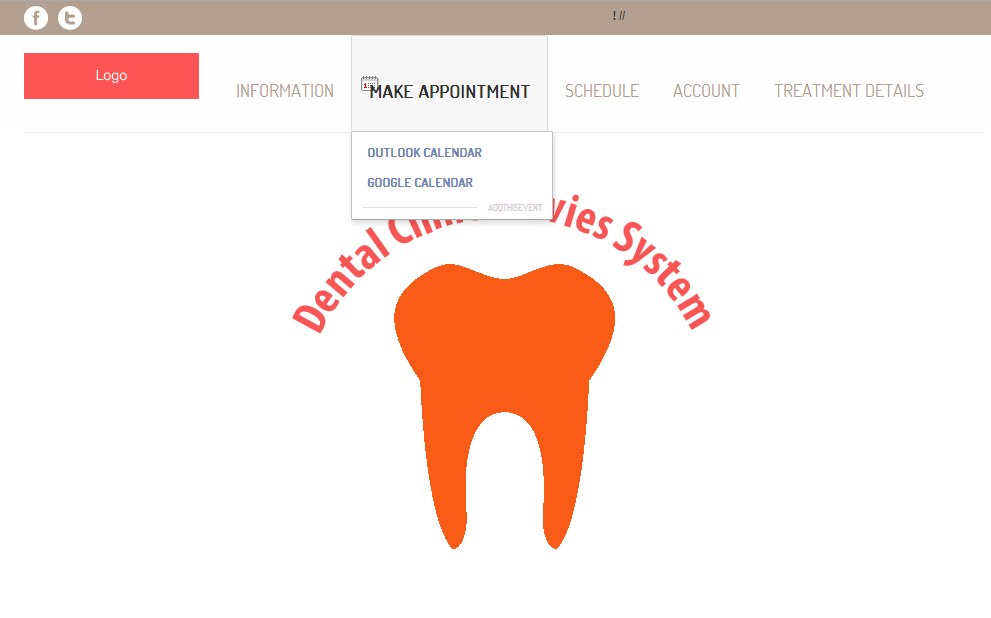


Figure 40: Shows the officer index page after officer login. Officer can make appointment google calendar, view schedule, and manage account.[UI-04]

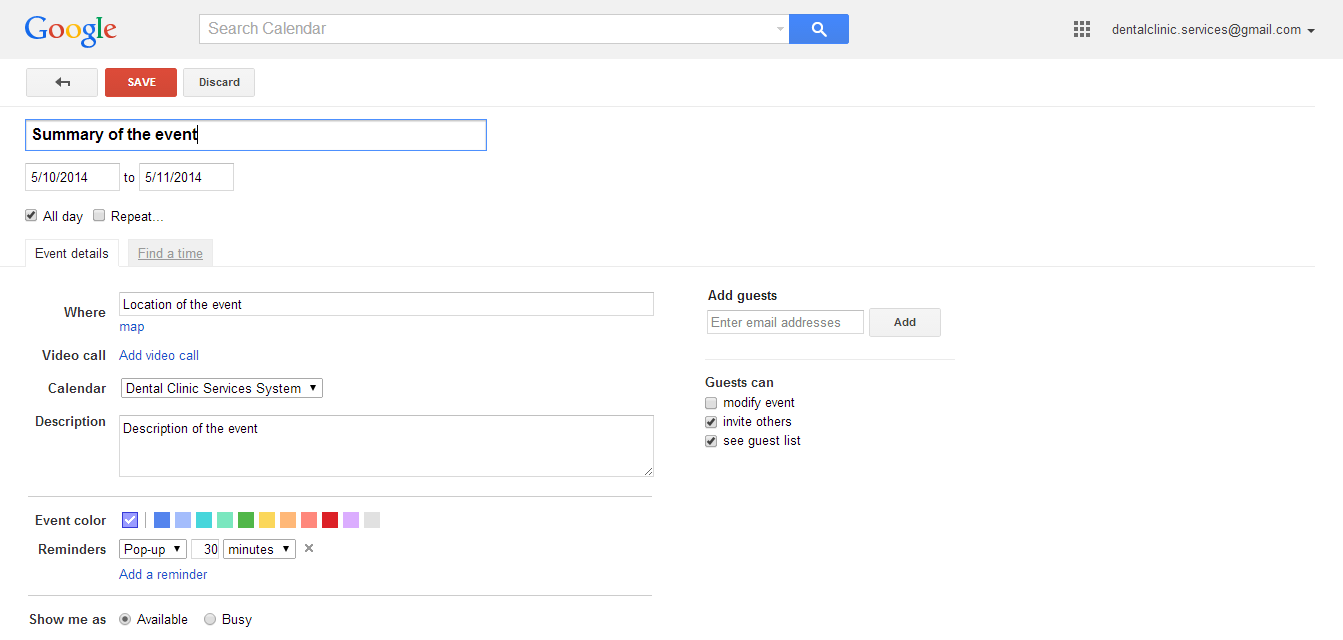


Figure 41: Shows page of google calendar to add the appointment to dental clinic calendar.[UI-05]

