# Manual Part 2-

* Tester – TCD, Test cases review, TCE, Test proof, Defect raised, Report

# Test Documentation –

**Q. What is your organization Test documentation?**

* Test Document hierarchy

Quality control – QC/

Testing Head -TH

Company Level document

**Test Scenario/ Case**

**Test Plane**

**Test Methodology**

**Test Strategy**

**Test Policy**

Test Strategist – TS & PM

Project manager – PM

TRM

Team Lead – TL

Team Lead – TL

Tester – Tester

Project Level document

**Test Procedure/Design**

**Test Script/ Execution (Test Proof)**

**Defect Report**

**Test Summary Report**

**Final Report/ Test closer Report**

**Test Policy-**

* Test policy documents defines objective of the project (Project renew generated)
* Test policy documents prepared by Testing Head -TH
* Test policy documents company level documents

**Test Strategy-**

* Test Strategy defines which Strategy/approaches we can apply for full fill the objective of the project
* Ex. Project 🡪 Automation testing 🡪 Java/ C#/ Python / etc + Selenium tool
* Test Strategy documents will be prepared by Test Strategist – TS & PM
* Test Strategy documents company level documents

**Test Methodology-**

* Test methodology defines 🡪 Environment follow/use for Strategy/approaches
* Test methodology documents will be **prepared by PM**
* Test methodology documents Project level documents
* Test methodology documents 🡪 PM will prepared the **TRM (Test responsibility matrix)**
* TRM defines **development stage mapping with testing factor**
* While preparing the TRM consider

1. Project requirement
2. Project scope
3. Risk in project

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Development Stage**  **Testing factor** | **Information gathering/ Analysis** | **Design** | **Coding** | **Testing** | **Maintains** |
| System & functional Testing | No | No | Yes | Yes | Yes |
| Security | No | No | No | Yes | Yes |
| Performance Testing | No | No | No | Yes | Yes |
| GUI/ UI | No | Yes | Yes | Yes | Yes |

**Test Plane-**

* Test plane will contains Resource allocation, Job allocation, estimation, etc
* Test plane documents prepared by Test lead
* Test plane documents Project level documents

**Identify test scenario & Test case design –**

* Tester will identity test scenario & Test cases design
* Test scenario & Test cases design prepared by Tester
* Test scenario & Test cases documents Project level documents

**Test cased execution & defect report-**

* In TCE, if we found defects then we will raised to developer
* Modified build we will perform regression & Retesting.

**Test summery report & Test closer report-**

* Test summery report will contains TCD, TCE & Test cases skip, ect
* Test summery report & Test closer report prepared by Test lead
* Test summery report & Test closer report documents Project level documents

# Software Testing Life Cycle (STLC) –

**Q. What is your organization Test process?**

**Q. What is STLC?**

Test Initiation

Test Plane

Test Scenario

Defect sent to developer

Test Case Execution- Test Proof

Test Case Design

Requirement Analysis

Test Closer/ Test Summery Report

**Test Initiation-**

* In Test initiation stage PM is working
* PM will prepared the **TRM** (Test responsibility matrix)

**Test Plane –**

* **Test plane will** prepared by test lead
* Test plane will contains **Job allocation, Resource allocation, Estimation**

**Test scenario-**

* **Test scenario will** prepared by **tester against US**
* Test scenariowill defines “**how to test”**

**Test cases design-**

* **Test cases will** prepared by **tester against Test scenario**
* Test **cases** will defines “**what to test”**

**Test cases execution & Defect report-**

* When we got build from developer then we will execute test cases
* If in TCE, when we **got the defect** then we will **create these defect in JIRA/ Azure Develops tool**
* Tester will inform to developer and developer has **fixed these defect**
* Tester will do **Re-testing & Regression**

**Test Closer/ Test Summery Report-**

* Test Closer/ Test Summery Report will prepped by test lead
* Test Summery Report will prepped against sprint
* Test Closer Report will prepped against Module

# Testing Process-

**BRS**

Testing

Development

**SRS/FRS/CRS**

**Design Test Initiation Stage- TRM**

**Coding Test Plane –Sprint**

**STLC**

**Unit Testing Test scenario & Test Case Design**

**SDLC**

**Integration Testing Test Case Execution & Closer**

**(Install Build)**

**Level 0** Sanity**/ Smoke Testing (**Check theStability of build)

**Level 1 BBT/ System & function testing (**Intern & External**)**

**(**If found **defect** sent to developer) (Inform throw JIRA/Azure)

**Level 2 Regression Testing on Modified built**

**Level 3 Final Regression Testing**

# Test Plane-

* Test plane will be **prepared by Test Lead**
* Test plane will be prepared **against the Sprint**
* While preparing these test plane, test lead focus on

1. Job allocation
2. Resource allocation
3. Estimation
4. Test plane main purpose to defines **start and end day of testing**

* In Test plane documents will contains

1. Test plane ID
2. Testing task
3. Feature to be tested
4. Feature not to be tested
5. Pass & Fail Criteria
6. Test delivery (entry and exit Criteria)
7. Test environments
8. Defect life cycle
9. Test risk
10. Responsibility
11. Signature & Approval

* I have **worked with Test Lead for preparation of Test Plane**

# Test Scenario-

**SRS/ FRS/ CRS**

**Use Case / User Story**

**Test Scenario**

**Test cases design**

**BA BA Tester Testers**

**SRS/ FRS/ CRS-**

* SRS defines system requirement specification
* SRS will be **prepared by BA**
* SRS will be **derived from BRA**
* SRS will **contains multiple Use cases**
* SRS will contain

1. Function requirements
2. Function flow diagram
3. Use Cases
4. Screenshot / Prototype

**Use Cases-**

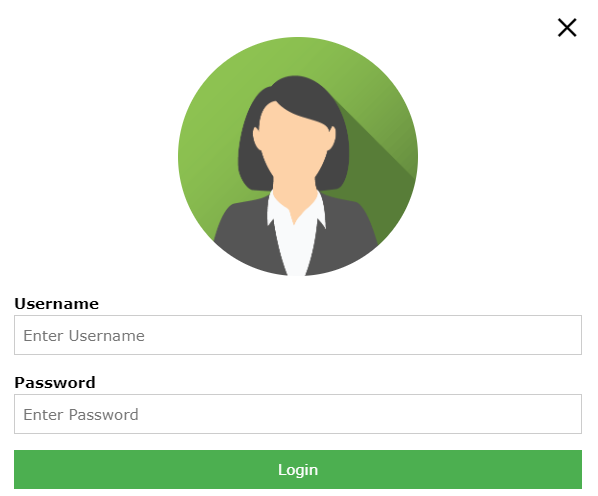
* Use Cases defines system single specification requirement
* Use Cases will be **prepared by BA**
* Use Cases will be **derived from SRS**
* Use Cases will **contains multiple Use stories**
* Use cases will contains

