UI AUTOMATION

Dissertation/Thesis submitted to the SASTRA Deemed to be University
in partial fulfilment of the requirements
for the award of the degree of

B. Tech. Electronics & Communication Engineering

Submitted by

Malavika Venkatanarayanan
(121004151)

June 2021

SCHOOL OF ELECTRICAL & ELECTRONICS ENGINEERING

THANJAVUR, TAMIL NADU, INDIA – 613 401

SASTRA

ENGINEERING · MANAGEMENT · LAW · SCIENCES · HUMANITIES · EDUCATION

DEEMED TO BE UNIVERSITY

(U/S 3 OF THE UGC ACT, 1956)

THINK MERIT | THINK TRANSPARENCY | THINK SASTRA

SCHOOL OF ELECTRICAL & ELECTRONICS ENGINEERING

THANJAVUR – 613 401

<u>Bonafide Certificate (Company)</u>

This is to certify that the thesis titled "**UI Automation**" submitted in partial fulfilment of the requirements for the award of the degree of B.Tech. Electronics & Communication Engineering to SASTRA Deemed to be University, is a bona-fide record of the work done by **Ms. Malavika Venkatanarayanan (Reg No: 121004151)** during the final semester of the academic year 2020-2021, while interning at the **Software Testing and Delivery Team (Fulfilment) of Jio Platforms Ltd**, under my supervision. This thesis has not formed the basis for the award of any degree, diploma, associateship, fellowship or other similar title to any candidate of any University.

Signature of Project Supervisor	: [illegible]
Name with Affiliation	: Mrityunjay Pandey
Date	:18th June 2021

SCHOOL OF ELECTRICAL & ELECTRONICS ENGINEERING

THANJAVUR – 613 401

Bonafide Certificate

This is to certify that the thesis titled "**UI Automation**" submitted in partial fulfilment of the requirements for the award of the degree of **B. Tech. Electronics & Communication Engineering** to the SASTRA Deemed to be University, is a bona-fide record of the work done by **Ms. Malavika Venkatanarayanan (Reg No: 121004151)** during the final semester of the academic year 2020-2021, in the **School of Electrical and Electronics Engineering**, under my supervision. This thesis has not formed the basis for the award of any degree,
diploma, associateship, fellowship or other similar title to any candidate of any University

Signature of Project Supervisor (Internal): Sri. Prasath R.K

Name with Affiliation **:** Sri. Prasath R.K
AP-III , SEEE, SASTRA University

Date **:** 27.05.2021

Project Viva-Voce held on 14/07/2021

EXAMINER-1 EXAMINER-11

SCHOOL OF ELECTRICAL & ELECTRONICS ENGINEERING

THANJAVUR – 613 401

Declaration

I declare that the thesis titled “**UI Automation”** submitted by me is an original work done by me under the guidance of **Mr.Mrityunjay Pandey , Jio Platforms Ltd** during my final semester of the academic year 2020-21, while interning at **Jio Platforms Ltd, Mumbai**. The work is original and wherever I have used materials from other sources, I have given due credit and cited them in the text of the thesis. This thesis has not formed the basis for the award of any degree, diploma, associate-ship, fellowship or other similar title to any candidate of any University.

`

Signature of Project Candidate	: **Malavika Venkatanarayanan**
Name of the candidate	**: Malavika Venkatanarayanan**
Register No	**: 121004151**
Date	: 27.05.2021

JIO PLATFORMS LIMITED

CIN: U72900GJ2019PLC110816

Certificate of Completion

TO WHOMEVER IT MAY CONCERN

This is to certify that **Miss. Malavika Venkatanarayanan** has completed the **"UI AUTOMATION"** project at **Jio Platforms Ltd** under the **Software Testing and Delivery Team (Fulfilment-Ericcson)** during the period **22/02/2021 to 28/05/2021** with my guidance successfully. She exceeded in completing all the tasks assigned to her within the period. We wish her success in her future endeavors.

Signature of the Supervisor: [illegible]

Name with Affiliation: Mrityunjay Pandey

Date: 02-07-2021

ACKNOWLEDGEMENTS

I would like to express my gratitude to Prof. **Dr. S. Vaidhyasubramaniam**, Vice Chancellor, and **Dr. S. Swaminathan,** Dean of Planning and Development, SASTRA Deemed to be University who provided all facilities and constant motivation throughout the course of my study. I extend my sincere thanks to Prof. **Dr. R Chandramouli**, Registrar, SASTRA Deemed to be University for providing the opportunity to pursue this project.

I extend my sincere thanks to **Dr. K Thenmozhi**, Dean (SEEE), **Dr. Sridhar K**, Associate Dean (ECE), **Dr. K.Vijayarekha**, Associate Dean (EEE), **Dr. R John Bosco Balaguru**, Dean (Sponsored Research) for their constant guidance both in and out of the classes.

I owe a debt of deepest gratitude to my internal guide **Sri. R.K Prasath** (AP-III/ECE/SEEE) and **Dr. Amirtharajan R.** (Professor/ECE/SEEE) for their valuable inputs with the report and their cooperation and constructive criticism throughout the duration of my project on the topic **"UI Automation"**. I take this opportunity to thank all my lecturers for introducing me to the wonderful subjects of Electronics and Communication Engineering.

I would like to thank **Mr. Mrityunjay Pandey** (Jio Platforms Ltd) and **Mr. Rajeev Si Kumar** (Jio Platforms Ltd) for giving me the opportunity to work under the OSS/BSS domain along with the team. It helped me learn a lot about software testing and delivery and yes, Software testing is not just for anybody. I would also like to thank **Ms. Priyal Mangala** for her special support and for constantly guiding me throughout the project.

Last but not the least, I would also like to thank my family, teachers and peers at SASTRA Deemed to be University for their constant cherishing and support which had led me to complete all the courses of my Bachelor's degree and this project successfully.

ABSTRACT

Keywords: BDD, Python, Selenium, Automation, Testing, OSS/BSS

Abstract:

Software automation is an important tool in the current era. With the ever-increasing customers and their needs and requirement, it is only fair that the Telecom companies such as Jio and Airtel keep updating their requirements and test the software. Using Python Programming language and Selenium packages one can create an excellent Web UI Testing Framework, allowing to run test scripts efficiently. Both Graphical User interface testing and API-driven testing are done to assert the software. The code driven testing is slowly amalgamated with the Behaviour development testing (BDD) using the Selenium library in python. This ensures efficiency and easy testing of software and lessens the maintenance hassles. The idea is to inculcate the code in BDD framework which has Automated acceptance testing with mapping rules.

The OSS/BSS side of the Telecom is invested in order fulfilment and assurance. Working in the Fulfilment side of Jio Pvt Ltd particularly in the Software Testing Team, I worked on the UI Automation testing along with my team.

Register No:	**Name**
121004151	Malavika Venkatanarayanan

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

BDD	Behaviour Driven Development
TDD	Test Driven Development
OSS	Operating Support Subsystem
BSS	Business Support Subsystem
GSM	global System for Mobile Communication
FCAPS	Fault, Configuration, Accounting, Performance and Security
CRM	Customer Relationship Management
BPM	Business Process Management
ITU	International Telecommunications Union
GB	Giga Byte

CHAPTER 1
INTRODUCTION

OSS/BSS is the part of Telecom that deals with the business and customer services of the industry. It is the holy grail of service providers. The basic idea behind the entire organization is to provide service at any time, anywhere over any network. Thus, to achieve these goals, a proper organization and systems are required. As times keep changing, new technologies force the designers to redesign all part of their networks. OSS/BSS is a part of the initially introduced GSM.

1.1 OSS

The Operations Support Systems are networks processing systems used to manage the communications networks. The OSS consists of Order management, Network security, Inventory management and Network Operations. The operating support subsystem basically fulfills an order that is placed. The support management functions of OSS include service provisioning , configuration and management.

They are designed to design, build, operate and maintain communications networks.

The key elements of OSS are:

- Processes
- Data
- Applications
- Technology

The OSS is connected to the Base station (BSC) and to the other equipment of the GSM.

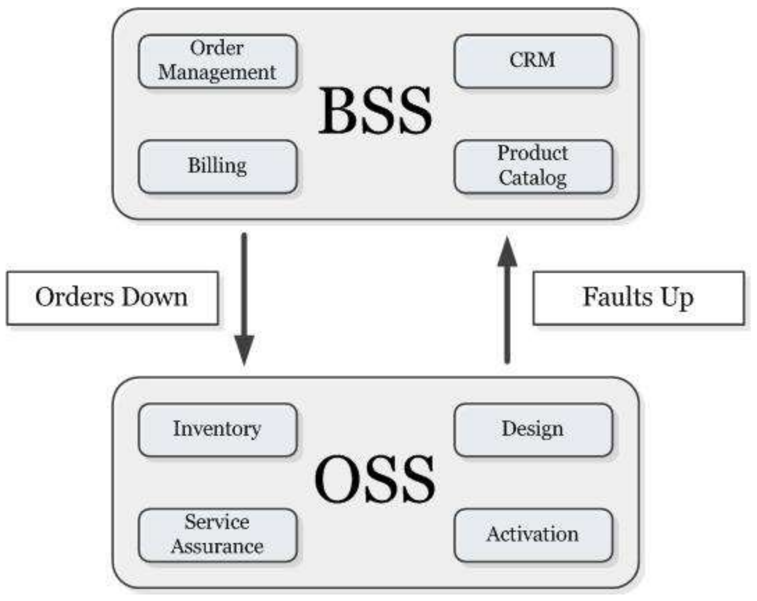


Fig 1.1 OSS/BSS Structure

In today's market and competition between the telecom companies, Customer Relationship Management (CRM) is an important part. It helps in answering customer queries and helps in service provisioning. There is Tier-1 and Tier-2 handling services in CRM. Tier-1 refers to solving a customer's issue in the first call and Tier-2 refers to a problem that involves more analysis and troubleshooting.

