༄༅།། འབྲུག་རྒྱལ་འཛིན་གཙུག་ལག་སློབ་སྡེ།།



ཚན་རིག་དང་འཕྲུལ་རིག་མཐོ་རིམ་སློབ་གྲྭ།

College of Science and Technology

***Find A Doctor App***

CASE STUDY REPORT

SOFTWARE ENGINEERING

ROYAL UNIVERSITY OFBHUTAN

Software Requirement Specification   
 Version 1.0

Android Project Version 1.0   
 October 3, 2017

Find a Doctor App

Team Members:

i) Deepika Suberi (0215505)

ii) Gajendra Ghalley (0216506)

iii) Karma Dorji (0215514)

iv) Khusant Chhetri (0215518)

v) Kinzang Pelden (0216512)

vi) Megnath Sharma (0215522)

vii) Sangay Lhaden (0216524)

viii) Tshering Penjor (0216536)

ix) Tula Ram Ghalley (0215542)

Project Guide:

Mr. Yeshi Jamtsho

Lecturer, Department of Information Technology College of Science & Technology, RUB

Submitted in partial fulfillment of Requirements by Jigme Dorji Wangchuk National Referral Hospital (JDWNRH)

Abstract

This Android App ***FindADoctorApp*** is to check the availability of doctors at JDWNRH. This App provides platform to avail the services facilitated by doctors and know the live status of their availability in the hospital if users have installed this app. There would be an administrator who controls and manages the whole system and also update the status of doctors daily. Beside Admin, there is a privilege provided to individual doctors to edit and update their own status during emergency.

1. Introduction

This case study mainly focus on the SRS (Software Requirement Specification) documentation for the currently named “Find the Doctor App” for JDWNRH. The current scenario of JDWNRH is that people have to stay very long time in a queue and too much crowd in the hospital. People spend lots of time waiting for doctors without the knowledge of their availability, which is in turn the wastage of expenditure and time consumption. On top of that the traffic congestion is also a main issue existing currently due to the large group of people visiting hospital simultaneously.

Therefore in order to overcome those existing problems at JDWNRH, we are developing an android application ***FindADoctorApp*** which will check the availability of doctors. There would be an Admin who update the information about doctors’ status and doctor will also have the authority to only edit their status after providing their credentials. But users do not have to enter any credentials to avail the services. To access the facility in your phone, users have to install this app and need the internet connection.

This app will help the JDWNRH to solve the existing problems and make their system to function in a systematic and efficient ways. It will also help the users to get treatment at right time from right doctors and safes their time as well as their expenditure.

1.1 Purpose

This android-based app is developed for JDWNRH, by a team of IT

Engineering under-grad students of College of Science & Technology, under the guidance of DITT, MoIC, Bhutan.

In the current scenario, people would have to wait in a queue, or at-least require going to the hospital, to know the availability of the doctor in the Hospital, or in their chamber.

With this simple app, people would no longer have to drive or go to the hospital, just to see the availability of the doctor, there-by reducing the traffic and crowd at the hospital.

1.2 Scope

The app after its development can be used by the people of Bhutan (mainly   
people living in the capital) to check the availability of the doctors in the   
hospital.

1.3 Document Conventions

Android - Operating System for Phones / Tablets by Google.

Admin - An administrative staff of JDWNRH responsible for updating the status of the doctors.

Doctor - A doctor employed at JDWNRH. GUI - Graphical User Interface

JDWNRH - Jigme Dorji Wangchuk National Referral Hospital SRS - Software Requirement Specification

User - A person who has installed this app on their phone.

1.4 Overview

The following parts of the document contains the Overall System

Description, and describes the Project Overview with illustrations and descriptions.

The document is followed by Functional and Non-Functional Requirements.

2. Overall System Description

This section of the document describes the features of the app based on the brainstorming by the team members and a requirement statement from JDWNRH. It contains the detailed tentative design plans for system and user interface, along with the data modeling.

2.1 Product Perspective

The app is used to see the live status of the availability of doctors in the hospital. The status of the doctor is updated timely by the admin. The individual doctor would also be able to set their status to “IN or OUT” from the app, but require to login for that.