1. Description
2. Acceptance Criteria

**Test Scenario –**

* Test scenario will be **prepped by Tester**
* Test scenario will be **derived/ identified from US**
* Test scenario defines **“Ways to test functionality”**
* Test scenario will be **written in +VE ways**
* Test scenario will **contains multiple test cases**
* **Ex. US =** VelocityLogin page – Username = email id/ Mobile number accept,

Password = 4 to 8 Charter combination (1 Upper letter, 1 small letter, 1 number, 1 Special no)

****

**Test Scenario** **-**

1. Verify the login page by passing email id data into Username test box
2. Verify the login page by passing mobile number data into Username test box
3. Verify the login page by passing data into password test box
4. Verify the login page by pressing Login with data into Username & Password test box

**Test Cases-**

* Test cases will be **prepped by Tester**
* Test cases will be **derived/ identified from Test scenario**
* Test cases defines **“How to test functionality”** OR “**Input, Process, Output**”
* Test cases will be **written in +VE ways & -VE ways**
* Test cases will **contains multiple steps for execution**

**Test Scenario** **-** Verify the login page by passing email id data into Username test box

**Test cases-**

1. Verify the login page by passing gmail id (abc@gmail.com) into Username test box
2. Verify the login page by passing yahoo id (abc@yahoo.com) into Username test box
3. Verify the login page by passing Outlook id (abc@outlook.com) into Username test box
4. Verify the login page by passing Hotmail id (abc@hotmail.com) into Username test box
5. Verify the login page by passing Redfmail id (abc@redfmail.com) into Username test box
6. Verify the login page by passing company id (priti.patil@ Infosys.com) into Username test box
7. Verify the login page by passing invalid gmail id ($#$&^@ gmail.com) into Username test box
8. Verify the login page by passing null/blank into Username test box

**Test Scenario** - Verify the login page by passing mobile number data into Username test box

**Test cases-**

1. Verify the login page by passing Airtel mobile into Username test box
2. Verify the login page by passing JIO mobile into Username test box
3. Verify the login page by passing BSNL mobile into Username test box
4. Verify the login page by passing VI mobile into Username test box
5. Verify the login page by passing MTNL mobile into Username test box
6. Verify the login page by passing invalid mobile (0000000000) into Username test box
7. Verify the login page by passing invalid mobile (9999999999) into Username test box
8. Verify the login page by passing null/blank mobile into Username test box

**Test Scenario** - Verify the login page by passing data into password test box

**Test Cases-**

1. Verify the login page by passing valid password combination (Abc@12) into password test box
2. Verify the login page by passing valid password combination (aBC@12) into password test box
3. Verify the login page by passing valid password combination (@Ab12) into password test box
4. Verify the login page by passing valid password combination (12Abc@) into password test box
5. Verify the login page by passing valid password combination (Ab@1) into password test box
6. Verify the login page by passing valid password combination (AAbb@@11) into password test box
7. Verify the login page by passing invalid password combination (ab@12) into password test box
8. Verify the login page by passing invalid password combination (A@12) into password test box
9. Verify the login page by passing invalid password combination (@1234) into password test box
10. Verify the login page by passing null/blank values into password test box

**Test Scenario** - Verify the functionality of Login button by passing data into Username & Password test box

# Test cases –

* Test cases will be written into **Excel sheet** OR **JIRA/ Azure DevOps tool**
* Test cases consider

1. Test cases should **cover business login**
2. Test cases should **cover all functionality**
3. Test cases should **cover GUI/ UI**

**Test cases write for following these –**

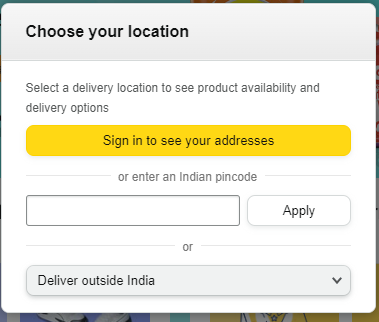
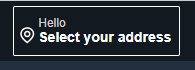
1. **Objects- Pen**, Chair, Table, Watch, Fan, **Bottle**, **ATM**, etc
2. **Whatup**, Instgram, Facebook, **Amazon**, Flipkard, etc
3. **Web side** - <https://www.heenakochhar.com/>, <https://worldofek.com/>, <https://emicalculator.net/>, <https://strfix.com/#!/>
4. **Scenario based** – **Left will stop on odd floors only**, Admission form page, etc

