CNG 491 - Senior Project and Seminar:Design

MIDDLE EAST TECHNICAL UNIVERSITY
NORTHERN CYPRUS CAMPUS

Test Plan Outline

Project
Smart Home Senior Project

Group Members

2041226 – Temirlan BAYESHOV
2016129 – Anıl PEKER

Table of Contents

Table of Figures

1. Introduction

This report contains all the test plan that will be implemented to “Smart Home Prototype Senior Project”. The plan that we consider consists testing strategies, test items, features to be tested & not to be tested, criteria that is about pass/fail, test deliverables and test schedules.

2. Testing Strategy

We decided to use unit test, integration test, smoke testing, validation testing, stress testing, system testing.

Software Part:

- In that project unit test must be tested because there are many function which is written with different software language.
- Project is containing many mobile application and backend process because of that, in these part of this project, we will test that with system and integration testing.
- Stress testing will be used for the backend and servers.
- Smoke testing is related with Smart Home since whole system should be tested when new components are added.
- Validation testing will give information about Smart Home Prototype Senior Project are satisfied or not.

Hardware Part:

- Smoke testing will be used when new components (sensors) connected to the Arduino.
- Also we will test compatibility of all of our sensors in the system.
- We will check functional and non-functional requirements for sensors and modules.

3. Test Items (Functions)

This section is giving information about test items that will be generate on the project.

1. Unit Test
2. System and Integration Testing
3. Smoke Testing
4. Stress and Performance Testing
5. Alpha/Beta Testing
6. Security Testing
7. Validation Testing

3.1. Unit Testing

- Participants: Temirlan Bayeshov, Anil Peker
- The Units: Functions which will compute statistical information correctly in R, functions that will send and retrieve data transaction in android, functions in backend about server abilities.
- Methodology: In R testthat library package and scripts will be used [1][2].After adding new codes we will be able to automatically re-run our testthat. Also it provides functions that will catch errors. Android will use JUNIT testing method, test folder will be created. PHPUnit framework will be used for server testing. We will mock our PHPUnit and JUNIT code using mockito framework.

3.2. System and Integration Testing

- Participants: Temirlan Bayeshov
- The Units: Data integration between database and Web server will tested.
- Methodology: Here participant start with bottom level units going to up, so bottom up strategy will be used. Before starting Integration Testing, Unit test should be successfully completed. We will check, if out data sets successfully arrived or delivered to Web Server from Database.

3.3. Smoke Testing

- Participants: Temirlan Bayeshov, Anil Peker
- The Units: Arduino Studio and Hardware.
- Methodology: QA team participants will manually test code written in Arduino Studio to verify work with sensors.

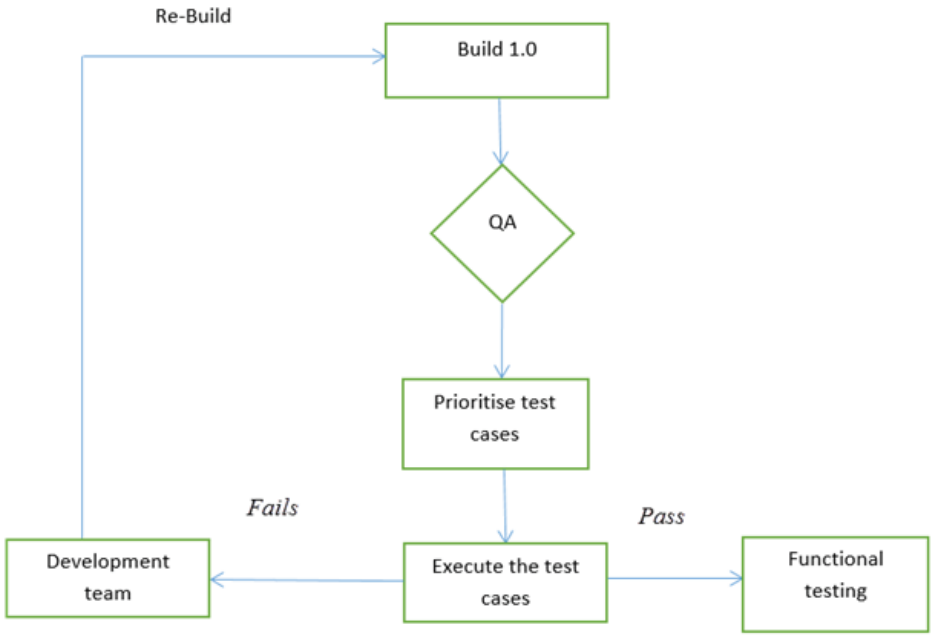

Figure 1: Smoke Testing Flowchart [4]

3.4. Performance and Stress Testing

- Participants: Temirlan Bayeshov, Anil Peker
- The Units: Web and Mobile applications, Arduino.
- Methodology: Stress testing automation scripts will be created. NeoLoad [3] tool will be generate huge amount of users to evaluate and analyze web/mobile application performance under stress condition. Response time will be tested with help of time stamps in Arduino requests to the server.

