Automating Desktop Application

Michelle Jariel-Francisco & Tome Bihis

Agenda

I. Problem
II. Solution
III. Prep Work
IV. Jumpstart Desktop Automation
V. Demo
VI. Tips and Tricks
VII. Q & A

I. Problem

Problem(s)

1. The web and mobile applications have been automated but there are issues in some of the fields being mapped to the backend application.

2. The project/product team have to perform redundant regression testing for a desktop application fortnightly due to change requests and bugfixes.

3. The company has lots of desktop applications which needs to be regression tested regularly but budget is a constraint so commercial tool is a big NO for test automation.

Test Automation

Web Application	Mobile Application	Desktop Application

II. Solution

How did we solve some or all of the problems?

"Test smarter, not harder"

-Design Driven Testing (book)

- ❑ Implemented an "end to end" test automation framework using a combination of tools
  - ✓ Web automation tool
  - ✓ Mobile automation tool
  - ✓ **Desktop automation tool**
- ❑ Selected tests which are repeatable and has critical business impact if it fails
- ❑ Reused existing code from Web and Mobile Automation

III. Prep Work

Popular Tools

Open Source	Freeware	Licensed
Winium	AutoIT	TOSCA
Sikuli		TestComplete
		HP UFT

Why Winium ?

- ❑ Selenium-based
- ❑ Free and open source
- ❑ Element locators (xPath) can be identified using tools like *inspect.exe* or *UISpy*
- ❑ Write tests using any language that's compatible with the driver: JAVA, C#, Javascript with Node.js, PHP, Python, Ruby
- ❑ Works with any popular Automation Framework: JUnit, TestNG, NUnit, XUnit, PyUnit
- ❑ Combine Selenium, Appium along with Winium to create an "end to end" framework

IV. Jumpstart Desktop Automation

Pre-requisites

✓Local Admin Rights

✓ Add the ***winium-webdriver*** dependency in your *pom.xml* or download the dependency

✓Download ***Winium.Desktop.Driver.exe***

- https://github.com/2gis/Winium.Desktop/releases

✓ Install Windows Software Development Kit (SDK) to get ***inspect.exe*.**

Steps to setup Winium:

----JAVA----

1. Add reference to `winium-webdriver` in your `pom.xml` file.

   Maven repo: https://mvnrepository.com/artifact/com.github.2gis.winium/winium-webdriver

```
<dependency>
    <groupId>com.github.2gis.winium</groupId>
    <artifactId>winium-webdriver</artifactId>
    <version>0.1.0-1</version>
</dependency>
```

----C#----

1. In NuGet Package Manager, Install `Winium.Elements.Desktop.`

Steps to setup Winium (c

```
ChromeOptions options = new
ChromeOptions();
options.addArguments("start-maximized");
```

----JAVA----

2. Initialize the instance of the specific Desktop Options class

```
DesktopOptions options = new DesktopOptions();
options.setApplicationPath("C:\\Windows\\notepad.exe");
```

----C#----

2. Initialize the instance of the specific Desktop Options class

```
DesktopOptions option;
option = new DesktopOptions();
option.ApplicationPath = "C:\\Windows\\notepad.exe";
```

Steps to setup Winium: (c

```
WebDriver driver;
WebDriver driver = new
ChromeDriver(options);
```

----JAVA----

3. Create an instance of the driver class

```
WiniumDriver driver;
```

```
WiniumDriverService service = new
WiniumDriverService.Builder()
        .usingDriverExecutable("path_to_driver_executable")
          .usingAnyFreePort()
          .withVerbose(true)
          .withSilent(false)
          .buildDesktopService();
```

```
driver = new WiniumDriver(service, options);
```

Steps to setup Winium: (cc

```
WebDriver driver;
WebDriver driver = new
ChromeDriver(options);
```

