BEHAVIOR-DRIVEN DEVELOPMENT (BDD) AND AUTOMATION TESTING WITH CUCUMBER

November 2013

KMS Technology: http://kms-technology.com

SAMPLE REQUIREMENT

Feature: login to the system.

As a user,

I want to login into the system when I provide username and password.

Scenario: login successfully

Given the login page is opening

When I input username into the username textbox

And I input valid password into the password textbox

And I click Login button

Then I am on the Home page

AUTOMATION TEST

```
@Test
public void fb_login_test() throws Exception {
    driver.get("https://www.facebook.com/");
    driver.findElement(By.id("email")).clear();
    driver.findElement(By.id("email")).sendKeys("bddtest@yahoo.com");
    driver.findElement(By.id("pass")).clear();
    driver.findElement(By.id("pass")).sendKeys("********");
    driver.findElement(By.id("u_0_e")).click();
}
```

How?

KMS TECHNOLOGY

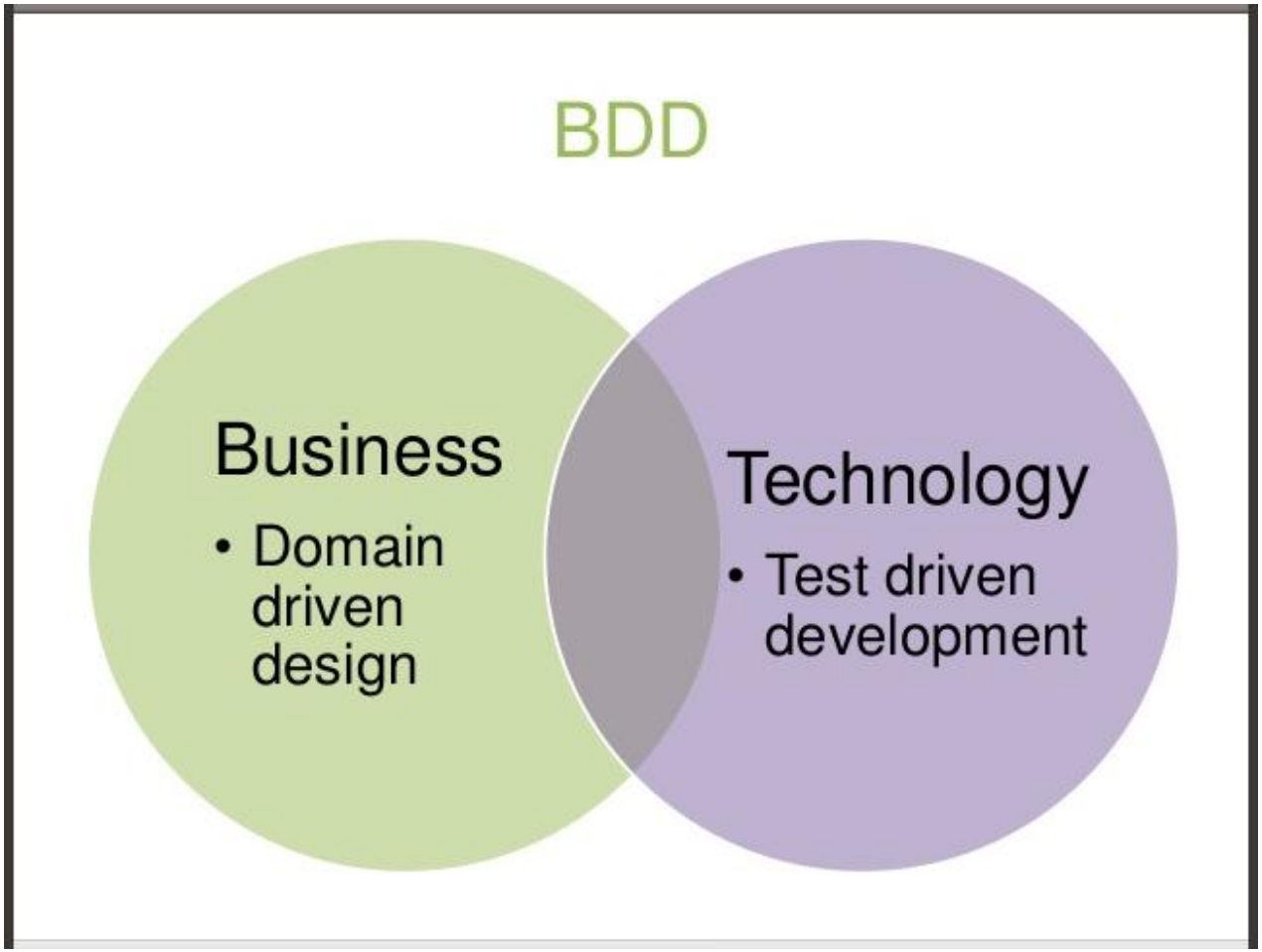
AGENDA

- BDD and Agile Development Practice
- Automation Testing with Cucumber
- Developing Cucumber-Based Automation Framework
- BDD/Cucumber – Pros & Cons

