CMPT 276 Project Report

1. Abstract

As a lot of self management mobile application is created. However, there is few health management mobile application can be found in the app store.This paper addresses the problem of using mobile application to manage a small clinic to handle the relationship among the patients, the doctors and the stakeholders, which is developed on the Google Android platform, and presents how this mobile application will work in the realistic cases. The application is based on the user stories from the daily life. The mobile application structure is developed by the class diagram and supported by the interaction model to help organize the group members better understanding the whole mobile application how it will be designed.

1.1. Introduction

Our mobile application name is called “Health manager” which is developed with Android API16 Jelly Bean Operating System.We mainly target at three types of users in our application, which is patients, doctors and stakeholders. All of these three users are assigned with different functions for their different roles. All of the functions are generated from the user stories. The major specialization of our mobile application is managing appointments between the patients and the doctors.

2. Software Management Plan

2.1. Project

Overview:

The project is developed by the scrum which is an agile management method. The duration of the Sprints was separated into three terms. The roles in the project was Scrum Master, Product Owner and Team Members. With a clear role assignment. The whole system is split into components before the project being started to develop(figure 1).

Figure 1

2.2. Project deliverables

The delivery of the project is divided into three terms with the iterations in every sprints. In the first sprint, the system structural model diagram(figure 2)was decided with the user requirements and the inner discussion among the developers.According to the limited time, the project was split into general features and desired features. The basic class perspective model is also directly created in the IDE(Eclipse) with named java class. In the second sprint, the majority of the project was concentrated on achieving the features have been listed on the system structural model diagram and ensure every basic components interact well as the initial diagram designed. The basic UI(user interface) was embedded in the the application. In the third sprint, the project was focusing on the optimization of the whole system. The UI optimization will be the most significant part in this sprint. Then the bug repair will be the same important as the UI optimization. The last part will be realize the desired feature. When the sprint 3 is done, the project will be deliverable. During the three sprints, project was kept reviewing by our customer with every sprints has been done. The test of the application was also kept by our test developers.

2.3. The Organization of the Project

2.3.1. Roles and Responsibility of software team member

The team members who are working on this project was separated into three roles, which is Scrum Master, Product Owner and team members. The Scrum Master is focusing on the management of the gitlab ,strategy suggestion,part of features developing and testing. The product owner was focusing on organizing the whole project process ,strategy suggestion, part

of features developing, UI designing, testing and talking to the customers. One of the team members was focusing on the documents, part of features developing, testing and strategy suggestion. The other one was focusing on the components combination and strategy suggestion

2.3.2. Tool and Techniques

- Eclipse IDE

- AVD-Android Emulator

- SDK-API 16(Jelly Bean)

- Java

- Junit

2.4. Software Management Plan

2.4.1. Tasks needed to complete

1. Basic feature should be done.
2. Junit Test should be done after each iteration.
   Team members should go to fix the issue when there is a issue coming up
White-box Test should be done by developers who is in charge for this part before and after fixing the issue.

2.4.2 Measurement

1. **Intended User Experience Measurement**
   Non-group members will be invited to test the App
2. **Feature Evaluation**
   In order to measure the completeness of feature implementation, developers will examine all features that have been implemented.
3. **Deadline Evaluation**
   The Product Owner will evaluate the deadlines for different components in order to measure the difficulty of the project and the project shortcomings, who will then adjust the development strategy to accommodate those results.

2.4.3. Risks Management

1. **High Risk:**
   A large number of rows in the Patients and Doctors database table will reduce the efficiency of selection query without a high-efficiency sorting algorithm.
2. **Moderate Risk:**
   The application is designed based on API 16 (Jelly Bean), may have the display issue with the latest Android device.

3. **Low Risk:**

The user interface may not align well when the new resolution portion comes out.