----C#----

3. Create an instance of the driver class

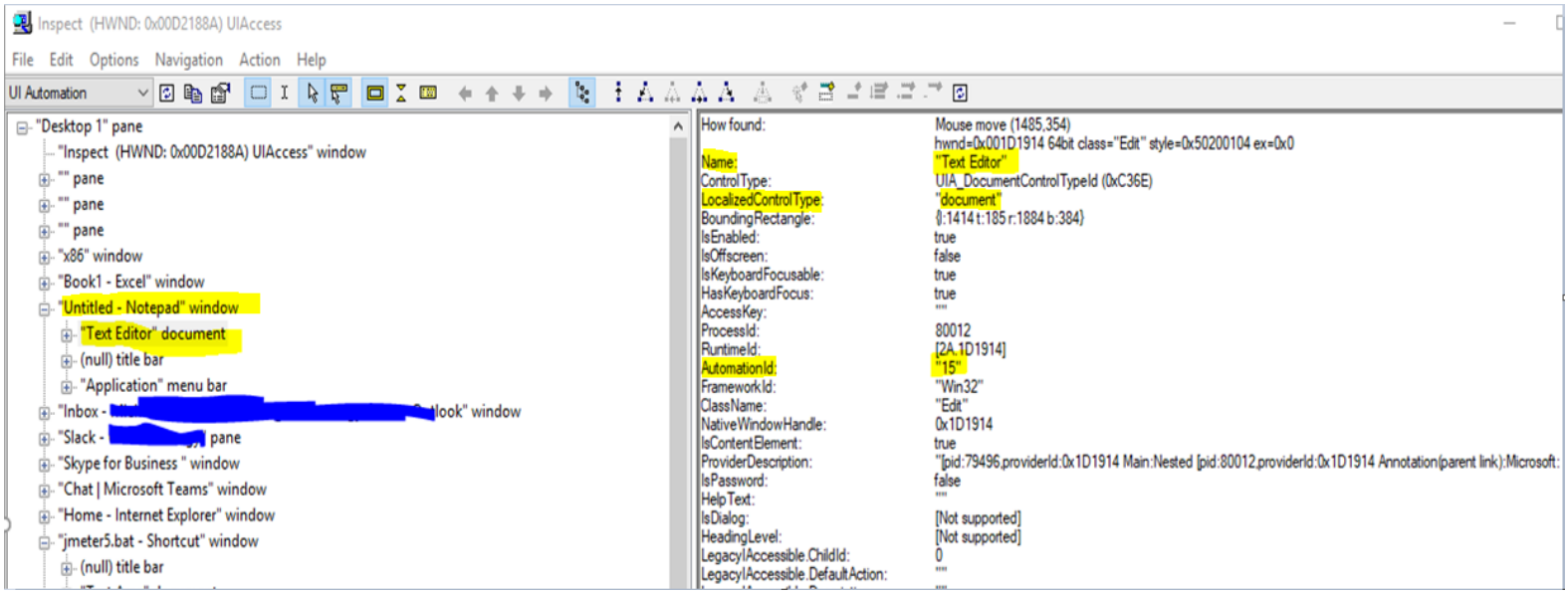
- `WiniumDriver driver = null;`
- `driver = new WiniumDriver("<pathofWinium.Desktop.Driver>", option, TimeSpan.FromSeconds(360));`

Inspect an Element

- ✓Install Windows SDK *(see slide 11: Pre-requisites)*
  - `C:\Program Files (x86)\Windows Kits\10\bin\10.0.17763.0\x86\inspect.exe`

