

National University of Computer and Emerging Sciences



UI Automation

Kanwar Taimoor 16L-4050

Rimsha Chaudhary 16L-4021

Fatima Mahnoor 15L-4372

Noor Fatima 16L-4028

Minahil Ahmed 16L-4150

Supervisor:Jawad Cheema

Software Testing (B)

Final Semester Project

December, 2019

Table of Contents

[UI Automation Testing 3](#_Toc26740916)

[Manual Testing vs Automated Testing 3](#_Toc26740917)

[Why Automated Testing 3](#_Toc26740918)

[Why Appium 4](#_Toc26740919)

[TODO Mobile Application 5](#_Toc26740920)

[What is this application about? 5](#_Toc26740921)

[What are the functionalities? 5](#_Toc26740922)

[Interface of the application 6](#_Toc26740923)

[Test cases based on the functionalities of the application 6](#_Toc26740924)

# UI Automation Testing

UI automation testing, is similar to manual testing, but instead of having a user click through your application, and visually verify the data. You create a script for each test case that you want to test.

## Manual Testing vs Automated Testing

|  |  |
| --- | --- |
| Manual Testing | Automated Testing |
| A typical test plan a manual tester might have could have easily have anywhere from 20 to 100 tests per screen. Taking them up to a few hours to days to thoroughly test all of those test cases. | Have a set of testers open up the application and try to methodically break the application. Which will work, but the larger your application gets, and the more changes that are made the more testing is required. |
| With every change in the application it can trigger the need to retest large portions of your application again and again. That can be a costly amount of man hours, which often leads people to test less often then they probably should. | The time saved in this testing, is more than just economical in the cost of man hours. It often means you can have the automated scripts re-run more often. Instead of waiting until right before release to re-test your application to see if everything runs smoothly. |

# Why Automated Testing

Automated Testing has a lot of benefits but the most important one and the reason why our team chose this is because there is a big window that is open in this field since people who can code opt to become developers and not pursue automated testing and on the other hand most of the QA people have not yet developed automated skills thus this part of Testing is what caught our attention. Other than this some more benefits of Automated Testing are given below,

* This improves communication among coders, designers, and Product Owners, and allows potential glitches to be immediately rectified.
* Owing to the quick implementation of automated testing, plenty of time is saved even for intricate and enormous systems. This allows for the testing to be carried out repeatedly, delivering faster results each time with lesser effort and time.
* Automated test cases are reusable and can hence be utilized through different approaches.
* The documentation of software defects becomes considerably easier for the testing teams. This helps increase the overall development speed while ensuring correct functionality across areas.
* It comes as no surprise that while the initial investment may be on the higher side, automated testing saves companies many a penny.
* Most of the testers have no or a little coding background as the automated testing requires writing test cases to test an application so testers tends to repel from this testing which gives the new coming testers more opportunities in this department.

# Why Appium

Appium is an open-source tool for automating native, mobile web, and hybrid applications on iOS mobile, Android mobile, and Windows desktop platforms. **Native apps** are those written using the iOS, Android, or Windows SDKs. Since we are testing an Android application thus, we chose Appium as our platform for automated testing.