3.5. Alpha/Beta Testing

- Participants: Anıl Peker
- The Units: User Interface and Optimization for Android and Web page.
- Methodology: In that testing case, according to survey that will be done, Android and web page will be developed by the participant. For the Android App, Google Developer Console will be used.

3.6. Security Testing

- Participants: Anıl Peker
- The Units: Android Authentication
- Methodology: For authentication, android app will generate token for every login. In that test case, participant will send request with random token and will check security of personal and smart home information reachable or not.

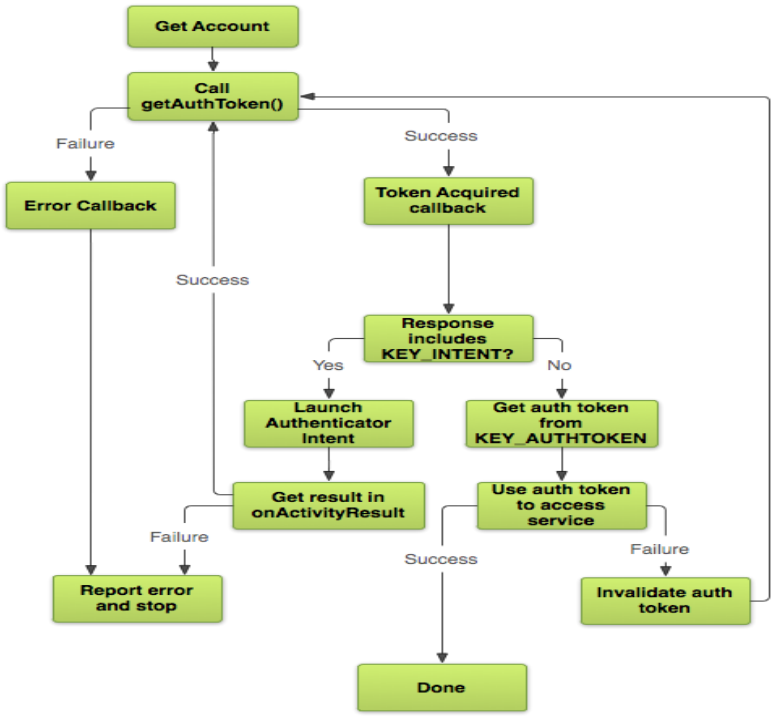

Figure 2: Token Generator Algorithm[5]

3.7. Validation Testing

- Participants: Anıl Peker, Temirlan Bayeshov
- The Units: Unit Test Result, Integration Test Result, System Test Result, User Acceptance Testing Result, functional and non-functional requirements.
- Methodology: According to test case result, participants will validate the project using V-Model. In addition we will check if our system validates with requirements.

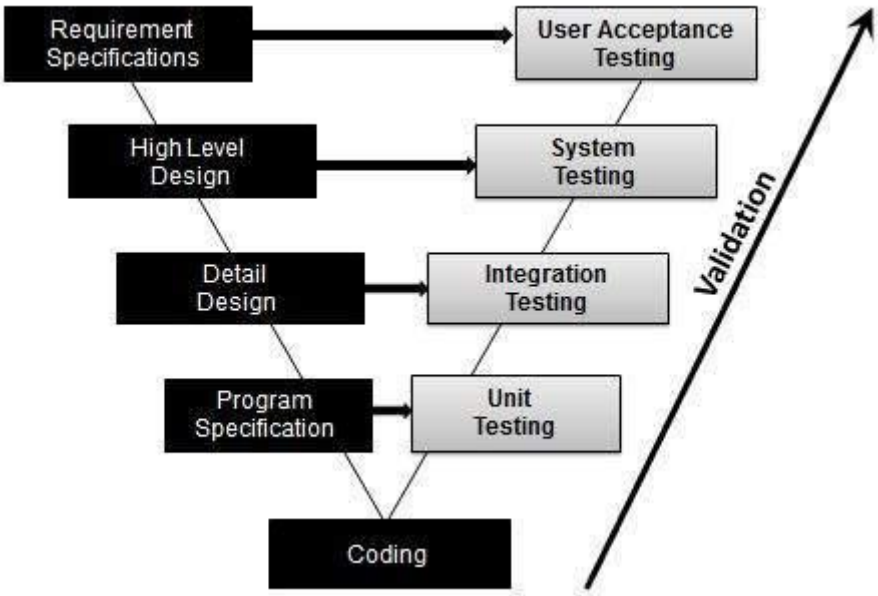

Figure 3: V-Model [6]

4. Summary of Features to Be Tested

- R statistical information in web server databases
- Database and Web Server integration
- Stress of web and mobile applications
- Response time of requests from servers and Arduino
- Sensors work with Arduino codes
- Android and Web pages screens
- Login credentials and tokens generation for each user

5. Summary of Features Not to Be Tested

- Collection of data in sensors
- Database connection
- Internet connection of Arduino

6. Item Pass/Fail Criteria

- **Unit testing:** All test cases should be completed as expected. Unexpected results will lead to fail unit tests. All database sets should be send and retrieved without any minor defects, because statistics should be computed in exact way from RStudio.

