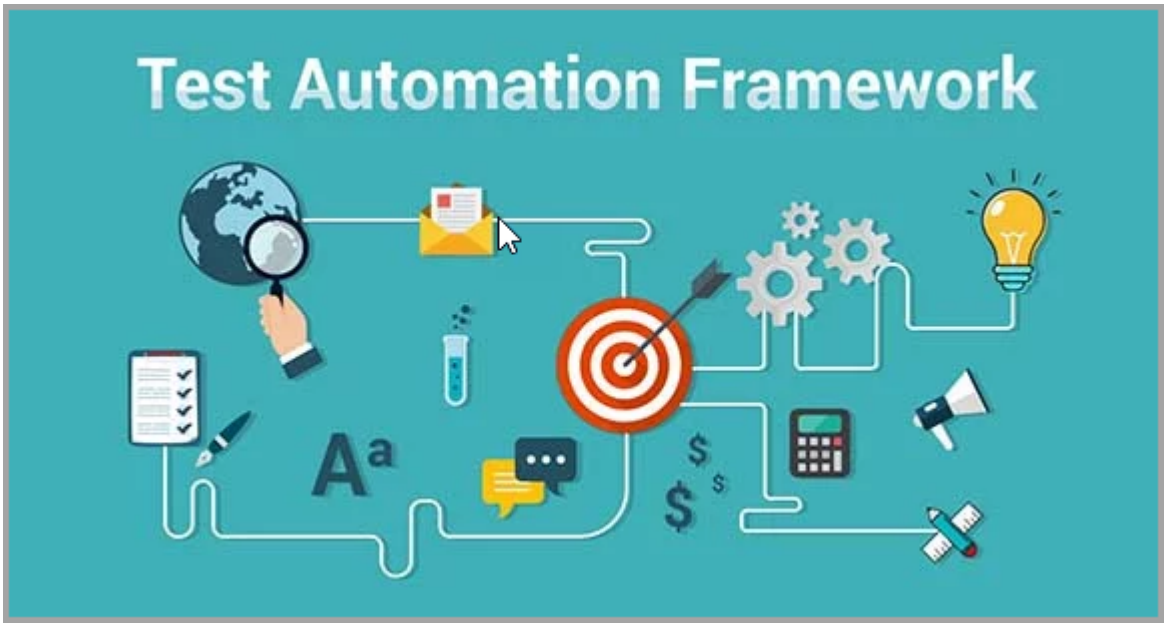
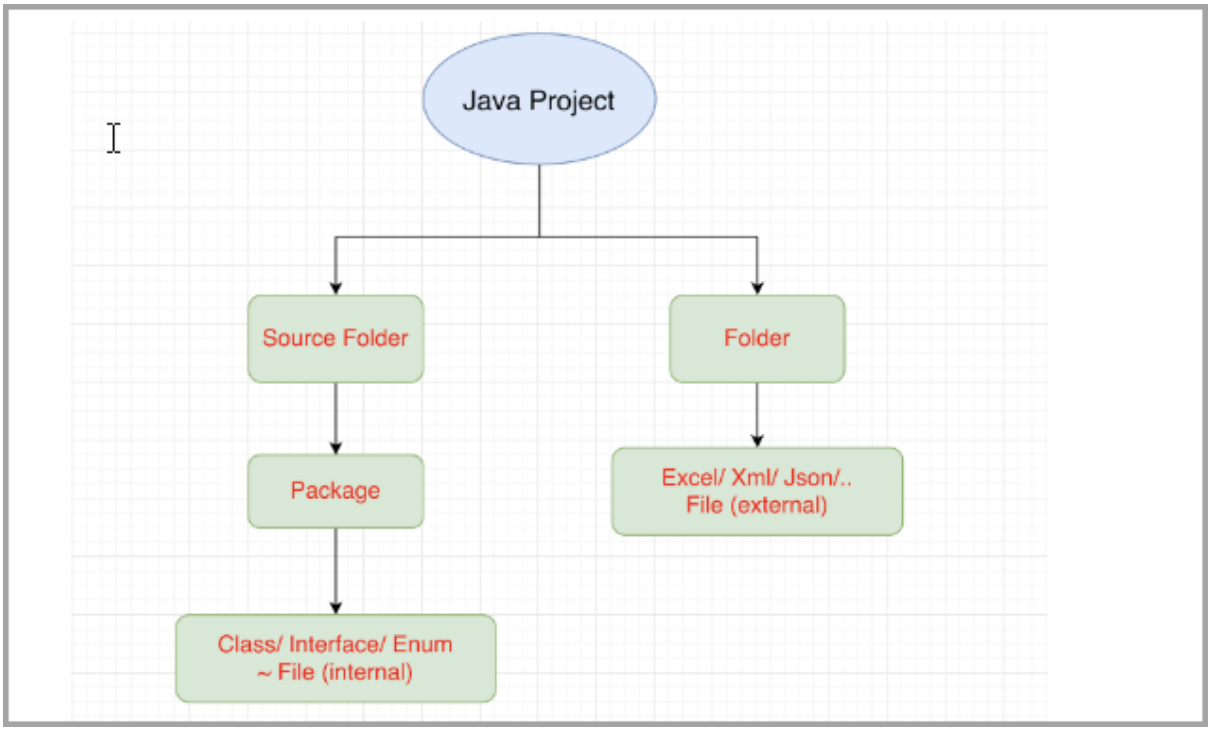
Trainer: LE DAC MINH
Phone: 0966152432
Email: ledacminh0305@gmail.com