**Web Side** - <https://artoftesting.com/chair>

**Bottle Test Cases-**

1. Verify the capacity of 1 liter bottle 🡪 EC = 1 liter water should be present
2. Verify the capacity of 2 liter bottle 🡪 EC = 2 liter water should be present
3. Verify the capacity of 5/10/25 liter bottle 🡪 EC = 2 liter water should be present
4. Verify the storage capacity of bottle by entering liquid (water) 🡪 EC = Bottle should store liquid
5. Verify the storage capacity of bottle by entering liquid (Oil) 🡪 EC = Bottle should store liquid
6. Verify the storage capacity of bottle by entering liquid (Chemical) 🡪 EC = Bottle should store liquid
7. Verify the storage capacity of bottle by entering liquid (Milk/medican) 🡪 EC = Bottle should store liquid
8. Verify the storage capacity of bottle by entering Gas (Oxygen) 🡪 EC = Bottle should store Gas
9. Verify the shape of bottle 🡪 EC = Bottle should in cylindrical, Rectangle, circle, etc format
10. Verify the transparency property of bottle 🡪 EC = Bottle should transparency and non transparency
11. Verify the shape of cap/ knob of bottle 🡪 EC = Bottle cap should be in round circle OR Some Bottle should present knob
12. Verify the base shape of bottle 🡪 EC = Bottle base should be in circle flat formate
13. Verify the material used in the bottle 🡪 EC = Bottle should made with Glass, Steel, Metal, Browns, Plastic, etc
14. Verify the size/ length/ width of bottle 🡪 EC = Bottle should in proper shape as per required
15. Verify the leakage property of bottle 🡪 EC = Bottle should not be leakage
16. Verify the cooling property of bottle 🡪 EC = Bottle should cooling
17. Verify the strengthens property of bottle 🡪 EC = Bottle should not be brake
18. Verify the look/ appearance of bottle 🡪 EC = Bottle look/ appearance should be normal
19. Verify the text present on bottle 🡪 EC = Bottle should contains text of company
20. Verify the details information present in bottle 🡪 EC = Bottle should contains details information a out company, date, packing, Price, etc
21. Verify the leakage property of bottle by entering hot water of 200 C🡪 EC = Bottle should melt
22. Verify the shirking property of bottle by entering cold water of -200 C🡪 EC = Bottle should shirking
23. Verify the leakage property of bottle by entering hot oil/ chemical/etc of 200 C🡪 EC = Bottle should melt
24. Verify the storage capacity of bottle by entering 2 liter volume into 1 liter capacity 🡪 EC= Bottle should not store 2 litter into 1 liter capacity bottle
25. Verify the storage capacity of bottle by entering solid of 1 kg volume into 1 liter capacity 🡪 EC= Bottle should not store 1 kg into 1 liter capacity bottle
26. Verify the strengthens property of plastic bottle after throwing from 10th floor🡪 EC = Bottle should be brake
27. Verify the strengthens property of brown/ metal bottle after throwing from 10th floor🡪 EC = Bottle should not be brake

**Amazon Test cases-**

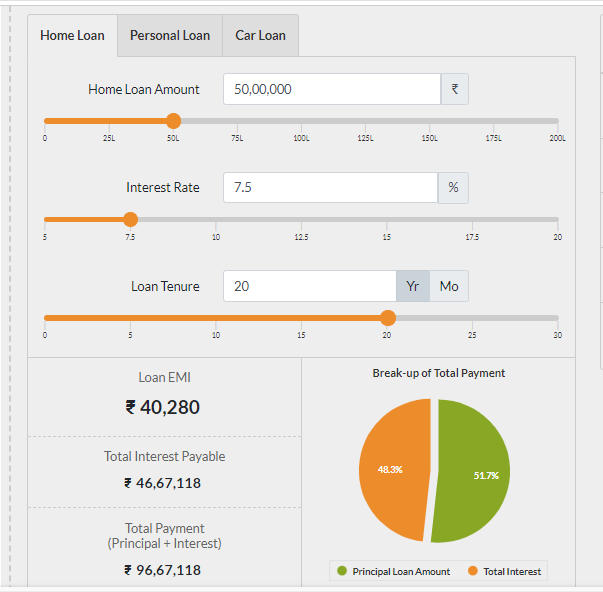
****

1. Verify the new icon present in Amazon home page 🡪EC = “Select your address” icon should be present
2. Verify new pop after clicking on new icon present in Amazon home page 🡪EC = “Choose your location” Pop should be display
3. Verify text present in a “Choose your location” pop 🡪 EC= Pop will contain test box, One button, Drop down
4. Verify after clicking on “Sing in to see your address” text box present in a “Choose your location” pop 🡪 EC= Sing page should be open
5. Verify by passing valid pin code (431001) into text box present in a “Choose your location” pop 🡪 EC= For valid pin code city name should be display
6. Verify by passing valid pin code (432100) into text box present in a “Choose your location” pop 🡪 EC= For valid pin code city name should be display
7. Verify by passing valid pin code (431095) into text box present in a “Choose your location” pop 🡪 EC= For valid pin code city name should be display
8. Verify by passing In-valid pin code (000000) into text box present in a “Choose your location” pop 🡪 EC= For In-valid pin code error message should display as – “Please enter a valid pincode”
9. Verify by values present in a drop “Delivery outside India” in a “Choose your location” pop 🡪 EC= “Sri lanka” values should be display
10. Verify text resent in “Choose your location” pop after changing country/ region outside India 🡪 EC = “Choose your location” pop will contains Button- Sing to see your address, Text box- enter a UK postcode, Drop down – Delivery outside the UK
11. Verify by passing valid UK postcode (SW10) into text box 🡪 EC= City name should be display
12. Verify by passing valid UK postcode (SW7) into text box 🡪 EC= City name should be display
13. Verify by passing In-valid UK postcode (431001) into text box 🡪 EC= Show the error – “ Please enter valid UK postcode”
14. Verify the values present in a drop “Delivery outside UK” in a “Choose your location” pop 🡪 EC= All country name/values should be display
15. Verify text resent in “Choose your location” pop after changing (India to Italy) country/ region outside India 🡪 EC = “Choose your location” pop will contains Button- Sing to see your address, Text box- enter a Italy postcode, Drop down – Delivery outside the Italy
16. Verify by passing valid Italy postcode (89898) into text box 🡪 EC= City name should be display
17. Verify by passing valid Italy postcode (88889) into text box 🡪 EC= City name should be display
18. Verify by passing In-valid UK postcode (431001) into text box 🡪 EC= Show the error – “ Please enter valid Italy postcode”
19. Verify the values present in a drop “Delivery outside Italy” in a “Choose your location” pop 🡪 EC= All country name/values should be display

**Scenario based test cases- Left will stop on odd floors**

|  |  |
| --- | --- |
| **Floor 1** | **Left moments** |
| **Floor 2** |  |
| **Floor 3** | **Y** |
| **Floor 4** |  |
| **Floor 5** | **Y** |
| **Floor 6** |  |
| **Floor 7** | **Y** |