- ✓Define an element location using: **ID, xPath, and name**

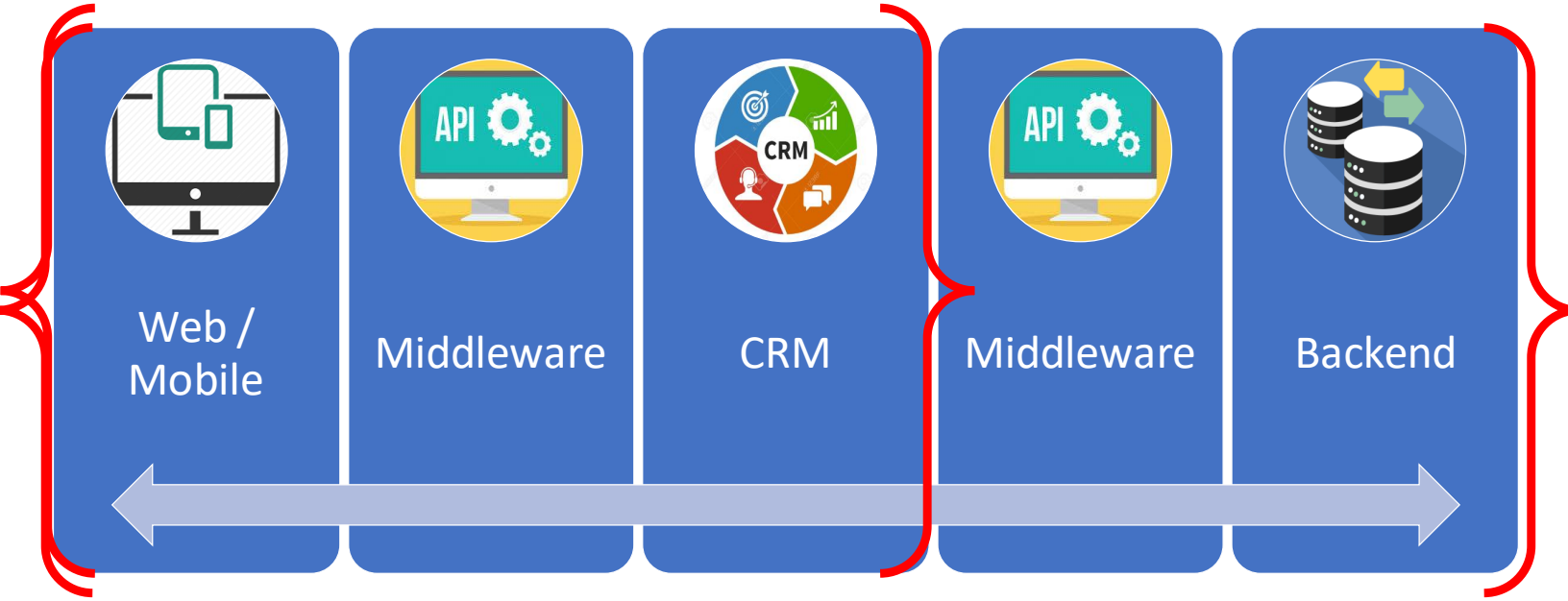
Inspect an Element (continued)

