Introduction

Good afternoon everyone

My name is Bill Liu and I am the software IT intern.

I am currently a masters student at NJIT studying computer science

Project

My project this summer is building a test automation system

The purpose is to implement a software system that automatically test our website and report any errors or bugs

First of all

What is test automation

Tests automation is a system of software that controls and execute test suites then report on the outcome.

Often these tests involve simulating moving, clicking and typing certain things on their browser as if the test is being done manually by a real person.

* CI/CD continuous integration and continuous deployment. This is something that has grown in importance tremendously in the past years. Its goal is to produce a seamless software development environment where code that the developers write is pushed to version control, then tested, then released to production all in an automated pipeline
* Continuous integration’s purpose is to allow the developers merge back to the main branch as often as possible so that it can be continuously released to the customers as often as possible.
* To achieve that, Continuous integration heavily focuses on test automation.
  + Includes unit tests integration tests then user acceptance tests

Why test automation

Test automation provide scheduled reassurances that our web pages and their functions are working properly.

Significantly reduce the amount of time the developer team spends on testing by having tests run themselves repeatedly

Along side high traffic pages, our website also has may transactional pages such as donations, and membership gifts. It is important to make sure that they are doing what they are suppose to.

Since our mission as a memorial and museum is so important to so many, we have to be sure our websites are always professional and proper and nothing messes up our look.

And you know how advertisements say “with one click of a button?”

Well our system does all the above with NO click of a button!

* Test automation

Contractors sending in their bids

There were Many possible directions we could have taken build this test system

The first part of my task was to test and select the best available tools that best fit for our needs.

* I was pretty overwhelmed at first. It wasn’t a subject that I had any experience in.
* There are so many tools and services out there available
* At first I was pointed to openQA
* The I found selenium. Selenium automates browsers. It mimics a real user
* It is written in java and has a stand alone server as well as webdriver bindings for different languages.
* Selenium running on local
* Codeception is the php testing framework that uses selenium webdriver to automate browser.
* backstopJs is a nodejs tool that tests website UI by comparing screenshots of webpages
* Jenkins vs travisci one is hosted one requires own server
* Using Jenkins, wasn’t really necessary because we are not in need of a build server. The most powerful of Jenkins functions are the build server.
* Switching to cron. Cron is simpler and easier to use

two approaches: User Acceptance Tests and Visual Regression Tests

And out of many there were two.

Two that fitted our needs and handles with just enough agility to evolve with the department and keep up with our tasks.

This is a model that shows the architecture of the test automation system

one on the right is codeception which runs user acceptance tests

and the left is backstopjs which is does visual regression tests

* Architecture
* From azure with linux vm
* Test reports run on apache2 server
* Codeception and backstop js and their stacks

User Aceptance Tests: Catching Errors

Codeception is a full stack testing framework written in PHP programming language

Can control your web browser to perform certain tasks on websites then check if the tasks were completed.

Generates reports showing any errors in the execution and provides snapshots of where the error occurred

* Codeception executes a series of steps written in php
* it uses webdriver or chromedriver to automate the browser
* can config screen size, mobile etc
* xvfb x virtual frame buffer

User Acceptance Tests Report

This picture shows the report after a successful test run. All green means all good

Codeception Video

This is a video shows what it looks like when the system is controlling the browser while performing these tasks. You can tell that the system is mimicking a real user

These tests are being performed on a consistent basis in the background

* Codeception runs on using css selectors to fill out forms
* Buttons and other elements all have ids
* Popups or other hidden elements that needs to be dealt with use jquery statements
* Code also checks assertions

Visual Regression Tests Report

The second half of our test suite is the visual regression tool called backstopjs

There the purpose is for the program to visually inspect a webpage pixel by pixel for any differences when compared to a reference snapshot

This is our calendar and events page and you can see that the testing system picked up changes to the page as new articles or events are added.

* Backstop takes web urls and configs such as screen size and interactions as input
* Create test screenshots of webpages
* Then compares the screenshots to reference screenshots
* If any changes detected are expected or by design, then approve and the reference screenshots are updated
* Use node and npm for packet manager

Visual Regression Tests Scrubber

This is a fun tool called the scrubber. It lets you drag your mouse back and forth and see the difference between the old page and new page like flipping through a flip book animation.

Visual regression Tests video

Here is a brief video of that fun scrubber tool

You caught a glimpse of the code running in the background but what is really cool is the report

We had an update on the website in July that slightly updated some forms. On the next automated run, My scripts caught the difference and reported to me. Of course, the change you saw are all intentional, but it proves that this system really works.

Automation test report webpage

Finally the test automation system I built spits out reports and posts them automatically on this website.

As you can see we can keep track of every test run we have had so if something goes wrong, you can track exactly when and where it happened.

and once again, NO CLICK OF A BUTTON

* This is the page run on an apache2 server

What I learned this summer/ summary

These are some pictures I took around the office. they represent the great memories working here this summer.

I got valuable industry experience being part of a development team.

This experience I gained working with test automation will be invaluable in my future career.

SPECIAL THANKS

Special thanks to Radha, Rui, Malavika, and Kenny for making this whole project possible!