The product would be an android app and an admin interface and control   
panel, which would be a web application. The product would require   
database to store the information and profiles of all the doctors, and the   
admin.

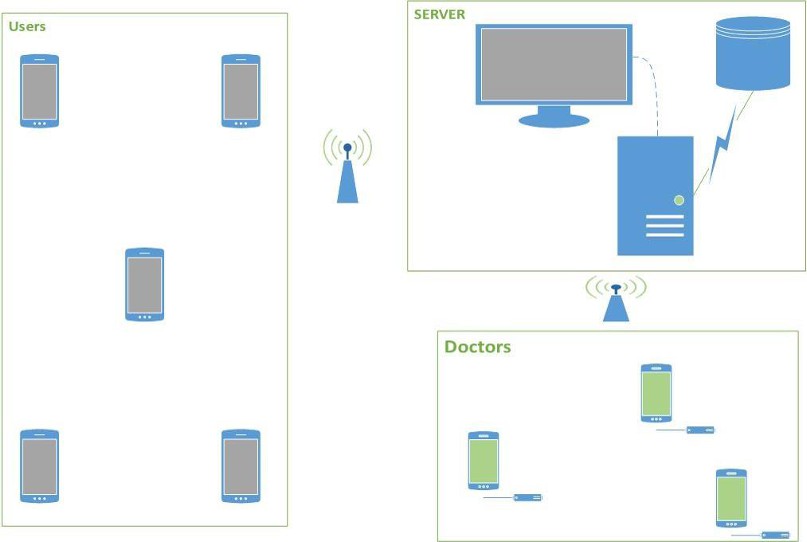
There won’t be a separate app for the doctor. Initially, after the

installation, the same interface of the app would be displayed to the doctor and a user on opening the app each time. A splash screen containing the hospital logo would be displayed for a moment, and follow by the category/departments of JDWNRH in the next screen, as illustrated in the tentative template as shown in Section 2.1.1.

On clicking a required department in the app, the user would see the list of   
all the doctor(s) in that department, and the status of the doctor (i.e. IN or   
OUT). By default, all the doctors would have the IN status from 0900 hrs till 1500 hrs. When they are out of town, or on emergency leave, either the   
doctor, or the admin would have to update the status to OUT.

For the doctor to login there would be an option in the app as illustrated in Section 2.1.1. The doctor would enter his/her credential, and would be able to update/change their status.

On changing the status of the doctor, by either the admin or the doctors themselves the information would be updated in the database Server. The   
users who are using the app would require an internet connection in their   
android phone/tablet. The app would automatically connect and retrieve the   
information from the database server, and display it to the app.



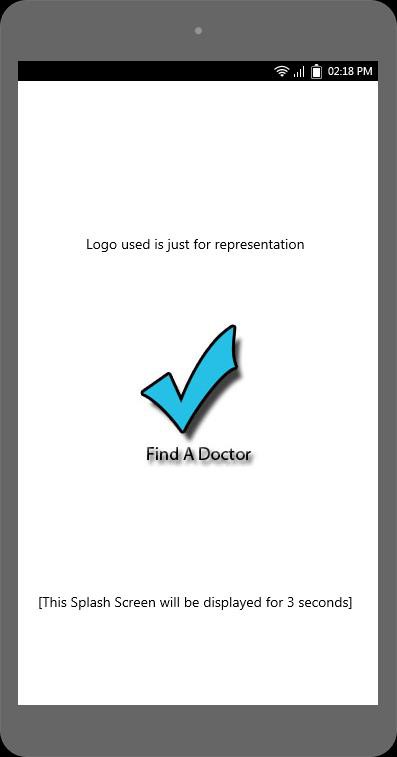
(Fig. 1)

The above figure represents the overall project architecture. Note that, the app for the users and doctor is the same. The doctor logs in using an   
interface as shown in figure 4 below.