2.5. Assignments

Name	Roles	Responsibility
Leo Shang	Scrum Master	
James He	Product Owner	
Zack Zhou	Team Member	
Mick Shu	Team Member	

3. Software Requirements Specification

3.1. Requirements Specification

3.1.1. External Interface Requirements

- User Interface (a warm-heart and comfortable design)
- System Platform (Android)
- Hardware Requirement (Android device)

3.1.2. Software Features

- Patient:

1. View treatment history:

Patients can check their disease history and have a general view of treatment

2. Appointment (make & cancel appointment), view doctor schedule :

Patients can make and cancel appointment after click “Appointment”. After patients make appointment, they can view doctor schedule so they will not miss the doctors’ working time.

3. Log out

After users finish their activity, they can click the “Log out” button to exit account.

- Doctor:

1. View my schedule:

Doctors can check their daily schedule after click the “schedule” button.

2. Start treatment:

After users click the “start treatment” button, the doctor can diagnose and prescribe patients. After finishing treatment, click the “finish treatment” button.

3. Log out

After users finish their activity, they can click the “Log out” button to exit account.

- Stakeholder:

1. View basic information of employees:

Stakeholders can check employee’s basic information.

2. Review medication repository:

Stakeholders can check medication repository and decide to buy new medicine.

3. View expense and profit:

Stakeholders will get a finance report of expense and revenue for the clinic.

4. Log out:

After users finish their activity, they can click the “Log out” button to exit account.

3.1.3. Software Attributes

- Security:

1. keep the information of employees private
2. keep patient personal information private

- Maintainability:

1. the system of the application is reusable
2. the application can add layer to enrich the features of the application

- Performance:

1. the response speed for an event is quick right now for a small quantity access.

- Availability:

1. available to small or medium-sized clinics

3.1.4. Database Requirements

1. Enable to record the treatment history of the patient.
2. Enable to interact the appointment management between the patients and doctors(including the exact time)
3. Enable to review the Medicine repository