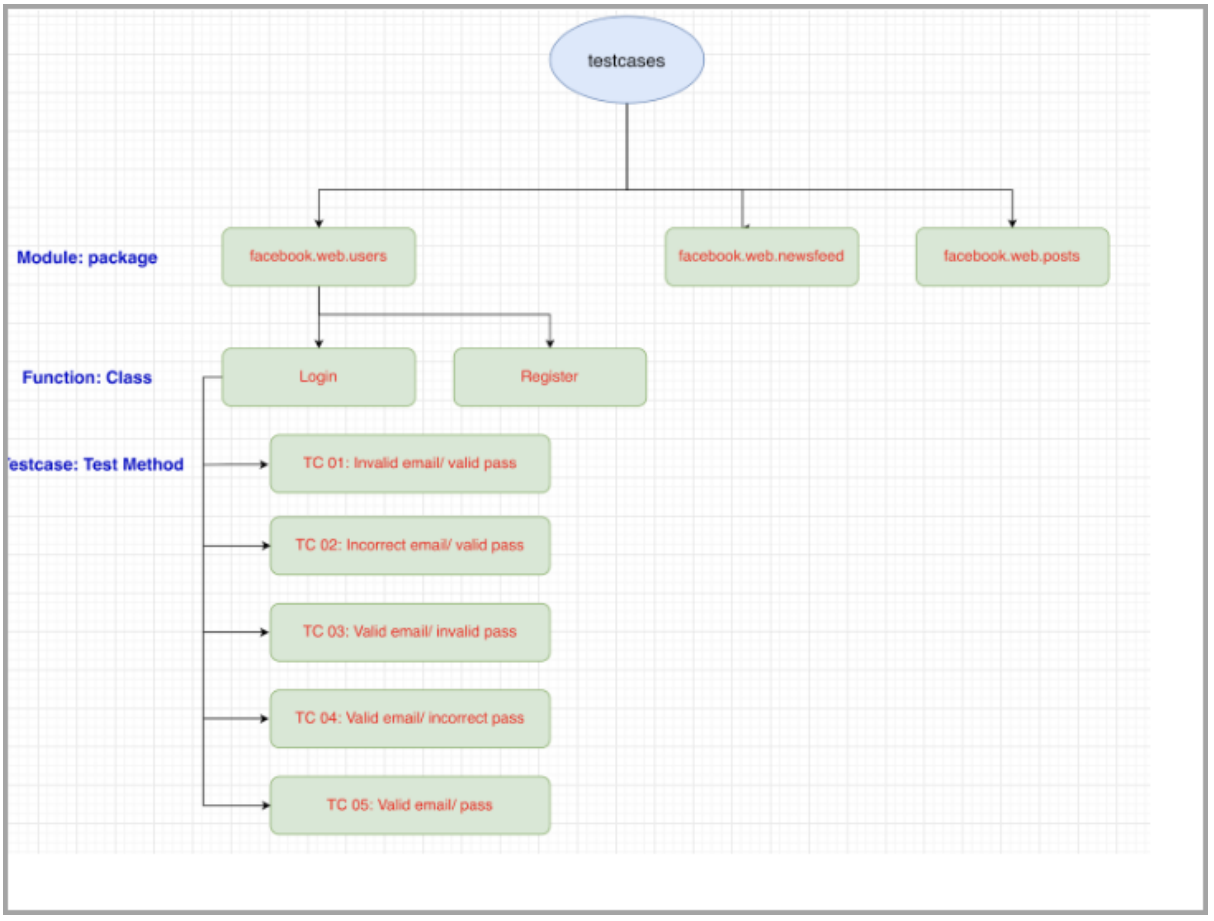
- **Target**
  - Building a framework
  - Building components/ layers
  - Building utilities
  - Running multiple browsers/ enviroments
  - A ware of settings/ generating reports (log)
  - Understanding and applying Design Parttern while implementing a Framework.
    - Page Object Parttern
    - Singleton Parttern
    - Parttern Object
    - Factory Parttern/ Abstract Factory
  - Intergating and Running the test scripts on Jenkins