- **System and integration testing:** All modules and units should work together in proper way. All lower level plans such as Unit Testing should be completed without any defects. Data integration should be completed without any minor defects, otherwise statically computation will fail.

- **Performance and stress testing**: After performing Unit and Integration tests, Arduino hardware and web/mobile application will be tested. With help of NeoLoad tools stress test will be performed to find boundaries of how many users will be allow to enter server. Failure will be found depends on users' traffic. Arduino response time performance test should be completed without any defects, in case of long response time test will be failed.

- **Smoke testing:** Each sensor of Arduino should work independently of each other. In case of failure of one sensor minor defects will be found. Specific sensor test will be failed. If all sensors will work according the code in Arduino Studio test will be passed.

- **Security testing:** After performing performance and stress testing, security testing should be completed as expected without any failures or minor defects. In case of generating error token, test will fail. Each of verification data should be matched. Personal data should be reachable for each user. Otherwise security testing will fail.

7. Test Deliverables

Deliverable Name	**Author(s)**
Test plan	Anıl Peker
Test cases	Temirlan Bayeshov, Anıl Peker
Error Logs	Temirlan Bayeshov
Test closure report	Anıl Peker, Temirlan Bayeshov
Problem reports and corrective actions	Temirlan Bayeshov, Anıl Peker

8. Test Schedule

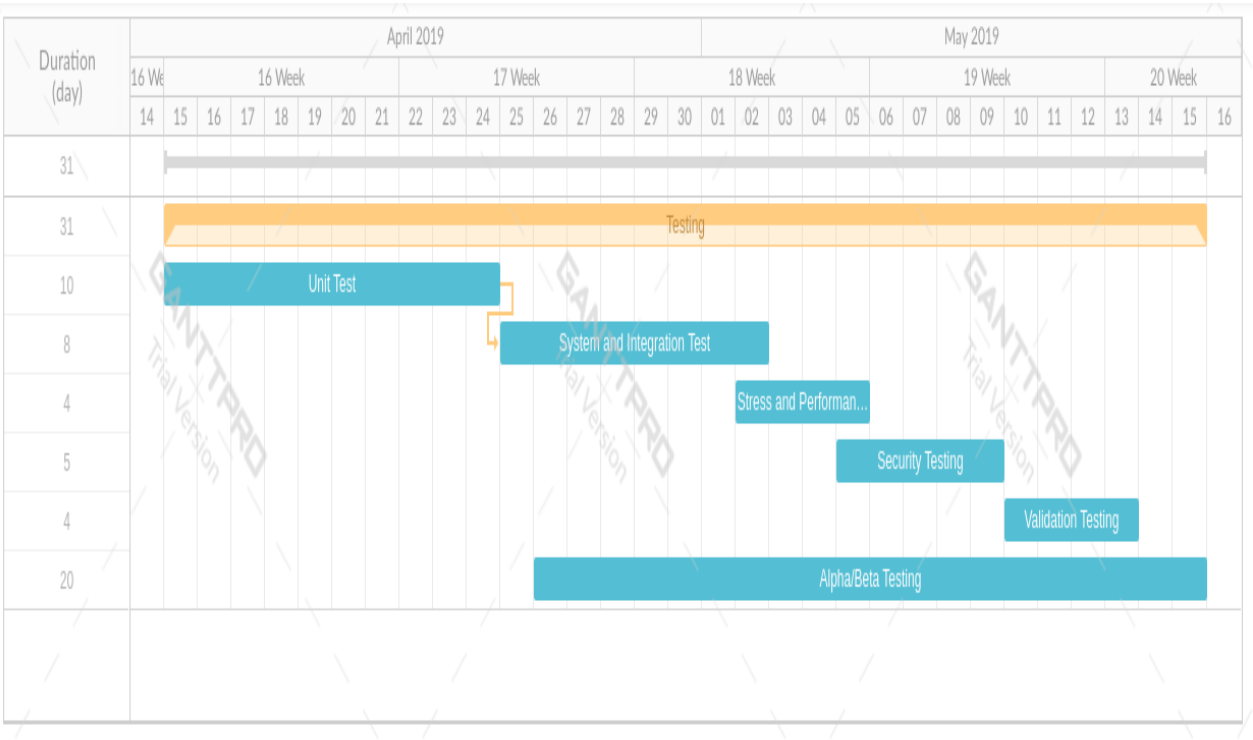

Figure 4: Test Schedule Gantt Chart

9. References

1. Testhat library, https://www.utest.com/tools/testhat
2. R test, https://www.r-bloggers.com/unit-testing-with-r/
3. NeoLoad, https://www.neotys.com/
4. Smoke Testing Flowchart, https://mysullys.com/flow-chart-of-series-test/smoke-testing-learn-examples/
5. Token Generator Algorithm, https://www.researchgate.net/figure/Flowchart-containing-the-steps-of-the-developed-compiler_fig5_236510290
6. V-Model, https://www.tutorialspoint.com/software_testing_dictionary/v_model.htm