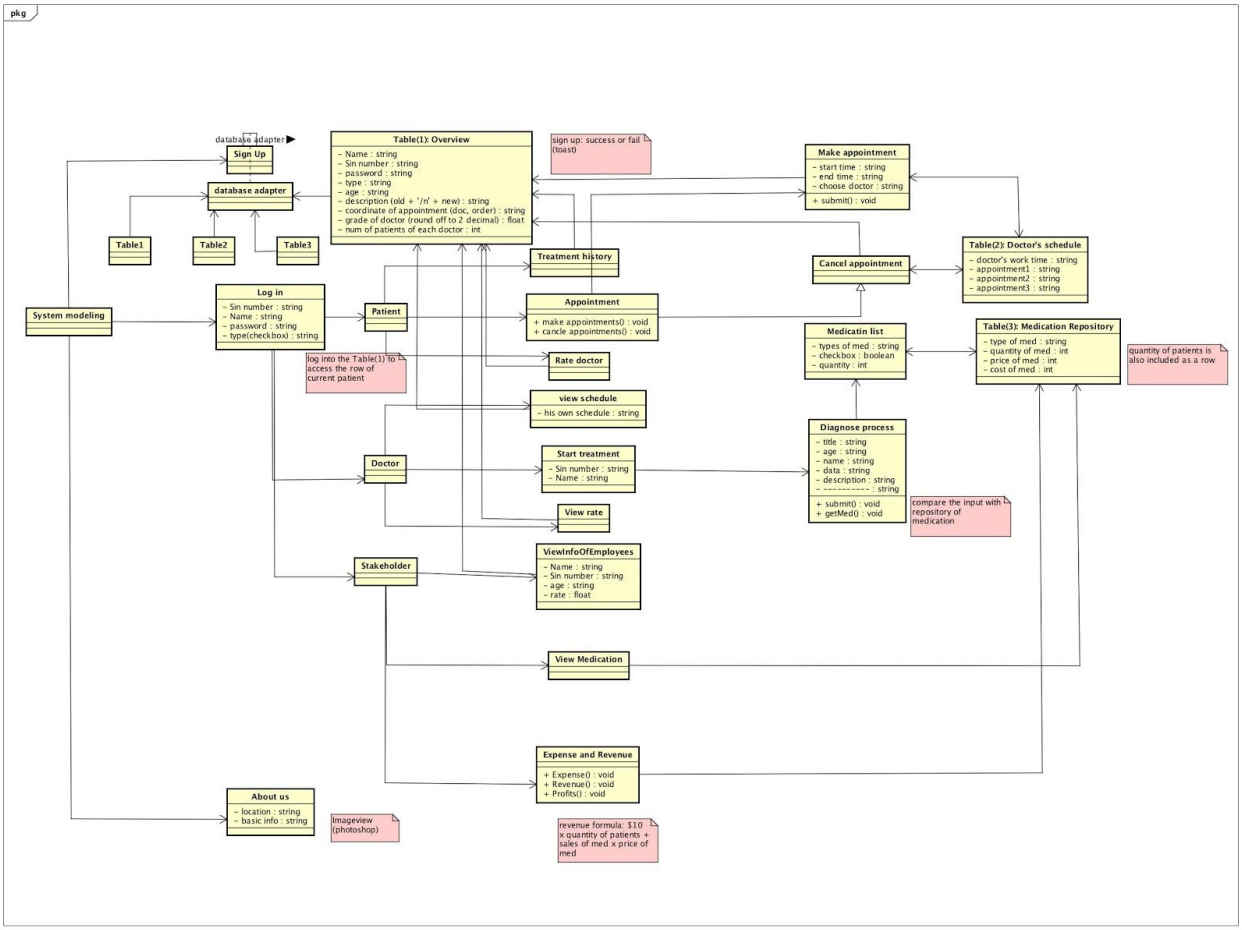
4. Software Design Documentation

4.1. Design Overview

- Object-oriented design

4.2. System Modeling

- The Perspective of Structural Modeling
- UML (class diagram)



4.3. System Architecture Design

Our software is using layered architecture, the advantage is that our software can be separated into four layers. When we want to add new functionality to the software, we could add button or edit text on user interface and make changes based on the interface.

- User interface: Our user interface is composed of sign_up page, log_in page, and about_us page. The majority of the application features such as making appointment, rating doctors are embedded within log_in activity.

- User interface management: To access to database, patient or stakeholder needs to sign up to authorize themselves to able to log into the system to interact with application

functionalities. For doctor, the info including name, sin number, and password are inserted into database by developer who have the access to the source code, in this case.

- Application functionalities: The majority of application features are embedded within log_in activity. The users with different identities will be led to corresponding pages.

- Database: The database in this app interconnects all functionalities in this app. There are three tables included in the database that contain the specific information for corresponding activity.

4.4. User Interface Design

Our user interface is user-friendly, we make all the background images on Photoshop. We choose blue and green as dominant hue, because these two colors will make user interface looks neat. We put button align center on sign up and log in interface, but we separate button and edit text on other interfaces such as doctor make new description and patient make appointment.

User Interface Images:

- Our app icon will appear on sign up and log in interface.

- The button background color is setting transparent because we make the button image into layout background.

- About us interface is scroll and background image is longer than other interface background image.

Input Objects and Actions (Buttons, Edit texts)

- On sign up interface we have four edit texts: name, age, SIN number and password, and sign up button.
- On log in interface we have three edit texts: name, SIN number and password, 3 different check-box for different types of clients: patient, doctor, stakeholder, and log in button.
- On Main activity interface we have three buttons: sign up, log in and about us.
- On patient activity interface we have four buttons : view history, appointment, rate my doctors and log out.
- On make appointment interface we have one buttons: view doctor's schedule, and two edit texts appointment start time and end time.
- On rate my doctor interface we have one edit texts: give doctor rate integer from 1 to 5, and submit button.
- On new description interface we have three edit texts: Patient's name, SIN number and description of diagnose, and two buttons: view medicine and submit.