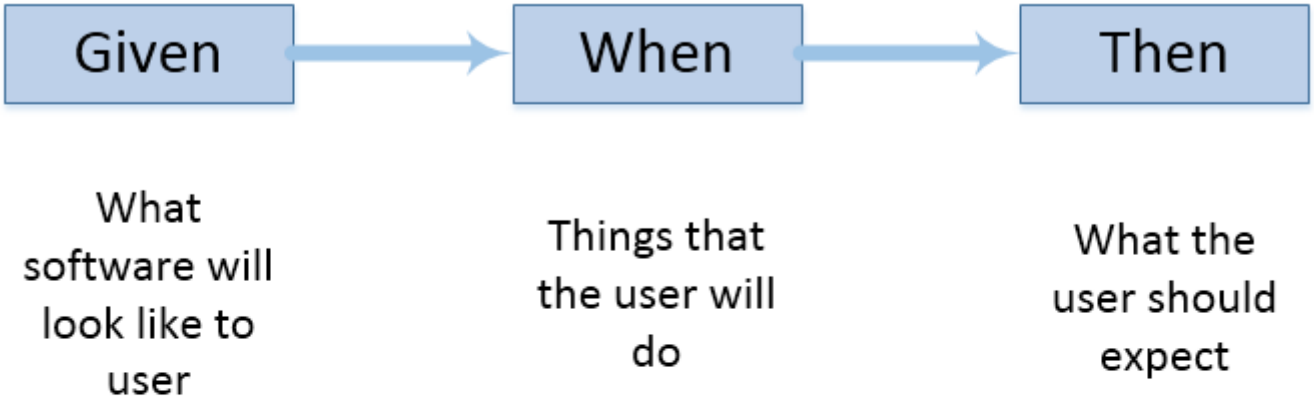
KMS TECHNOLOGY

BDD & AGILE DEVELOPMENT PRACTICE

Behavior-driven development



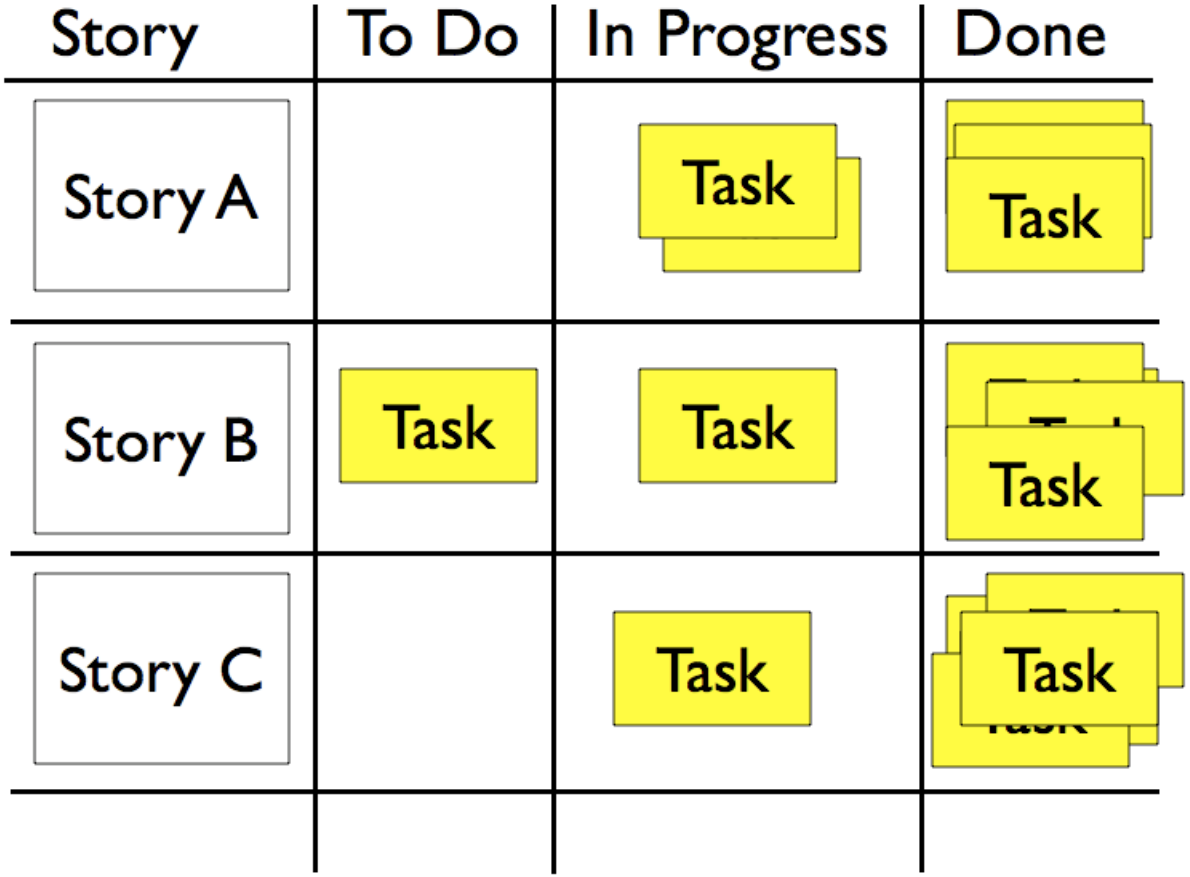
GIVEN, WHEN, THEN



Given The login page is opening
When I input username, password and click Login button
Then I am on the Home page

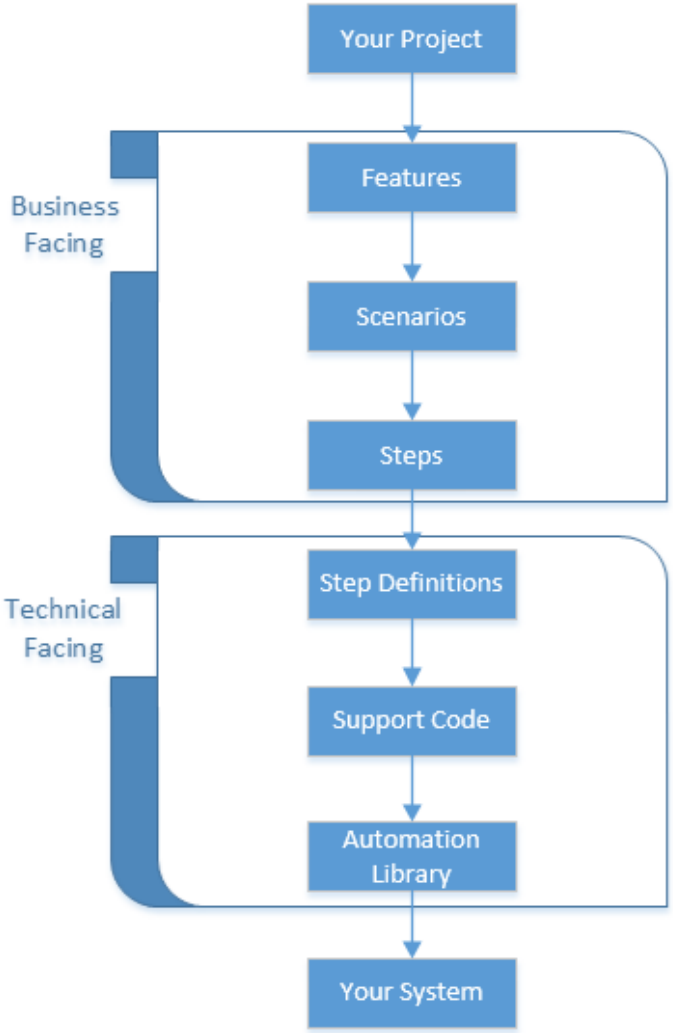
KMS TECHNOLOGY

BDD AND AGILE DEVELOPMENT PRACTICE



AUTOMATION TESTING WITH CUCUMBER

CUCUMBER APPROACH



Feature: login to the system.

As a user, I want to login into the system when I provide username and password.

Scenario: Verify that user can login gmail

Given I launch "*https://accounts.google.com*" page
When I fill in "Email " with "*kms.automation.2013@gmail.com*"
And I fill in "Passwd" with "*kms@2013*"
And I click on "signIn" button
Then I am on the "*Home*" page

```
Given /^I launch "([^\"]*)" page$/ do |page|
  visit(page)
End

When /^I fill in "([^\"]*)" with "([^\"]*)"$/ do |field, value|
  fill_in field, :with => value
end
```

Scenario Example in BDD Format

Feature: login to the system.

As a user, I want to login into the system when I provide username and password.

@tag_login_email

Scenario Outline: Verify that can login gmail

Given I launch "*https://accounts.google.com*" page

When I fill in “Email " with “**<Email >"**

And I fill in “Passwd" with "**<Password>** "

