

EC Doctor Mobile Application

R. Arun Ram Deepaoli¹, K. Gousik², M. Ruthrashree³

^{1,2,3}Student, Department of Information Technology, Bannari Amman Institute of Technology, Erode, India

Abstract: In today's world if someone wants to consult a Doctor we need carry the patient record. This consumes papers and insecure. Sometimes the record may find missing. The objective of this project is to build a system that will provide the online maintenance of records of the patient. The patient or student will able to find his record online anytime, anywhere to his/her use. The doctor will upload the details and prescription to be provided in the cloud. The system will save paper work and records. It will be eco-friendly. In pharmacy student may able to buy the medicines by telling his/her register number with a proof. The parents of the students are also able to view the records of their children. It helps parents to track the student's health. The system will prove to be useful for patient (student) as he/she can check his details whenever and from wherever he wants from his mobile phone.

Keywords: EC doctor

1. Introduction

EC Doctor is an online android application for easily providing the prescription for the patients. Here, the patients are the students since it is designed for hospital in college. Anyway past system using papers for prescription is not secure and not eco-friendly since it uses papers. Anyone can misuse the prescription and create some illegal problems. By eliminating these issues, this project is aimed at developing an online application for doctor prescription. In this project doctor enters the prescription and not in the paper. The patients who are the students have to view their prescription in their login. In Pharmacy, only the tablets are shown. Parents can also see the students' things. Doctor, Pharmacy, Students and Parents are provided with Login credentials. Students cannot view other person details. Doctor have to login with his login credentials. After logging into doctor panel, he/she can enter the details and prescription of the student and upload it to the cloud storage. Similarly, the pharmacy has to the respective login credentials to see the students' prescription. The Students have to enter their respective login credentials. Also, Parents are provided with login credentials to safe login.

2. Proposed method

This software can be readily used by non-programming personal, avoiding human handled chance of error. This project is used by three types of users namely, Doctor, Pharmacy, Students and Parents.

A. Doctor

The system includes the first type of user as Doctor. Only Doctor is able to control that system. Update any system information such as register student, view student, delete student. There is no involvement of any user other than doctor to update the information due to security purpose.

B. Pharmacy

The second types of user, Pharmacy has limited access. He/she can view only the Tablets provided by the doctor to the student. The Pharmacy can view the students' prescription by his/her register number only when the student shows the respective Identity proof.

C. Students

The third type of user is Student. He/she can login with their respective credentials. By logging into student panel able to view only their respective details. Students can view all the details that doctor have entered about them including prescription, current check-up date, next check-up date.

D. Parents

The fourth type of user is Parents. They can track their children's health by logging into parents' panel. The students' panel and the parents are almost the same.

E. Flow Diagram

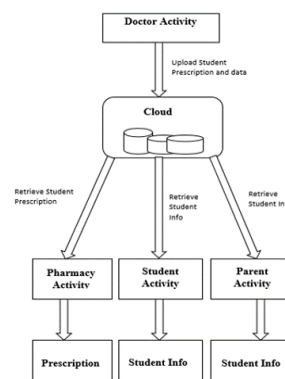


Fig. 1. Flow diagram

3. Module description

- Main window
- Doctor window
- Pharmacy window
- Student window

- *Main window:* The main window of the application has four panels namely Meet Doctor panel, Go pharmacy panel, Student view list, Parent view list seeking patient name with respective login credentials.
- *Doctor window:* In Doctor Window, the details of the students like student register number, tablet suggestion, Description of the problem, current checkup date, next checkup date. Doctor also has to enter his name in this window. Once the details are uploaded to the cloud, Successful alert message is shown.
- *Pharmacy window:* After entering the pharmacy login credentials, pharmacy window opens. In this window the pharmacist has to enter the student register number. Then the tablets suggested to the respective student by the doctor is shown.
- *Students window:* After entering the student login credentials, it enters into the Student Window. Here student can view their entire details given by the doctor.
- *Parents window:* The parents are asked to enter the login credentials given to them. They can see their son/daughter 's details. Parents have to enter the register number of their children.

4. Testing

Testing is a procedure, which uncovers blunders in the program. It is the real quality measure utilized amid programming improvement. Amid testing, the program is executed with a lot of conditions known as experiments and the yield is assessed to decide if the program is executing of course. Programming testing is the way toward testing the usefulness and accuracy of programming by running it. Procedure of executing a program with the expectation of finding a mistake. A decent experiment is one that has a high likelihood of finding a so far unfamiliar blunder. An effective test is one that reveals an up 'til now unfamiliar mistake. Programming testing is generally performed for two reasons, Defect location and Reliability estimation

A. Testing objectives

Testing is a procedure of executing a program with the goal of finding a mistake. A great experiment is one that has a high likelihood of finding a so far unfamiliar. An effective test is one that reveals a so far unfamiliar blunder.

B. Testing principles

All tests ought to be discernible to client prerequisites. Tests ought to be arranged vast before testing starts. Testing should start "In the Small" and advancement towards "In the Large".