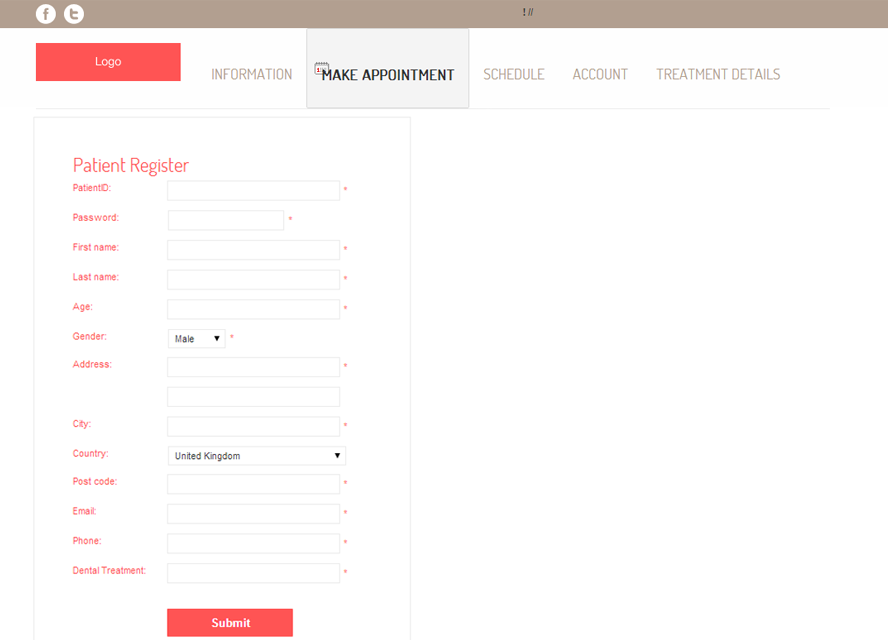


Figure 42: Shows the patient registration page.[UI-06]

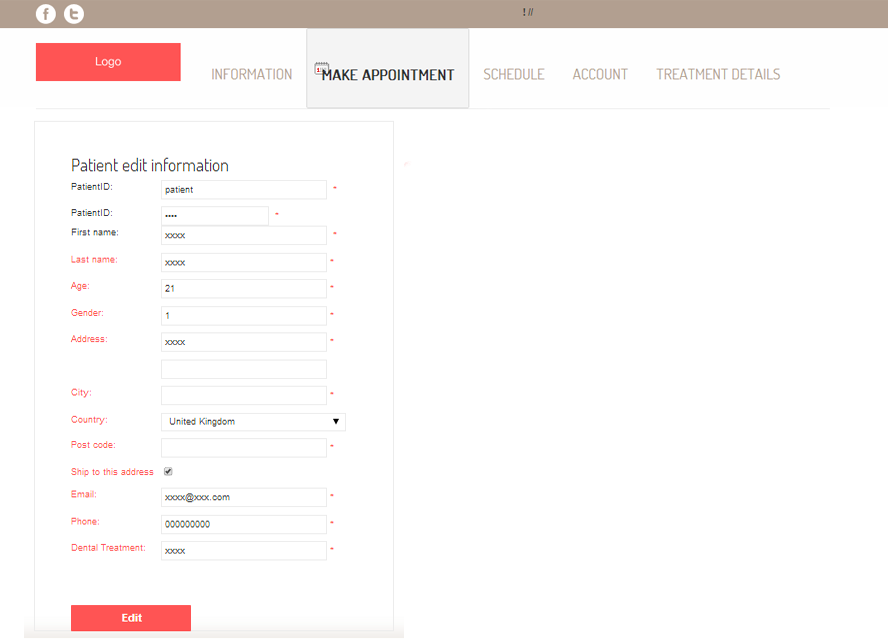


Figure 43: Shows the edit patients’ details page.[UI-07]