Trainer: LE DAC MINH
Phone: 0966152432
Email: ledacminh0305@gmail.com

- **Java project structure**

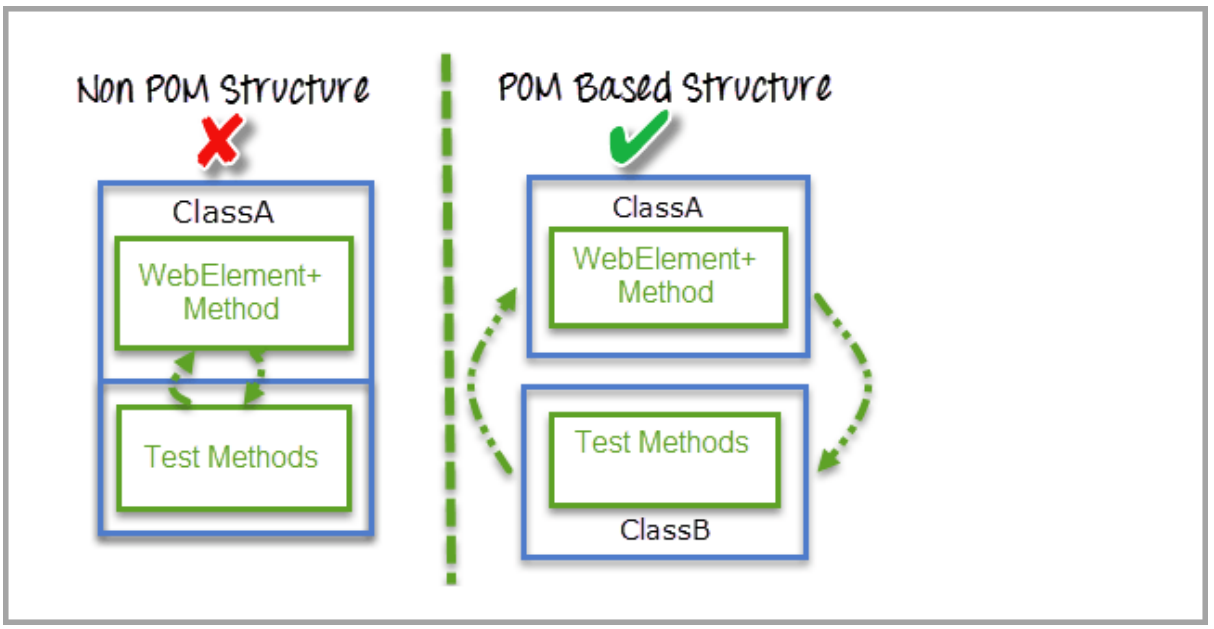


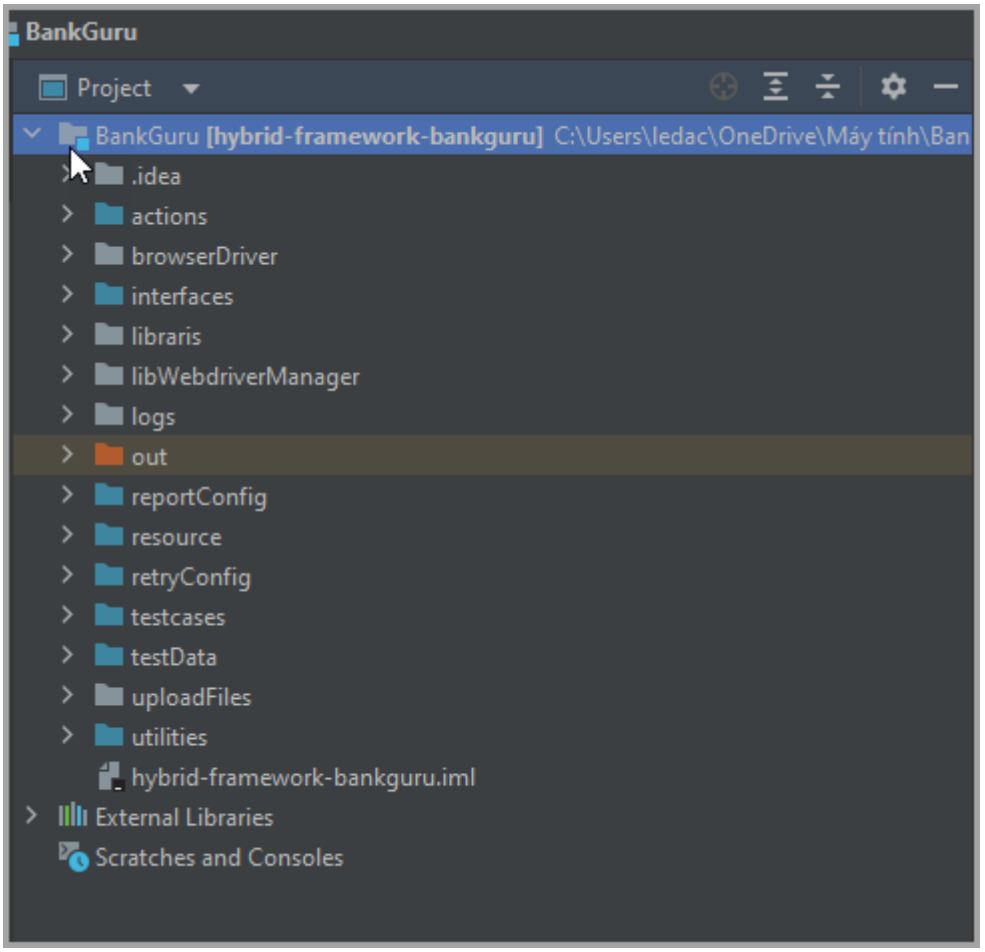
- **Java project automation structure**



Trainer: LE DAC MINH
Phone: 0966152432
Email: ledacminh0305@gmail.com

- **POM**





Trainer: LE DAC MINH
Phone: 0966152432
Email: ledacminh0305@gmail.com

FUNCTIONS/ ACTIONS

- Commons: Managing functions using for test cases/ pages
- AbstractPage/ BasePage: using for pageObjects
    - click/ type/ select…..
    - inEnabled/ isDisplayed/ isSelected …..
    - waitForDisplayed/ waitForControl….
    - handle alert/ window popup/ user interactions….
    - checked/ unchecked on checkbox….
    - …..
- AbstractTest/ BaseTest: Including functions using for test cases layer
    - open browser
    - close browser
    - verify true/ false/ equals
    - random number
    - get date/ month/ year
    - …..
- Report/ log….
- Global Constants: URL Server, user/ pass, driver path….
- pageObjects: Managing actions in individual page

INTERFACES LAYER

- Managing UI/ HTML/ Locators of the pages in the project (each page in action's layer will have one page [UI] correspondingly in interfaces' class.
    - LoginPageUI
    - HomePageUI
    - NewCustomerPageUI
    - EditCustomerPageUI
    - NewAccountPageUI
    - LogoutPageUI
    - …..

Trainer: LE DAC MINH
Phone: 0966152432
Email: ledacminh0305@gmail.com

RESOURCES LAYER

- Managing all of the external files in project
- TestNG: file .xml to run/manage test cases
- Log4j (library support): file .xml/ .properties
- Excel: file data .xlsx

THIRD PARTY TOOL

- **Libraries**
  - Managing libraries for the project
    - log4j.jar
- browserDriver: chrome/ ie/ firefox driver
- uploadFiles
  - Image files
  - AutoIT files
- downloadFiles
- browserLogs
  - Firefox/ Chrome/….logs
- reportHTML
- autoITConfig
- …..

Trainer: LE DAC MINH
Phone: 0966152432