1. Verify the moment of left by pressing floor 3 from floor 1 🡪 EC= Left should be stop on floor 3
2. Verify the moment of left by pressing floor 5 from floor 1 🡪 EC= Left should be stop on floor 5
3. Verify the moment of left by pressing floor 7 from floor 1 🡪 EC= Left should be stop on floor 7
4. Verify the moment of left by pressing floor 2 from floor 1 🡪 EC= Left should NOT be stop
5. Verify the moment of left by pressing floor 4/6 from floor 1 🡪 EC= Left should NOT be stop
6. Verify the moment of left by pressing floor 5 from floor 3 🡪 EC= Left should be stop on floor 5
7. Verify the moment of left by pressing floor 7 from floor 3 🡪 EC= Left should be stop on floor 7
8. Verify the moment of left by pressing floor 1 from floor 3 🡪 EC= Left should be stop on floor 1
9. Verify the moment of left by pressing floor 2/4/6 from floor 3 🡪 EC= Left should NOT be stop
10. Verify the moment of left by pressing floor 7 from floor 5 🡪 EC= Left should be stop on floor 7
11. Verify the moment of left by pressing floor 1 from floor 5 🡪 EC= Left should be stop on floor 1
12. Verify the moment of left by pressing floor 3 from floor 5 🡪 EC= Left should be stop on floor 3
13. Verify the moment of left by pressing floor 2/4/6 from floor 5 🡪 EC= Left should NOT be stop
14. Verify the moment of left by pressing floor 1 from floor 7 🡪 EC= Left should be stop on floor 1
15. Verify the moment of left by pressing floor 3 from floor 7 🡪 EC= Left should be stop on floor 3
16. Verify the moment of left by pressing floor 5 from floor 7 🡪 EC= Left should be stop on floor 5
17. Verify the moment of left by pressing floor 2/4/6 from floor 7 🡪 EC= Left should NOT be stop

**Web side-**

**Test scenario-**

1. Verify the functionality of “Home Loan amount” text box and slider chart
2. Verify the functionality of “Interest Rate” text box and slider chart
3. Verify the functionality of “Loan Tenure” text box and slider chart
4. Verify the calculation present in a EMI calculator
5. Verify the functionality of PI chart

**Test Scenario- Verify the functionality of “Home Loan amount” text box and bar chart**

**Test cases –**

1. Verify the input passed into Home loan amount text box 🡪EC= Only integer
2. Verify the min and max values passed into Home loan amount text box
3. Verify the separation present in a values passed into Home loan amount text box
4. Verify the unit present near to Home loan amount text box
5. Verify the by passing –ve values/ charter/ decimal passed into Home loan amount text box
6. Verify the min and max values passed into Home loan amount text box
7. Verify the color present in slider of Home loan amount
8. Verify the range present in slider of Home loan amount
9. Verify the movement of slider of Home loan amount
10. Verify the slider movement by passing some 50,00,000 into of Home loan amount
11. Verify the slider movement by passing some 50,000 into of Home loan amount
12. Verify the text values after moving slider into right side of Home loan amount
13. Verify the text values after moving slider into left side of Home loan amount

**Test Scenario- Verify the functionality of “Loan tenure” text box and slider char**

**Test cases –**

1. Verify the input passed into Loan tenure text box
2. Verify the min and max values passed into Loan tenure text box
3. Verify the separation in a values passed into Loan tenure text box
4. Verify the unit present near to Loan tenure text box
5. Verify the by passing –ve values/ charter/ decimal passed into Home loan amount text box
6. Verify the min and max values passed into Loan tenure text box
7. Verify the color present in slider of Loan tenure
8. Verify the range present in slider of Loan tenure
9. Verify the movement of slider of Loan tenure
10. Verify the slider movement by passing some 20 Year into of Loan tenure
11. Verify the slider movement by passing some 5 year into of Loan tenure
12. Verify the text values after moving slider into right side of Home loan amount
13. Verify the text values after moving slider into left side of Home loan amount
14. Verify the slider movement by passing some 5 year into of Loan tenure
15. Verify the range and values if by passing values in month format and click on year format
16. Verify the range and values if by passing values in year format and click on month format

**BDD (Behavioral driven development)-**

1. **Test Cases –**

**Feature File/ US-**

**Given** the value entered in the Number text box is not numerical  
**When** the Form is submitted  
**Then** an error message “Please enter a numerical value” appear  
Given the User is logged in ← Condition  
**And** the value in the Number text box changes ← Trigger

**Login text box -**

**Submit**

1. **US OR Feature file -**

**Given** I’m at the sign up form  
And all these mandatory fields are entered

* First Name
* Last Name
* Email
* Password

**When** I submit the Form  
**Then** an account is created  
**And** account name is set as my Email  
And a confirmation email is sent to me

