UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA
Faculty of Engineering, Built Environment and Information Technology

EPI-USE® LABS
value through innovation

COS 301 - SOFTWARE ENGINEERING

HOME SECURITY SYSTEM (ARGUS)

TESTING POLICIES

Initials and Surname	Student Number
B.A. Zietsman	u15228194
B.R. Fourie	u16024002
J.K. Lutz	u15323413
J.S. Stadler	u16008554
S. Vermeulen	u16078625
B. Chu (Deregistered)	u18176951

September 17, 2020

Contents

Introduction

To ensure we deliver a product of the highest possible quality we put a great emphasis on testing each and every part of our system separately and together. To ensure correct and thorough testing this document was created.

Testing Process

To ensure testing is done and that it is done in a proper fashion, effectively and efficiently we have automated all of our unit testing and integration. The goal of our testing process is to detect any faulty code, classes and interfaces to help reduce the amount of failures, bugs and errors the user might encounter during usage.

2.1 Spring Testing

Testing our Spring-Boot backend, we implemented 2 types of testing in our system, namely:

2.1.1 Unit Testing

For the unit testing in the Spring Boot backend, we test to see if all of the functions work properly. This test can be ran by using the following command: *"mvn test"*

2.1.2 Integration Testing

For our integration testing, we test various functions working in conjunction with spring. We test the SQL database connection on spring, email connection and SMS connection through specified tests. This test can be run by using the following command: "*mvn test"*

2.2 Web Application

Testing our mobile application is performed with Karma and Jasmine that are packaged along with Angular 9. There is 1 test being run on the web application.

2.2.1 Unit Testing

Unit testing is used to ensure that everything is displayed correctly on the website. This test can be triggered by using the following command: "*ng test"*

Approach to Test Process Improvement

Tests should be reviewed alongside code reviews after a sprint. Tests should be on the same quality standard as the production code released to end-users.

Testing Tools

4.1 Manual Testing

It is recommended that developers run the tests manually before committing code to Github.

4.1.1 Spring

Running each test individually on the spring backend when testing each function.

4.2 Automated Testing

To ensure that tests are performed whenever changes are made we decided to use Github and TravisCI.

4.2.1 Github

Github does more than just managing our code base, it also provides us with an area to report issues missed by tests which were then encountered by users and it then allows us to effectively identify, track and resolve these errors. It also automatically invokes our next tool Travis CI to automatically run tests on each and every commit as well as each pull request.

4.2.2 Travis CI

Travis CI is our main testing tool. There is a testing located in the root directory of our repository. Upon each commit and pull request this testing is invoked. The testing creates a small virtual machine and installs all the necessary dependencies to test our code. It then runs each and every test and sends the results to Github as well as our emails.