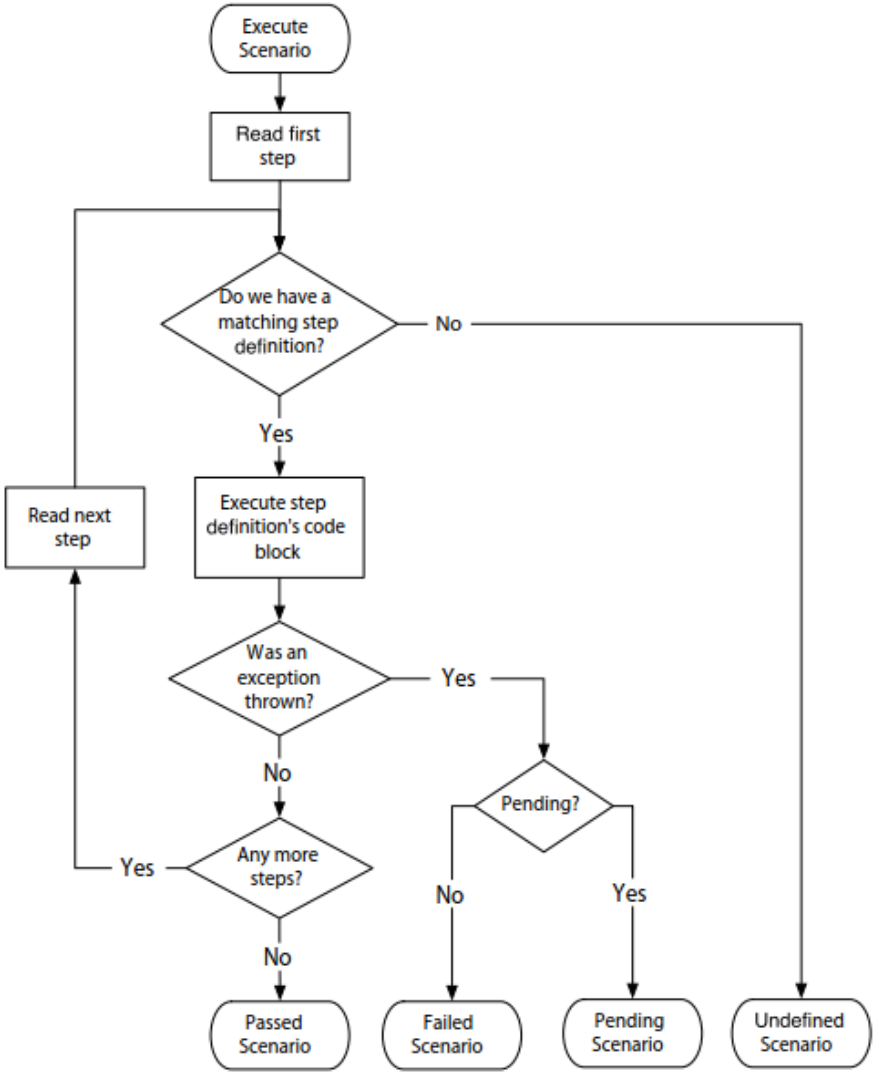
And I click on "signIn" button

Then I am on the “*Home*” page

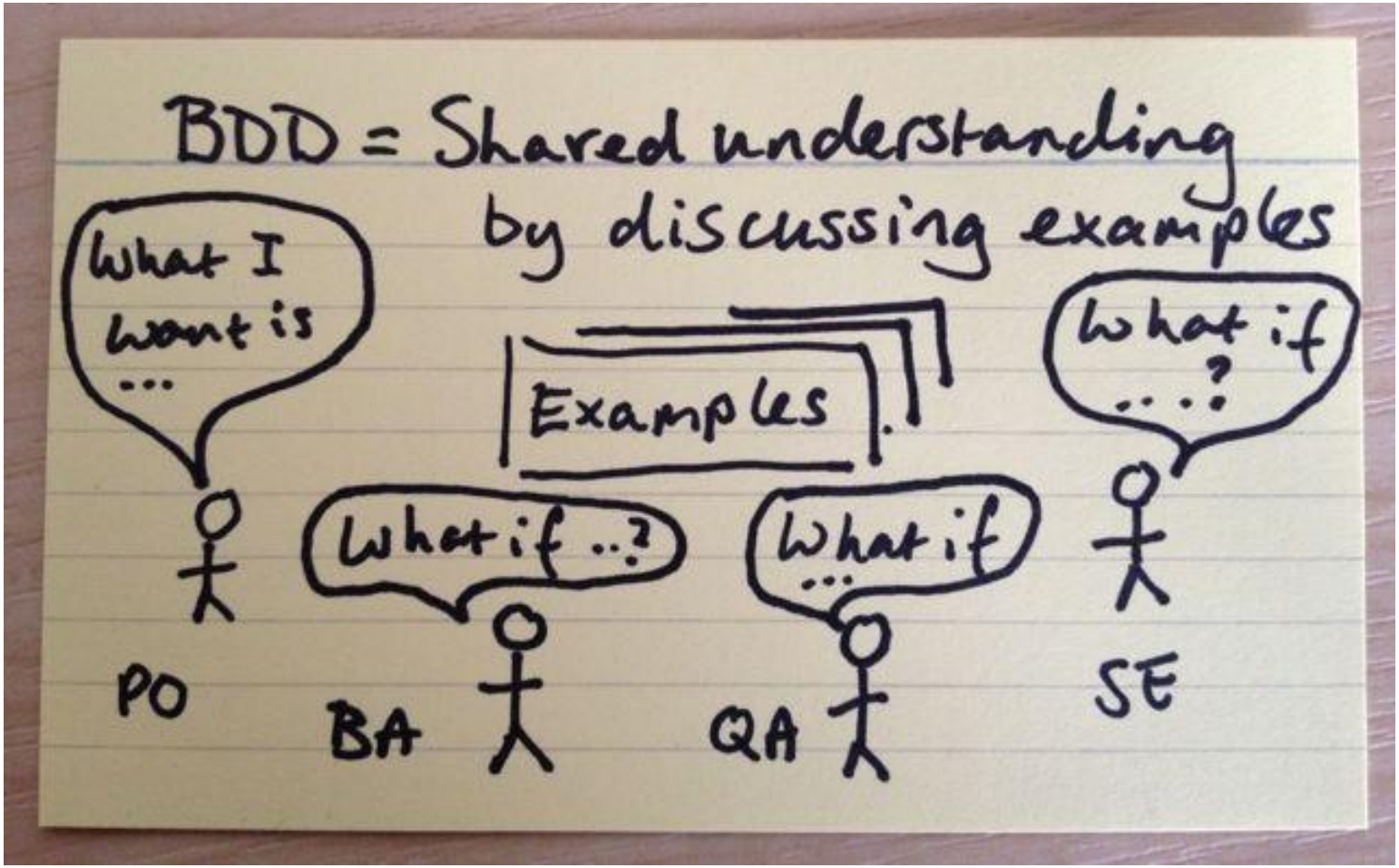
Scenarios:

Email	**Password**
kms.admin@gmail.com	*kms@2013*
kms.user@gmail.com	*kms@1234*

How Cucumber execute a Scenario



SUMMARY


BDD = Shared understanding
by discussing examples
What I want is ...
Examples
What if ...?
What if ...
What if ...?
PO
BA
QA
SE

Testing Area

- **Web application**
- **Web services**
- Desktop application
  - http://pragprog.com/book/idgtr/scripted-gui-testing-with-ruby
- Mobile application
  - http://www.moncefbelyamani.com/ios-automated-testing-with-calabash-cucumber-ruby/

CASE STUDY

Complete 70% automation testing following agile process for a Healthcare Data Provider

- **Client:**
  - Health Market Science (HMS)
- **Business Challenges:**
  - 90% automation testing must be done for every 4-week sprint
  - Big Data Technology – Since MySQL, Oracle to NoSQL, Cassandra, Storm, Hadoop
  - Complicated test data creation

INSTALLATION

- Ruby:
  - Windows: http://rubyinstaller.org/
  - Linux: https://rvm.io/rvm/install
- Cucumber:
  - gem install cucumber
    http://rubygems.org/gems/cucumber