**First Name\*-**

**Last Name\*-**

**Email\*-**

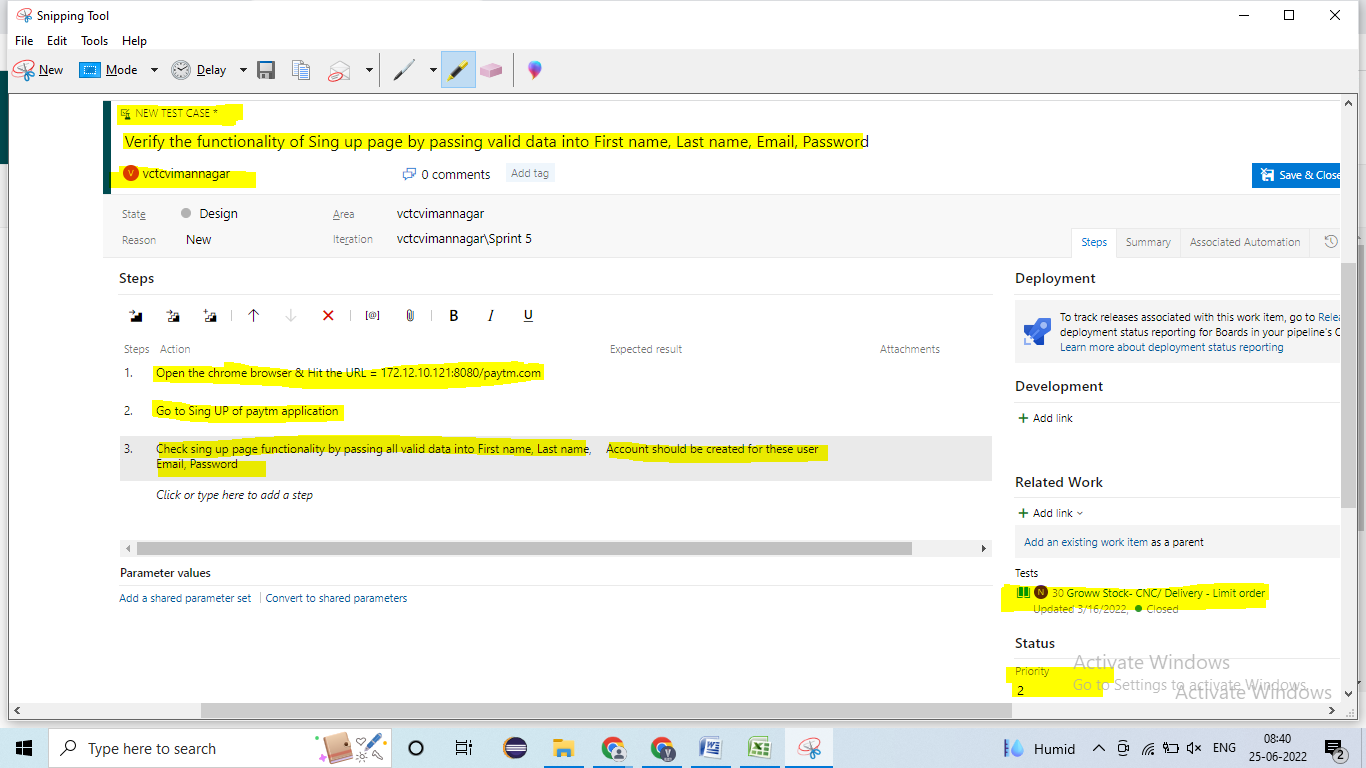
**Password\*-**

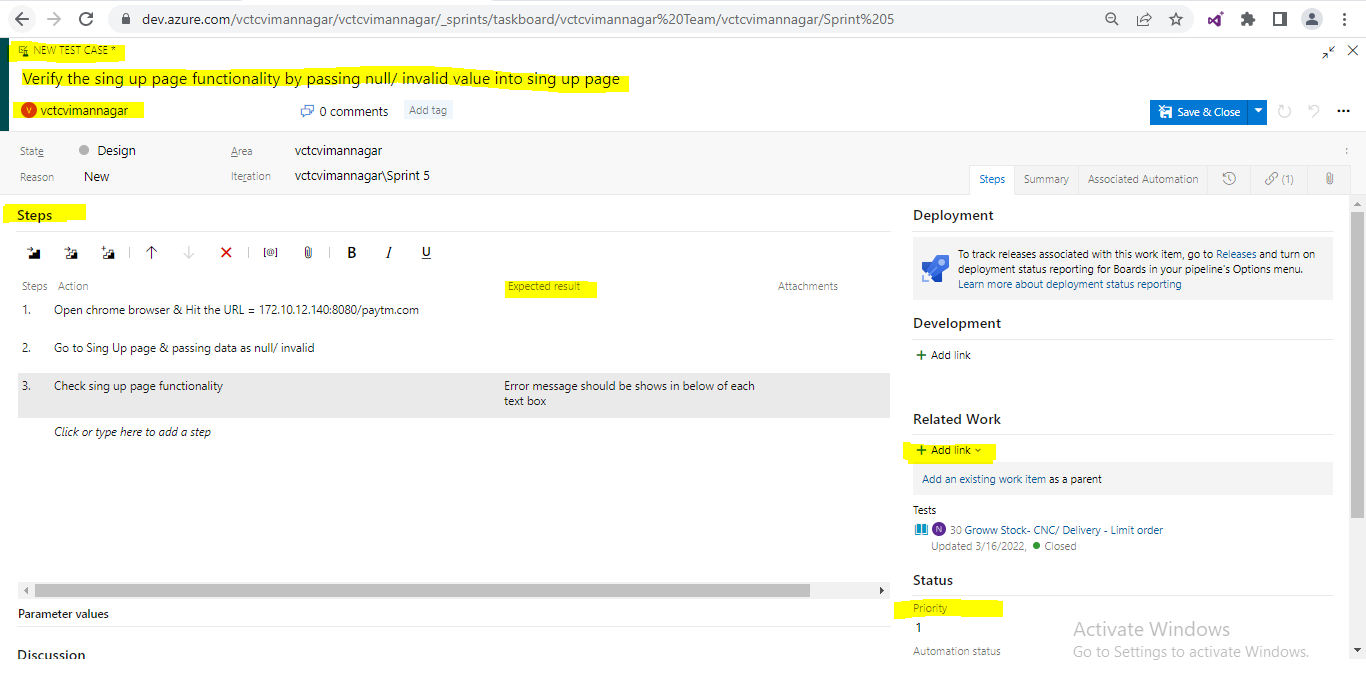
**Sing Up**

**Test cases written –**

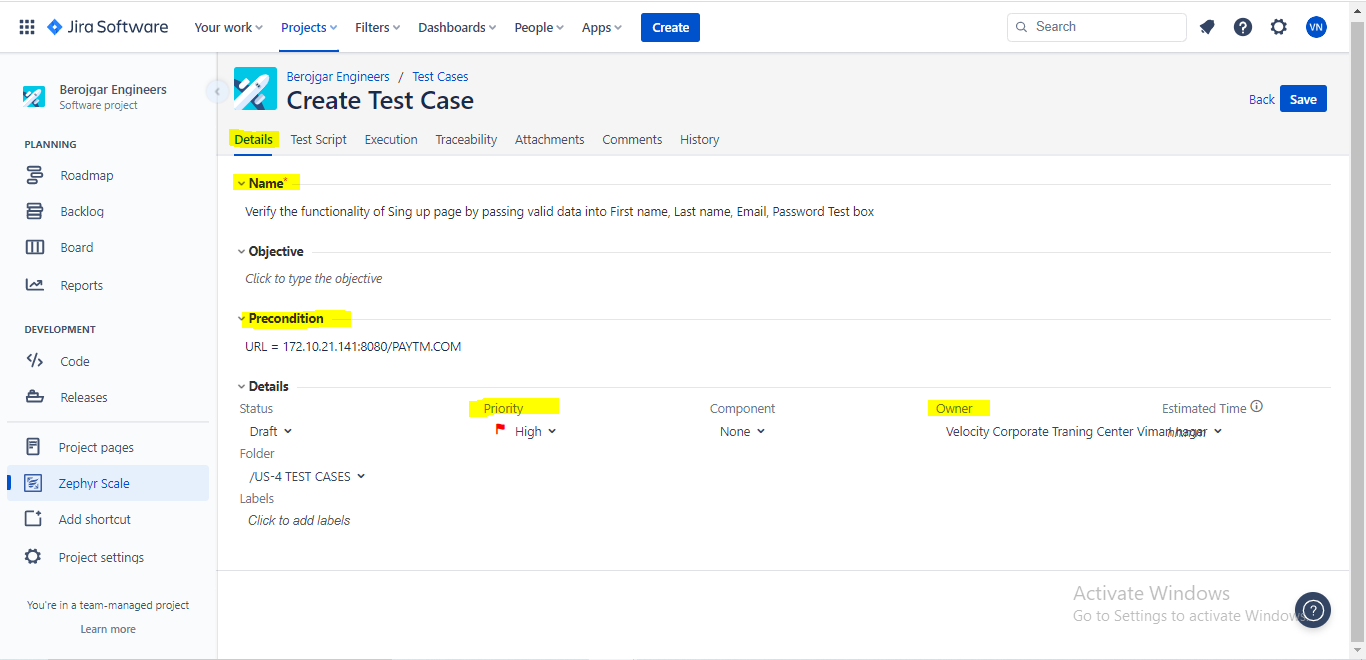
* Test cases written into Excel sheet OR **directly into** **JIRA/ Azure Devops tool**