1.2 BSS

The BSS or the Business Support Subsystem deals with the accepting orders , payment issues etc. to keep the network operational and to keep it profitable. It is essentially a customer facing organ. It supports the following four processes:

- product management,
- order management,
- revenue management
- customer management.

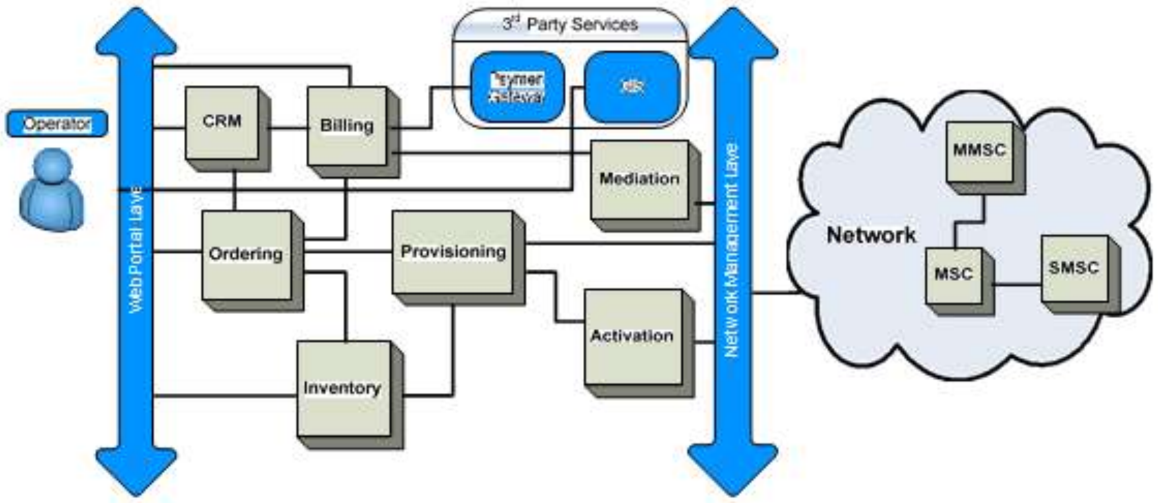


Fig 1.2. OSC

1.3 Telecommunications Management Network (TMN)

TMN was brought forth in the early days by the ITU-T (International Telecommunication Union. It is a basic integration of the OSS. Like all BSS/OSS Models all of it's processes also involve fulfillment and assurance.

The layers of TMN Model are given below:

- Business Management Level (BML)
- Service Management Level (SML)
- Network Management Level (NML)
- Element Management Level (EML)

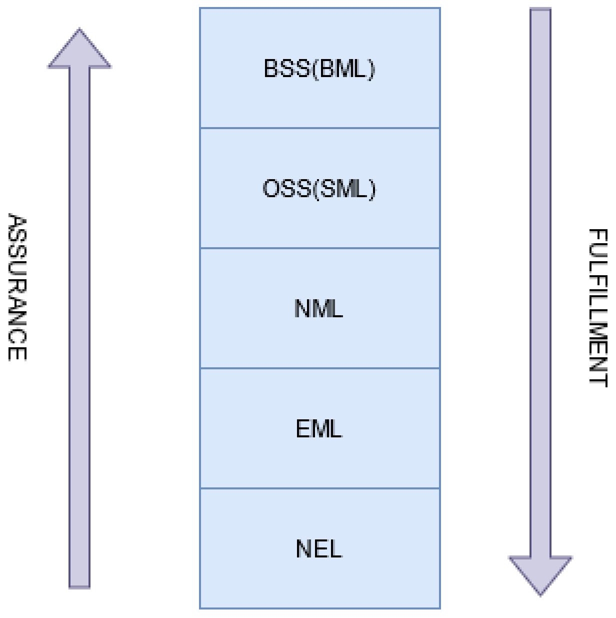


Fig 1.3 TMN

The Network management layer was further developed with the advent of the FCAPS Model [Fault, Configuration, Accounting, Performance and Security].

1.4 limitations of the system

The limitations of the traditional OSS/BSS system are the following**:**

- Point to point integration
  The integration is time consuming as it has to be incorporated in a lot of systems . Hence the system is very fragile.
- Tightly Coupled
  The system is not flexible enough. For example: The business systems are coupled with the software operations and hence a change in a business system requires a change in the entire organization.
- Integration with External Systems
  The system development is a huge process and so external help is required.
- Complex Transaction Management
  Even though the change needs to be incorporated to a single system, in reality and in actual process it needs to be incorporated to multiple systems thus it is time consuming and is a waste of energy.

1.5 OSS/BSS Integration-TM Forum

The integration of the OSS/BSS is an important process. OSS/BSS is a profit engine and it brings the buyers and the sellers together. Thus, OSS/BSS allow network operators to offer services effortlessly to a number of users. To resolve all the limitations of the traditional system integration, a new structure is introduced for efficient transaction as follows. The fig 1.2 shows the original integration. But due to the multiple limitations of the system, a different approach is taken here as shown in fig 1.3. As we all know BSS captures the order and the OSS basically fulfils it. The OSS and BSS together straddle the Service assurance, catalogues and management. Here we have the BPM (Business Process Management) layer. The components of this integration are explained below:

Business Process Management layer(BPM) : Business processes are set in the BPM, it is a part of the BSS structure

Application Queue: The business process workflow is started by placing a message or a request in the application queue. It provides asynchronous communication. The reason for that is that in telecommunication sector, due to the huge market and the time to collect data, synchronous communication is almost not possible especially in India.

Connector: The connector consists of the In Adapter, Out Adapter, webservice, transformer and the queue. The connector is basically invoked by placing a request.

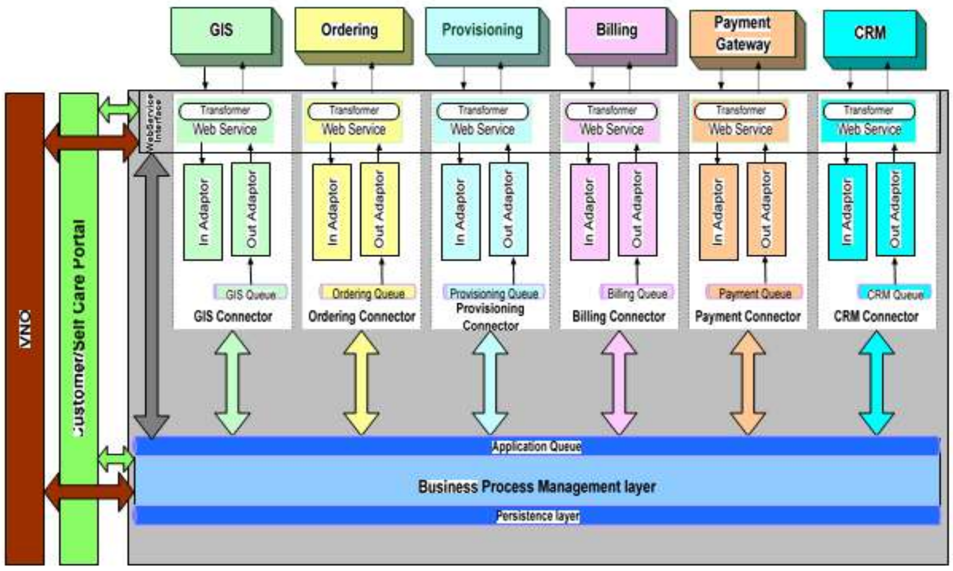


Fig 1.4 OSS/BSS Integration *(via Google)*

In Adapter: The In Adapter is a connector. It places a request in the app queue that invokes the workflow in the BPM.

Out Adapter: The Out Adapter is used to process the requests that are placed in the connector queue.

Transformer: It is used to convert the object of a third-party system into another and vice versa. It uses only transformation logic.

To make sense of the above logic, let us take an example: If the CRM of an account is to be updated then the request is placed by the In Adapter in the queue and the request proceeds to BPM to fulfil the order.

1.6 Telecommunication Management Forum

TM Forum introduces frameworks in OSS and BSS.
A set of models that provide standardized approaches are called Frameworks.
Frameworks includes:

- Information model
- Process model
- Application model

Application model (the Telecom Applications Map or TAM) – It helps in providing consistency and compatibility between the customers and the sellers.
Process model (the enhanced Telecom Operation Map, or eTOM) – The ETOM provides a common language or a path between the different layers
Information model (the Shared Information / Data model, or SID) – defines the essential entities, relationships and attributes of data objects widespread in telecommunications sector . It also provides an established language for use by OSS developers / integrators.

