Name: Muhammad Usama

ID: 13935

Course Name: Software verification and validation

Date: 15/4/2020

Testing tool: Ranorex Studio

- **Ranorex Studio:-**

  Ranorex Studio is a GUI test automation framework provided by Ranorex GmbH, a software development company. The framework is used for the testing of desktop, web-based and mobile applications.

- **Pros and Cons of Ranorex Studio:-**

  - Pros :-
    - This is by far a great product that is easy, cost effective and a winner in my book, the only thing is that my team was not up to par to utilize it for the simple tasks I had originally looked at.
    - As I described above - Ranorex is highly useful for creating and executing stable mobile tests on our devices (Android, iOS). In future we plan to use this tool for web automation, too.

- It seems as if everything a junior quality assurance enginner would need were already there. For anything that was not easy to find out how it worked, the support team was of great assistance.
- I am super happy to be a user of this tool, as I mentioned it is life changes for me to learn this tool and make my job in a better shape and love to work in.

➢ Cons :-
  - At times I had a hard time identifying why a step failed.
  - Not Mac compatible so it has to be ran on a VM. The performance is slow while recording which makes the data entry a little difficult and requires the steps to be modified.
  - It runs very poorly with our in-house editor. We get WPF errors often that force us to restart our primary tool.
  - The software is finicky and difficult to use and understand; it can sometimes be frustrating.

- **Functionalities of Ranorex Studio**:-
  - Ranorex is Cross platform testing. Once the test cases are written, they can be executed on different mobile devices like Android and iPhone.
  - It offers user interface object recognition model that is reliable and is totally dependent on Ranorex XPath technology.
  - It provides a toolset that allows the user to automate the UI testing by recording the UI actions without writing any code.

- It offers a spy engine which tracks the details of each and every user interface element present on the application's screen. Its object recognition feature helps the user to identify the UI elements of the application.
- It directly records the test cases from the real mobile devices. While editing the recorded steps, there is no need to write a single line of code.
- It supports text validation of any text view present on the screen. For this validation, a variable is specified and attached to the test case.
- It allows the Ranorex script to be executed from anywhere in between the commands of a particular test case.
- It supports image-based validation i.e. the images in the application can be compared with an expected image as whole or subset of an image. The image under test can be the screenshot of the application or it can be directly uploaded in the software.

- **Supporting languages of Ranorex Studio**:-
  - Python
  - C#
  - PHP
  - Tcl
  - C++
  - Javascript
  - Perl
  - ASP.net

- Haskell
- ColdFusion Markup Language (CFML)

- **Supporting tests of Ranorex Studio:-**

1) **Desktop Application Testing**

   The broadest technology support, unbeatable object recognition, full IDE, open API, code and codeless options, and cross-technology functionality make Ranorex Studio the only choice for desktop testing.

2) . **Automated Web Testing**

   Accelerate your web application testing with comprehensive tools for test automation

3) **Mobile Testing Tools**

   Use Ranorex Studio's tools to automate mobile and mobile web testing by setting end-points on both real and emulated Android and iOS devices. Incorporate your mobile tests into absolute cross-platform test suites and eliminate complicated context switching.

4) **Cross-Platform/Cross-Technology Testing**

   Work across applications, platforms, devices, and technologies without complicated context switching. Use centralized repositories to collect all your controls in one spot, set endpoints to real mobile devices, and have absolute cross-platform testing.

- **Code :-**

```
struct SimpleRwLock
{
  const int RwWait = 1;
  const int RwWrite = 2;
  const int RwRead = 4;
  ....
  public void EnterReadLock()
  {
    var sw = new SpinWait();
    do
    {
      while ((_rwlock & (RwWrite | RwWait)) > 0)
        sw.SpinOnce();

      if ((Interlocked.Add(ref _rwlock, RwRead)
          & (RwWait | RwWait)) == 0)               // <=
        return;

      Interlocked.Add(ref _rwlock, -RwRead);
    } while (true);
  }
  ....
}
```