



Project Proposal

Project: Unit Testing on “Pill Reminder” Mobile Application

Course: CSE427

Course Title: Software Testing and Quality Assurance

Section: 01

Semester: Spring 2019

Submitted to: Shaikh Shawon Arefin Shimon

Submitted by: Group 06

1. Noshin Islam 1521733642

2. Asik Azad 1510612042

1. Introduction

Taking medication is occasionally frustrating, but ultimately necessary. Thankfully, smartphones can make the experience a little easier. The mobile application that we have decided to test the functionality off is named “Pill Reminder, an app that can be used to decrease the stress of remembering to take the pills on a predefined scheduled time. Staying healthy and getting your medications exactly when your body needs them is vital, but sometimes you just forget.

2. App Description

Our application helps people to take medicine in due time as it is an integral part of our medication. Taking an untimely dose of medicine could lead to many long term or short term health issues. The “Pill Reminder” app takes a medicine name as input along with a specified time which a patient can select by roving around a digital clock. A visual representation is given below:

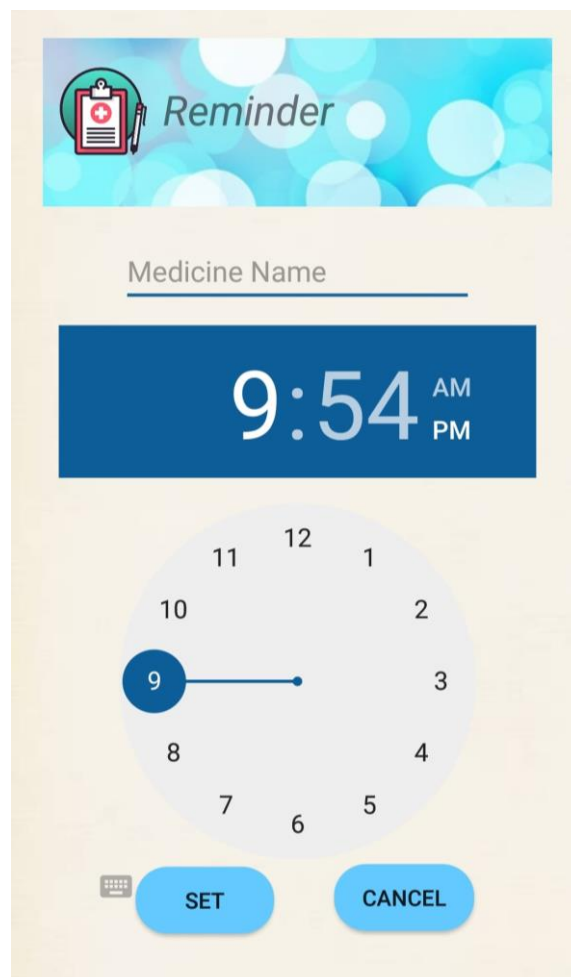


Figure 1 Pill Reminder App

3. Functionality

1. Easy and understandable User Interface

2. No authentication required
3. Add Medicine names and set a specified time to get the alert
4. A pop-up message is shown when it's reminder time

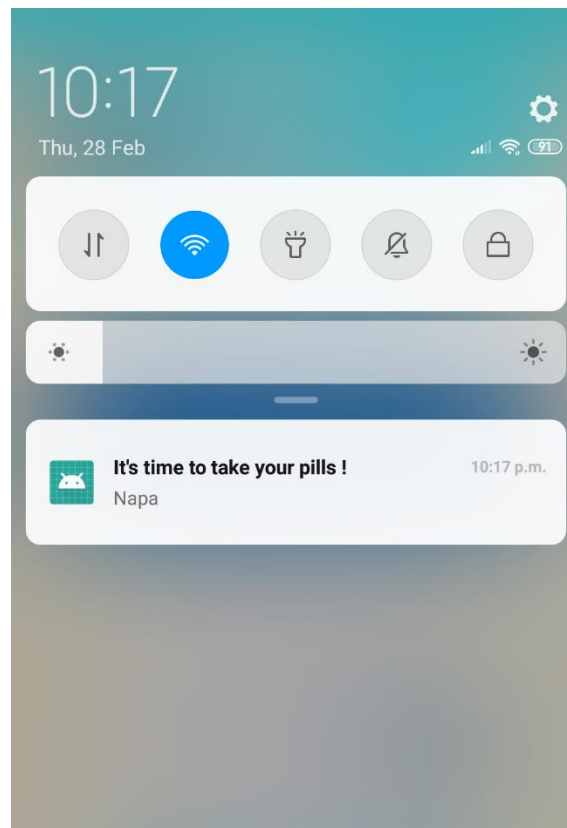


Figure 2 Pop-up message

5. Patients can add another medicine and set up a reminder

4. Software Unit Testing:

TDD (Test Driven Development) is an effective way of developing the system by incrementally adding the code and writing tests. Unit tests can be written using the Mockito framework

Mocking is a way of producing dummy objects, operations, and results as if they were real scenarios. This means that it deals with no real database connections and no real server up and running. However, it mimics them so that the lines are also covered and expect the actual result. Thus, it can be compared with the expected result and asserted.

Mockito is a framework that facilitates mocking in the tests. Mockito objects are kind of proxy objects that work on operations, servers, and database connections. We would be using dummy, fake, and stub wherever applicable for mocking.

We will use JUnit with the Mockito framework for our unit tests

5. Features:

We are planning on adding more features such as:

1. At this moment, the app only excepts one medicine reminder at a time, we will try to implement multiple medicine reminder
2. We will try to implement a registration panel and an account

6. Time Distribution:

10 th February- 21 st February	Learning the fundamentals of JUnit
21 st February-28 th February	Study about the mockito framework
1 st March-5 th March	Start implementing the test cases for the existing features
6 th March	Meeting with our faculty to review the test cases
6 th March-9 th March	Try to add new features
10 th March- 12 th March	Complete unit testing of all the features
13 th March-14 th March	Complete the documentation and submission

7. Software requirements:

1. Language: JAVA
2. App design and Implementation IDE: Android Studio
3. Testing: Junit 5
4. Framework: Mockito

8. Deliverables:

1. Full project with Testing codes uploaded in GitHub
2. Documentation