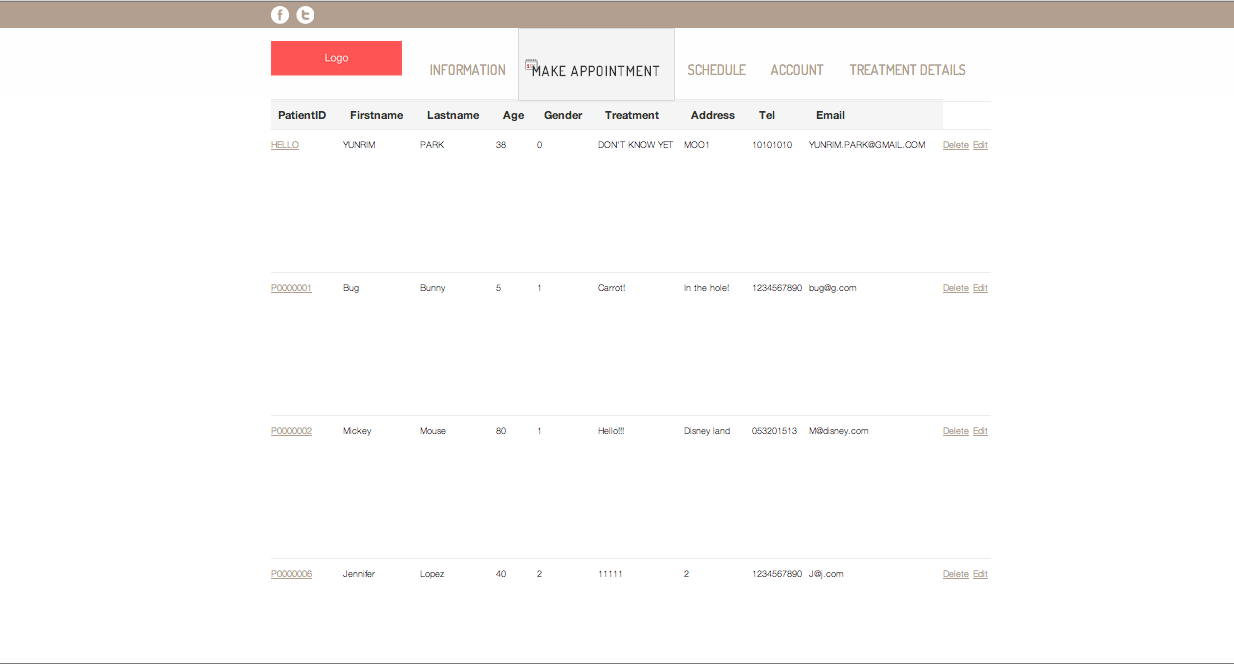
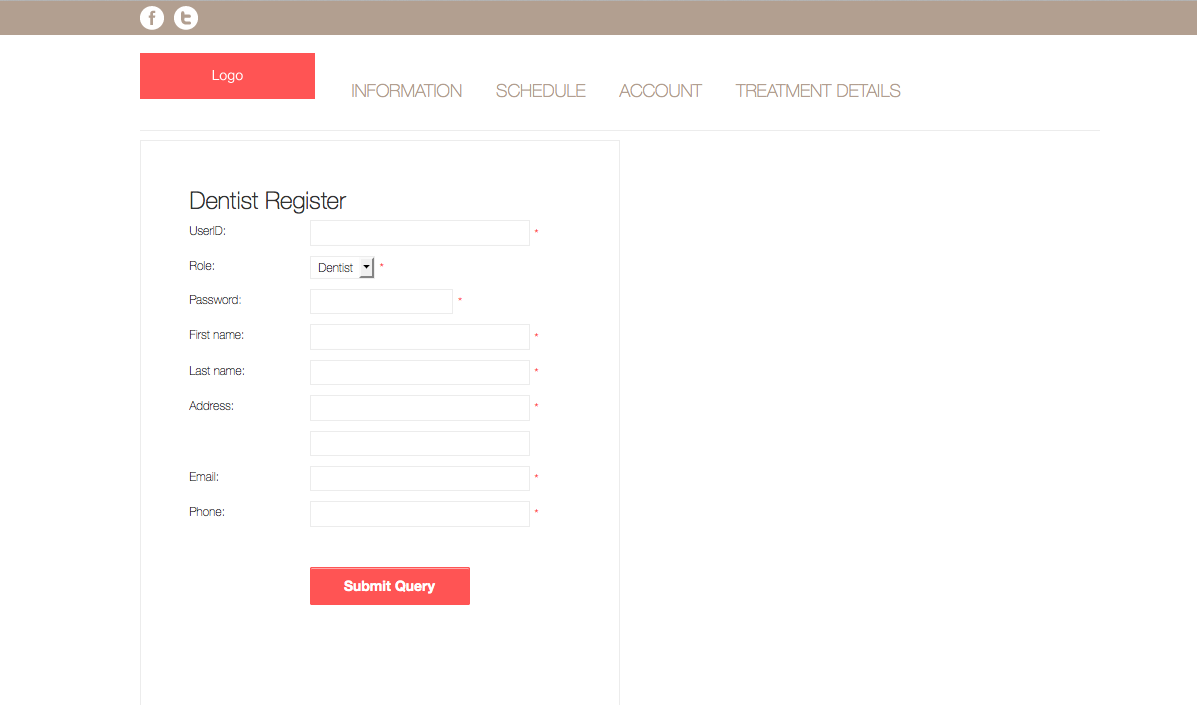
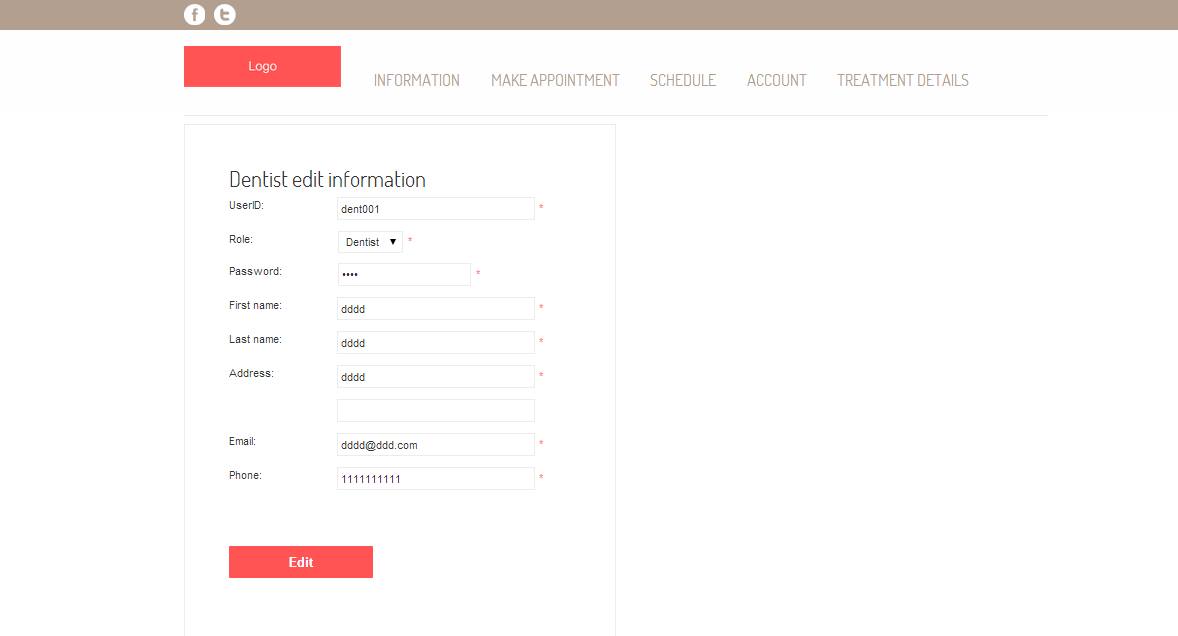