Email: ledacminh0305@gmail.com

SELENIUM PAGE FACTORY PAGE GENERATOR MANAGER

- **FindBy/ FindBys/ FindAll/ PageFactory/ CacheLookup**

```
import org.openqa.selenium.support.FindBy;
import org.openqa.selenium.support.FindBys;
import org.openqa.selenium.support.FindAll;
import org.openqa.selenium.support.PageFactory;
import org.openqa.selenium.support.CacheLookup;
```

- **FindBy:** Using for locating 1 or many elements to satisfy the condition.

```
@FindBy(how = How.XPATH, using = "//form[@name='frmLogin']")
private WebElement loginForm;

@FindBy(how = How.NAME, using = "uid")
// @FindBy(name = "uid")
private WebElement userIDTextbox;

@FindBy(how = How.CSS, using = "input[name='btnLogin']")
// @FindBy(css = "input[name='btnLogin']")
private List <WebElement> loginButton;
```

- **FindBys**: Using for locating 1 or many elements to satisfy the conditions (AND) 1 in 2 incorrect will not be found.

```
@FindBys({
        @FindBy(how = How.NAME, using = "uid"),
        @FindBy(how = How.NAME, using = "password")
})
private List <WebElement> bothCriteria;
```

- **FindAll**: Using for locating 1 or many elements to satisfy the conditions (OR) 1 in 2 correct will be found.

```
@FindAll({
            @FindBy(how = How.NAME, using = "uid"),
            @FindBy(how = How.NAME, using = "password")
        })
private List<WebElement> eitherCriterion;
```

- PageFactory.initElements(driver,this);
    - When initialing - WebElements will have not been initialed. The initial only creates the links between elements and locators (FindBy). Once WebElements is used then Elements will be found based on WebElements declared above.

Trainer: LE DAC MINH
Phone: 0966152432
Email: ledacminh0305@gmail.com

    - If WebElement wants to be used many times on one page that will be found many times. If we only want to be found one time then use again - using the annotation @CacheLookup (increasing the performance).