KMS TECHNOLOGY

DEMO

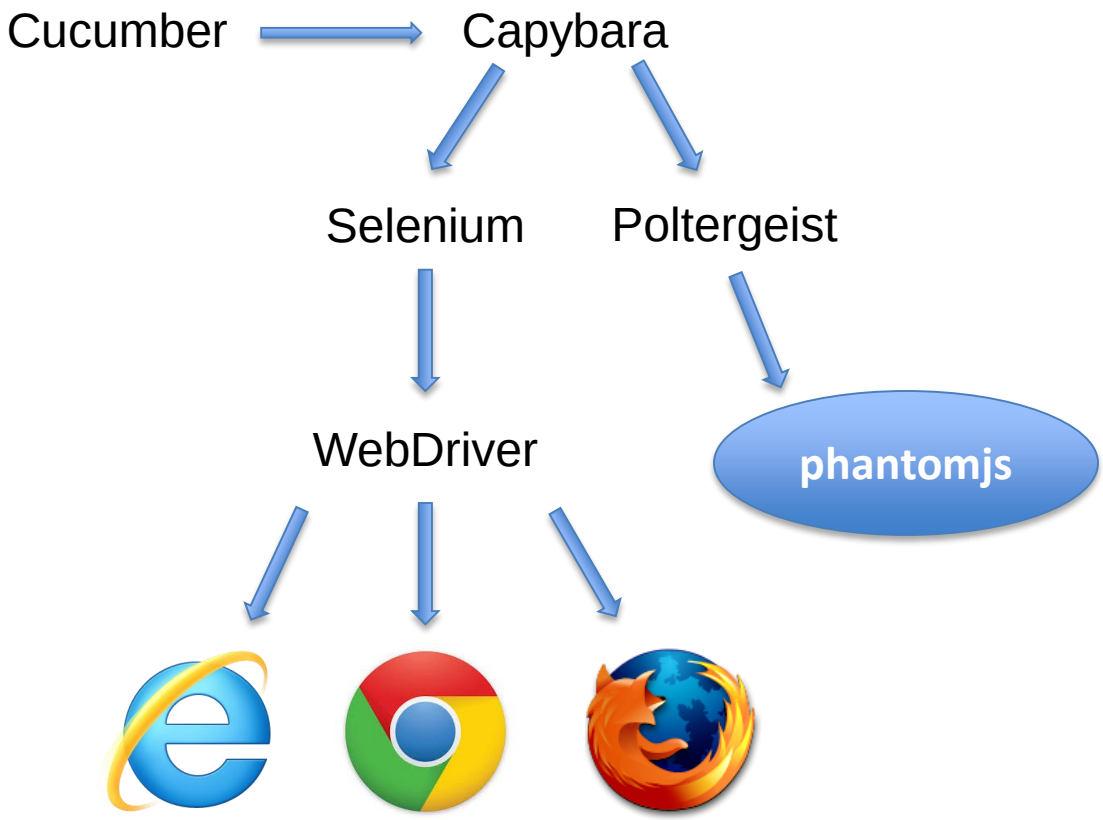
- **Web application:**

  Cucumber + Capybara

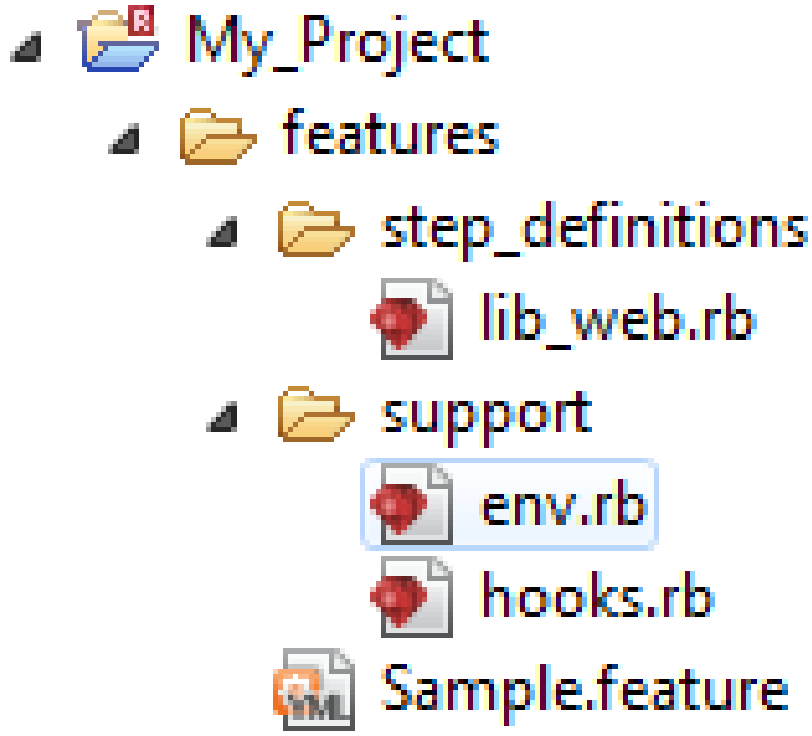
- Web Services:

  Cucumber + HTTParty + Savon

Demo: Web Application



Demo: Web Application



DEMO: WEB APPLICATION

```
Feature: login to the system.
  As a user, I want to login into the system when I provide username and password

@gmail_1
Scenario: Verify that can login gmail
Given I launch "https://accounts.google.com" page
When I fill in "Email" with "kms.automation.2013@gmail.com"
And I fill in "Passwd" with "kms@2013"
And I click on "signIn" button
Then I should see text "Primary email address.*kms.automation.2013@gmail.com"
```

Navigating

- (Object) **visit**(url)
  - The request method is **always** GET.
  - url (String) - The relative/absolute URL to navigate to.
  - Examples:
    - visit('http://google.com')

DEMO: WEB APPLICATION

```
Feature: login to the system.
  As a user, I want to login into the system when I provide username and password

@gmail_1
Scenario: Verify that can login gmail
Given I launch "https://accounts.google.com" page
When I fill in "Email" with "kms.automation.2013@gmail.com"
And I fill in "Passwd" with "kms@2013"
And I click on "signIn" button
Then I should see text "Primary email address.*kms.automation.2013@gmail.com"
```

Navigating

Action

- (Object) click_link(locator, options = {})
  - Finds a link by id or text and clicks it.
- (Object) click_button(locator, options = {})
  - Finds a button by id, text or value and clicks it.
- (Object) fill_in(locator, options = {})
  - Locate a text field or text area. The field can be found via its name, id or label text.

DEMO: WEB APPLICATION

```
Feature: login to the system.
  As a user, I want to login into the system when I provide username and password

@gmail_1
Scenario: Verify that can login gmail
Given I launch "https://accounts.google.com" page
When I fill in "Email" with "kms.automation.2013@gmail.com"
And I fill in "Passwd" with "kms@2013"
And I click on "signIn" button
Then I should see text "Primary email address.*kms.automation.2013@gmail.com"
```

Navigating

Action

Check Point

- (Boolean) has_content?(content) /
- (Boolean) has_no_content?(content)
  - content (String) - The text to check for
  - Examples:
    - page.has_content?('hello')
    - page.has_no_content?('hello')

DEMO: WEB APPLICATION

```
Feature: login to the system.
  As a user, I want to login into the system when I provide username and password

@gmail_1
Scenario: Verify that can login gmail
Given I launch "https://accounts.google.com" page
When I fill in "Email" with "kms.automation.2013@gmail.com"
And I fill in "Passwd" with "kms@2013"
And I click on "signIn" button
Then I should see text "Primary email address.*kms.automation.2013@gmail.com"
```

- RSpec magic matchers
  - page.should **have**_selector('table tr')
  - page.should **have**_no_selector(:content)
  - page.should **have**_css('table tr.foo')
  - page.should **have**_content('foo')

DEMO: WEB APPLICATION

```
Feature: login to the system.
  As a user, I want to login into the system when I provide username and password

@gmail_1
Scenario: Verify that can login gmail
Given I launch "https://accounts.google.com" page
When I fill in "Email" with "kms.automation.2013@gmail.com"
And I fill in "Passwd" with "kms@2013"
And I click on "signIn" button
Then I should see text "Primary email address.*kms.automation.2013@gmail.com"
```

```
Given /^I launch "([^\"]*)" page$/ do |page|
  visit(page)
end

When /^I click on "([^\"]*)" button$/ do |button|
  click_button button
end

When /^I fill in "([^\"]*)" with "([^\"]*)"$/ do |field, value|
  fill_in field, :with => value
end
```

DEMO: WEB APPLICATION

```
Feature: login to the system.
  As a user, I want to login into the system when I provide username and password

@gmail_1
Scenario: Verify that can login gmail
Given I launch "https://accounts.google.com" page
When I fill in "Email" with "kms.automation.2013@gmail.com"
And I fill in "Passwd" with "kms@2013"
And I click on "signIn" button
Then I should see text "Primary email address.*kms.automation.2013@gmail.com"
```

Regular Expression Regular Expression

```
Then /^I should( not)? see text "([^\"]*)"$/ do |negate, text|
  negate = negate.strip if negate != nil
                                Regular Expression
  if negate == 'not'
    page.has_no_text?(Regexp.new(/#{text}/)).should == true
  else
    page.has_text?(Regexp.new(/#{text}/)).should == true
  end
end
```