1.7 Business Process Framework (eTOM)

This level of standardization aims to simplify the lines of communication between service providers and associated systems integrators.
It covers the following customer-centric flows :

- Request to Answer (R2A)
- Order to Payment (O2P)
- Problem to Solution (P2S)

Additionally, we also have the following :

- O2A- Ordered to activate
- P2B-Plan to build
- U2C- Users to cash

Here generally we work with granite, siteforge and ordercare. Granite is used for inventory and siteforge helps in workflow. All flows like TFN are done using these platforms.

The Business Process Framework model consists of processes at five levels as follows:

- Infrastructure, Strategy and Product
- Network Operations,
- Level-3,
- Level-4.

The Business Process Framework (eTOM) model consists of rows and columns(refer fig 1.5), which denote very specific business processes as follows:

- The top row includes customer activities such as marketing,
- While the bottom row includes supplier and support activities.

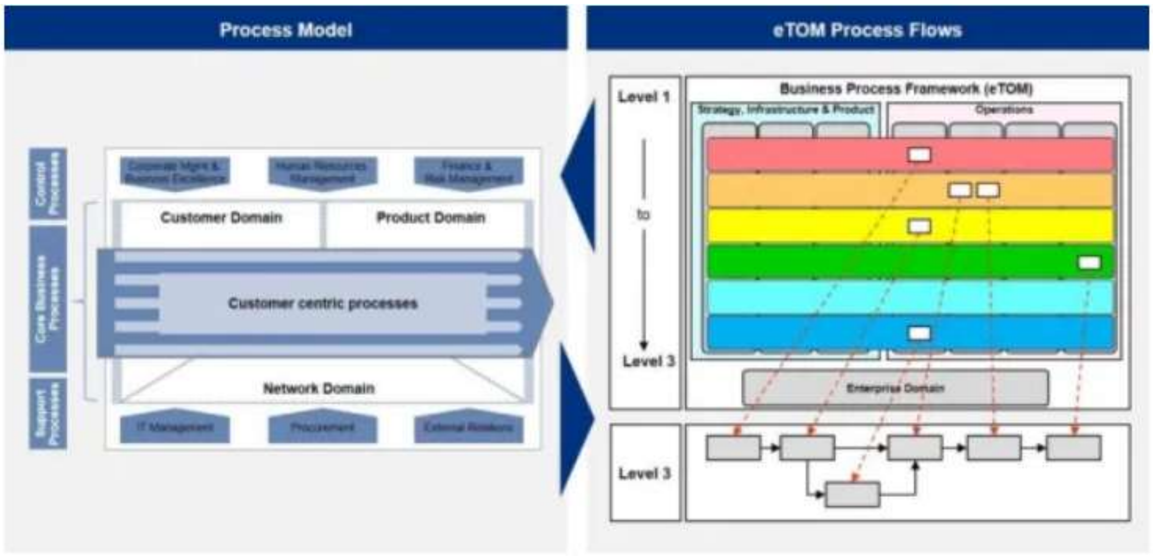


Fig 1.5 TM Framework -process model *(via Google)*

One brilliant feature of OSS/BSS structures is that they are capable of collecting an unbelievable amount of data about networks, services , service providers and customers. But like everything, the cost is heavy : with a huge amount of data, it is challenging to figure out how to measure and use them. Here, TM Forum has lent us a helping hand by compiling GB988.

1.8 OSS/BSS STRUCTURE:

The main components of the OSS/BSS structure are order fulfilment and assurance.
The OSS/BSS straddle both of these functions. My work is in the service fulfilment area, particularly in the software testing and development team. The fig 1.6 below shows the three main sectors of OSS/BSS.

SERVICE FULFILLMENT

Fulfilment is responsible for making services available to the users and for fulfilling their requirements. It is a form of customer service. To achieve these, service fulfillment platforms take the following into consideration:

a. Data transparency
b. Process automation
c. Management of Inventory
d. Asset monetization

SERVICE ASSURANCE

Service assurance is defined as those processes provided by a Communications Service Provider (CSP) to secure those services offered over networks meet a pre-established quality of service for an ideal subscriber experience. It includes fault management and performance management.

BILLING AND CUSTOMER CARE

Billing is the monetary handling part of the company. It also involves customer care services that is sometimes synchronous. It helps the customers to trust the company's services to be in such a close contact to the service managing structure

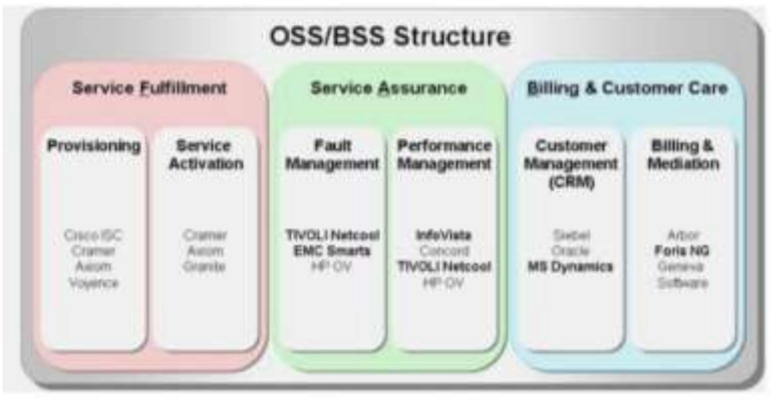


Fig 1.6. OSS/BSS Integration *(via Google)*

CHAPTER 2

SOFTWARE TESTING-BDD

2.1 BDD

Behavior Driven Development is a branch of TDD. BDD basically uses Ubiquitous language (English) descriptions of the user requirements of the software. This type of testing is mainly focused on the business value of the software rather than the software goals. It’s swift and easy. Problems that take 6 months to implement can be implemented in hours. BDD encourages collaboration between the developers and business participants. BDD is where all the participants of a team can communicate to understand the requirements of the order/project. The need for the project is specified and naturally the developer and the testers create a venture to fulfill the requirements.

BDD is a testing practice that follows the ideas of specification by example. It explains the behavior of an application in a very simple focused way. It allows to create multiple applications that are easy to maintain.

The user requirements are written in simple English using the language Gherkin..

The advantages of using this framework are :

i. Strong collaboration
ii. Increases business value
iii. Ubiquitous language
iv. increases the developer’s confidence
v. Lower costs

vi. Continuous testing

Here, BDD is implemented in Python in the PyCharm IDE.

2.2 Python Testing Framework

Python testing framework is a dynamic framework based on Python, known for its ease of use in web development and test automation. The various frameworks: Behave, Lettuce, Robot, raddish. Behave is widely used in BDD Testing. It fully supports the Gherkin language. BDD is implemented in Python using PyCharm

The packages used are: Selenium and behave. The Selenium Web driver is downloaded for all the browsers The folder structure typically consists of a feature folder and a steps folder within it.

The feature files are written under the Feature folder (.feature format) and the corresponding steps file are written in steps folder (.py format). The feature files(.feature) must be under the features directory and the step files (.py) must be under the steps directory . Fig 2.1 shows the folder format.

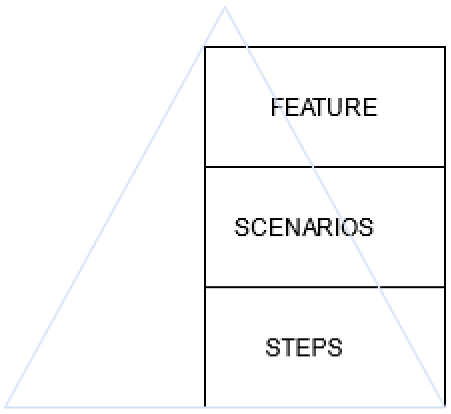


Fig 2.1 Folder Hierarchy

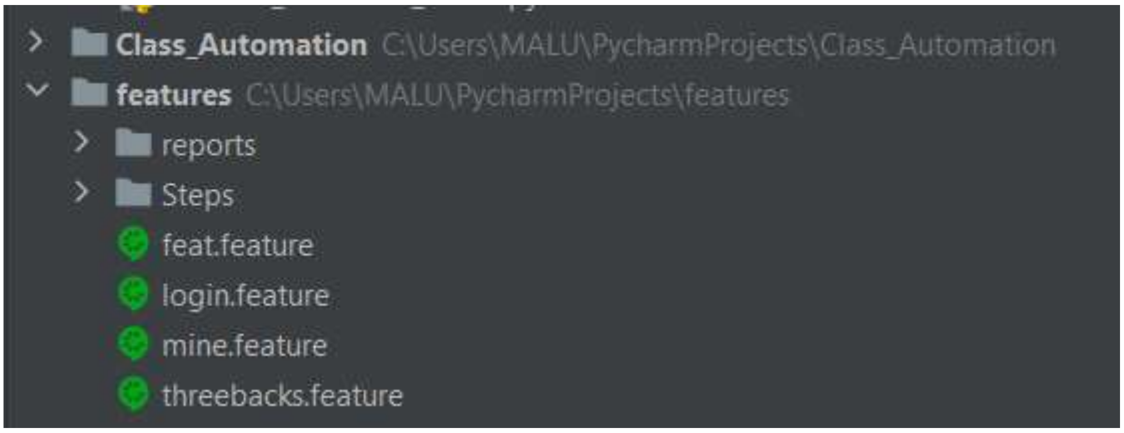

Class_Automation C:\Users\MALU\PycharmProjects\Class_Automation
features C:\Users\MALU\PycharmProjects\features
reports
Steps
feat.feature
login.feature
mine.feature
threebacks.feature