Figure 44: Shows patient list. [UI-08]

Figure 45: Create dentists’ account. [UI-09]****

Figure 46: Dentist edit information page[UI-10].

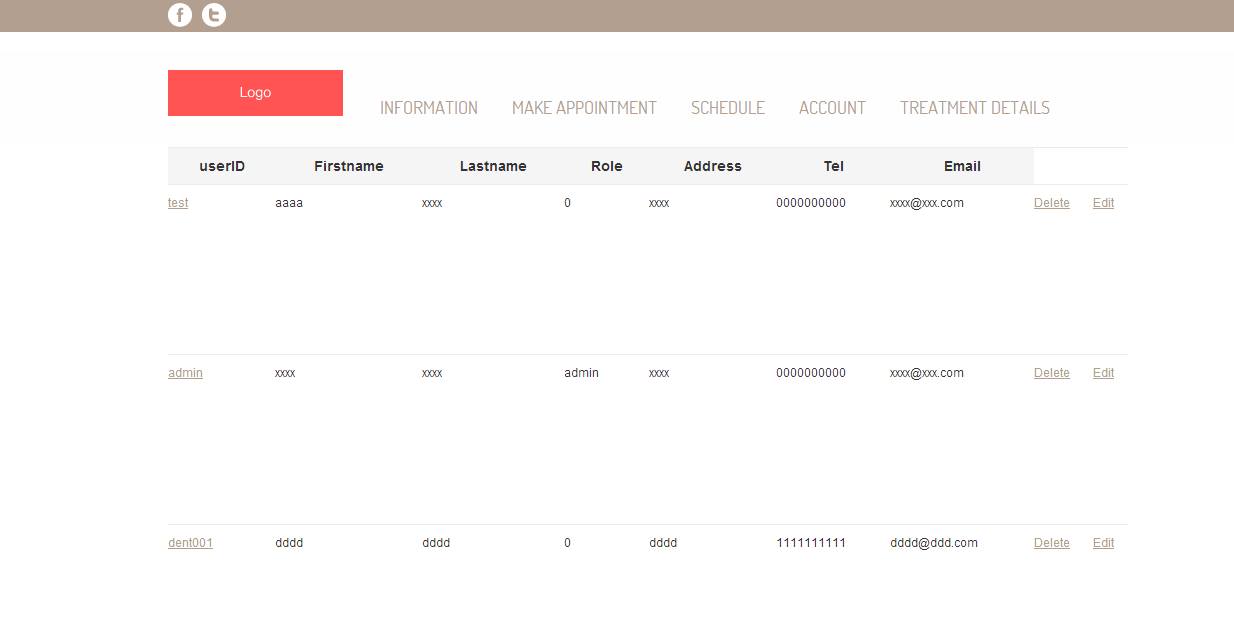


Figure 47: Dentist list page[UI-11].