- **CacheLookup**
    - It allows the element will be found one time (in case of the element is always permanent on a page.
- **How**
    - CLASS_NAME
    - CSS
    - ID
    - ID_OR_NAME
    - LINK_TEXT
    - NAME
    - PARTIAL_LINK_TEXT
    - TAG_NAME
    - XPATH

- **Advantaged**
- **Disadvantaged**

DYNAMIC LOCATOR

- **By Locator Refactor**
    - Currently, BasePage is supporting Xpath Locator - Requiring is to be used with others locators of selenium.
    - Selenium Locator has 8 types.
        - ID
        - Classname
        - Name
        - Tagname
        - LinkText
        - PartialLinkText
        - Css
        - Xpath

- **Xpath vs Css**
- **findElement**

Trainer: LE DAC MINH
Phone: 0966152432
Email: ledacminh0305@gmail.com

```
public By getByXpath(String locator) {
    return By.xpath(locator);
}

public WebElement getWebElement(WebDriver driver, String locator) {
    return driver.findElement(getByXpath(locator));
}

public List<WebElement> getWebElements(WebDriver driver, String locator) {
    return driver.findElements(getByXpath(locator));
}

public void clickToElement(WebDriver driver, String locator) {
    getWebElement(driver, locator).click();
}

public void sendkeyToElement(WebDriver driver, String locator, String value) {
    getWebElement(driver, locator).clear();
    getWebElement(driver, locator).sendKeys(value);
}
```

- **Performance/ Function**
  - By.ID/ Class/ Name
    - Run fastest
  - By.CSS
    - run faster Xpath
    - work with IE browser better than Xpath
    - don't work with text (dynamic locator)
    - apply for basic project/ HTML defined clearly = > Good
  - By. Xpath
    - runs slower than other locators (id/ class/ name/ CSS)
    - cover all locators: id/ class/ name/ CSS/ link..
    - can reverse to parent/ ancestor node
    - handle complex cases: data-table/ data-grid (Build Framework)
  - Compare
    - There is nothing to be best
    - we can use both of them and base on the context

- **Issue**
  - We have many locators having the same format - Only the value is the different
    - //a[text()=' Manager']
    - //a[text()= 'New Customer']
    - //a[text()=' Edit Customer']
  - Using String. format and %s
    - https://gpcoder.com/2352-huong-dan-su-dung-string-format-trong-java/#Dinh_dang_kieu_chuoi_String

Trainer: LE DAC MINH
Phone: 0966152432
Email: ledacminh0305@gmail.com

```
public class DynamicLocator {

	public static void main(String[] args) {
		String NEW_ACCOUNT_LINK = "//a[text()='New Account']";
		String DEPOSIT_LINK = "//a[text()='Deposit']";
		String NEW_CUSTOMER_LINK = "//a[text()='New Customer']";
		String HOMEPAGE_LINK = "//a[text()='Manager']";
		String DYNAMIC_LINK_1_PARAM = "//a[text()='%s']";
		String DYNAMIC_LINK_2_PARAM = "//a[text()='%s']//a[text()='%s']";
		String DYNAMIC_LINK_3_PARAM = "//a[text()='%s']//a[text()='%s']//a[text()='%s']";
		String DYNAMIC_EDIT_TABLE = "//td[@data-key='females' and
text()='%s']/following-sibling::td[@data-key='country' and text()='%s']/following-sibling::td[@data-key='males' and
text()='%s']/preceding-sibling::td[@class='qgrd-actions']/button[@class='qgrd-edit-row-btn']";
		String DYNAMIC_DELETE_TABLE = "//td[@data-key='females' and
text()='%s']/following-sibling::td[@data-key='country' and text()='%s']/following-sibling::td[@data-key='males' and
text()='%s']/preceding-sibling::td[@class='qgrd-actions']/button[@class='qgrd-remove-row-btn']";

		clickToElement(NEW_CUSTOMER_LINK);
		clickToElement(NEW_ACCOUNT_LINK);
		clickToElement(DEPOSIT_LINK);
		clickToElement(HOMEPAGE_LINK);

		clickToElement(DYNAMIC_LINK_1_PARAM, "New Account");
		clickToElement(DYNAMIC_LINK_1_PARAM, "New Customer");
		clickToElement(DYNAMIC_LINK_1_PARAM, "Manager");
		clickToElement(DYNAMIC_LINK_1_PARAM, "Deposit");

		clickToElement(DYNAMIC_LINK_2_PARAM, "Female", "Afghanistan");
		clickToElement(DYNAMIC_LINK_2_PARAM, "Male", "Albania");

		clickToElement(DYNAMIC_LINK_3_PARAM, "Female", "Afghanistan", "Male");
		clickToElement(DYNAMIC_LINK_3_PARAM, "Male", "Albania", "Female");

		clickToElement(DYNAMIC_DELETE_TABLE, "12253515", "AFRICA", "12599691");
		clickToElement(DYNAMIC_DELETE_TABLE, "384187", "Afghanistan", "407124");

		clickToElement(DYNAMIC_EDIT_TABLE, "12253515", "AFRICA", "12599691");
		clickToElement(DYNAMIC_EDIT_TABLE, "384187", "Afghanistan", "407124");
	}

	public static void clickToElement(String pageName) {
		System.out.println(pageName);
```