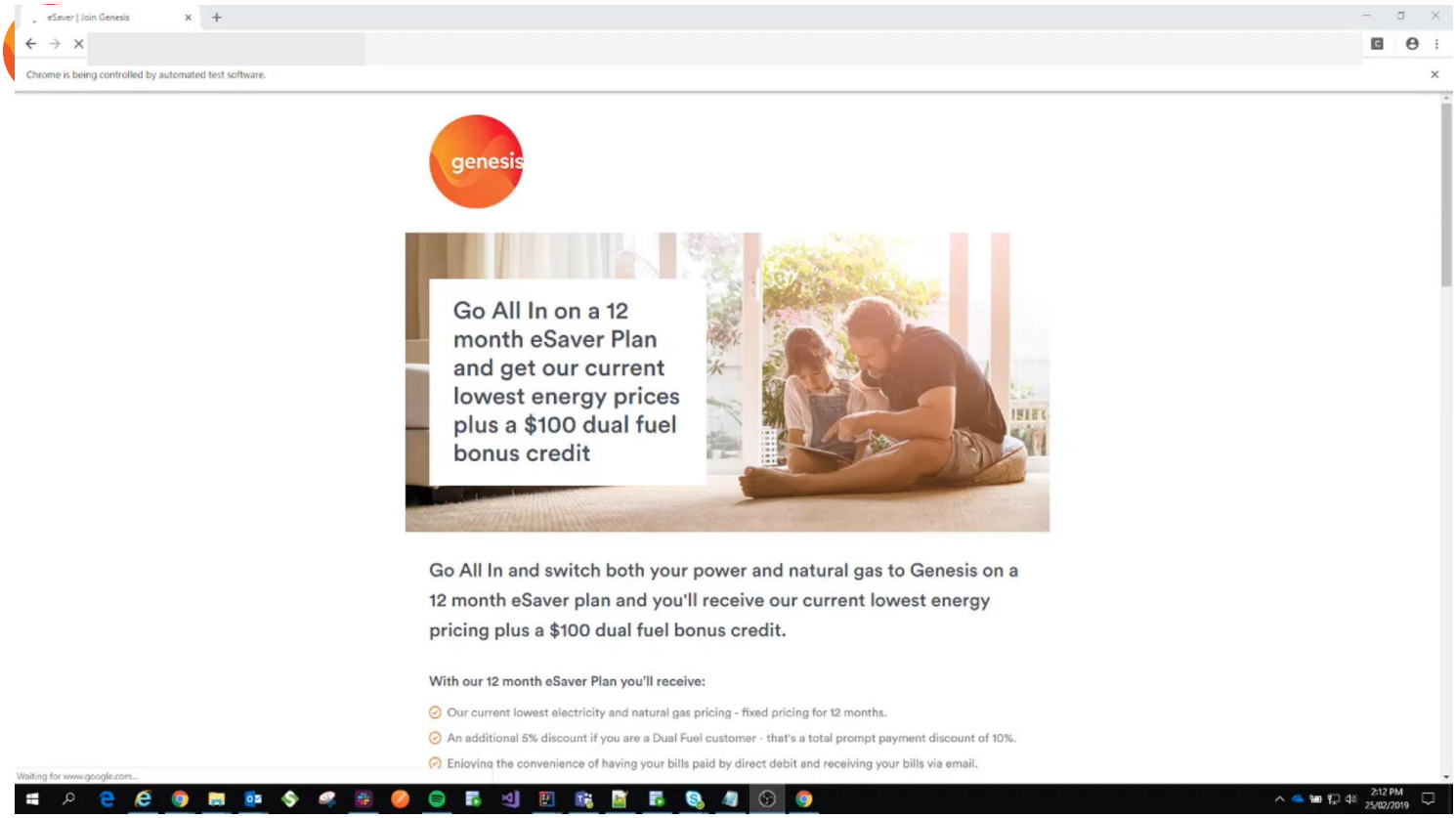


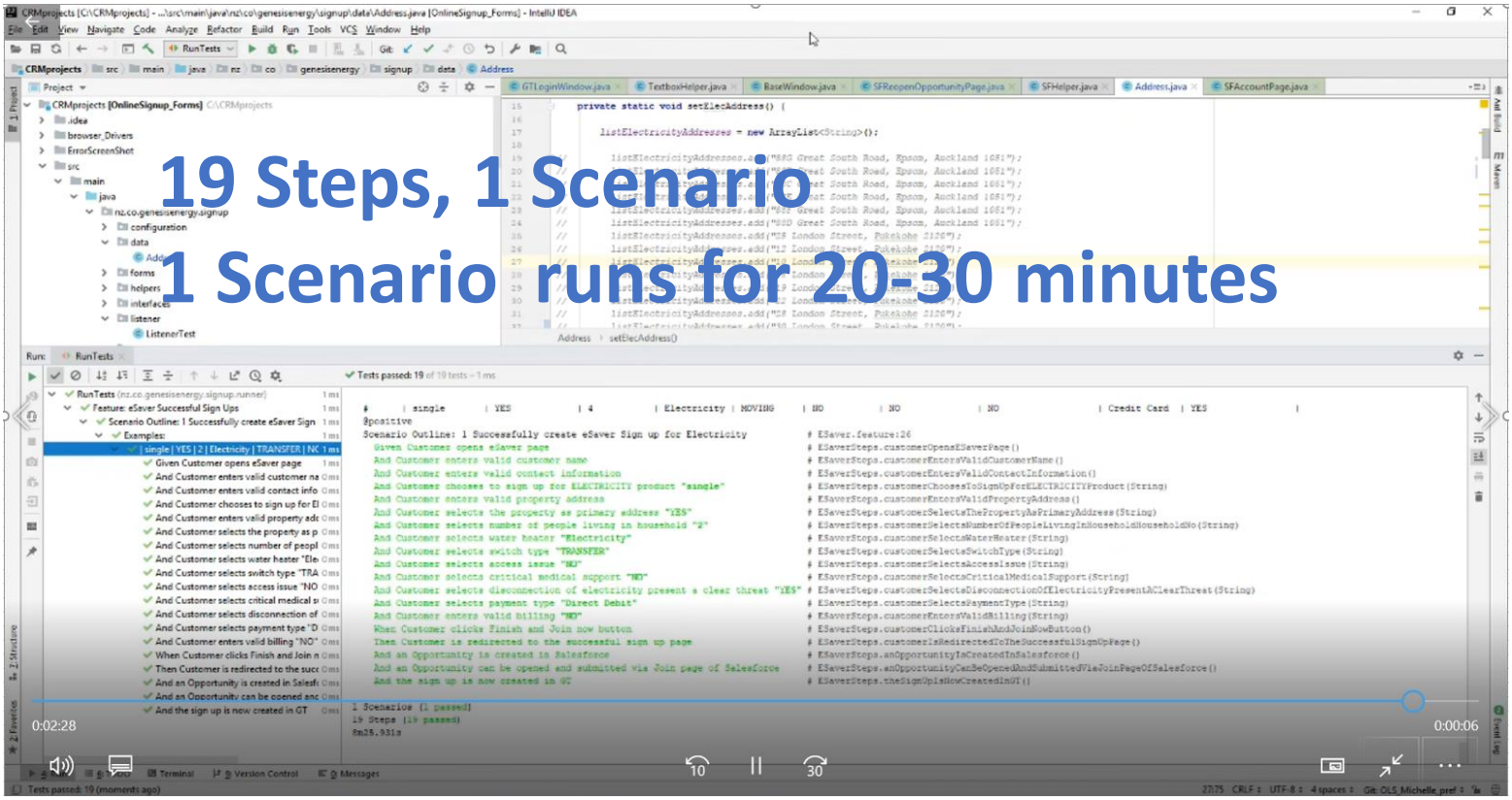
- `By.id("15")`
- `By.xpath("//*[@LocalizedControlType='document' and @Name='Text Editor']")`
- `By.name("Text Editor")`