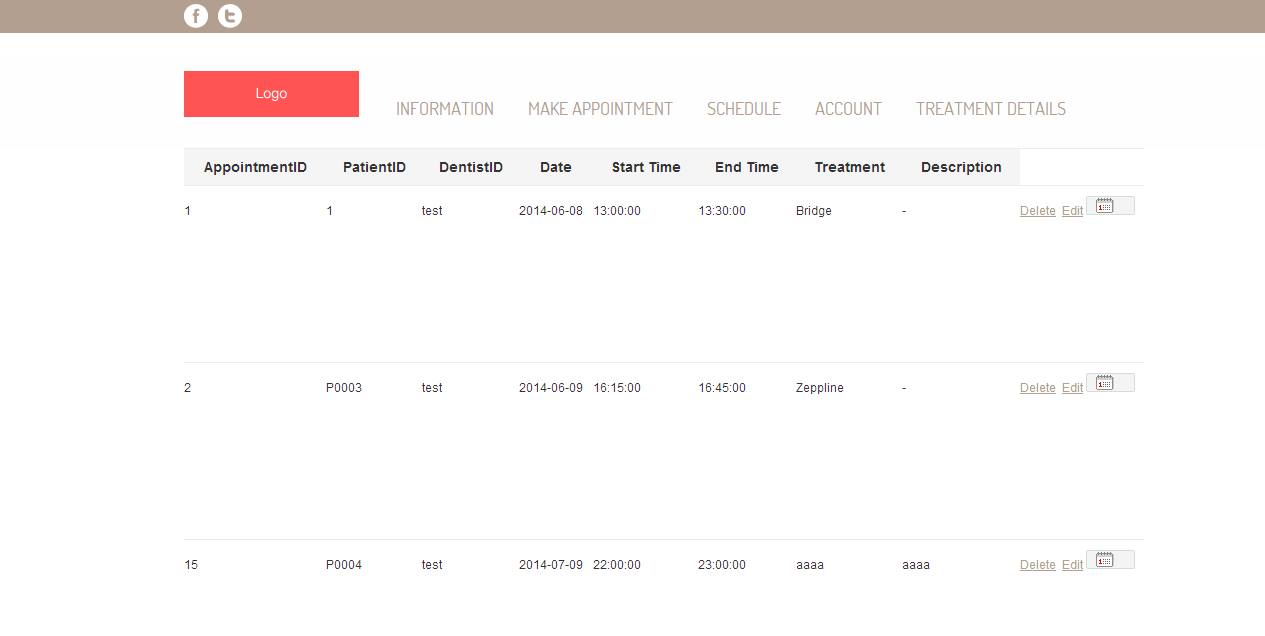
****

Figure 48: Show appointment list[UI-12].

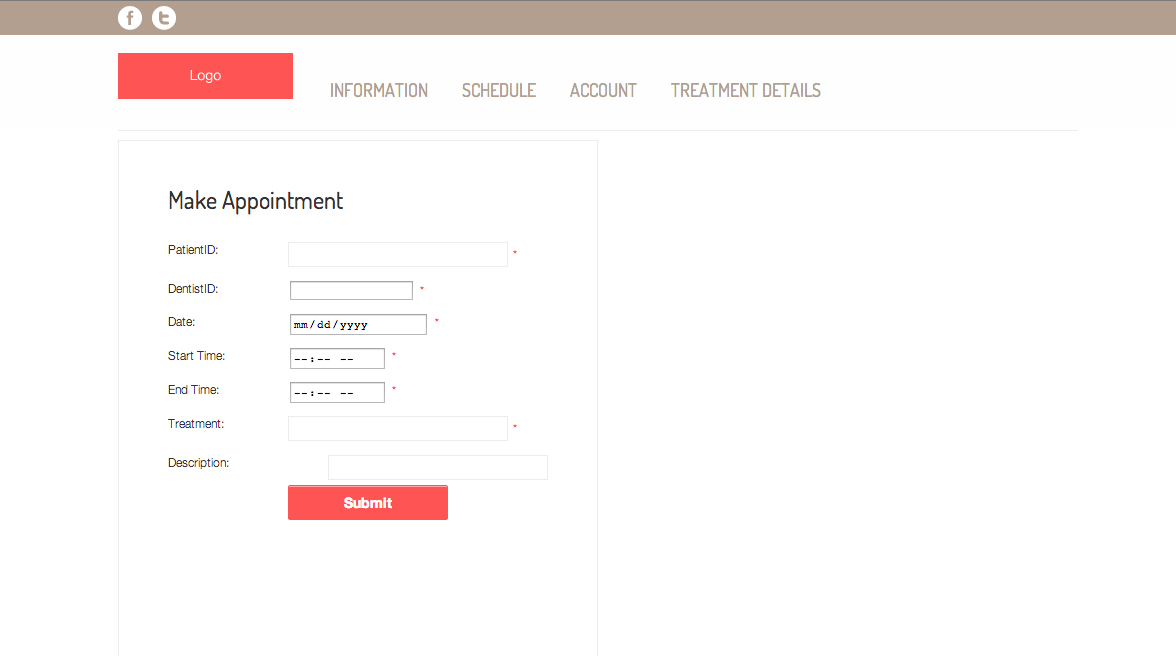


Figure 49: Create appointment page[UI-13].