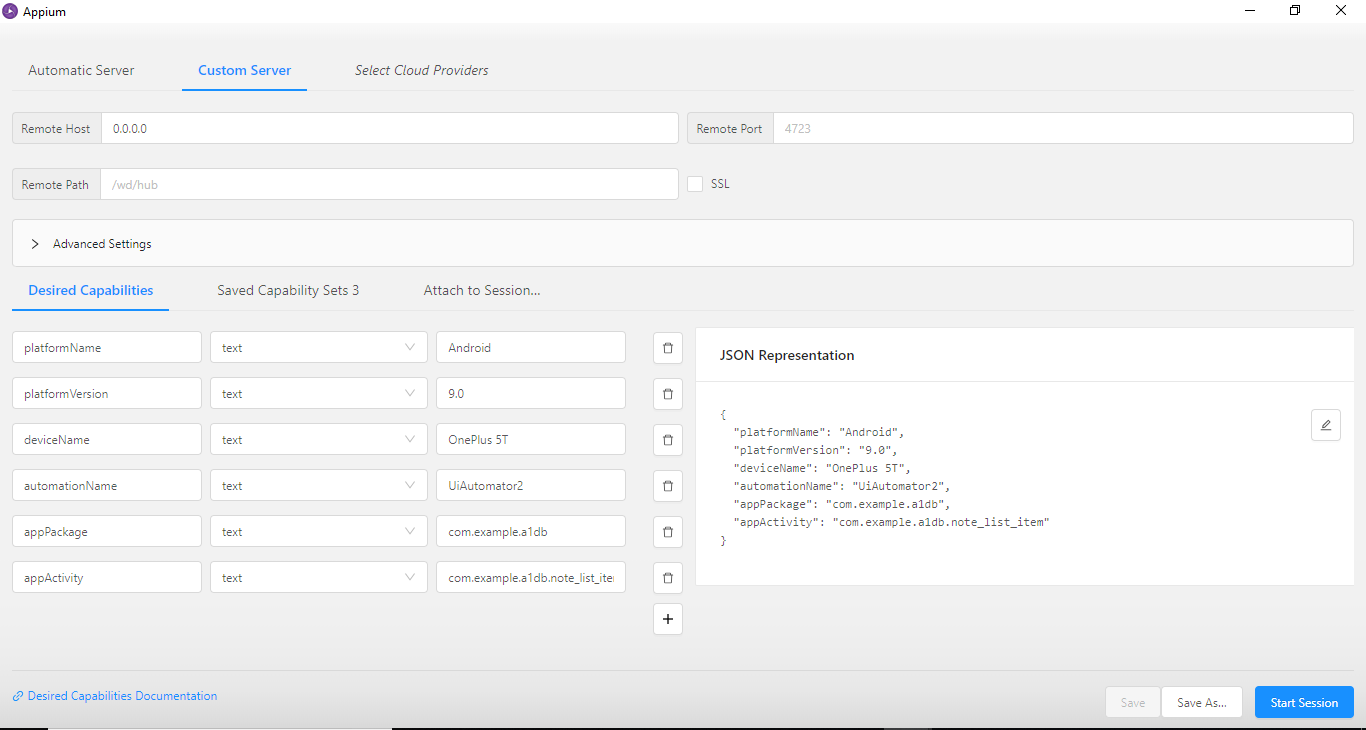
****

Figure 1 Setting up

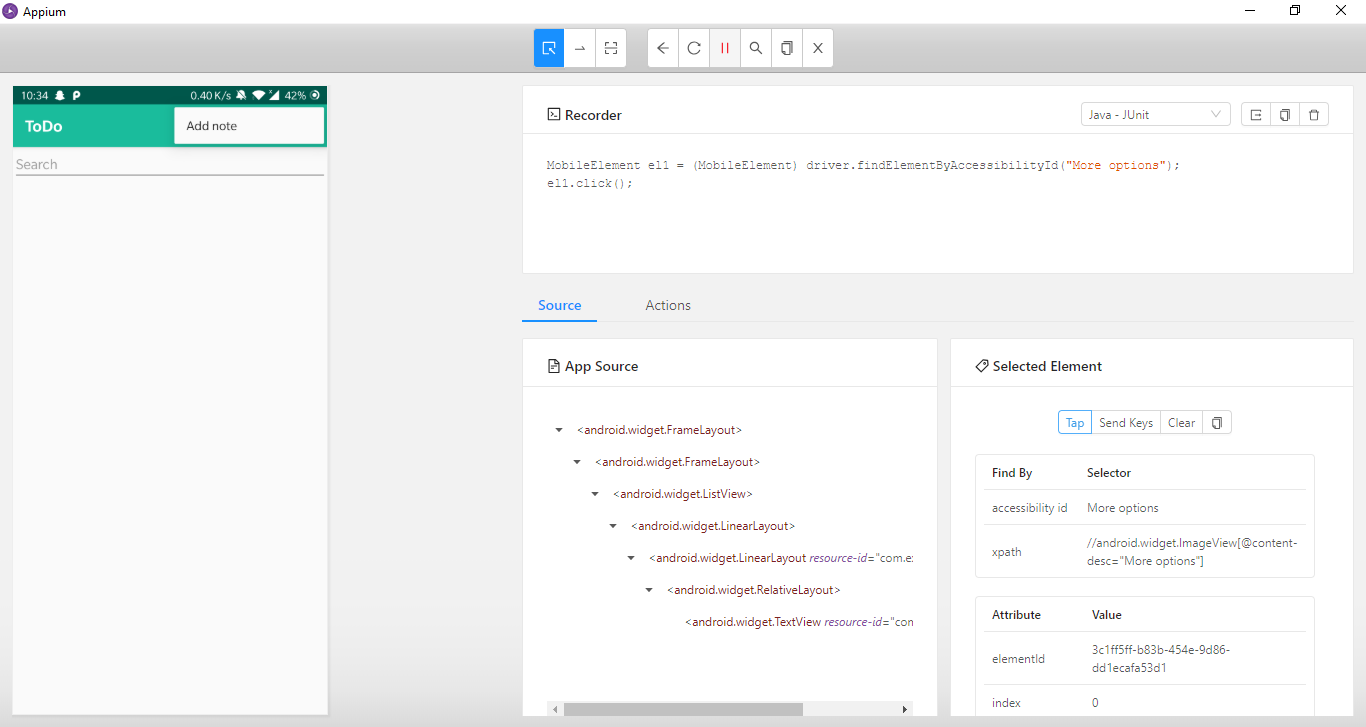
****

Figure 2 UI Automation

# TODO Mobile Application

## What is this application about?

Todo is a note saving application. The purpose of this application is to allow the user to make new notes of the tasks they have to do in future. They assign priority to a note relative to the importance of each task and assign them an expected task completion date and time. The colors of each displayed note indicate the priority of the task to the user. The three priorities have different display colors.

## What are the functionalities?

1. To save a new note with desired title, description, date, time, priority and even customized priority color
2. View the list of notes with priority colors, title, description, time and date in display
3. Search from a title any desired note
4. Update information i.e. title, date, time, description, priority of any note
5. To delete any desired note

## Interface of the application

|  |  |  |
| --- | --- | --- |
| Figure 3 Display Notes Page | Figure 4 Add new note page | Figure 5 Update or delete note page |