V. DEMO

Problem 1: The web and mobile applications have been automated but there are issues in some of the fields being mapped to the backend application.




eSaver | Join Genesis
Chrome is being controlled by automated test software.
genesis
Go All In on a 12 month eSaver Plan and get our current lowest energy prices plus a $100 dual fuel bonus credit
Go All In and switch both your power and natural gas to Genesis on a 12 month eSaver plan and you'll receive our current lowest energy pricing plus a $100 dual fuel bonus credit.
With our 12 month eSaver Plan you'll receive:
Our current lowest electricity and natural gas pricing - fixed pricing for 12 months.
An additional 5% discount if you are a Dual Fuel customer - that's a total prompt payment discount of 10%.
Enjoying the convenience of having your bills paid by direct debit and receiving your bills via email.
Waiting for www.google.com...
2:12 PM
25/02/2019


19 Steps, 1 Scenario
1 Scenario runs for 20-30 minutes

VI. Tips and Tricks

Disadvantages of using Winium

- The window for application under test MUST always be opened. Test execution would fail if the window application is minimized during execution

- Desktop MUST always be active while your automation scripts are running

- Winium driver clicks wrong coordinates if not defaulted to screen scale of 100%

Tips and Tricks

1. Cancel both display power off and hibernation mode. For Windows 7 and above, enable Auto login utility to maintain active desktop

2. Don’t forget your display settings
   - Scaling should be at 100%
   - Default monitor being used (avoid monitors with touch screens)

3. Only 1 Winium Driver can be used, so ensure you check task manager and kill any pending winium driver before running your tests

Tips and Tricks (continued)

4. Use Actions to enable clearing of a textbox

```
Actions act = new Actions (driver)
act.sendKeys(" ").build().perform();
```

5. Execute the test on different machine

```
WiniumDriver driver = new WiniumDriver
(new Uri("http://<host_ip_address>:9999/"),
options);
```

Summary

- Selenium knowledge is an advantage when automating using Winium
- Winium, Selenium and/or Appium is a powerhouse tool to create an end to end framework
- Winium is a better tool when automating stand alone desktop application

Test Automation

VII. Q & A

References

- https://github.com/2gis/Winium.Desktop
- https://github.com/2gis/Winium.Desktop/releases
- https://mvnrepository.com/artifact/com.github.2gis.winium/winium-webdriver/0.1.0-1
- https://suyati.com/blog/regression-testing-windows-with-winium/
- https://github.com/2gis/Winium/tree/master/java
- https://asifahmedsarja.wordpress.com/2017/08/02/windows-application-desktop-automation-with-winium/