Fig 2.2 Folder Structure

2.3 Feature File and Gherkin Keywords

The feature file is a text file which uses Gherkin keywords to describe the requirements. The feature file consists of various scenarios to be implemented. The common Gherkin keywords are:

- Feature ,
- Rule,
- Scenario,
- Scenario Outline,
- Background,
- When,
- Then,
- And,
- Given,
- But

The General syntax for the Feature file is as follows:

```
Scenario: Title/Short Description
Given [A Precondition]
When [Some Event]
Then [Some Outcome]
```

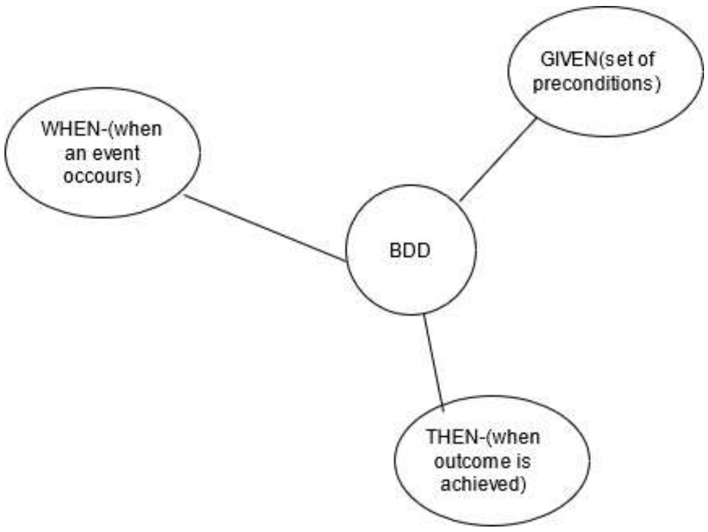
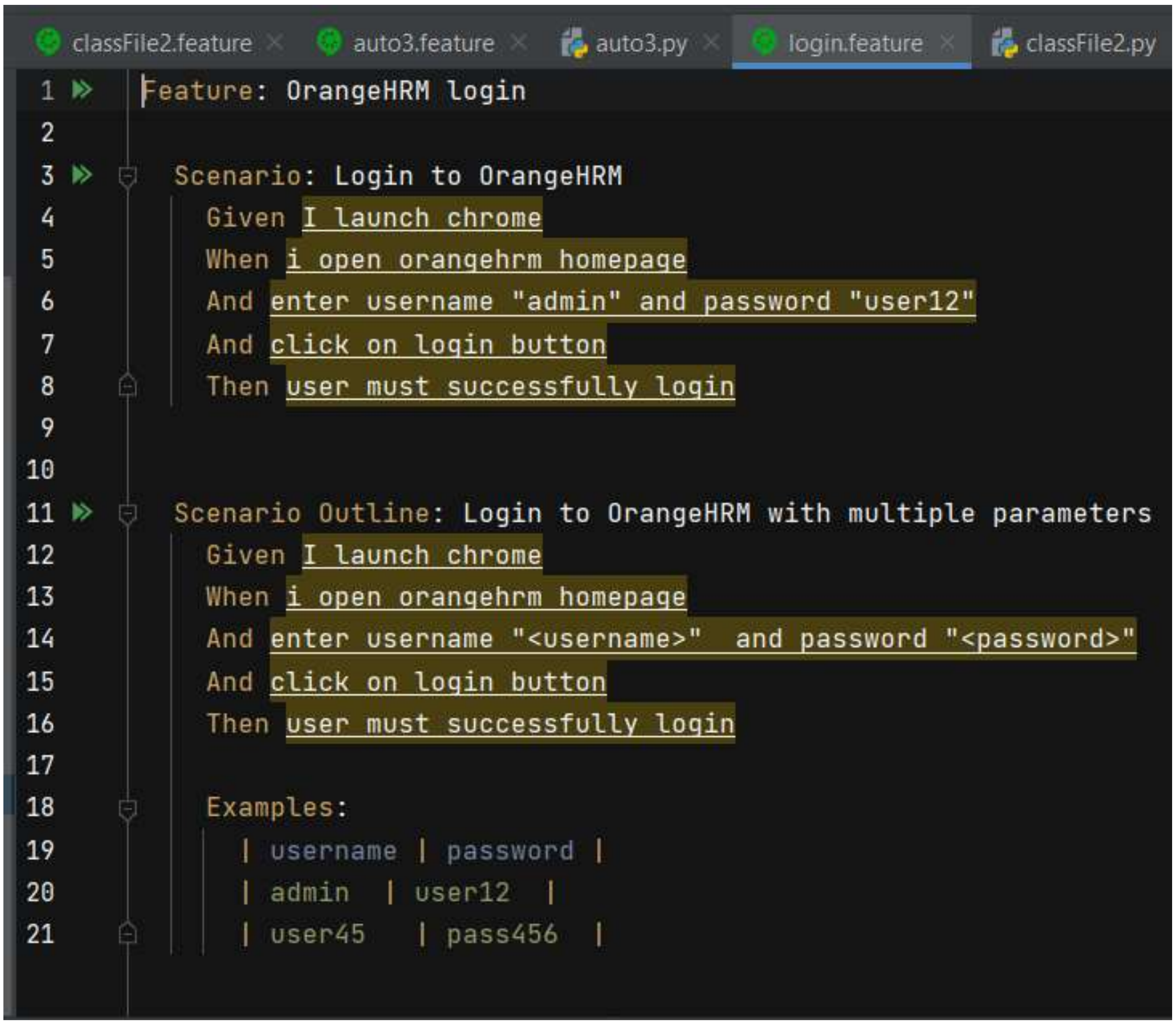

GIVEN(set of preconditions)
WHEN-(when an event occours)
BDD
THEN-(when outcome is achieved)

Fig 2.3 Gherkin Keywords

GIVEN basically puts the opponent in a known state, in a very familiar environment. WHEN takes the key actions and THEN is specified to observe the outcome



```
Feature: OrangeHRM login

  Scenario: Login to OrangeHRM
    Given I launch chrome
    When i open orangehrm homepage
    And enter username "admin" and password "user12"
    And click on login button
    Then user must successfully login


  Scenario Outline: Login to OrangeHRM with multiple parameters
    Given I launch chrome
    When i open orangehrm homepage
    And enter username "<username>"  and password "<password>"
    And click on login button
    Then user must successfully login

    Examples:
      | username | password |
      | admin    | user12   |
      | user45   | pass456  |
```

Fig 2.4 Selenium BDD Sample structure

Multiple inputs are given under Scenario Outline. The inputs are taken from the feature file and are passed as arguments in the .py file. The common steps are written under Background

2.4 Execution

A very simple way to sum up the entire process is as follows:

- Specify and write the tests and watch them fail
- Create feature to increase the pass rate
- Debug and refactor the code to make is better
- Keep repeating the cycle

The feature file is run in the terminal

```
behave <features/filename.feature>
example:  behave  search.feature
```

The above statements generates the code for the steps control file. The controls file defines the steps. The overall implementation process are as follows:

1. Create a Project
2. Create a Feature File
3. Configure the settings (behave.ini | setup.cfg)
4. Create Helper functions
5. Create environmental or the python file (environment.py)
6. Write Step definitions in .py file for each scenarios in feature file
7. Execution

2.5 Allure Report

Allure Report tool is a flexible test report tool. It basically provides the no of pass and fail test cases and gives us an overall outlook of the project .

Allure package needs to be installed as follows:

- pip install allure-pytest

To execute test cases & generate report files(.json):

- behave -f allure_behave.formatter:AllureFormatter -o reports/ feature

To Generate Allure report :

- allure serve reports/

The report files are stored in .json format under the reports directory and it is also generated as shown in fig 2.5. Here, we see the report for 4 test cases for an initial test case wherein only one has passed.