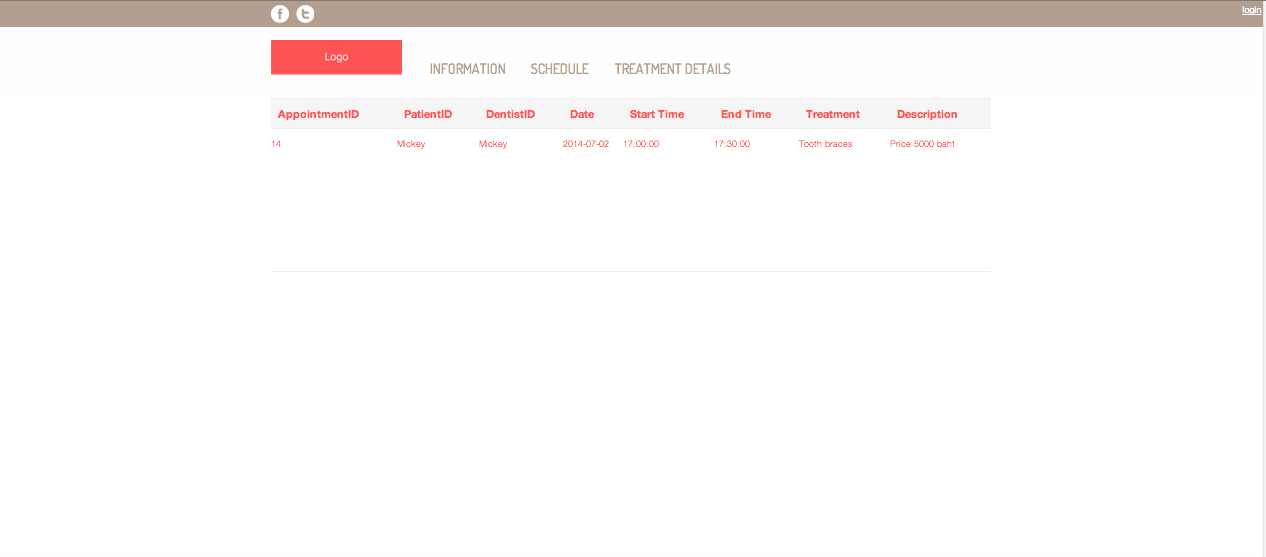


Figure 50: Show patients’ own appointment[UI-14].

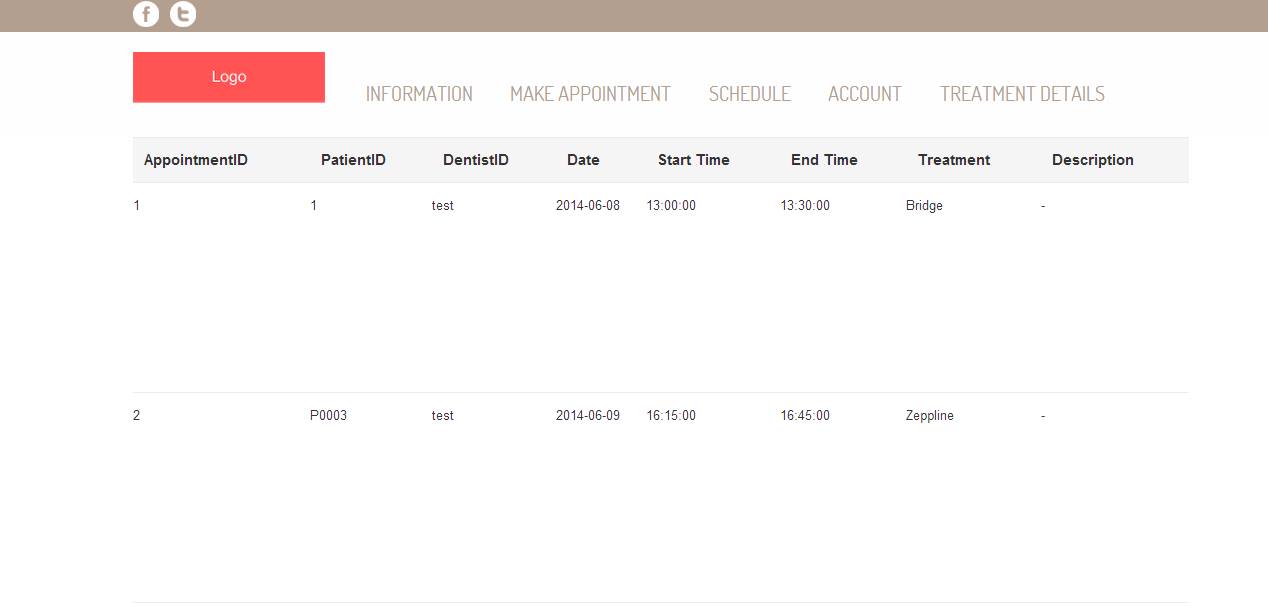


Figure 51: Show dentists’ own appointment[UI-15].