2.1.1 Product Functions

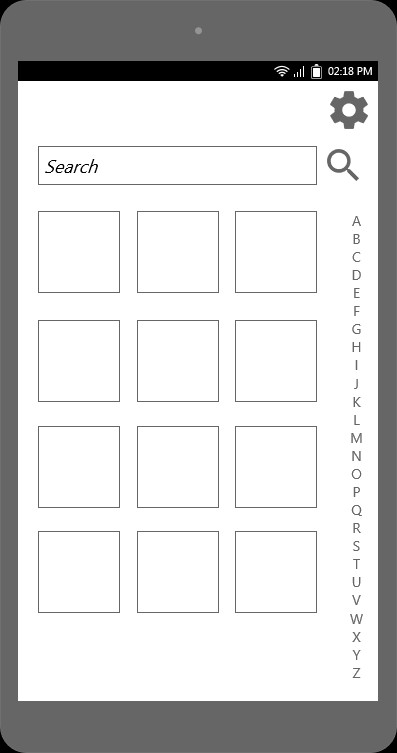
This section describes the major functionality of the app.

i) User



(Fig. 2)

The above is the splash screen that appears before the main screen loads. It would display the logo and name of JDWNRH, along with the app name.

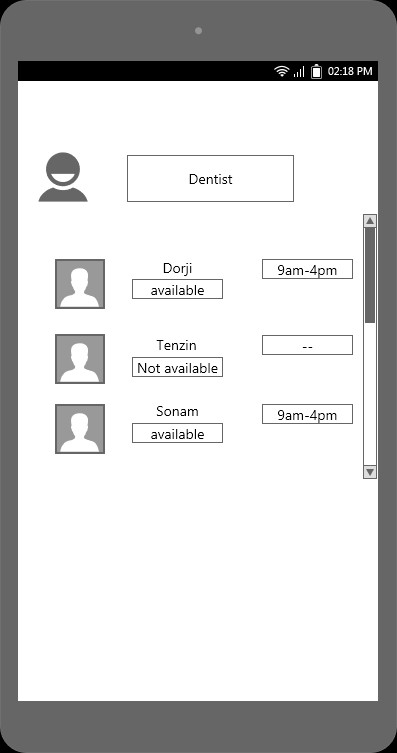


(Fig. 3)

The above screen is the first screen seen by the user/doctor. The empty boxes would contain the name and a representational picture of each Department of JDWNRH.

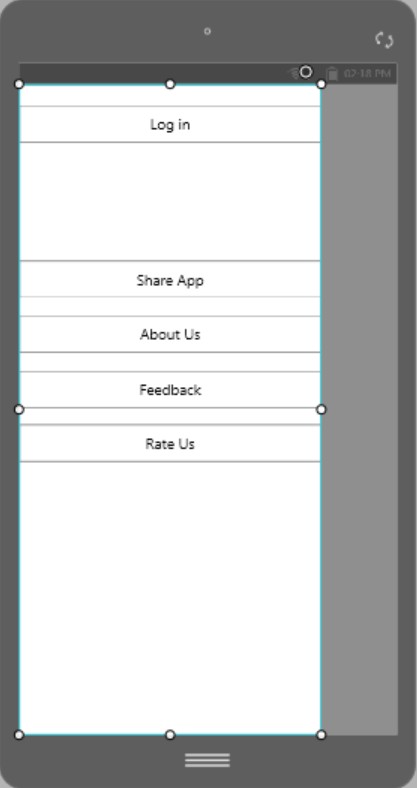
It contains a search-box where a user can search the app entering the doctor’s specialization (eg: Ophthalmologist) or the name of a   
particular doctor.

On clicking on a tile (Say Dental Department), the following screen will be displayed where the names of all the doctors of the dental department would be displayed and their status would be displayed along-side their name.



(Fig. 4)

On clicking the settings icon in the home page, the following drawer screen would be loaded.