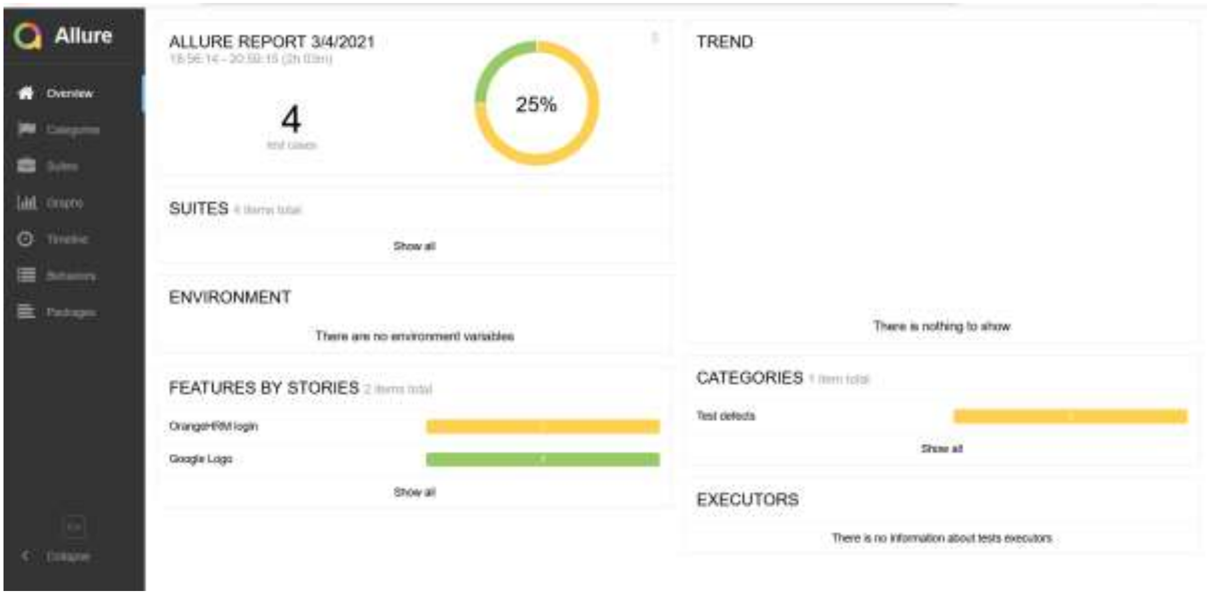


Fig 2.5 Allure Report for a Test Case

CHAPTER 3

EXPERIMENTAL WORK

3.1 UI Automation

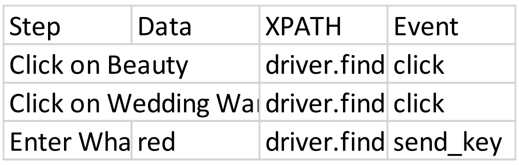
A User Interface (UI) is the front end of any application. It acts as a correspondence to the user. UI Automation Testing is to test whether the front end is up and running correctly.
Here the automation is performed using tools that facilitate easy flows without any manual intervention. We focus on every bit of the UI including but not limited to button clicks, data entries and links. Tests scripts are written for each and every test case and a report is generated at the end.In Manual testing, the testers create and the test the UI by hand which is time consuming and error prone. Automated play comes into play here. It is much better at detecting errors and the expense is less. It is also flexible and consumes less time. Automated testing is best suited for end -to-end testing, regression testing and Testing stable UI versions.

At Jio, the team tests internal web application. As a part of the software testing team, I worked on the UI Automation Project.
This framework enables engineers to perform codeless automation based on excel inputs.
This is quite a broad framework that can be used on normal websites as well.

3.2 DATA

The python code was written using the UI Automation Framework. The input is given in the form of csv files. The scenarios along with the flags (Y or N) are given as the data in the csv files. If the flag is ‘Y’ , the scenario gets executed and if it is ‘N’ it doesn’t get executed. A loop is run in the code to find out the particular website to be opened for the scenario. Any web driver can be installed (Firefox, Chrome, Safari). In this code, a Chrome web driver was used to open the website and the particular operation was performed.

So basically, we have two data files : Scenario.csv and vogue_scenario.csv (example files).
The Scenario.csv files have the scenarios and the flags. For example the first scenario is called ‘vogue_scenario’ and if the flag is ‘Y’, then the code will search for a file with the name ‘vogue_scenario’. The vogue_scenario.csv file contains the Step,Data,Xpath and the Event. Here the necessary steps are performed. All the functions to be performed are stored here on the csv. For example :

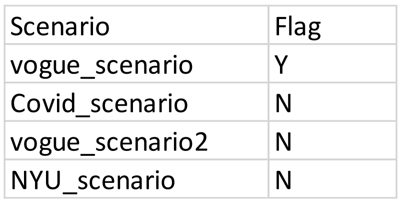


Step	Data	XPATH	Event
Click on Beauty		driver.find	click
Click on Wedding Wa		driver.find	click
Enter Wha	red	driver.find	send_key

Table 3.1 vogue_scenario.csv

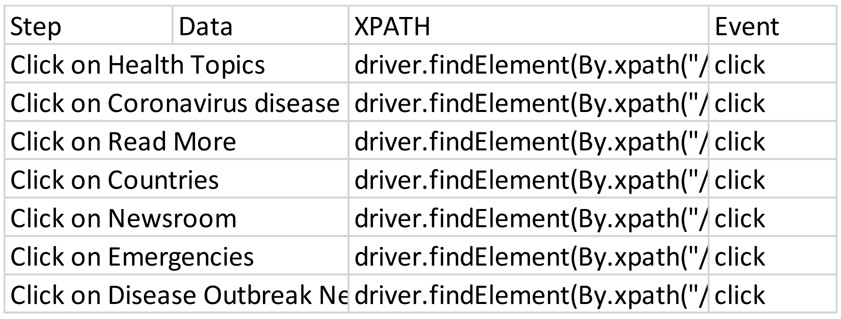
Here, the step says the function to be performed, Data is any information to be given as an input by the user and it is entered wherever required. The XPATH of the button or the link is given in XPATH and the Event is the method to be called. The entire csv file with the XPATH is found using the Chropath extension in Google Chrome.

The several definitions/methods in the code include: open_app, enable_log, takeScreenshots, pass_fail_watermark, testResultPngtoPdf, scenario_file_path, construct_xpath, run_script.



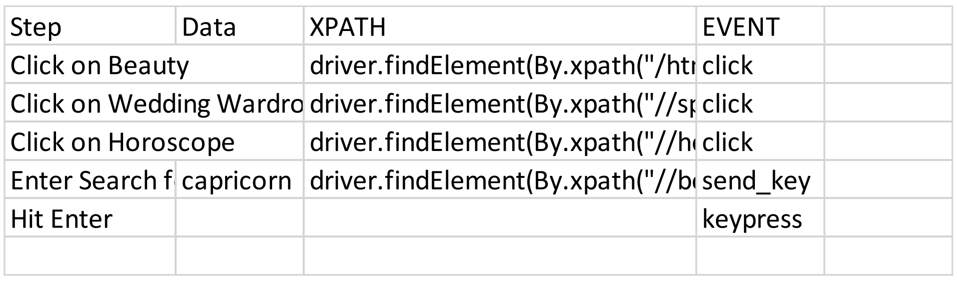
Scenario	Flag
vogue_scenario	Y
Covid_scenario	N
vogue_scenario2	N
NYU_scenario	N

Table 3.2 Scenario.csv



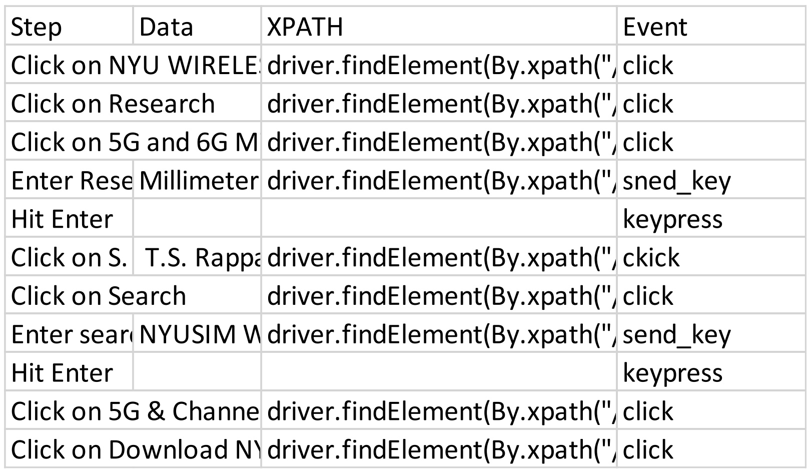
Step	Data	XPATH	Event
Click on Health Topics		driver.findElement(By.xpath("/	click
Click on Coronavirus disease		driver.findElement(By.xpath("/	click
Click on Read More		driver.findElement(By.xpath("/	click
Click on Countries		driver.findElement(By.xpath("/	click
Click on Newsroom		driver.findElement(By.xpath("/	click
Click on Emergencies		driver.findElement(By.xpath("/	click
Click on Disease Outbreak Ne		driver.findElement(By.xpath("/	click

Table 3.3 Covid_scenario.csv



Step	Data	XPATH	EVENT	
Click on Beauty		driver.findElement(By.xpath("/htr	click	
Click on Wedding Wardro		driver.findElement(By.xpath("//sp	click	
Click on Horoscope		driver.findElement(By.xpath("//h	click	
Enter Search f	capricorn	driver.findElement(By.xpath("//b	send_key	
Hit Enter			keypress	

Table 3.4 vogue_scenario2.csv



Step	Data	XPATH	Event
Click on NYU WIRELE		driver.findElement(By.xpath(",	click
Click on Research		driver.findElement(By.xpath(",	click
Click on 5G and 6G M		driver.findElement(By.xpath(",	click
Enter Rese	Millimeter	driver.findElement(By.xpath(",	sned_key
Hit Enter			keypress
Click on S.	T.S. Rappa	driver.findElement(By.xpath(",	ckick
Click on Search		driver.findElement(By.xpath(",	click
Enter sear	NYUSIM W	driver.findElement(By.xpath(",	send_key
Hit Enter			keypress
Click on 5G & Channe		driver.findElement(By.xpath(",	click
Click on Download NY		driver.findElement(By.xpath(",	click

