1. Due to the issue which I mentioned in the email, I did not complete creating the automation test cases for “Creating new account” and “Forgotten password” features yet.

I implemented some test cases using Selenium Web Driver, TestNG in Java.

In scope in this test, I don’t develop modularization, test driver, repository… These test cases simply were developed as module based type.

But from my point of view, Create new account is not a case should be automated and executed in Production environment.

An automation test will be performed repeatedly many times, so the user information (phone number and email) will be duplicated in the later run. Of course, we can generate these information for each run (a solution will make Production environment messy) or remove created user from system (in cases the application support this function) but we should avoid changing Production data.

The negative cases for this feature are possible

1. 1. Benefit of automation test:

* **Test engineer productivity**: because of its reusability, we spend the effort to create scripts one time, then can reuse them over and over later (enhance the script when needed). . So it is almost used in Regression test.

Automation can run the test parallel in different environments.

Automation helps to save the resource if need to test with large of different data input sets in a same test. It is higher coverage. So it is used in Data Driven test also.

Automation script can run nightly. Save time.

* **Increased test effectiveness**: People become bored and make mistakes when doing the same tasks over and over. Automation test reduces repetitive work which is a tedious to do manually. So it qualifies the human mistake.

It can do the works which manual test cannot: compare 2 large data files, create large users, transactions… So it is used in Performance test, Load test, Stress test and Volume test

* **Reliable**: we can know exactly how many test cases were run with pass/fail status in the report. And also can give the evident about our workload to the other stakeholders (such as client)

There are more benefits of automation test actually. But above is my experience.

2. In my opinion, it is subjective when judging an automation tool is good or bad. The important thing is choose a suitable tool for our needs:

* Budget for the testing: to choose a license tool or outsource tool
* What is your application: web-based, mobile app, window app. Example if your application is Window app, Selenium is excluded. If your web-based uses AngularJS, Quick Test Pro cannot support….
* Size of the system: do you want to perform just an application, or a bigger system? Do you need to integrate to other systems? how you want to organize the report?
* What testing type we intend to automate: functionality test or performance test? There are specific tools for each of them.
* Resource skills: some tools require resources are good at technical, some do not. Also, which language do your resource familiar with??

But technically, I prefer a tool which flexible enough to build many types of automation framework (Module based, Library Architecture, Data Driven, Data keyword, Hybrid), easy to maintain, scale up, upgrade to keep it up to date with new technology, ability to integrate with others system (Build system, Defect management system, Test management system), support multiple languages… Additional, it should support indicate objects