Trainer: LE DAC MINH
Phone: 0966152432
Email: ledacminh0305@gmail.com

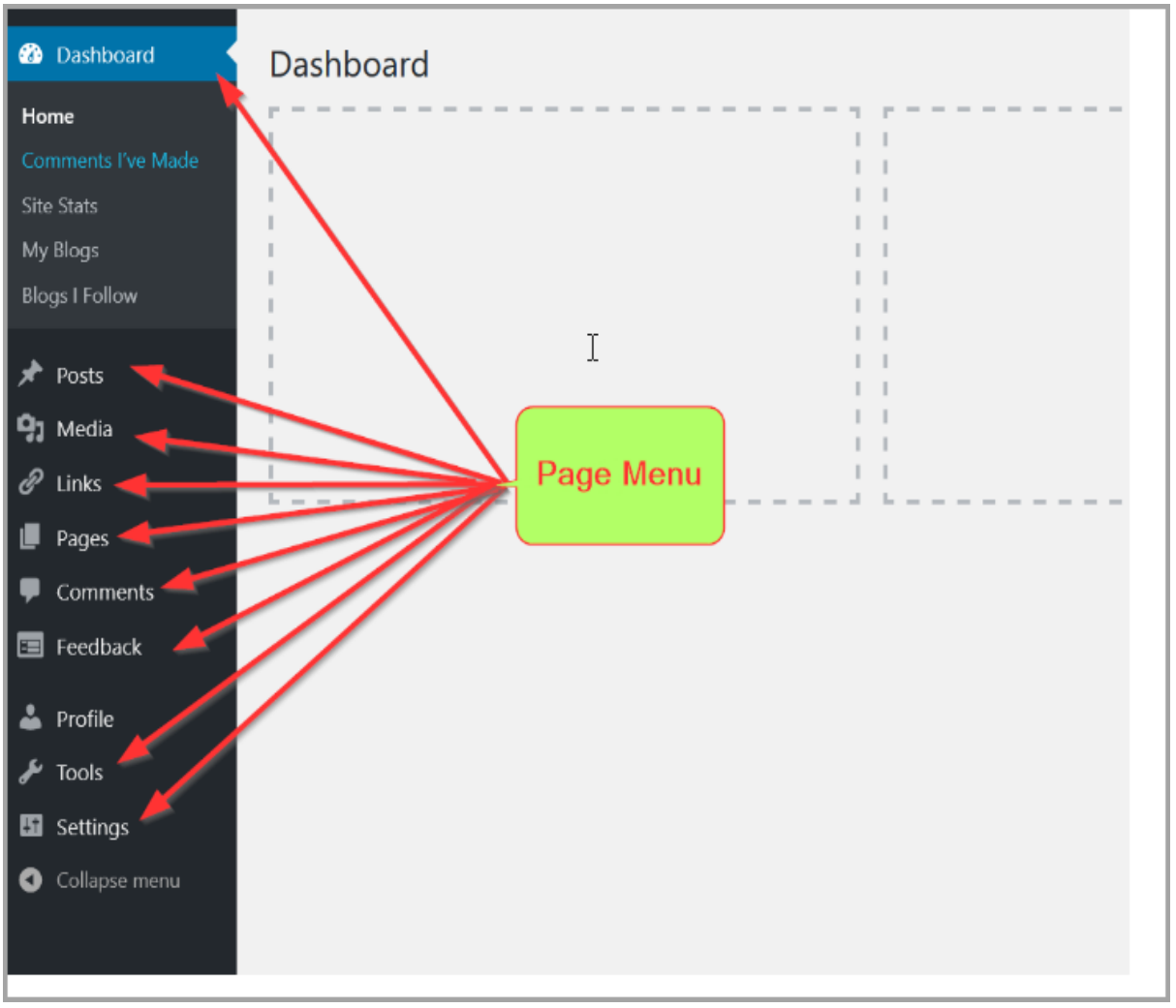
```
	}

	public static void clickToElement(String pageName, String dynamicValue) {
		System.out.println(String.format(pageName, dynamicValue));
	}

	public static void clickToElement(String pageName, String dynamicValue_01, String dynamicValue_02) {
		System.out.println(String.format(pageName, dynamicValue_01, dynamicValue_02));
	}

	public static void clickToElement(String pageName, String dynamicValue_01, String dynamicValue_02, String
dynamicValue_03) {
		System.out.println(String.format(pageName, dynamicValue_01, dynamicValue_02, dynamicValue_03));
	}

	public static void clickToElement(String locator, String... dynamicValue) {
		System.out.println("Click to Element: " + String.format(locator, (Object[]) dynamicValue));
	}
}
```