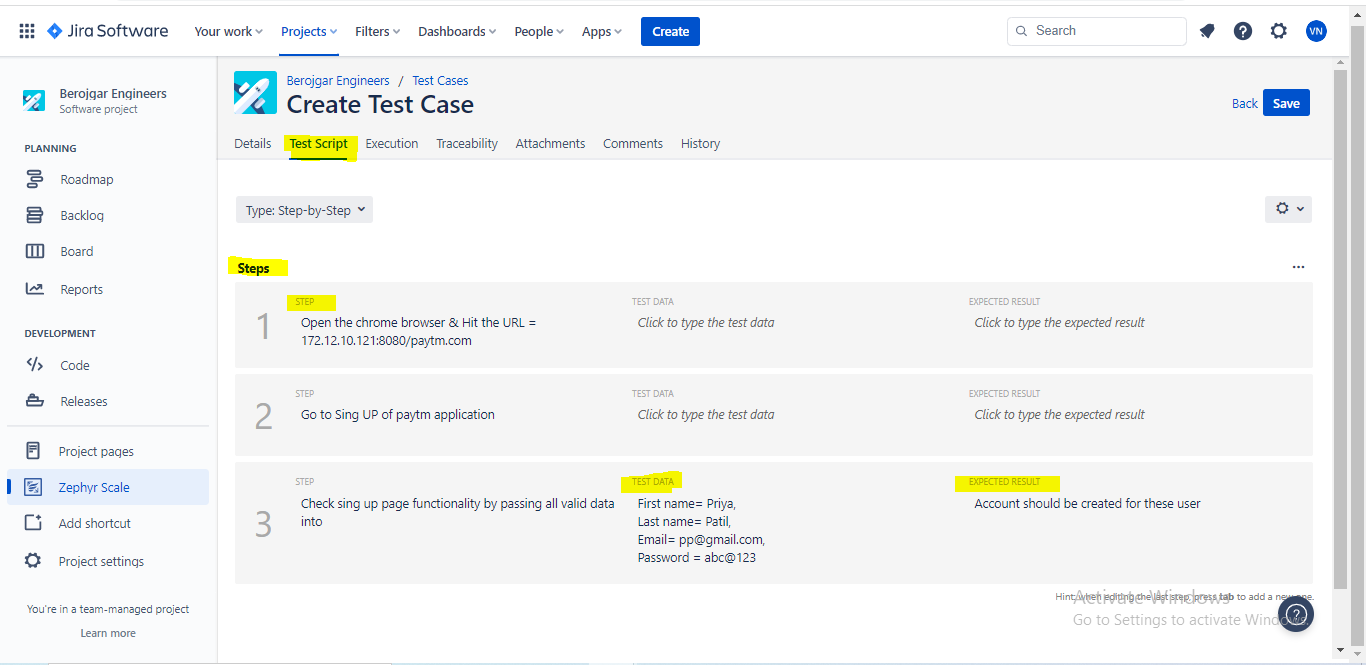
**Azure Devops tool-**

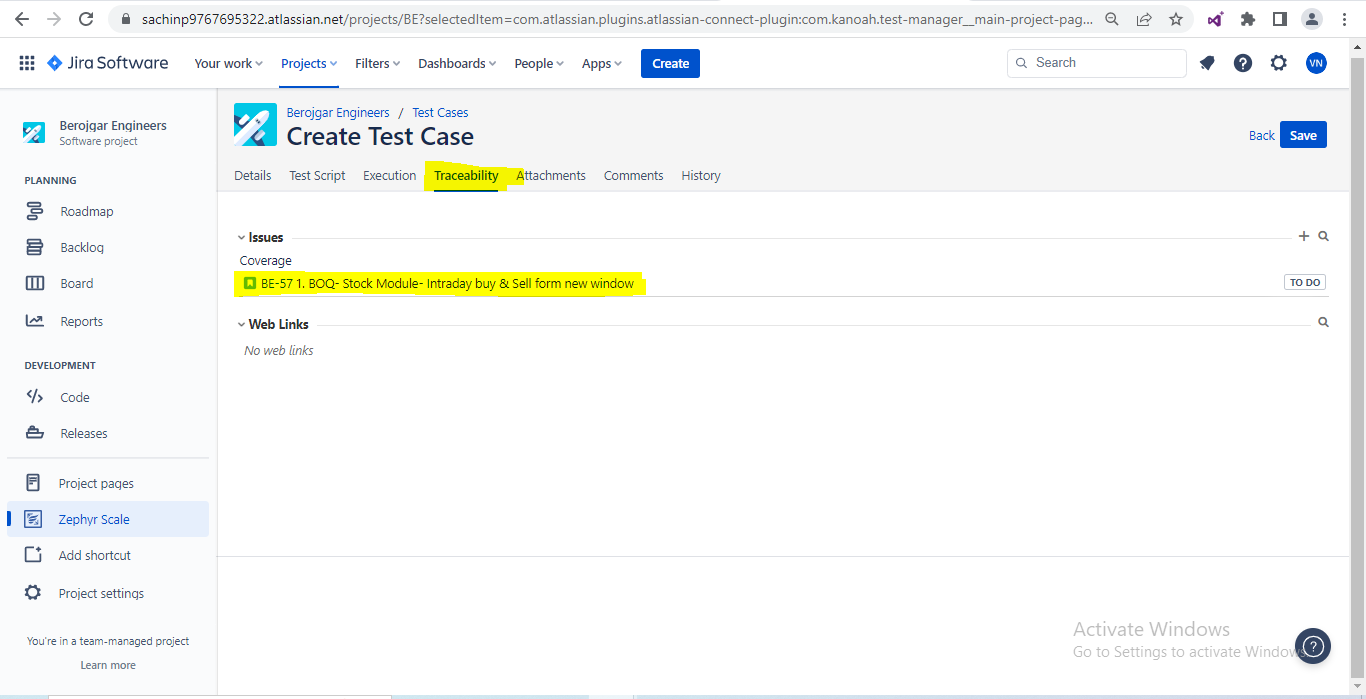
****

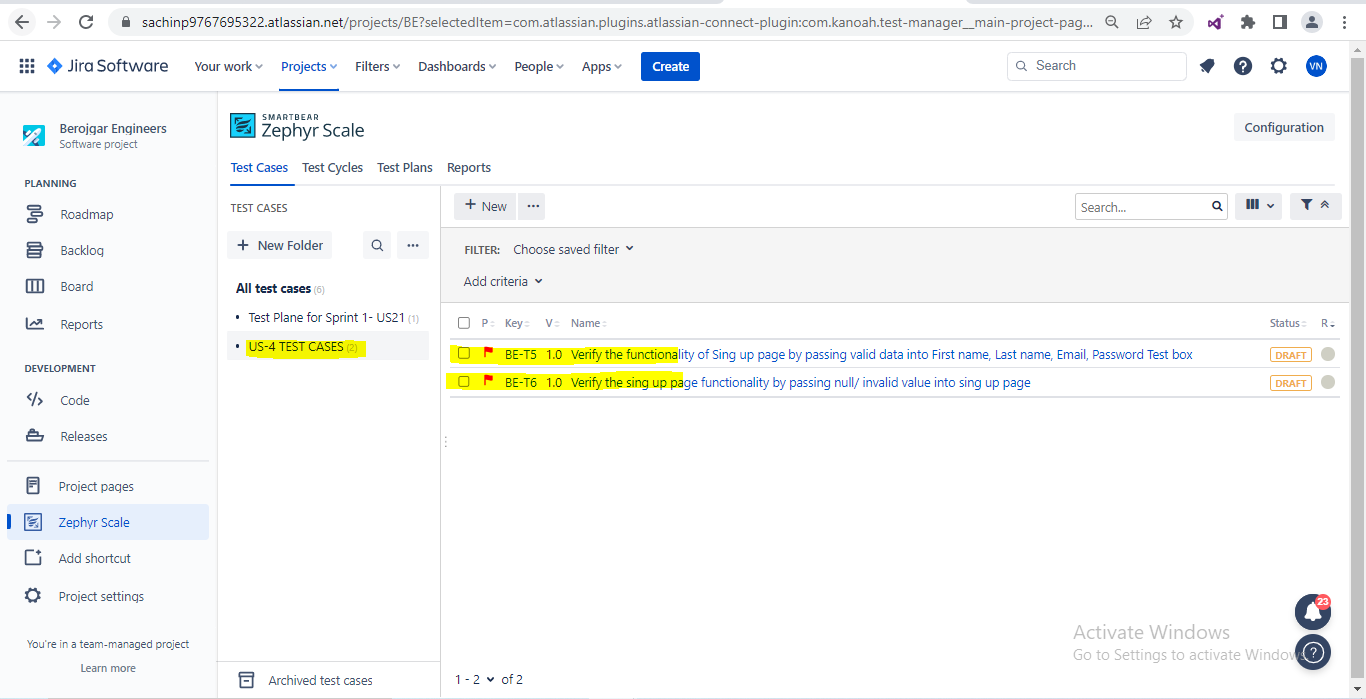
****

**JIRA Tool-**

****

****

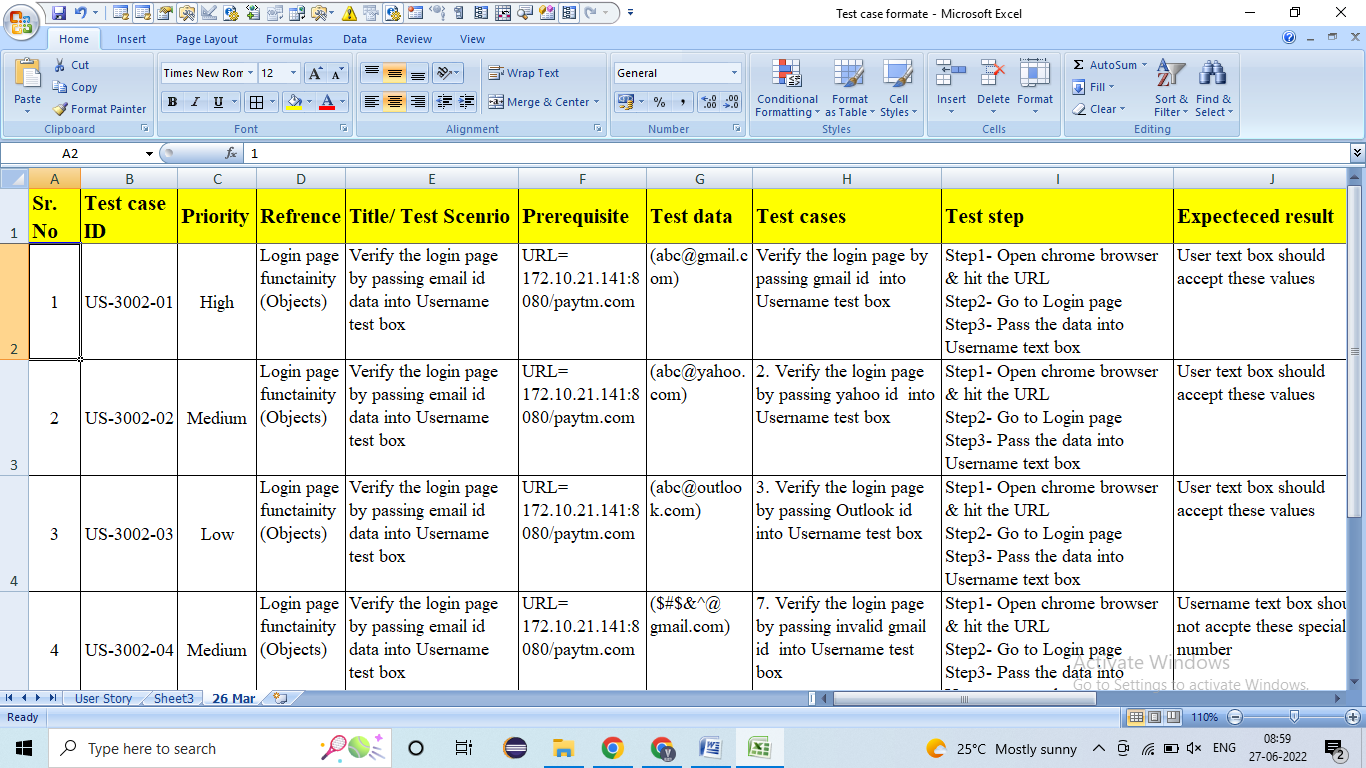
****

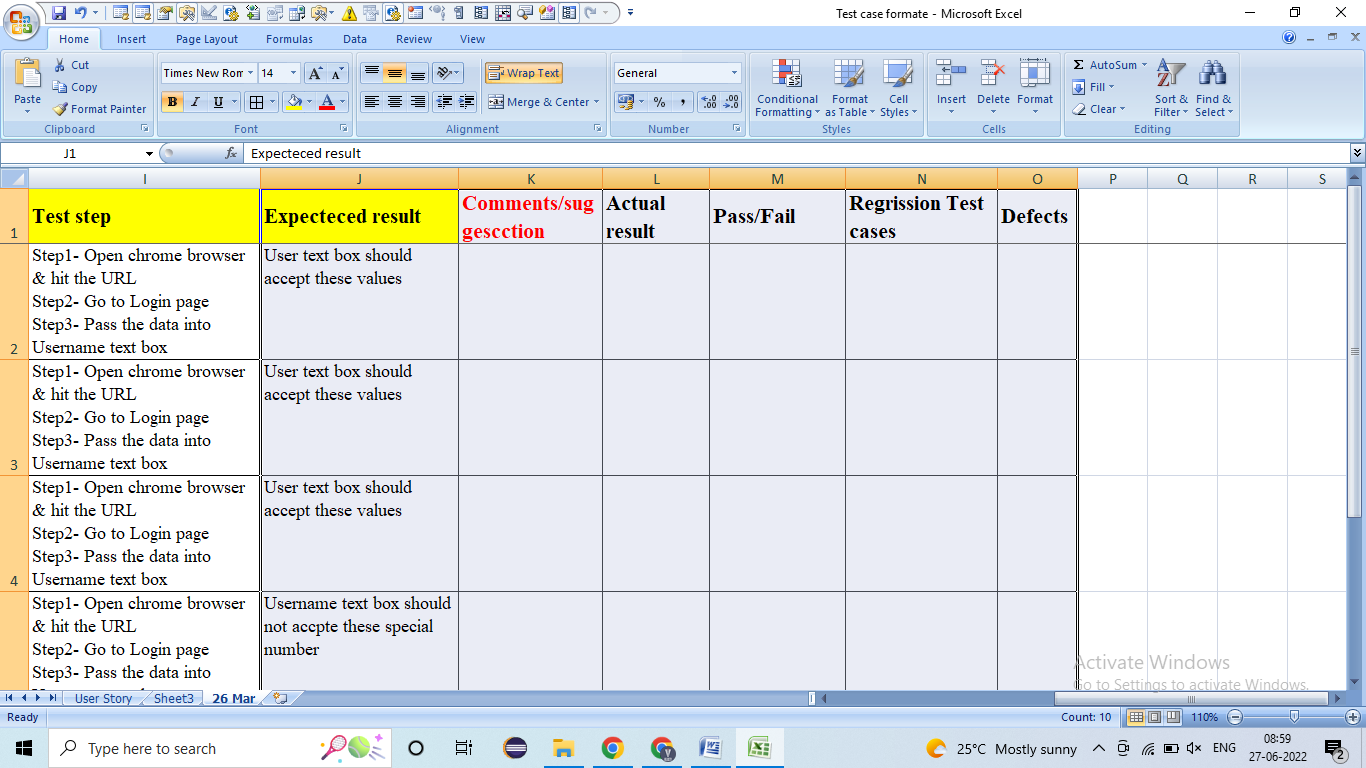
****

**Test cases written into excel sheet -**

1. **Sr. No**
2. **Test case ID**
3. **Priority**
4. **Reference**
5. **Title/ Test Scenario**
6. **Prerequisite**
7. **Test data**
8. **Test cases**
9. **Test step**