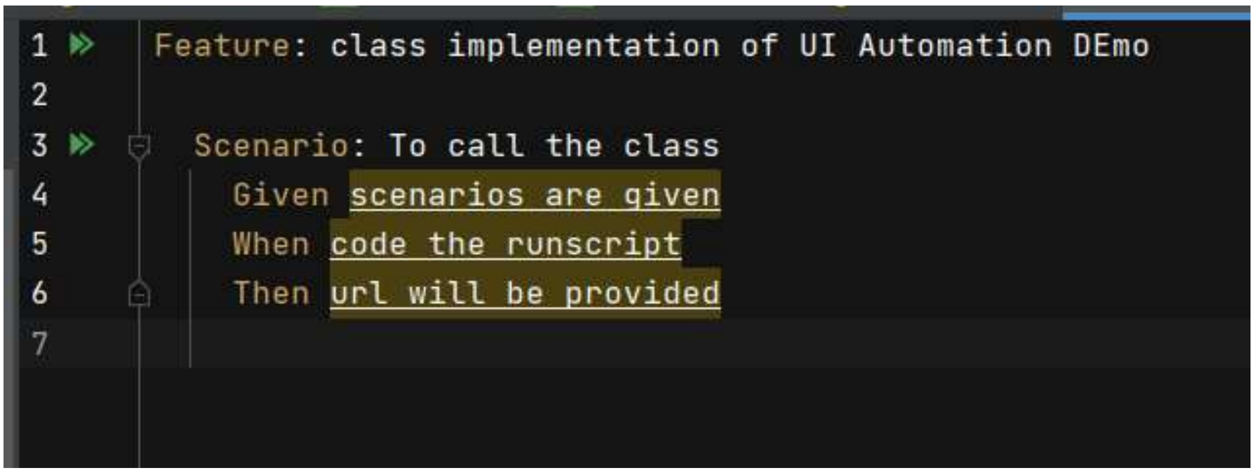
Table 3.5 NYU_scenario.csv

3.3 BDD Implementation

The UI Automation Framework is excellent but the testing can be done more efficient in a BDD Framework. My task was to implement the UI Automation in a BDD Framework incorporating class functions in it. As explained before BDD (Behaviour Driven Development) is an expansion of Test Driven Development (TDD). It is a very agile Framework. TDD is very technical and users find it difficult to understand and so BDD comes into play here. It introduces a User-friendly language. Here Each scenario is considered as a Given-When -then structure.

First, a simple BDD without the class structure was carried out. The BDD structure(feature file) of the class implementation is shown in Fig 3.1. Multiple and multiline inheritance is used in the class implementation without the BDD Framework. The implementation became more complex with the additional functions. The same was implemented using BDD which was executed seamlessly.

I implemented BDD in PyCharm module with the help of selenium packages. The structure was such that the functions were separated for ease so the code can be refactored for any use or for future automations. The Result of the BDD Class implementation is shown in the Allure Report. All the scenarios were passed and we had zero fail cases. The script below shows the basic class implementation. In both the class and the normal BDD implementation, all the test cases passed.



```
Feature: class implementation of UI Automation DEmo

  Scenario: To call the class
    Given scenarios are given
    When code the runscript
    Then url will be provided
```

Fig 3.6 UI Automation Class feature file

CHAPTER 4

RESULTS AND DISCUSSION

4.1 Results

From the implementation of the testing, we see that BDD is simpler. The automation flows with each and every segment can be easily understood by the user. A simple test flow chart is shown in the fig 4.2. For example: when the software automation using UI Automation and BDD is implemented for the Vogue scenario, we see the following screenshot with the watermarked "PASS" in green colour as an indication that the test case has passed.

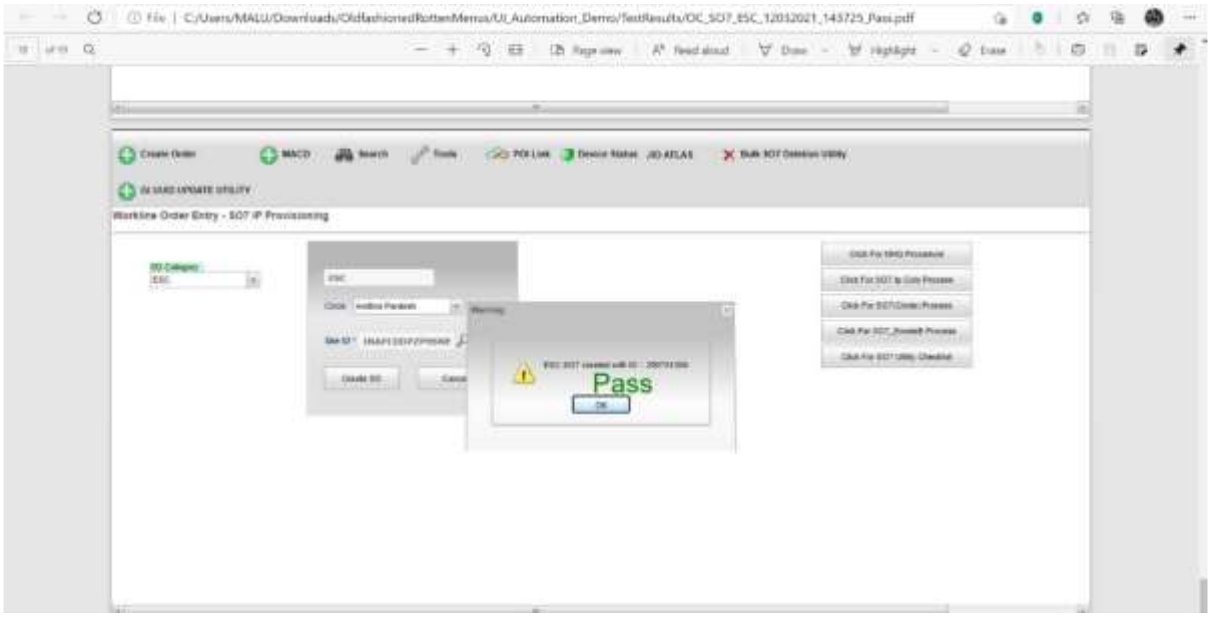

File | C:/Users/MALU/Downloads/OldfashionedRottenMenus/UI_Automation_Demo/TestResults/OC_SO7_ESC_12032021_143725_Pass.pdf
Page view
Read aloud
Draw
Highlight
Erase
Create Order
MACD
Search
Tools
Device Status
Workline Order Entry - SO7 IP Provisioning
ESC
Warning
Pass
OK