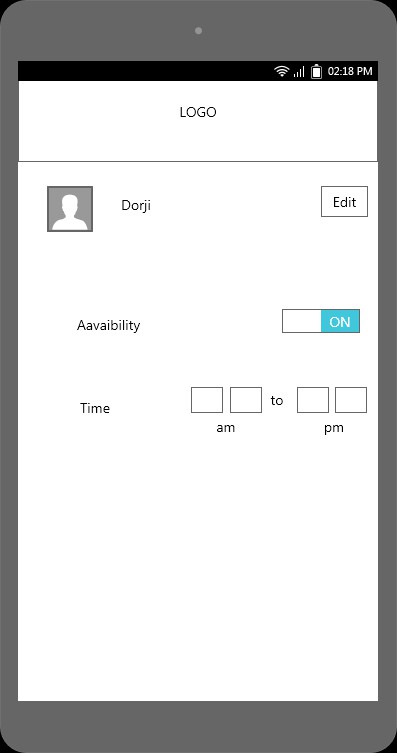
(Fig. 5)

The above buttons would work as it is named.

The “Log-In” is particularly the most important one. On clicking the   
Login Option, the login screen for the doctor would appear as below.

(Fig. 6)

The doctor would enter his/her credentials and get to update his/her   
IN or OUT Status. Upon successful Login, the following screen would   
appear.

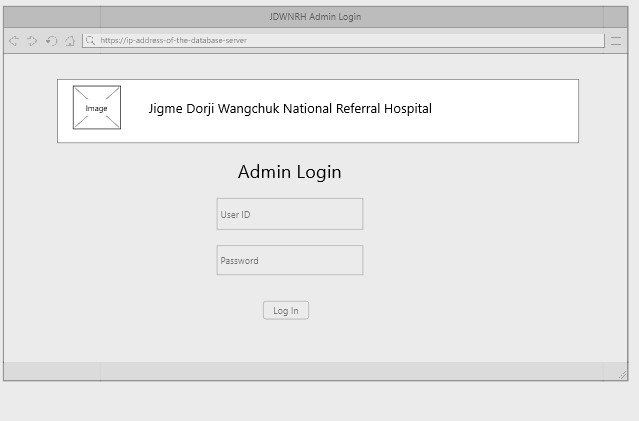


(Fig. 7)

Note: The screen-shots used above are used for representational purpose only. However, it would convey the basic functionality of the   
app.

ii) Admin

The admin would have their user-interface as a web application, on a   
desktop computer. It would be run through a browser, and typically be connected to the Database Server 24x7. The admin would be required to login for each session, and has the authority to ADD, REMOVE, & EDIT the information of each doctor. Upon logging in, the admin would have the control panel containing the list of all the names of the doctor.



(Fig. 8)

2.1.2 Hardware Interface

- Android Phone/Tablet for app users and doctor.

- Computer System running a web browser for Admin.

2.1.3 Software Interface

Requirement for development:

Operating System Environment - Windows   
Tools - Android SDK, WAMP Server   
Platform / Framework - Android Studio

Technology - Java, XML, PHP, HTML, CSS, JavaScript Database - MySQL, SQLite

Browser - Google Chrome / Mozilla Firefox Version Control System - Git

2.1.4 Communication Interface

Android API Packages for Communication would be implemented for the session initiation, and communication with the server.

2.2 User Characteristics

Any person using an android-based Mobile Phone/Tablet can install and use the app. However, the app would be only supported in Devices running Android API Level 17 and above.

2.3 Constrains

Internet Connection is required to obtain the live information.

Login for Admin and Doctors.

Admin is required to update the database daily, or maybe few   
 times a day.

Doctors may update their information, if they are going out for   
 longer duration.