C. Unit testing

It is also known as component testing refers to tests that verify the functionality of a specific section of code, usually at the function level. These types of tests are usually written by developers as they work on code, to ensure that the specific function is working as expected. One function might have

multiple tests, to catch corner cases or other branches in the code. Unit testing alone cannot verify the functionality of a piece of software, but rather is used to ensure that the building blocks of the software work independently from each other. In my system unit testing is done by testing various java script functions. For example, I write a java script code for validating user input in sign up page. When a user wants to register without having any password I want that system will show an error message. I do this using java script. Likely password less than four characters, first name or last name contain any digit, invalid account no user name already used etc. That's unit testing for my system.

D. Integration testing

Integration testing is any type of software testing that seeks to verify the interfaces between components against a software design. Software components may be integrated in an iterative way or all together. So in integration testing I tested the whole module as different parts are combined together. Execute all the circles at their limits and inside their operational limits.

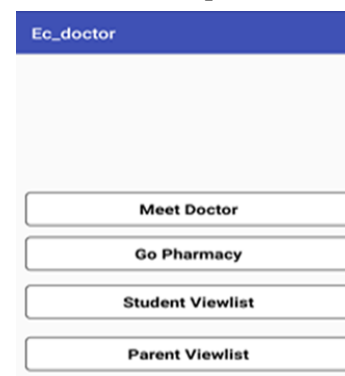
E. Component interface testing

The practice of component interface testing can be used to check the handling of data passed between various units, or subsystem components, beyond full integration testing between those units. The data being passed can be considered as "message packets" and the range or data types can be checked, for data generated from one unit, and tested for validity before being passed into another unit. For component interface testing, when I complete student module test its integrated parts then I combine its result with doctor module.

F. System testing

Once all bricks are combined into a system, this system may not be functioning as expected, even if integration tests haven't revealed anything wrong. Here, system tests come into play. System testing, or end-to-end testing, tests a completely integrated system to verify that it meets its requirements. For this testing as the whole system is completed I checked whether the whole system works fine. If there are any contradiction among the modules.

5. Output



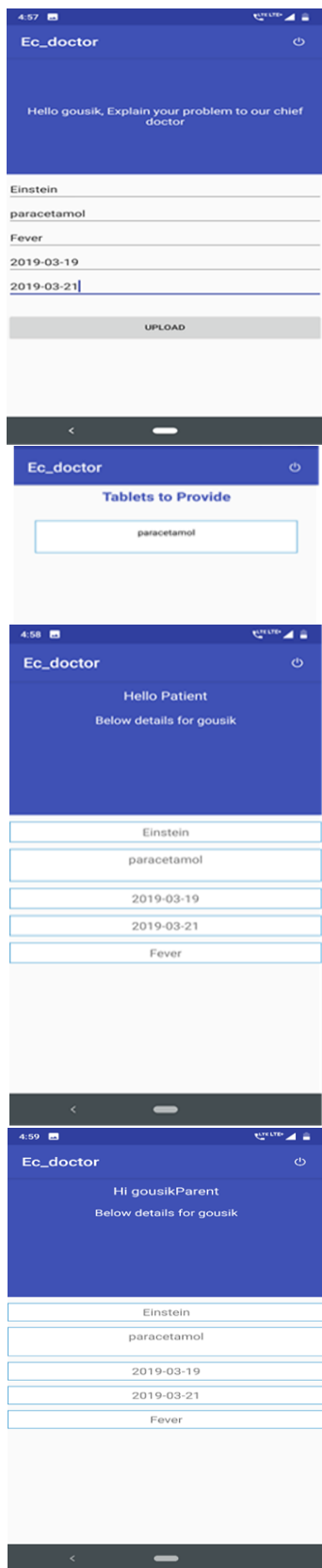


Fig. 2. Process

6. Conclusion

After finishing up my project I have to state that, I tried my best in order to develop the software in the most suitable, helpful and easiest method that can be used by the user easily. Users can easily pick up their expected information based upon their privilege. This is android based software and it will be going to accessible from any computer by using internet. I have tested the system using several techniques to determine the system flexibility. I tried to develop automated doctor appointment and doctor help management system software which recovers as possible the drawbacks and limitations compared to this types of existing software's as well as for efficient use for users. It will save money, time and energy to appoint a doctor. Provide safety and security to data enable the system administrator to authenticate all of the users through user names and password so that unauthorized users do not get access to the hospitals data.

References

- [1] <https://firebase.google.com/docs/storage/android/start>
- [2] <https://developer.android.com/studio/write/firebase>
- [3] <https://stackoverflow.com/>
- [4] <https://www.raywenderlich.com/51114-firebase-tutorial-for-android-getting-started>
- [5] <https://www.sitepoint.com/creating-a-cloud-backend-for-your-android-app-using-firebase/>
- [6] <https://firebase.google.com/docs/storage/>
- [7] <https://www.youtube.com/tutorials>
- [8] <https://www.tutorialspoint.com/android/>
- [9] <https://developer.android.com/training/>
- [10] <https://codelabs.developers.google.com/codelabs/>