**6.2Mobile application design**

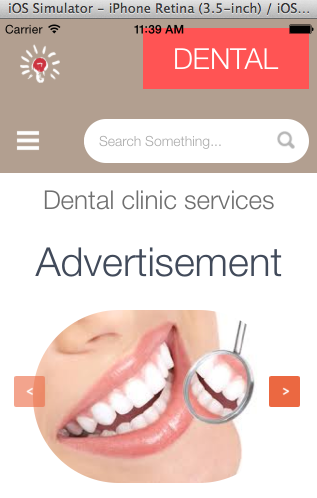


Figure 52: Mobile application index page[UI-16]

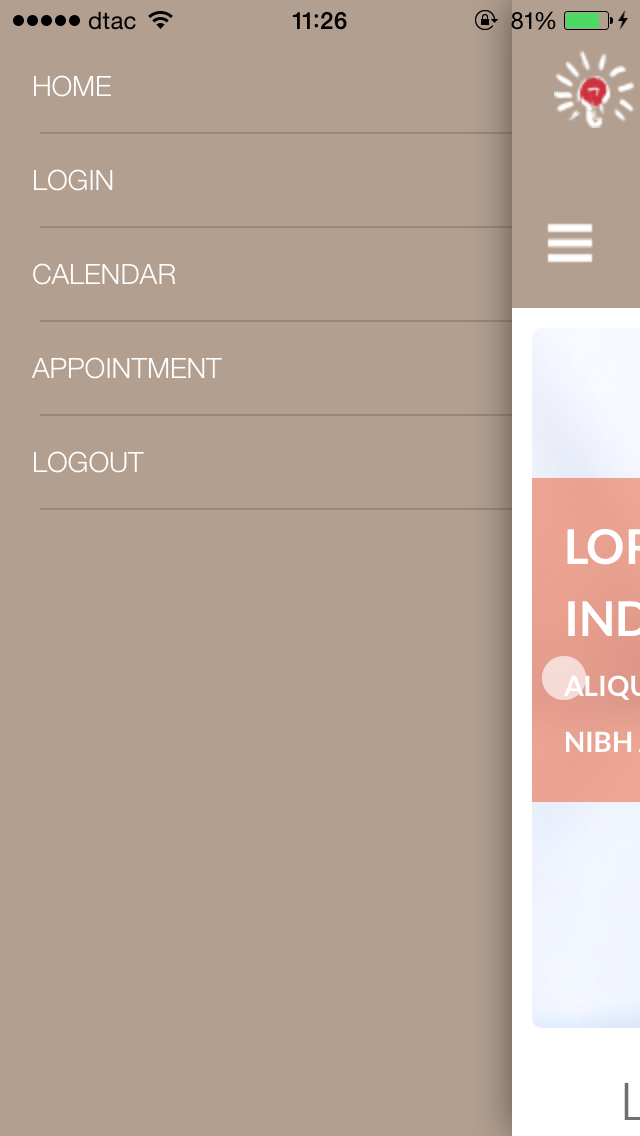


Figure 53: Mobile application menu [UI-17]

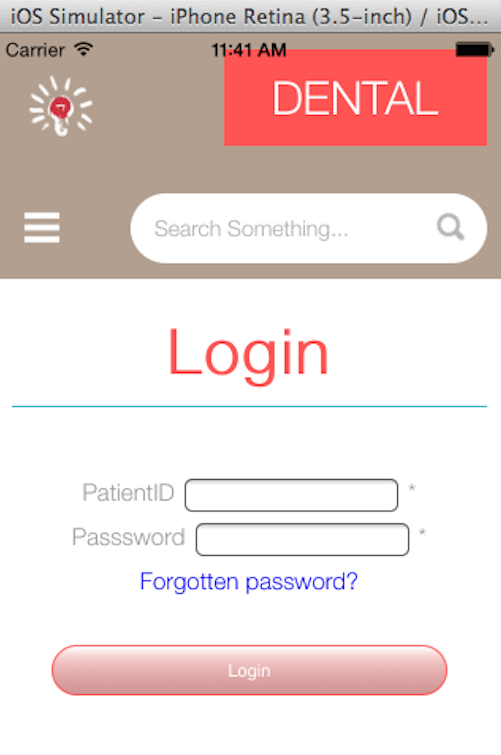


Figure 54: Login page for patient [UI-18]