## Test cases based on the functionalities of the application

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Id** | 001 | **Test description** | Save new note functionality with all valid information |
| **Prerequisites** | - | | |
| **Testing Data** | **Title**= “test 1” **Description**= “test desc 1”  **Priority= “**high”, Date= “8th December 2019”, Time= “2 pm”, | | |
| **Test Scenario** | Verify on entering all the information and valid date and time, the note is saved | | |
| **Step Details** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| Click menu | Menu should open | As per expected | Pass |
| Click Add new note option | Add new note page should open | As per expected | Pass |
| Fill all the fields i.e. title, description, priority, data, time and click save | A new note should be saved | As per expected | Pass |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test ID** | 002 | **Test description** | Save new note functionality without title |
| **Prerequisites** | - | | |
| **Testing Data** | **Title**= “ ” **Description**= “test desc 2”  **Priority= “**high”, Date= “8th December 2019”, Time= “2 pm”, | | |
| **Test Scenario** | Verify on entering invalid information (leaving title field empty), the note is not saved | | |
| **Step Details** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| Click menu | Menu should open | As per expected | Pass |
| Click Add new note option | Add new note page should open | As per expected | Pass |
| Fill the fields i.e. description, priority, data, time and click save except title of note | An error message should pop and ask user to add title of the note | As per expected | Pass |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test ID** | 003 | **Test description** | Save new note functionality with space instead of title name |
| **Prerequisites** | - | | |
| **Testing Data** | **Title**= “ ” **Description**= “test desc 3”  **Priority= “**high”, Date= “8th December 2019”, Time= “2 pm”, | | |
| **Test Scenario** | Verify on entering invalid information (Adding space on title field only), the note is not saved | | |
| **Step Details** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| Click menu | Menu should open | As per expected | Pass |
| Click Add new note option | Add new note page should open | As per expected | Pass |
| Fill the fields i.e. description, priority, data, title, time and click save | An error message should pop and ask user to add a valid title of the note | Note saved, with spaces only in title | Failed |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test ID** | 004 | **Test description** | Save new note with invalid date functionality |
| **Prerequisites** | - | | |
| **Testing Data** | **Title**= “Test 4” **Description**= “test desc 4”  **Priority= “**high”, Date= “8th December 2007”, Time= “2 pm”, | | |
| **Test Scenario** | Verify on entering invalid date (past date), the note is not saved | | |
| **Step Details** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| Click menu | Menu should open | As per expected | Pass |
| Click Add new note option | Add new note page should open | As per expected | Pass |
| Fill the fields i.e. description, priority, time and date click save | An error message should pop and ask user to add a valid date | Note saved | Failed |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test ID** | 005 | **Test description** | Search note functionality |
| **Prerequisites** | App must be logged in | | |
| **Testing Data** | Text entered in search bar= “Test” | | |
| **Test Scenario** | Verify on entering valid note name that exists in saved notes | | |
| **Step Details** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| 1. Navigate to Homepage | Homepage containing all existing notes should be displayed | As per expected | Pass |
| 1. Click on search bar | Cursor should appear on search bar | As per expected | Pass |
| 1. Enter keyword | Titles starting with the letters entered must start displaying until Title name matches keyword entered | As per expected | Pass |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test ID** | 006 | **Test description** | Search note functionality |
| **Prerequisites** | App must be logged in | | |
| **Testing Data** | Text entered in search bar= “Excellent” | | |
| **Test Scenario** | Verify no notes displayed on entering invalid note name that does not exist in saved notes | | |
| **Step Details** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| 1. Navigate to Homepage | Homepage containing all existing notes should be displayed | As per expected | Pass |
| 1. Click on search bar | Cursor should appear on search bar | As per expected | Pass |
| 1. Enter keyword | Titles starting with the letters entered must start displaying until Title name no longer matches keyword entered | As per expected | Pass |
|  | After complete word is entered no results are displayed | As per expected | Pass |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test ID** | 007 | **Test description** | Search note functionality |
| **Pre-requisites** | App must be logged in | | |
| **Testing Data** | Text entered in search bar= “” (space) | | |
| **Test Scenario** | Verify on entering ambiguous note name that exists in saved notes | | |
| **Step Details** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| 1. Navigate to Homepage | Homepage containing all existing notes should be displayed | As per expected | Pass |
| 1. Click on search bar | Cursor should appear on search bar | As per expected | **Pass** |
| 1. Enter keyword | Note titles starting with a space must start displaying until Title name no longer matches keyword entered | As per expected | Pass |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test ID** | 008 | **Test description** | Update note functionality |
| **Pre-requisites** | App must be logged in | | |
| **Testing Data** | - | | |
| **Test Scenario** | Verify proper functioning of Update Note functionality | | |
| **Step Details** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| 1. Navigate to Homepage | Homepage containing all existing notes should be displayed | As per expected | Pass |
| 1. Click on one existing note | Update button should be active on page | As per expected | Pass |
| 1. Make changes in note | Changes must be displayed | As per expected | Pass |
| 1. Click update button | Changes must be saved in note | As per expected | Pass |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test ID** | 009 | **Test description** | Update note functionality |
| **Pre-requisites** | App must be logged in | | |
| **Testing Data** | - | | |
| **Test Scenario** | Verify proper functioning of Update Note functionality | | |
| **Step Details** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| 1. Navigate to Homepage | Homepage containing all existing notes should be displayed | As per expected | Pass |
| 1. Click on one existing note | Update button should be active on page | As per expected | Pass |
| 1. Make no changes in note | Note remains visibly unchanged | As per expected | Pass |
| 1. Click update button | Button must be clickable | As per expected | Pass |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test ID** | 010 | **Test description** | Delete note functionality |
| **Pre-requisites** | App must be logged in | | |
| **Testing Data** | - | | |
| **Test Scenario** | Verify proper functioning of Delete Note functionality | | |
| **Step Details** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| 1. Navigate to Homepage | Homepage containing all existing notes should be displayed | As per expected | Pass |
| 1. Click on one existing note | Delete button should be active on page | As per expected | Pass |
| 1. Click on Delete button | Note must be deleted | As per expected | Pass |
|  | Note should no longer be displayed in existing notes | As per expected | Pass |

# Code

The code for UI automation is attached with the document in GitHub repository.