Test description & steps

Test verifies error message when credit card gets declined.
If error message text is correct than test will pass.

Steps:

- Get to home page
- Login
- Choose flight airports, dates and passengers
- Choose first available date for flight and proceed to payment page
- Populate all the required fields
- Populate invalid card details
- After clicking on [PayNow] button given error is validated

Technical details

There are Cucumber and Junit versions of this script, both share same PageObject classes, and are created as part of a Maven project.

Programming language used is Java, developed in Eclipse IDE.

Page Object Pattern/Page Object Factory class design was used for grouping of the elements required.

As for element locators, mostly XPATH is used, and sometimes IDs and names if they were present.

Reporting is implemented for Cucumber runs ("Pretty" report, **path /site/cucumber-html-report**).

Since this is a Maven project, **pom.xml** file contain configuration details required to run these tests.

Test Environment details:

Google Chrome Version 64.0.3282.186 (Official Build) (64-bit)
Eclipse Oxygen.2 Release (4.7.2)
Windows 10 (64-bit)

Code

Methods implemented are mostly based on loops and conditions since there was a lot of conditional elements on the page which were also quite similar one to another, so just one locator would not do the trick.
Often getting a list of elements is used.

EXAMPLE METHOD

```
	public void selectFirstAvailableFlight() throws Exception {
		Thread.sleep(500);

		boolean flightFound = false;
		int position = 2;
		while (!flightFound) {
			String xpathAvailableFlight =
String.format(availableFlight_XPATH, position++);
			WebElement availableFlightDate =
driver.findElement(By.xpath(xpathAvailableFlight));
		availableFlightDate.click();
			Thread.sleep(1000);
			List<WebElement> availableFlights = driver.findElements(By
					.xpath(availableFlights_XPATH));
			for (int i = 0; i < availableFlights.size(); i++) {
				if
(ExpectedConditions.elementToBeClickable(availableFlights.get(i)) != null) {
					availableFlights.get(i).click();
					Thread.sleep(500);
					flightFound = true;
					break;
				}
			}
		}
	}
```

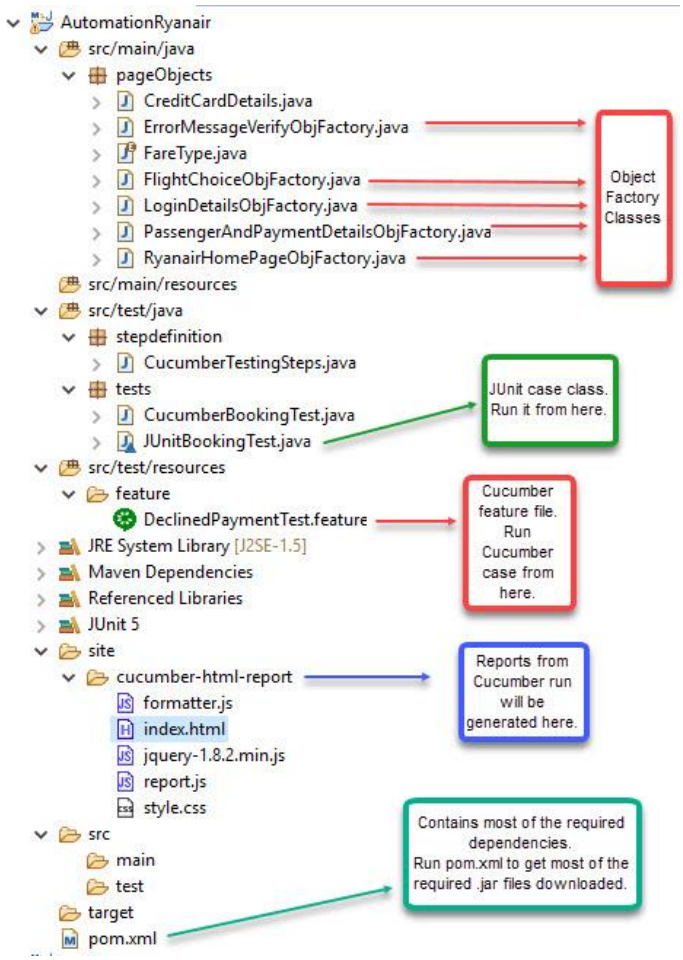
Method above will find the first available flight after we have chosen all the details on the home page.

It will go through date slider to find first available date. This is why String is used:
`String.format(availableFlight_XPATH, position++);.`

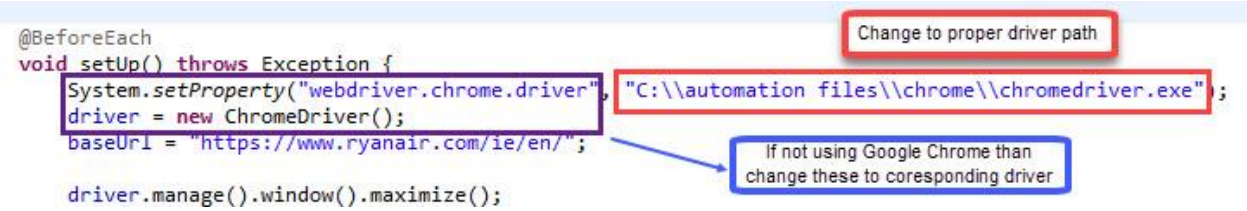
`position` in this case would replace the **%s** part of the original availableFlight_XPATH String in each iteration with corresponding position, until date on slider have valid flights.

When the date on slider is clicked, it will pull all the elements under it and verify if any of those can be clicked on. If that happens that means that valid flight is found and it will break out of the loop. If not, next date on the slider is clicked.

Project structure & how to run



Before the run update driver details in '*JUnitBookingTest.java*' and '*CucumberTestingSteps.java*'!



Run Junit version:

- '*JUnitBookingTest.java*' run as JUnit test.

Run Cucumber version:

- 'DeclinedPaymentTest.feature' run as Cucumber Feature.
- Report is saved in **index.html** file.
- *'CucumberBookingTest.java'* class is used for gluing the '*CucumberTestingSteps.java*' with Feature file.