5. Software Testing Documentation

5.1. Test Approach
5.2. Test Plan:
5.2.1. The software features to be tested:
1.signup 2. 3 different types log in 3. Patient view history 4. Patient appointment 5. Patient rate doctor 6. Doctor view schedule 7. Doctor start treatment 8.Doctor view own rate 9. Stakeholder view Doctor's information 10. Stakeholder view medication repository 11. Log out
5.2.2. The software features not to be tested:
1.Doctor check medication repository 2. Stakeholder view expense and profit
5.2.3. Test Environment:
1.JUnit environment 2.Eclipse
5.3. Test Cases:
Patient and Stakeholder sign up. First, patient login , check the history. Patient make an appointment and cancel it. Patient make an appointment again, view doctor schedule, choose a specific doctor and enter appointment start time and end time. Patient log out. Second, Doctor login check schedule to see only one appointment this time. Doctor meet Patient and start treatment, enter Patient's name and SIN number, enter patient treatment description, submit the treatment description. Doctor logout. Patient login and rate the doctor who made the diagnose. Patient log out. Doctor login and view the rate given by patient. Doctor logout. Third, Stakeholder log in , check doctor's information , check medication repository. Stakeholder log out.
5.3.1. Purpose of the test case:
Test signup and login matched, test history display correctly, test no more than one appointment made for patient, test cancel appointment, test appointment time correctly.
Test view schedule, test access and edit patient history, test treatment description submit correctly. Test patient's rate and doctor view rate matched.
Test view doctor's information,test check medication repository
5.3.2. Inputs
1. Sign up enter name, age, SIN number and password. 2. Log in enter wrong information. 3.Log in enter correct information. 4. Click view history 5. Click appointment and make an appointment 6. Click make an appointment again. 7. Click cancel the appointment.8. Click view doctor's schedule 9. Enter start time and end time exceed doctor's work time. 10. Enter start time and end time in doctor's work time 11. Click view own schedule 12. Enter patient's information contains name ,SIN number and description and submit 13. Click view history 14. Enter rate number out of range integer 1 to 5 15. Enter rate number in range inter 1 to 5 16. Click view my rate 17. Click view doctor's information 18. Click check medication repository.
5.3.3. Expected outputs (pass / fail criteria)
1. Sign up success pass 2. Warning login fail because wrong information 3. Log in pass 4. Nothing display pass, have history fail because it's a new customer. 5. Start choose doctor pass, nothing change fail. 6. Warning make an appointment fail because only make one appointment each time 7. Can start a new appointment pass, cannot start a new appointment fail 8. Pop-up window pass, nothing change fail 9. Warning start time and end time fail because exceed doctor's work time 10. Make an appointment success pass 11. View 2 appointments fail because cancel appointment does not work, view 0 appointment fail because make appointment does not work, view 1 appointment pass 13. Nothing display fail because doctor's start treatment does not work, new description display success pass 14. Warning rate doctor fail because does not accept

any number out of range integer 1 to 5 15. Success rate pass 16. Display number out of range integer 1 to 5 or not same as patient's rate fail because it allows number error or the numbers are not matched, display number in range and same as patient's rate pass 17. View doctor's information contains name and rate in range 1 to 5 pass, only name fail because rate does not display. 18. Pop up window pass , nothing change fail

5.3.4. Results

All the tests mentioned above have already been executed. All feature are tested except 1.Doctor check medication repository 2. Stakeholder view expense and profit. JUnit environment is required for simplified debugging. During the test, we meet some problems. Patients cannot cancel the appointment if they did not make appointment before. New description was not add to the patient's history.

System would crush if doctor submit the new description without fill in anything. Patients gave rate to doctor and the rate exceed range from integer 1 to 5. Those problem were founded in the testing and fixed. We test activities separately, and for some different activities involve same database we test them together. Finally we combine all the components together and test on SDK-API16. Although the user interface image has small conflict with context, other features are will perform on the advanced programming interface.

6. Software Ethnics

6.1. Intellectual Property Rights

6.2. References Lists:

- Source code:
- User Interface Images: