

Learning Outcome 4 Supporting Document

Gaps and Omissions in the Testing Process:

- Lack of Respeck data - One of the major limitations in testing the HARTy system, is the lack of adequate Respeck data. When thoroughly testing the HAR, daily activity breakdown, and the step counting algorithms, we need a large amount of data obtained from the Respeck sensor. A lack of data, can result in accurate results during testing. This can lead us to make incorrect conclusions about the functionality of the HARTy system we are testing. To solve this, we need to collect more data from the Respeck sensor to use for testing purposes. The data should be collected from a large number of people, to remove the influence of individual body characteristics on the data. Collecting this data would ensure that the testing is carried out with a larger and more diverse set of data, which will lead to more accurate results during testing.
- Absence of Tools - Another limitation testing the HARTy system, is the absence of specialist tools. An example of this limitation is that the Virtual Device Emulator within Android Studio does not support Bluetooth functionality. This means that within the HARTy system all the different features with Bluetooth functionality cannot be tested through the emulator and must instead be tested on physical Android devices. Due to this, testing the Bluetooth functionality cannot be automated and must be done manually. This makes it time consuming, labour intensive, inconsistent and error prone. To solve this we could try to obtain tools that support us in testing functionality in an emulated environment. This would help make the testing process efficient, and more accurate. Another example of this limitation is that, is the limited access to the Respeck sensor. This limited me from being able to write more Respeck centric tests or rather more data. To solve this issue, one solution would be to invest in additional Respeck sensors or obtain more time with the existing sensor.
- Limited computational resources - Another limitation testing the HARTy system, is the access to limited computational resources. Testing machine learning algorithms is very computationally expensive and requires a large amount of computational resources. Especially in terms of HARTy, where we are dealing with a vast amount of datasets and using two different HAR algorithms. This limited the number of tests I could run, as an increased amount of time is needed to run each test; therefore reducing the efficiency and accuracy of the testing process. To solve this issue, we could invest in additional computational resources or use cloud-based computing resources (such as Google Colab running a GPU or TPU) to run the testing of the ML algorithms. This will ensure that the testing process of the ML models is more efficient and accurate.
- Limited time budget - Another limitation testing the HARTy system, is the limited time budget allocated towards it. This limits the thoroughness of testing the system, and only allows us to test some parts of the system in great detail. To solve this issue, we could allocate more time towards testing the system.

Target Coverage / Performance Levels:

While testing the HARTy system, it is important to establish target coverage and performance levels for the different testing procedures. Ensuring it will result in the system having a high level of quality and reliability.

For functional testing, the target coverage level should aim to thoroughly test all of the features, functions and requirements of the HARTy system. This includes the user interface, user

authentication, step counting, activity classification, and so on. The goal of this is to make sure that all the features work as intended without any errors / bugs.

For performance testing, the target coverage levels should focus on the measurable attribute of the system, such as the speed, scalability, reliability and availability. An example of this, could be that the HARty system should respond quickly and smoothly to user input, even when dealing with a large amount of data.

While setting these target levels, it is important to consider the requirements and constraints of the system. In the case of HARty, the choice of using a minimum Android OS of 11, will impact the performance, scalability, and compatibility of the HARty system. Therefore, constraints like this should be taken into account. It is also important to consider that quality measurements are more challenging to test for than functional requirements, as it is hard to accurately measure attributes such as speed, and response time.

Using this information above, I managed to identify the target coverage and performance levels for the test plans designed in LO2:

Email Address Change Test:

- Target Coverage / Performance Levels
 - Functionality - 100% coverage of all the functional requirements, such as email validation, Firebase authentication update and user notifications.
 - Measurable Attributes - The target performance level should be within an acceptable limit. This includes a quick response time, low memory and CPU utilisation, and accuracy of the email address validation.
- Comparison to Actual Test Case
 - The functionality test coverage was achieved, as our test cases tested all the different parts of the email address change functionality and reached a code coverage of 100%.
 - Our test cases do not currently test for the performance values.
- Steps to achieve Target Levels
 - Modify our test plan to include tests for measuring and testing if the response time, validation accuracy, memory and CPU utilisation of the HARty Android application are below a certain threshold, while the email address is being changed.

Password Change Test:

- Target Coverage / Performance Levels
 - Functionality - 100% coverage of all the functional requirements, such as the password change, Firebase authentication update and user notifications.
 - Measurable Attributes - The target performance level should be within an acceptable limit. This includes a quick response time, low memory and CPU utilisation, and accuracy of the password validation.
- Comparison to Actual Test Case
 - The functionality test coverage was achieved, as our test cases tested all the different parts of the password change functionality and reached a code coverage of 100%.
 - Our test cases do not currently test for the performance values.
- Steps to achieve Target Levels
 - Modify our test plan to include tests for measuring and testing if the response time, validation accuracy, memory and CPU utilisation of the HARty Android application are below a certain threshold, while the password is being changed.

Authentication Test:

- Target Coverage / Performance Levels
 - Functionality - 100% coverage of all the functional requirements, such as signing in and out, and making Firebase authentication calls.
 - Measurable Attributes - The target performance level should be within an acceptable limit. This includes a quick response time, low memory and CPU utilisation, and accuracy of the email address and password validation.
- Comparison to Actual Test Case
 - The functionality test coverage was achieved, as our test cases tested all the different parts of the authentication functionality and reached a code coverage of 100%.
 - Our test cases do not currently test for the performance values.
- Steps to achieve Target Levels
 - Modify our test plan to include tests for measuring and testing if the response time, validation accuracy, memory and CPU utilisation of the HARty Android application are below a certain threshold, while the user is signing in or out of the Android application.

Account Creation Test:

- Target Coverage / Performance Levels
 - Functionality - 100% coverage of all the functional requirements, such as creating a user account, Firebase authentication update and user notifications.
 - Measurable Attributes - The target performance level should be within an acceptable limit. This includes a quick response time, low memory and CPU utilisation, and accuracy of the email address and password validation.
- Comparison to Actual Test Case
 - The functionality test coverage was achieved, as our test cases tested all the different parts of the account creation functionality and reached a code coverage of 100%.
 - Our test cases do not currently test for the performance values.
- Steps to achieve Target Levels
 - Modify our test plan to include tests for measuring and testing if the response time, validation accuracy, memory and CPU utilisation of the HARty Android application are below a certain threshold, while the user is creating an account through the Android application.

Account Deletion Test:

- Target Coverage / Performance Levels
 - Functionality - 100% coverage of all the functional requirements, such as deleting a user account, Firebase authentication update and user notifications.
 - Measurable Attributes - The target performance level should be within an acceptable limit. This includes a quick response time, low memory and CPU utilisation, and accuracy of the email address and password validation.
- Comparison to Actual Test Case
 - The functionality test coverage was achieved, as our test cases tested all the different parts of the account deletion functionality and reached a code coverage of 100%.
 - Our test cases do not currently test for the performance values.
- Steps to achieve Target Levels
 - Modify our test plan to include tests for measuring and testing if the response time, validation accuracy, memory and CPU utilisation of the HARty Android application are below a certain threshold, while the user is deleting their account through the Android application.

Conclusion:

In conclusion, we evaluated the Target Coverage and Performance Levels for some of the test plans implemented in LO2. It's important to note that determining the threshold for testing the performance values above is subjective. The author who is implementing the tests should decide an appropriate threshold, which allows the HARty Android application to meet the specified requirements.

Each evaluation of the test plans above has been treated independently, so that it can reflect the real-world testing where multiple testers handle different requirements. To provide as much detail as possible, some of the information has been repeated in multiple evaluations.