TAKE A REST

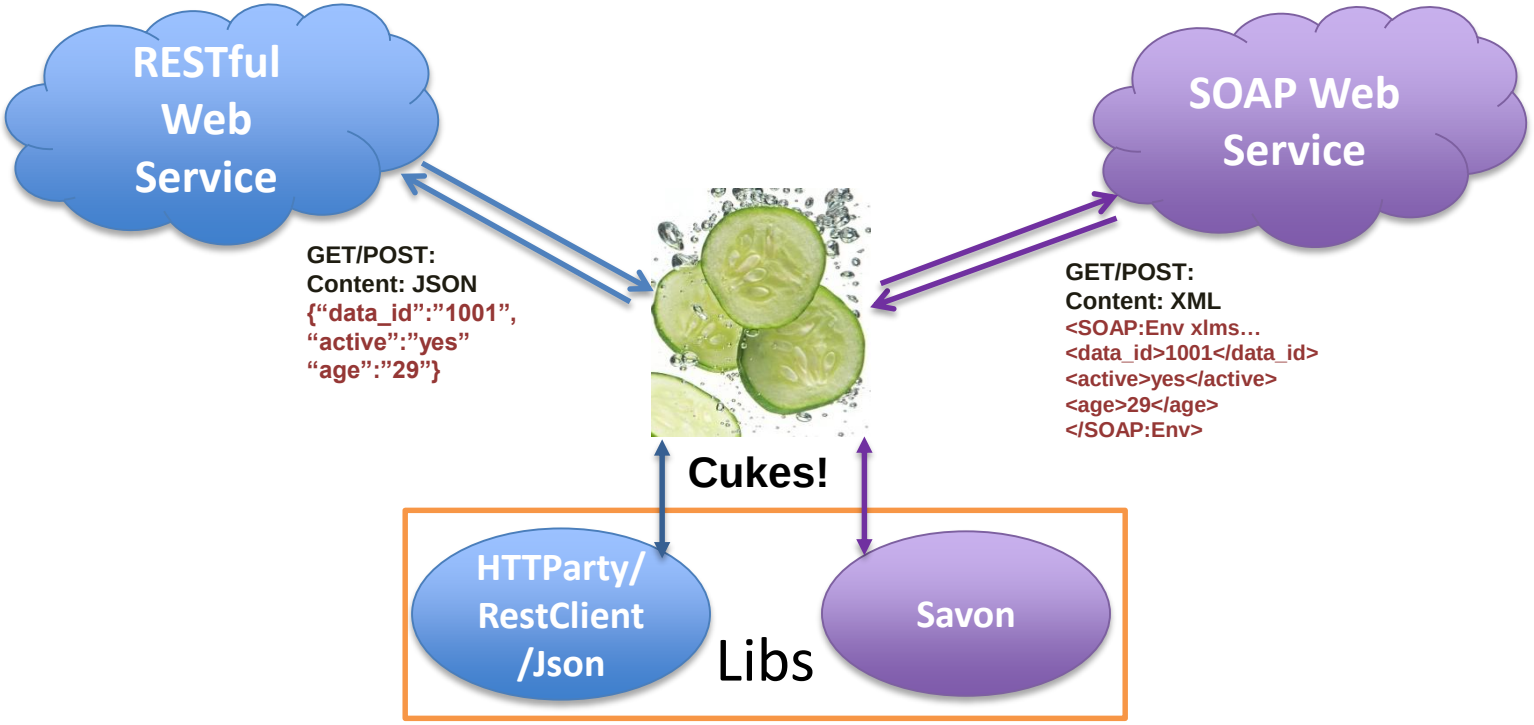
KMS TECHNOLOGY

DEMO

- Web application:
  Cucumber + Capybara
- **Web Services:**
  Cucumber + HTTParty + Savon

Demo: Web Services

- At a glance:



DEMO: WEB SERVICES

HTTParty

HTTParty.**get** 'http://myexample.com/newfeeds‘

⇒ **Response.body=**{“id”:”100”,”Message”:”Hello”}

HTTParty.**post** 'http://myexample.com/updateStatus',
{:date => “Nov-11-2013”, :status => “Full of energy today”}

⇒ **Response.body=**{“update”:”success”,”id”:”425”}

HTTParty.**put** 'http://myexample.com/status/message’, {:timeout => 5000}

⇒ **Response.body=**{“success”:”true”,”message”:”message_1377057406713”}

HTTParty.**delete** 'http://myexample.com/status/message_1377057406714‘

=> **Response.body=**{“success”:”false”,”reason”:”non-exist”}

JSON

ActualRslt = JSON.parse(response.body)
Msg1 = ActualRslt[“message”] #= “Hello”

ActualRslt = JSON.parse(response.body)
Msg1 = ActualRslt[“update”] #= “success”

ActualRslt = JSON.parse(response.body)
Msg1 = ActualRslt[“message”]
#= “message_1377057406713”

ActualRslt = JSON.parse(response.body)
Msg1 = ActualRslt[“success”] #= “false”
Msg1 = ActualRslt[“reason”] #= “non-exist”

KMS TECHNOLOGY

DEMO: WEB SERVICES

Savon

```
# create a client for the service
client = Savon.client(wsdl:
'http://service.example.com?wsdl')
client.operations
=> [:find_user, :list_users]
# call the 'findUser' operation
response = client.call(:find_user,
message: { id: 42 })
```

```
actualRes = response.xml
<?xml version="1.0" encoding="utf-
8"?><soap:Envelope....XMLSchema">
<soap:Body><findUserResponse
xmlns="http://service.example.com">
<findUserResult><Success>true</Success>
<LastName>Smith</LastName>
<FirstName>John</FirstName>
<City>Mountain View</City>
<State>CA</State><WeatherID>
</findUserResult></findUserResponse>
</soap:Body></soap:Envelope>
```

DEMO: WEB SERVICES

Feature: Get Weather status to display on webpage

As an owner of Traveling Service, I want to be able to get weather status of any location based on address of client's request

Scenario Outline: Get weather status by address of client's requests

Given The check IP location and weather web services are running

When I **send request** to get location detail of address "<Address>"

Then I should have ZIP code and "<Country>" of that location

And I **send request** to get weather status of that location by its ZIP code

Then I should **receive** current "Temperature, Wind, RelativeHumidity" and "<City_Name>" and "<State>" of that location

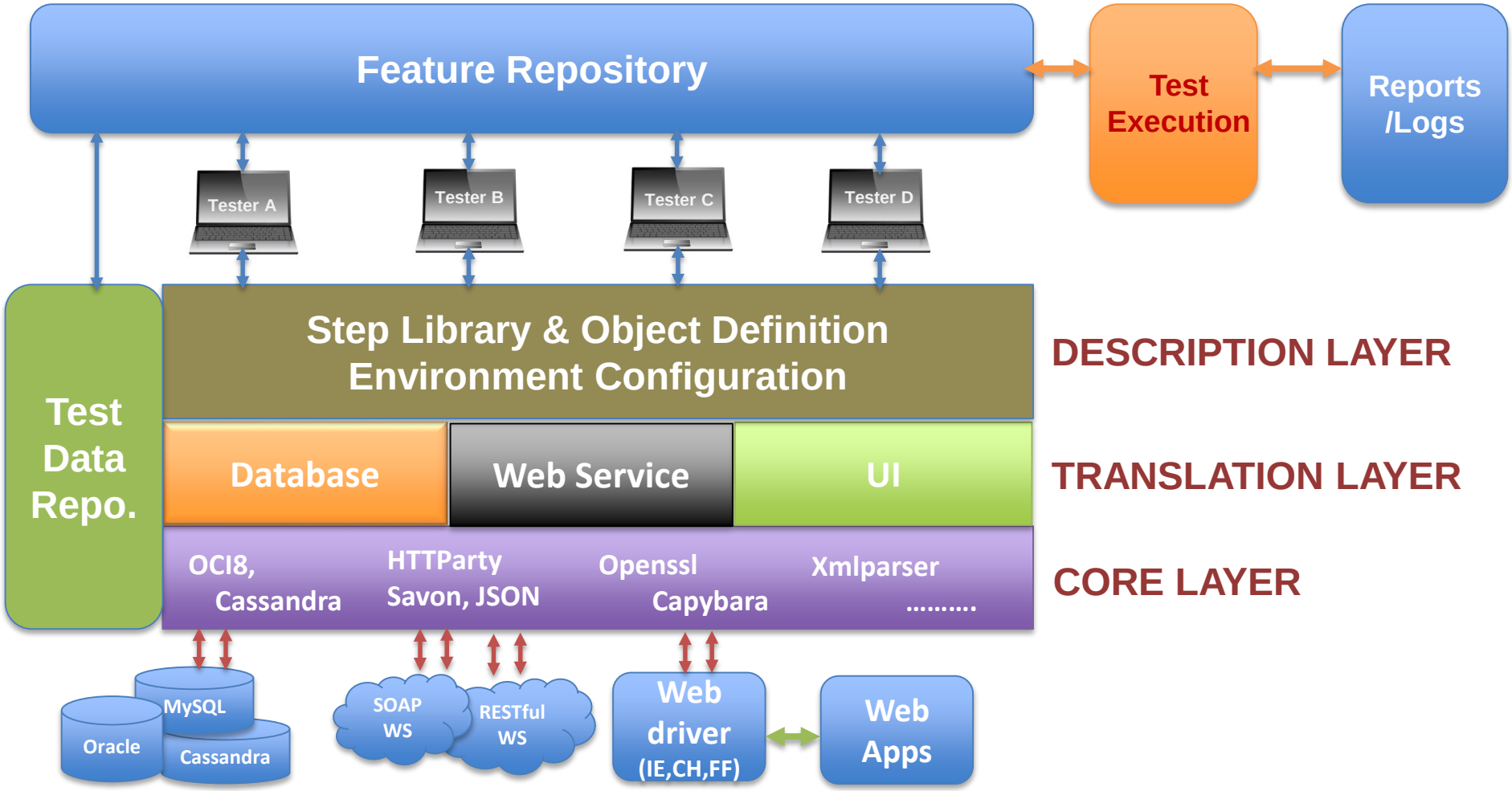
Scenarios:

