SOFTWARE TESTING ANALYSIS

Team No. 21

Team Members : Akash Verma, Vishal Verma, Archit Goyal, Priyanshu Madaan

This document talks of the different levels of testing done on the software project.

UNIT TESTING

Unit testing is a level of software testing where individual units of a software are tested. The purpose is to validate that each unit of the software performs as designed. Here, these tests are conducted by the specific developer who developed the feature, after implementing a particular feature, or a unit. A unit is the smallest testable part of any software, that has a few inputs and usually a single output. The unit usually is a part of the implementation of some feature. Since the feature is quite small in most cases, testing is done manually. It is performed by using the White Box Testing method. The testing of the android app was done in this phase. The parts of Android UI can also be tested under the unit testing phase. Also the individual hardware was tested in R1. We have used Android studios inbuilt APK analyser for flaws in functions and memory leaks.

INTEGRATION TESTING

Integration testing is a level of software testing where individual units are combined and tested as a group. The purpose of this level of testing is to expose faults in the interaction between integrated units. These are conducted by the testers (most probably the test designers). The different approaches here are: • Big Bang : all or most of the units are combined together and tested at one go. • Top Down : top-level units are tested first and lower level units are tested step by step after that. • Bottom Up : bottom level units are tested first and upper-level units step by step after that. • Sandwich/Hybrid : a combination of Top Down and Bottom Up approaches. Here, we used the big bang approach. Here the integration tests are done once a complete use case is completed, or a checkpoint has been reached. Here, all the units corresponding to the test case are combined together and tested at a go. We have done it manually . The integration of Server with the android app can be tested. The Hardware integration could not be performed due to inability to access the sensors and IOT devices.

SYSTEM TESTING

System testing is a level of software testing where a complete and integrated software is tested. The purpose of this test is to evaluate the system’s compliance with the specified requirements. Normally individual testers perform this system testing. Here, this testing is conducted by all the team members after the integration and before his release. Here we use Black Box Testing method. We test each use case as a black box (end-end scenario), to ensure that the software works in all the intended target systems.

ACCEPTANCE TEST

Acceptance testing is a level of software testing where a system is tested for acceptability. The purpose of this test is to evaluate the system’s compliance with the business requirements and assess whether it is acceptable for delivery. This is performed at multiple levels:

• Internal Acceptance Testing: Internal acceptance testing was done by our mentor from client side Simran Singhal. She reviewed the Apps UI and flows.The method used for testing was Black Box Testing as the internal working of the app was not known to the tester.

• External Acceptance testing: This was done by our client Mr. Vishal Garg. He is also a faculty at IIIT-H and one of the users who will be using this system. According to him UI was good and flow was as expected. The App can not be tested by other potential users as the System is yet not fully integrated due to accessibility issues. Here, we use Black Box Testing method. Here, the internal structure/design/implementation of the item being tested is not known to the tester.

The steps followed here are as follows:

• First we make an acceptance test plan

• Then, we list down all the test cases, and the baseline when it would get accepted

• Then we perform the acceptance test.

For example, while testing for the use case of controlling lights, we followed the scene selection procedure on the App and after it was done we queried the backend for the current status of lights and matched if the new state was a valid and app generated state.