- **Polymorphism & Rest Parameter**
  - The number of arguments is the same, but the type must be different.
  - The number of arguments is different, and the type is not necessary differently.

```
public static void clickToElement(String pageName) {
public static void clickToElement(String pageName, String dynamicValue) {
public static void clickToElement(String pageName, String dynamicValue_01, String dynamicValue_02) {
public static void clickToElement(String pageName, String dynamicValue_01, String dynamicValue_02, String dynamicValue_03) {
public static void clickToElement(String locator, String... dynamicValue) {
```

Trainer: LE DAC MINH
Phone: 0966152432
Email: ledacminh0305@gmail.com

- **Handle Dynamic Page**
  - Apply for Menu/ Footer/ Header



- **Global Constants Class**
  - This class includes all of the data used for the whole system and it can be accessed at any position in the framework.

Trainer: LE DAC MINH
Phone: 0966152432
Email: ledacminh0305@gmail.com

```
// System Infor
public static final String PROJECT_PATH = System.getProperty("user.dir");
public static final String OS_NAME = System.getProperty("os.name");

// App Infor User
public static final String DEV_USER_URL = "http://dev.techpanda.org/";
public static final String STAGING_USER_URL = "http://staging.techpanda.org/";
public static final String LIVE_USER_URL = "http://live.techpanda.org/";

// App Infor Admin
public static final String DEV_ADMIN_URL = "http://dev.techpanda.org/index.php/backendlogin";
public static final String STAGING_ADMIN_URL = "http://staging.techpanda.org/index.php/backendlogin";
public static final String LIVE_ADMIN_URL = "http://live.techpanda.org/index.php/backendlogin";

public static final String ADMIN_USERNAME = "user01";
public static final String ADMIN_PASSWORD = "guru99com";

// Wait Infor
public static final long SHORT_TIMEOUT = 10;
public static final long LONG_TIMEOUT = 30;

// Download/ Upload file
public static final String UPLOAD_PATH = PROJECT_PATH + "/uploadFiles/";
public static final String DOWNLOAD_PATH = PROJECT_PATH + "/downloadFiles/";

// Retry Case Failed
public static final int RETRY_NUMBER = 3;

// Browser Logs/ Extension
public static final String BROWSER_LOG_PATH = PROJECT_PATH + "/browserLogs/";
public static final String BROWSER_EXTENSION_PATH = PROJECT_PATH + "/browserExtensions/";

// HTML Report Folder
public static final String REPORTNG_PATH = PROJECT_PATH + "/htmlReportNG/";
```

ASSERT vs VERIFY

- **Arrange - Action - Assert**
  - Arrange (Pre-Condition) - set up the testing objects and prepare the prerequisites for your test.
    - Arrange inputs and targets
    - Arrange steps should set up the test case
    - Does the test require any objects or extraordinary things?
    - Does it need to prep a database?
    - Does it need to log into a web app?
    - Handle all of these operations at the start of the test cases.
  - Action - perform the actual work of the test
    - Act steps should cover the main thing to be tested

Trainer: LE DAC MINH
Phone: 0966152432
Email: ledacminh0305@gmail.com

        - This could be calling a function or method. calling a REST API, or interacting with a web page.
        - Keep actions focused on the target behavior
    - Assert - verify the result
        - Act steps should elicit some sort of response
        - Assert steps verify the goodness or badness of the response
        - Assert are as simple as checking numeric or string values
        - Assertions will ultimately if the test passes or fails.
    - Behavior Driven Development follows the Arrange/ Act/ Assert pattern by another name
        - Given (Arrange) - When (Action) - Then (Verify)
- **Pre-condition**
    - Init Browser/ Driver
    - Connect to DB (Database)
    - Init Data Test
    - Init Page Object
- **Test case (s)**
    - Actions
        - Open page link
        - Click to button/ checkbox…(elements)
        - Select dropdown
        - …..
    - Verify
        - Checkpoint (Input = Output)
            - assertEquals (expected, actual)
            - Condition (true/ false)
            - Condition (null, not null)
            - Condition (instance/ data type/ …)
- **Post-Condition**
    - Clean/ Clear Browser/ Driver (executable)
    - Disconnect to DB

HARD ASSERT (TestNG)

- **The function to Assert**
    - Assert. assertTrue(condition false)
    - Assert. assertFalse(condition false)
    - Assert. assertEquals(actual type, expected type)tot
        - String - String
        - boolean - boolean
        - Array - Array

Trainer: LE DAC MINH
Phone: 0966152432
Email: ledacminh0305@gmail.com

        - Collection (ArrayList/ LinkedList/ Set/…)
    - import static org.testng.Assert.assertFalse;
        - assertFalse(....);
        - assertTrue(...);
        - assertEquals(...);
- **Best practice**
    - The function returns true/false we use assertTrue/ False
    - The function returns text we should use assertEqual:
        - getCurrentUrl/ getTitle/ getText/ getAttribue/ getSize/….
- **Assert**: Stop the following steps if the current step is failed
    - Advantaged
        - getting the result fast
        - suitable with levels: Unit/ API Testing
    - Disadvantaged
        - Must have to rerun many times through test cases/ steps or must have to retest many times step/ case.
        - No suitable with UI Testing