Fig 4.1 Final Screenshot with Pass Watermark

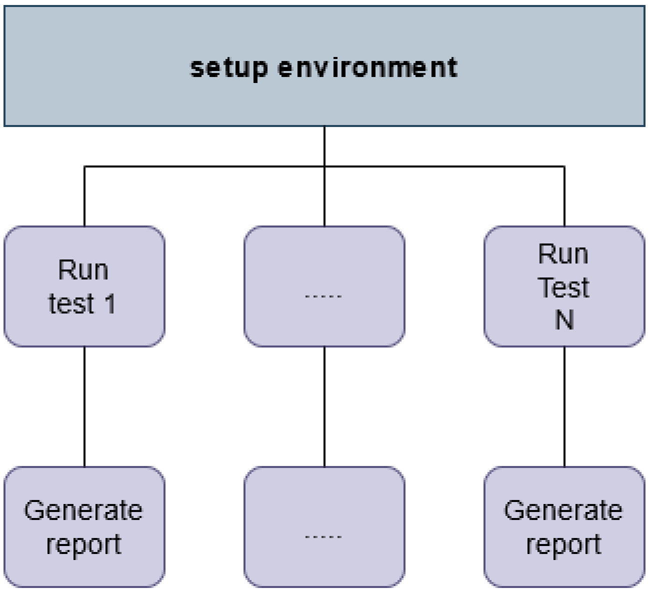


Fig 4.2 Test Run Flowchart

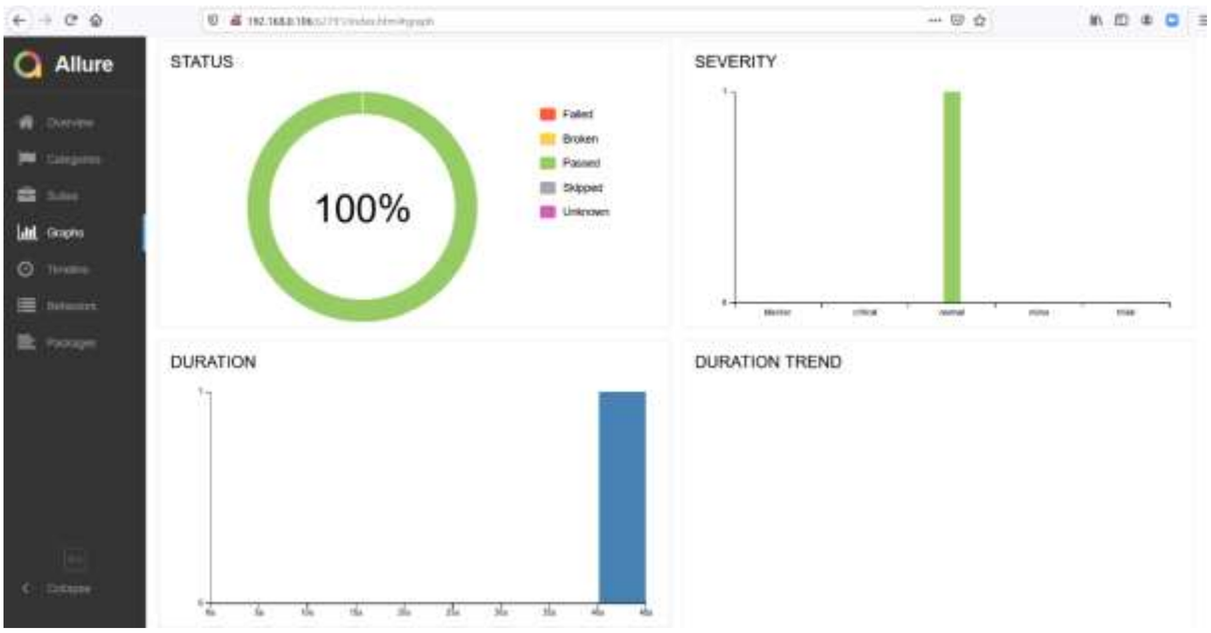
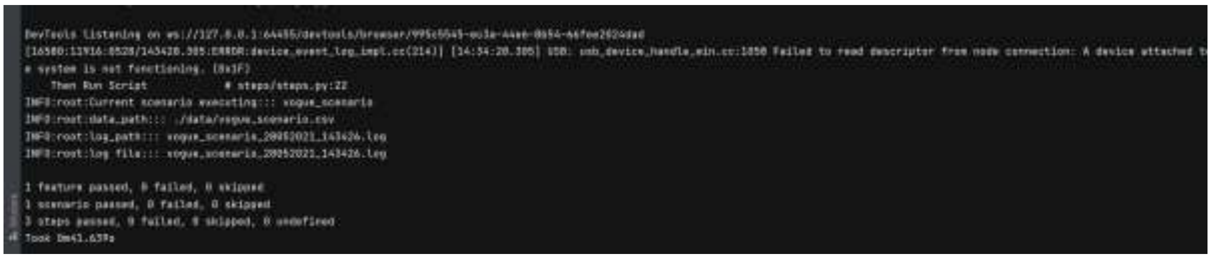


Fig 4.3 Allure Report – vogue_scenario test case



```
DevTools listening on ws://127.0.0.1:64455/devtools/browser/995c5545-ec1a-44ee-8654-46fee2624dad
[16580:11916:0528/143428.305:ERROR:device_event_log_impl.cc(214)] [14:34:28.305] USB: usb_device_handle_win.cc:1058 Failed to read descriptor from node connection: A device attached t
e system is not functioning. (0x1F)
    Then Run Script           # steps/steps.py:22
INFO:root:Current scenario executing::: vogue_scenario
INFO:root:data_path::: ./data/vogue_scenario.csv
INFO:root:log_path::: vogue_scenario_28052021_143426.log
INFO:root:log file::: vogue_scenario_28052021_143426.log

1 feature passed, 0 failed, 0 skipped
1 scenario passed, 0 failed, 0 skipped
3 steps passed, 0 failed, 0 skipped, 0 undefined
Took 0m41.639s
```

Fig 4.4 Terminal

The fig 4.2 shows the Allure Report for one test case i.e. with the vogue_scenario.csv with the ‘Y’ flag and the others with a ‘N’ flag. Fig 4.4 shows the screenshot with the ‘PASS’ watermark in green indicating a successful automation. When the same is performed with two test cases (Covid_scenario is marked as ‘Y’ along with vogue_scenario), we have the screenshot of that with a fail watermark as shown in fig 4.3.

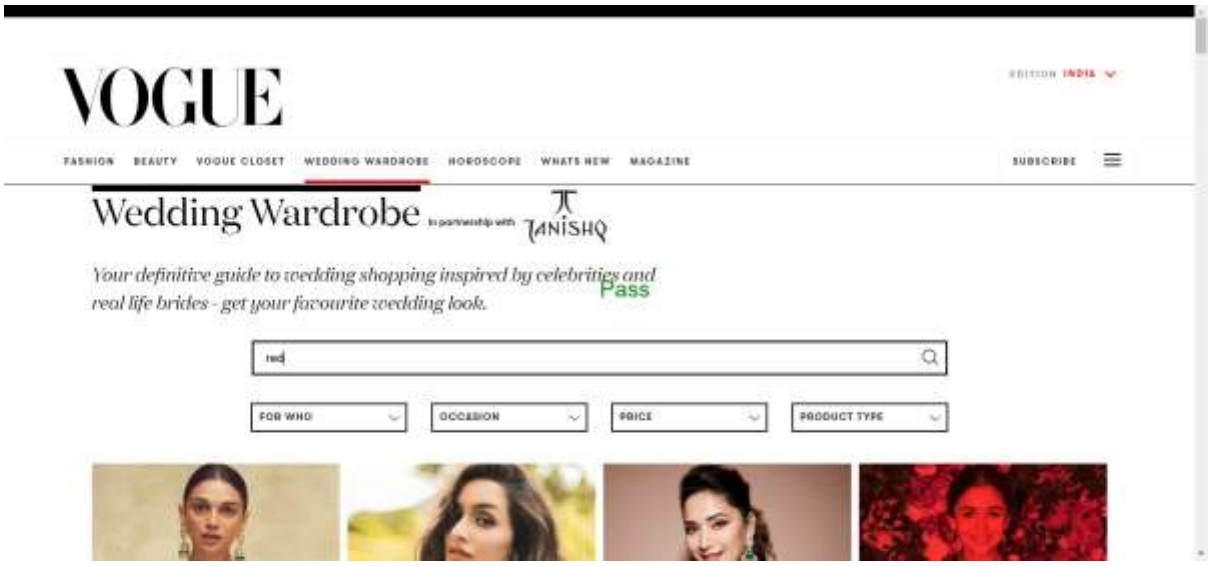

VOGUE
EDITION INDIA
FASHION
BEAUTY
VOGUE CLOSET
WEDDING WARDROBE
HOROSCOPE
WHATS NEW
MAGAZINE
SUBSCRIBE
Wedding Wardrobe in partnership with TANISHQ
Your definitive guide to wedding shopping inspired by celebrities and real life brides - get your favourite wedding look.
Pass
red
FOR WHO
OCCASION
PRICE
PRODUCT TYPE

Fig 4.5 Vogue-Pass watermark

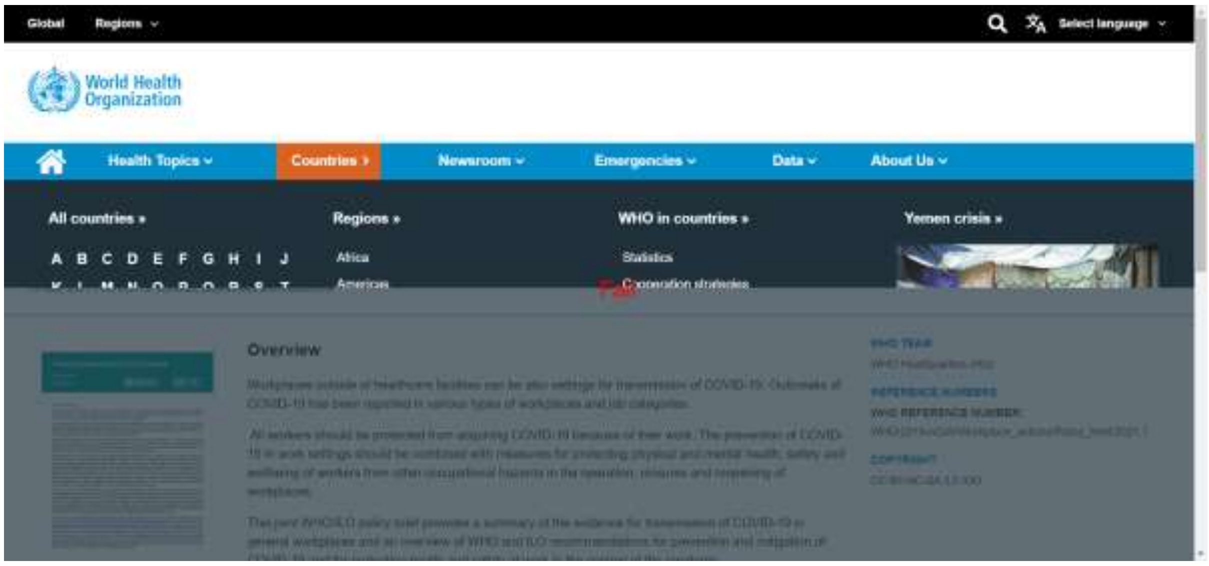
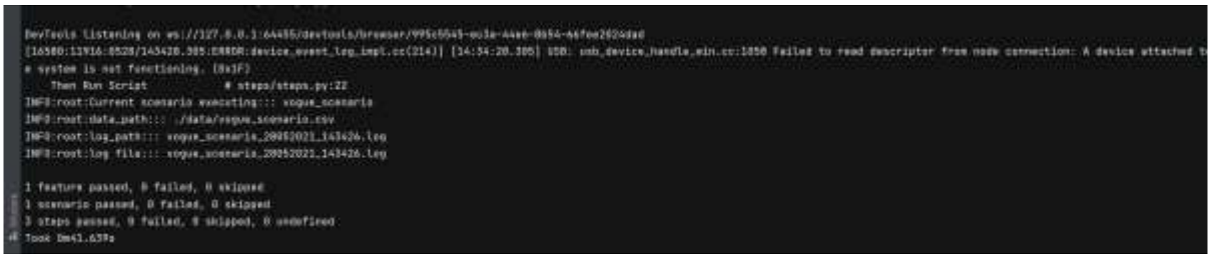