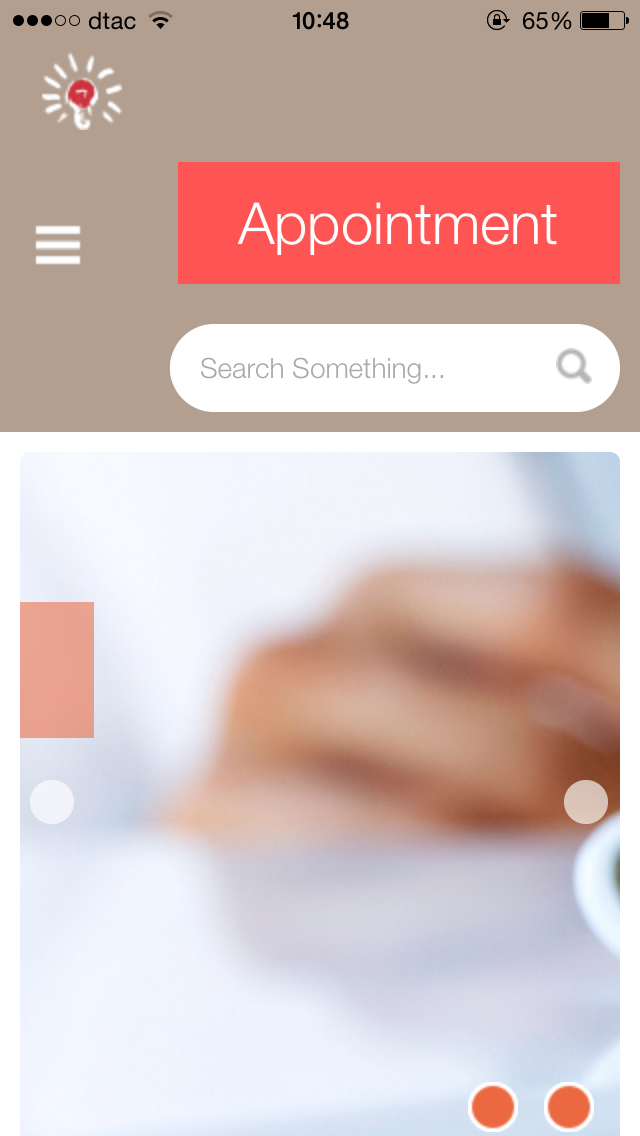
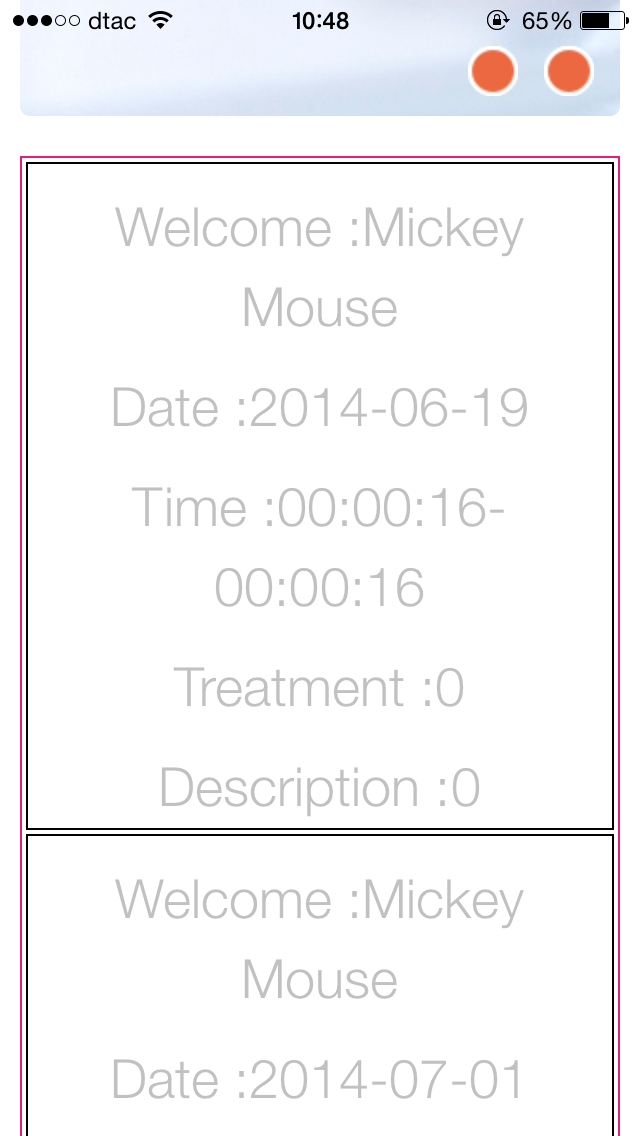


Figure 55: Display patients’ appointment [UI-19]

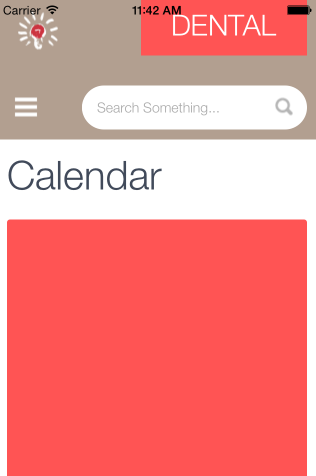




Figure 56: Dental clinic calendar [UI-20]