SOFT ASSERT (TestNG)

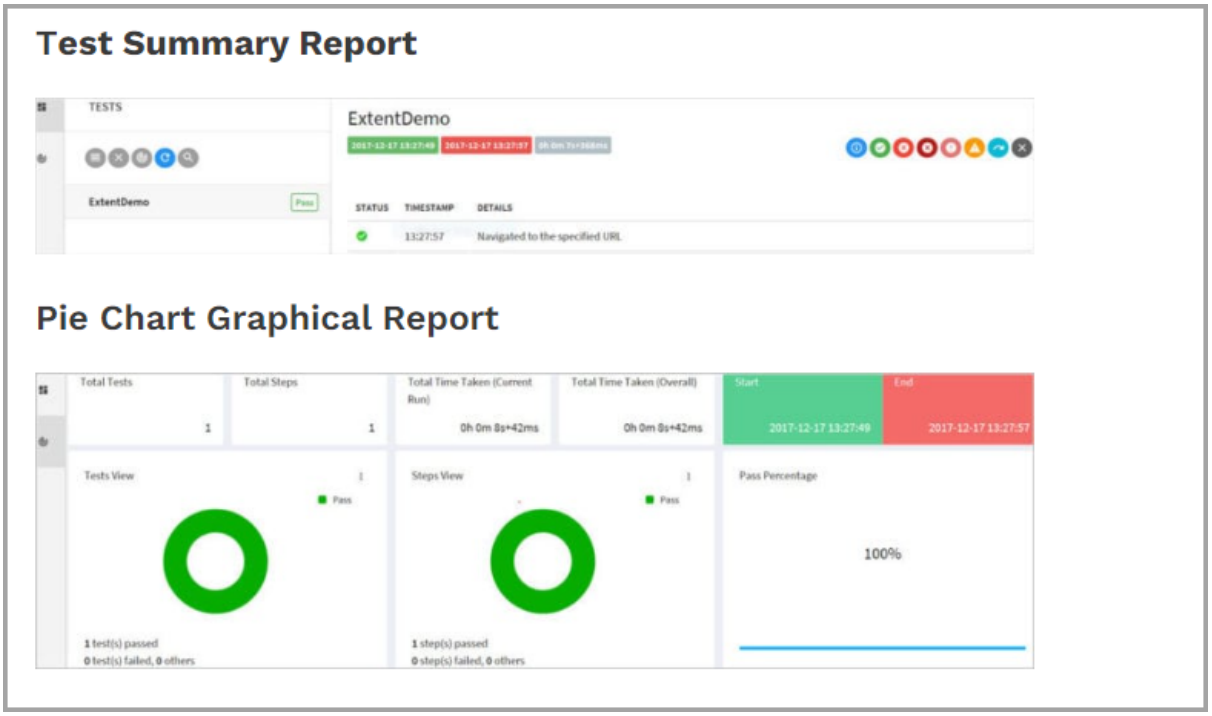
- To satisfy the condition if there is a step getting failure the remaining steps continue running to the end of the test case.
- Using assertAll hard to investigate - the only reference to the step assertAll - no reference to individual step error.

- Verify (Custom Hard Assert)
    - Advantaged:
        - runs all of the test cases/ steps - when having resulted - known of how many test cases got failed in a test case.
        - suitable with UI/ E2E testing
    - Disadvantaged
        - giving the result slow

Trainer: LE DAC MINH
Phone: 0966152432
Email: ledacminh0305@gmail.com

```java
@Test
public void TC_01_Assert() {
		System.out.println("TC_01 - Step 01: Open New Customer Page");
		newCustomerPage = homePage.openNewCustomerPage(driver);

		System.out.println("TC_01 - Step 02: Verify New Customer page displayed");
		Assert.assertTrue(true);

		System.out.println("TC_01 - Step 03: Verify New Customer form not displayed");
		Assert.assertTrue(false);

		System.out.println("TC_01 - Step 04: Verify Home Page not displayed");
		Assert.assertTrue(true);
}

@Test
public void TC_02_Verify() {
		System.out.println("TC_02 - Step 01: Open New Customer Page");
		newCustomerPage = homePage.openNewCustomerPage(driver);

		System.out.println("TC_02 - Step 02: Verify New Customer page displayed");
		verifyTrue(true);

		System.out.println("TC_02 - Step 03: Verify New Customer form not displayed");
		verifyTrue(false);

		System.out.println("TC_02 - Step 04: Verify Home Page not displayed");
		verifyTrue(true);
}
```

EXTEND REPORT



Trainer: LE DAC MINH
Phone: 0966152432
Email: ledacminh0305@gmail.com