Address	**Country**	**City_Name**	**State**
www.google.com	United States	Mountain View	California

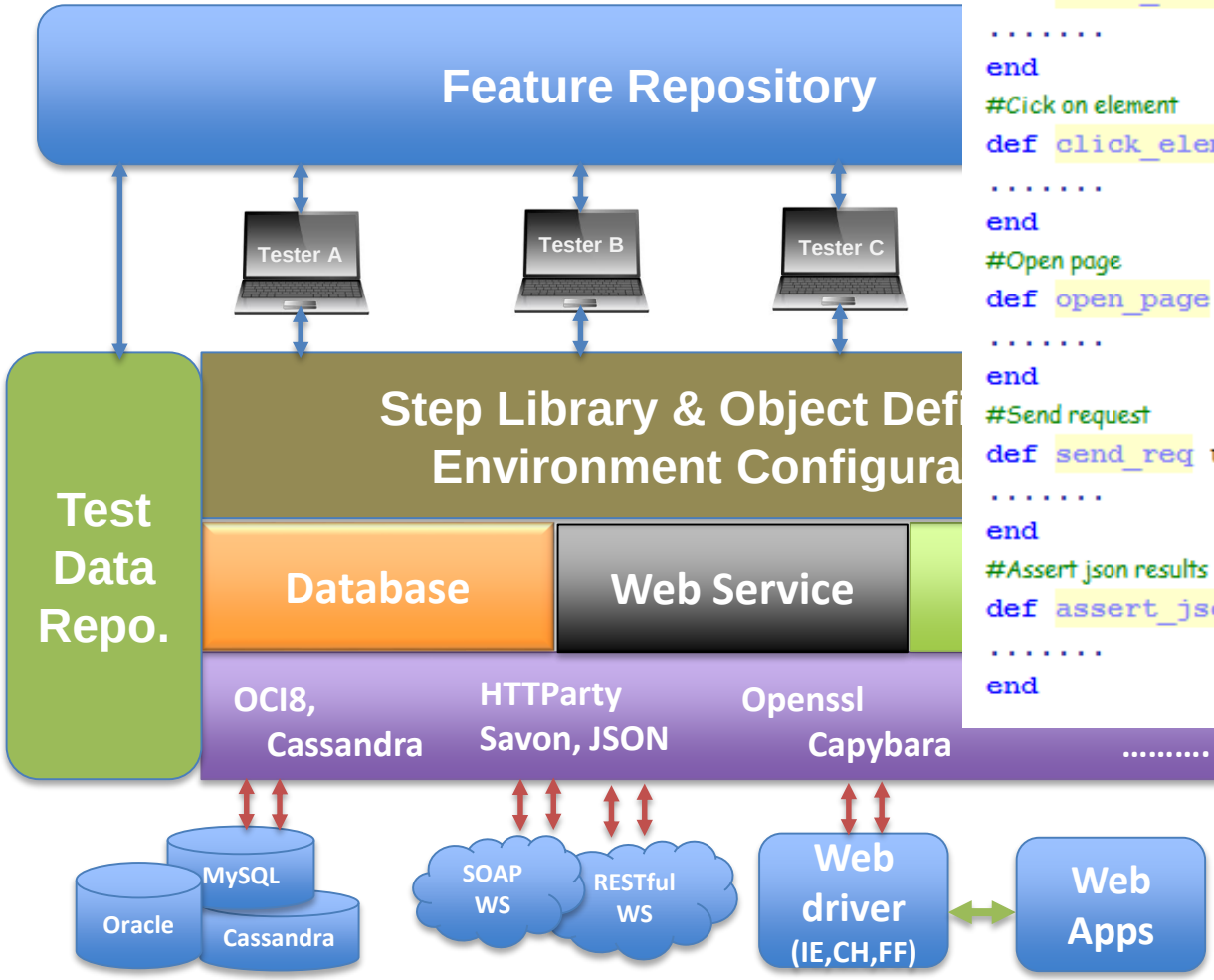
KMS TECHNOLOGY

DEVELOPING CUCUMBER-BASED AUTOMATION FRAMEWORK

DEVELOPING CUCUMBER-BASED AUTOMATION FRAMEWORK

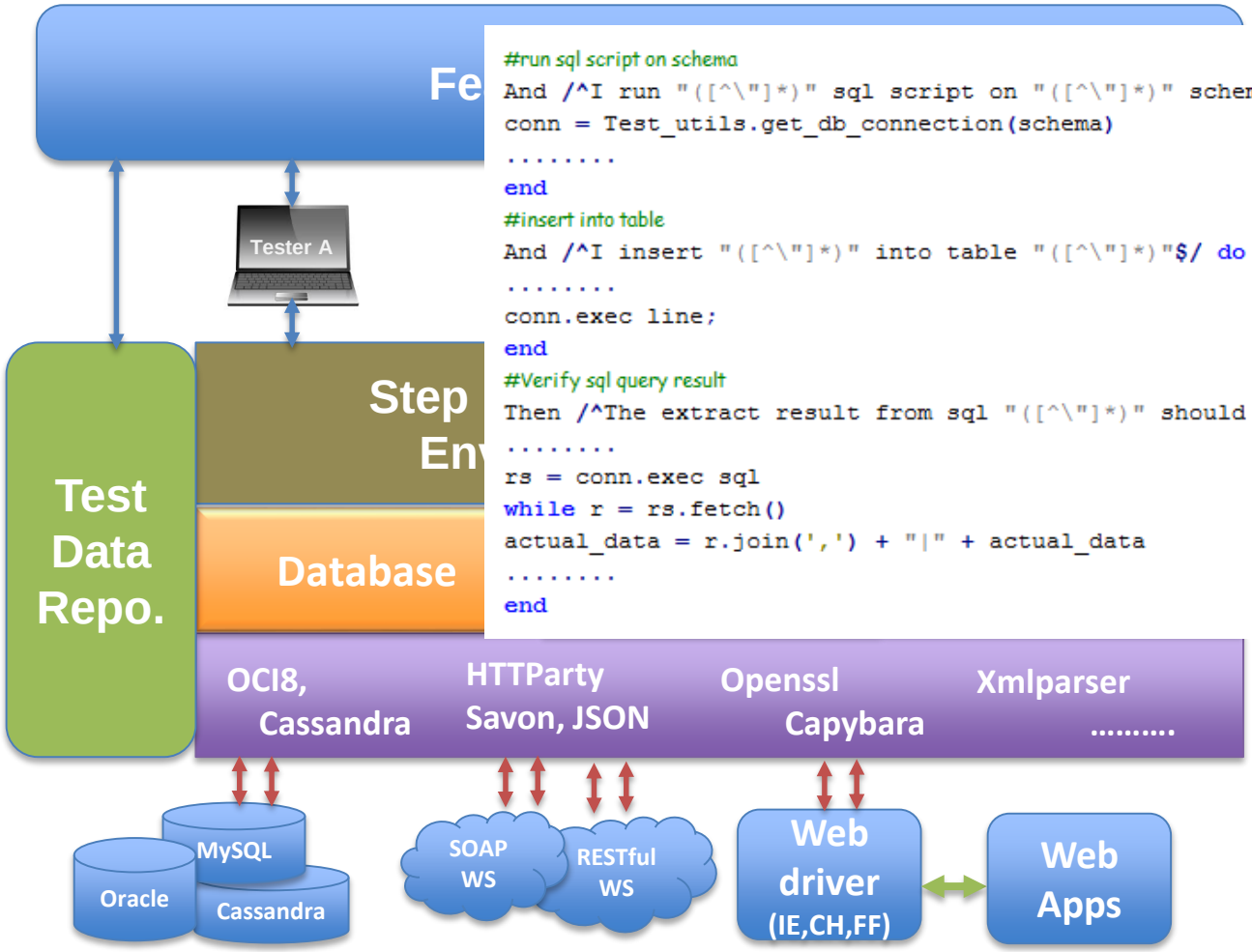


DEVELOPING CUCUMBER-BASED AUTOMATION FRAMEWORK




```
#Verify text on element
def check_text element, text
.......
end
#Cick on element
def click_element element
.......
end
#Open page
def open_page url
.......
end
#Send request
def send_req url, action, param
.......
end
#Assert json results
def assert_json_json expectedObj, actualObj