3. Specific Requirement

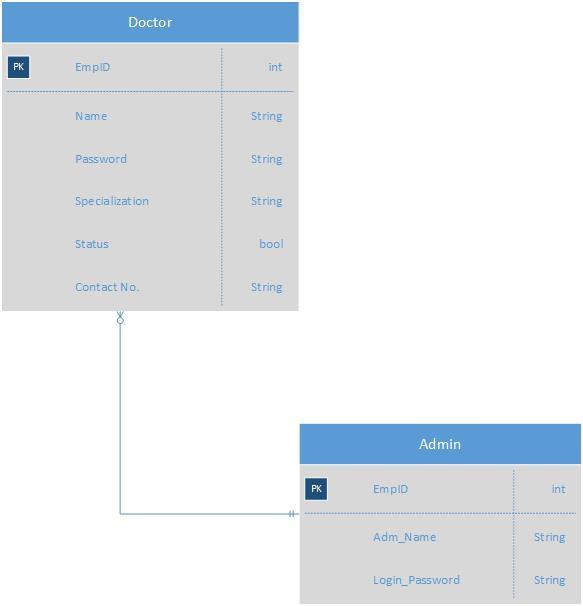
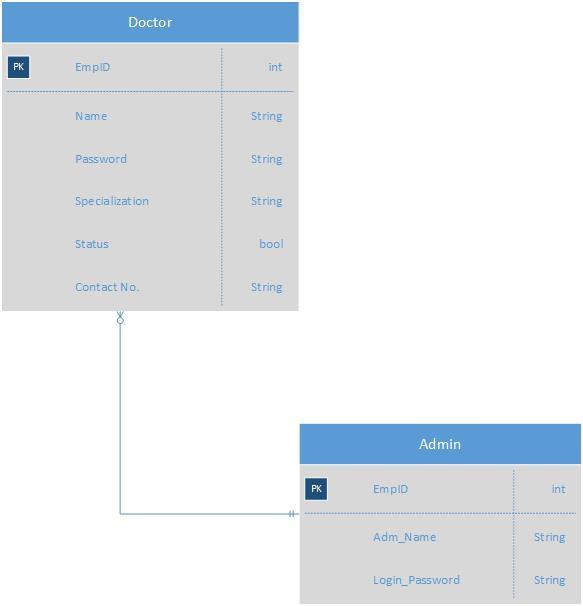
3.1 Functional Requirements

3.1.1 User Interface

The user interface would be similar to the screen shots from figure 2 through 7.

3.1.2 Data Modelling

The following diagram is an Entity Relationship diagram using CrowFoot Notation for RDBMS model, for the database of the app.



3.2 Non-Functional Requirements

3.2.1 Reliability

The app needs to be reliable, as the users would make decisions based   
on the information available on the app. The information must not be   
stale and it must be updated regularly, by both the doctor and admin.

3.2.2 Availability

The app must be available in all the common places freely, so that anyone willing to install has an easy access.

3.2.3 Maintainability

The app must be updated based on reviews and feedback.

3.2.4 Portability

The app must be light, and must not consume a lot of space, in both primary storage and secondary storage. Efficient coding techniques and effective choice and implementation of algorithms is necessary to achieve this state of the app.

**Methodology**

Discussion on data requirements

Data collection

Internet

Interview

Brainstorm

Research

B

**Data Objects**

Requirement Analysis

Entity Relationship Diagram

|  |  |
| --- | --- |
| **Entity Name** | **Description** |
| Admin | It contains the details about the Admin where they have  to provide their Admin\_id and password to login and  has the authority to control the whole management system. |
| Doctors | It contains the details of all the doctors with their name and specialization where each doctor has to enter their credentials Like Doctor\_id and Password to login. Basically it provides doctors the privilege to edit their profile and status. |

**Data Dictionary:**

**Admin**

|  |  |  |
| --- | --- | --- |
| **Sl. No** | **Attributes** | **Description** |
| 1 | **Emp\_id** | Unique id given to the Admin which identify them. |
| **2** | Admin\_Name | Name of Admin |
| **3** | **Login\_Password** | This is for security and authentication to access the  Databases. |

**Doctor**

|  |  |  |
| --- | --- | --- |
| **Sl.No** | **Attributes** | **Description** |
| 1 | Emp\_id | It is a unique number given to the entire employee to identify them. |
| 2 | Name | Name of Doctors |
| 3 | Password | For security and authentication purpose |
| 4 | Specialization | This is to provide in which field doctors are specialized. |
| 5 | Status | It will update the information about doctors’ availability in the campus with time. |
| 6 | Contact no | This is for more information about doctors and also during emergency. |

Conclusion

This is the end of version 1.0 of the SRS Document for the “Find the Doctor App”.

THANK YOU