Fig4.6 Covid-Fail Watermark

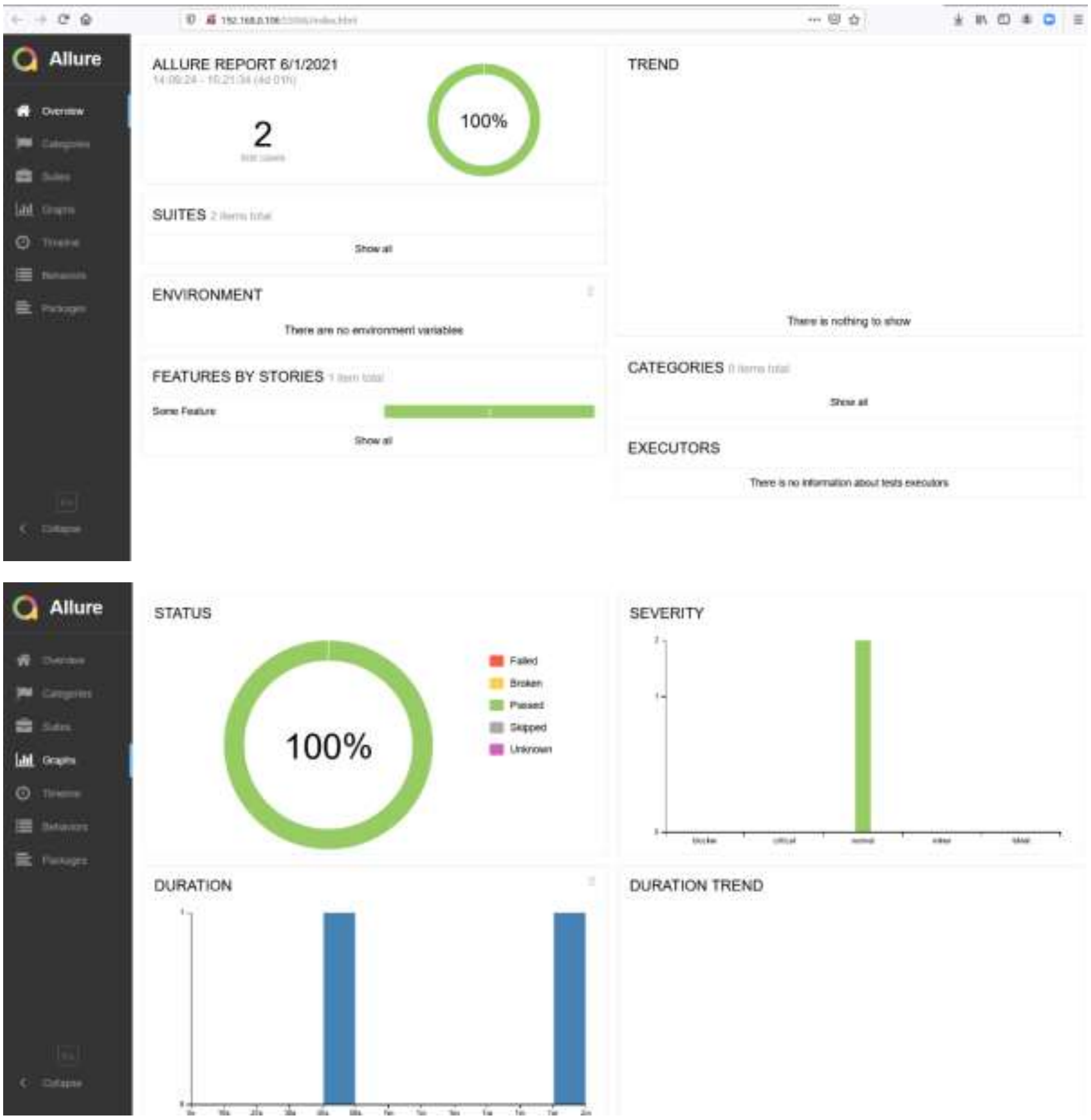
The reason for the failure of the previous scenario is that the scenarios in the scenario.csv file were not iterated under the behave step file. As a result only the scaenario whose flag was last changed, kept running. There were a few errors due to wrong XPATHs also. This was finally found out and rectified. Leading to all the scenarios getting passed and running successfully**.** The below figures show the results of the project.



```
DevTools listening on ws://127.0.0.1:64455/devtools/browser/995c5545-ea1a-44ee-8b54-46fee2624dad
[16580:11916:0528/143428.385:ERROR:device_event_log_impl.cc(214)] [14:34:28.385] USB: usb_device_handle_win.cc:1050 Failed to read descriptor from node connection: A device attached t
# system is not functioning. (0x1F)
    Then Run Script            # steps/steps.py:22
INFO:root:Current scenario executing::: vogue_scenario
INFO:root:data_path::: ./data/vogue_scenario.csv
INFO:root:log_path::: vogue_scenario_28052021_143426.log
INFO:root:log file::: vogue_scenario_28052021_143426.log

1 feature passed, 0 failed, 0 skipped
1 scenario passed, 0 failed, 0 skipped
3 steps passed, 0 failed, 0 skipped, 0 undefined
Took 0m41.639s
```

Fig 4.7 Terminal after final run



There are no environment variables

There is nothing to show

There is no information about tests executors

Fig 4.8 (top and bottom) Allure report of the project

4.2 Contributions

Specific Contribution:

- Worked on UI Automation class code
- Implemented UI Automation code using BDD in Python

Specific Learning:

- Various Automation Techniques
- Behaviour Driven Development Testing

Technical Limitation and Ethical Challenges Faced: *None*

CHAPTER 5
SOURCE CODE

5.1 Code

Due to confidentiality purposes, I have been asked not to put up the code in this report.

CHAPTER 6

CONCLUSIONS AND FURTHER WORK

6.1 CONCLUSION

Thus, the above project can be used by all software testers to test a web applications. It can also be further developed to incorporate all forms of functions.

The implementation of BDD has made it simple for us to refactor our code according to the need and it makes debugging easy. We can combine both TDD and BDD (T-BDD) to change the automation testing in companies from implementation based to behaviour based.

More BDD based tests can be used to speed up regression based tests. Since BDD optimeses to make the most of each and every single approach, it takes a holistic way of testing. In the future, researchers could implement additional mapping rules for BDDs. The existing mapping rules in the BDD toolkits only map scenarios to code. Feature sets might
be mapped to packages also so that the test classes of a scenario can be located.

REFERENCES

[1] Carlos Solís, Xiaofeng Wang, “A Study of the Characteristics of Behaviour Driven Development”, DOI:10.1109/SEAA.2011.76 , Software Engineering and Advanced Applications (SEAA), 2011 37th EUROMICRO Conference

[2] Melanie Diepenbeck, Ulrich Kühne et al , “Behaviour Driven Development for Testsand Verification”,

[3] Mohsin Irshad,Ricardo Britto,Kai Petersen ,2021/03/07, “Adapting Behavior Driven Development (BDD) for Large-scale Software Systems”

[4] Stefania Bruschi, Le Xiao, Mihir Kavatkar, Gustavo Jimenez-Maggiora, “Behavior Driven Development (BDD) A Case Study in Healthtech”

[5] Arlinta Christy Barus 2019 J. Phys.: Conf. Ser.1175 01211, “The implementation of ATDD and BDD from Testing Perspectives”

[6] Srashti Lariya, Dr. Sameer Shrivastava, Er. Sumit Nema, “Automation Testing using Selenium Web Driver &Behavior Driven Development(BDD)” , JSRD -International Journal for Scientific Research & Development| Vol. 6, Issue 03, 2018| ISSN (online): 2321-0613

[7] P. V. Sagar, R. Paruchuri and S. Vemuri, "Testing using selenium web driver," IEEE, 23 November 2017.

[8] Chandraprabha, A. Kumar and S. Saxena, "Data Driven Testing Framework using Selenium," International Journal of Computer Applications, vol. 118, no. 18, pp. 18-23, May 2015.

[9] https://hsc.com/DesktopModules/DigArticle/Print.aspx?PortalId=0&ModuleId=1215&Article=155

[10] https://iopscience.iop.org/article/10.1088/1742-6596/1175/1/012112/pdf

[11] http://passionateaboutoss.com/background/what-are-oss-bss/

[12] CodeUtopia. What’s the difference between Unit Testing, TDD and BDD? | CodeUtopia [Internet]. 2019 [cited 2019 May 6]. https://codeutopia.net/blog/2015/03/01/unit-testing-tdd-and-bdd/