.......
end
```

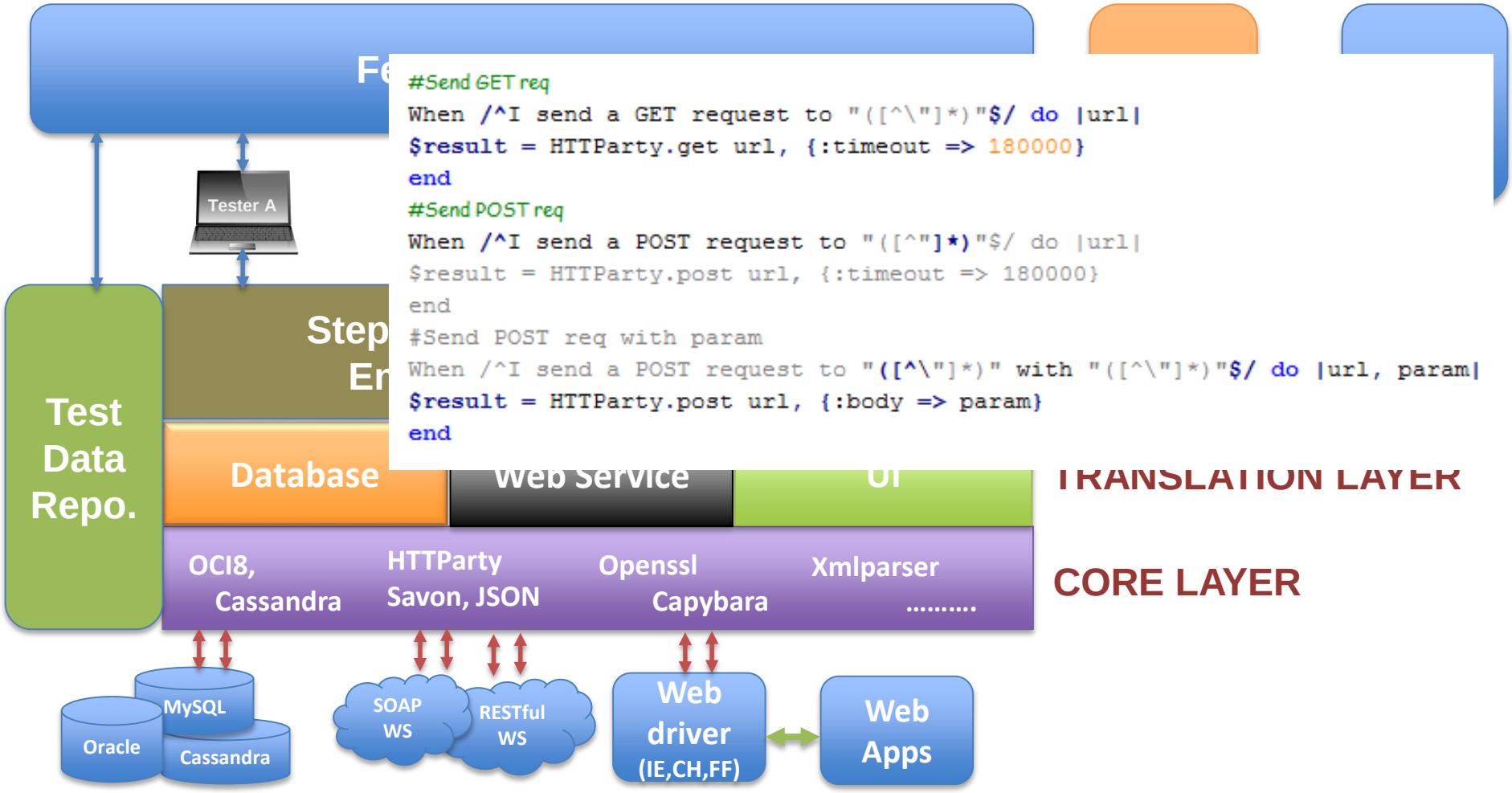
DEVELOPING CUCUMBER-BASED AUTOMATION FRAMEWORK




```
#run sql script on schema
And /^I run "([^\"]*)" sql script on "([^\"]*)" schema$/ do |sql, schema|
conn = Test_utils.get_db_connection(schema)
........
end
#insert into table
And /^I insert "([^\"]*)" into table "([^\"]*)"$/ do |data, table|
........
conn.exec line;
end
#Verify sql query result
Then /^The extract result from sql "([^\"]*)" should be as "([^\"]*)"$/ do |sql, expected|
........
rs = conn.exec sql
while r = rs.fetch()
actual_data = r.join(',') + "|" + actual_data
........
end
```

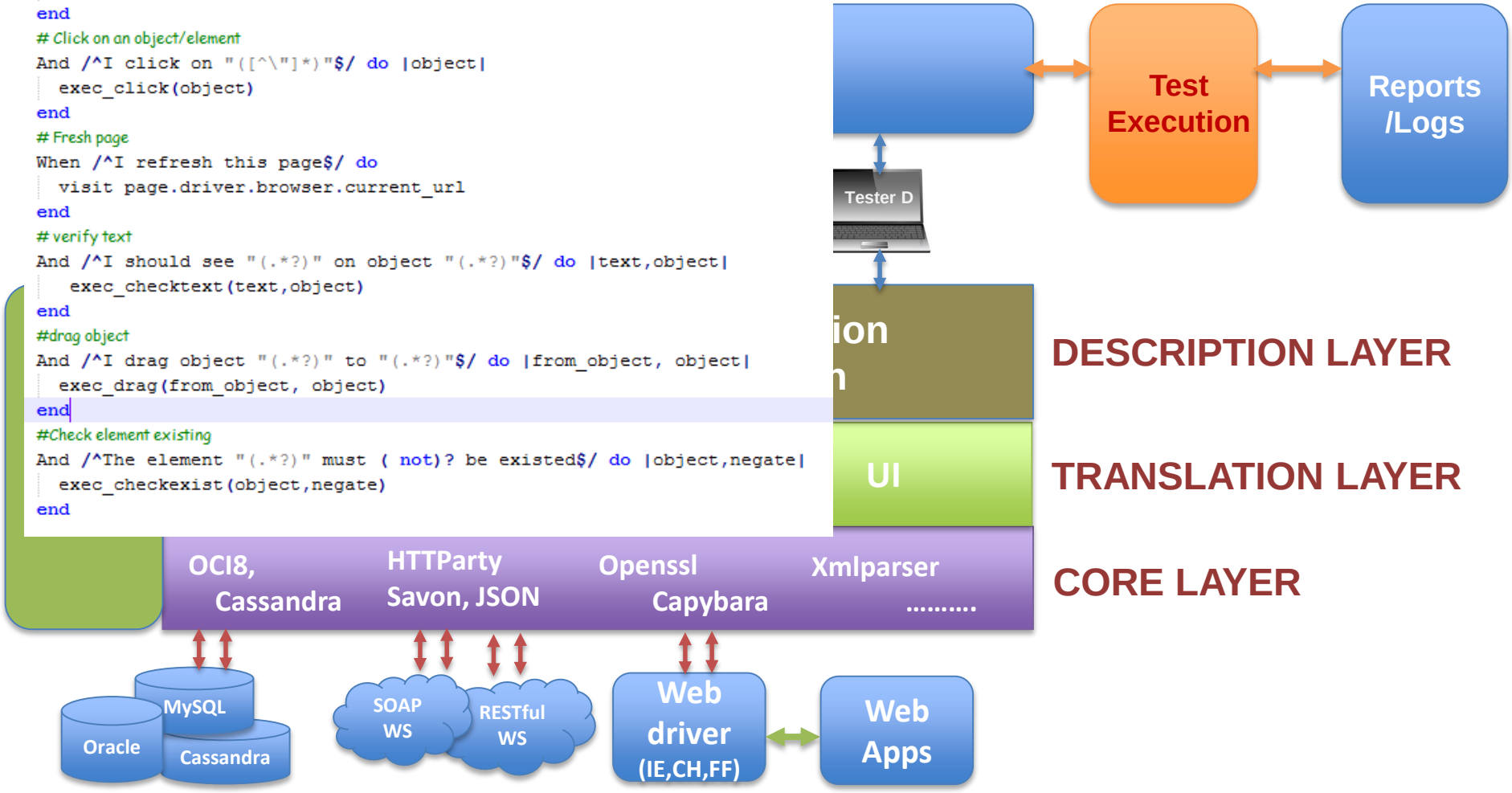
CORE LAYER

DEVELOPING CUCUMBER-BASED AUTOMATION FRAMEWORK



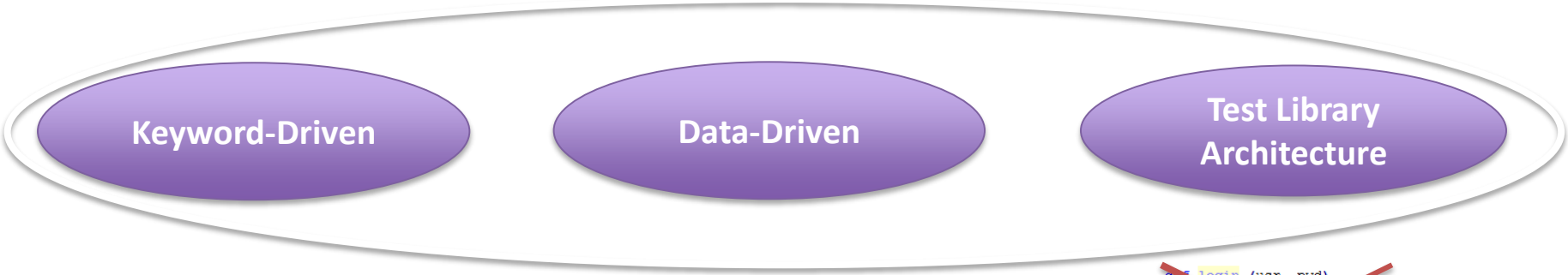
DEVELOPING CUCUMBER-BASED MATION FRAMEWORK

```
#Open page
Given /^I open page "([^\"]*)"$/ do |page|
  exec_openbrowser(page)
end
# Click on an object/element
And /^I click on "([^\"]*)"$/ do |object|
  exec_click(object)
end
# Fresh page
When /^I refresh this page$/ do
  visit page.driver.browser.current_url
end
# verify text
And /^I should see "(.*?)" on object "(.*?)"$/ do |text,object|
   exec_checktext(text,object)
end
#drag object
And /^I drag object "(.*?)" to "(.*?)"$/ do |from_object, object|
  exec_drag(from_object, object)
end
#Check element existing
And /^The element "(.*?)" must ( not)? be existed$/ do |object,negate|
  exec_checkexist(object,negate)
end
```



DEVELOPING CUCUMBER-BASED AUTOMATION FRAMEWORK

A Hybrid Test Automation Framework



Object	Action	Argument
Login.txt_usr	set	myname
Login.txt_pwd	set	mypass
Login.btn_sign	click	

```
$_RP_OBJECT['login.txt_usr'] = {name:'j_usr'}
$_RP_OBJECT['login.txt_pwd'] = {name:'j_pwd'}
$_RP_OBJECT['login.btn_sign'] =.
{xpath_selector:'//button/span/span[text()="Login"]'}

When /^I login by "(.*)" and "(.*)"$/ do |usr, pwd|
step %{I launch my page}
step %{I set text on "login.txt_usr" with "#{usr}"}
step %{I set text on "login.txt_pwd" with "#{pwd}"}
step %{I click on "login.btn_sign"}
end
```

```
--- # Expected data for orders - ord.yaml file
- table-name: "order_detail:"
table-id: orderDetail
ignore-columns: []
expected-data:
    - Order ID    Name
    - 10000001    Canon 5D Mark II
    - 10000002    Apple Ipad Air
    - 10000003    Iphone 5S
    - 10000004    Samsung Note 3
    - 10000005    LG Opt G
```

```
Then /^My orders in "(.*?)" should be as "(.*)"$/ do |object,file|
  found_element = find_object(object)
  exectue_checkYamlRows found_element,file
end
```

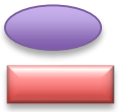
```
def login (usr, pwd)
#...more code lines here
visit(url)
fill_in("user_name",:with=>usr)
fill_in("password",:with=>pwd)
click_button "Sign in"
end
```

Functional Decomposition

```
def execute_openbrowser url_site
#..more code lines here
browser = Capybara.current_session.driver.browser
browser.manage.delete_all_cookies
visit(url_site)
end

def execute_settext element, text
foundElement = find_object(element)
foundElement.set(text)
#..logging codes here
end

def execute_click element
foundElement = find_object(element)
page.driver.browser.mouse.click(foundElement.native)
#..logging codes here
end
```



Framework

Method

BDD/CUCUMBER – PROS & CONS

BDD/Cucumber - Pros

- BBD is FRIENDLY and UNDERSTANDABLE by non-technical User
- Great support from RUBY community - Automation framework based BDD Cucumber is NOT REALLY HARD to develop and maintenance
- Support on MULTIPLE PLATFORM, OS and different browsers

BDD/Cucumber - Cons

- Incompatibility among GEM versions
- Lacking of tool for managing Features and Scenarios effectively

Q&A

THANK YOU

© 2013 KMS Technology

KMS TECHNOLOGY

REFERENCES

- All about BDD Cucumber: http://cukes.info/
- BDD cucumber book: http://www.amazon.com/The-Cucumber-Book-Behaviour-Driven-Development/dp/1934356808
- Ruby gems: http://rubygems.org
- Ruby programming: http://www.ruby-lang.org