10. **Expected result**
11. **Comments/suggestion**
12. **Actual result**
13. **Pass/Fail**

**Test cases written into excel sheet-**

****

****

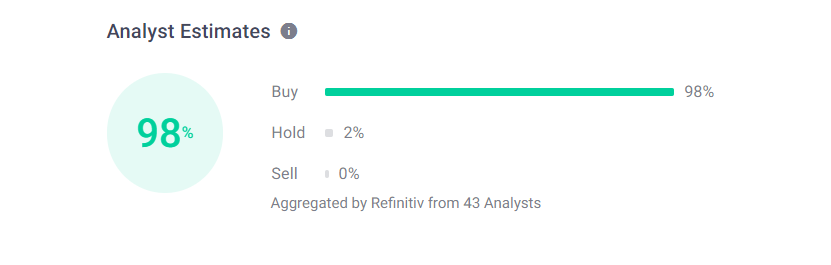
**Sample Test cases-**

**US1- Stock- Analyst estimation chart in Stock module**

**Description-** Analystestimation chart will show data for user. User will use for suggestion how it is stock moving. Take a holistic view of all the weekly/monthly options and their movement for indices and equities. Real time market movement to help you trade faster with ease. Extremely easy to use with real time tracking of Profit/Loss. Excellent features to help you trade easily in any segment you like.

**Acceptance criteria-**

1. Analystestimation chart will **show only for Nifty 100**
2. Chart will show for **Buy, Hold & Sell pattern**
3. Shows **Percentages values** for buy, hold & Sell
4. **Greater buy, hold & Sell values** will shows in Circle
5. Color will shows for **Buy= Green, Hold= Blue, Sell= red**
6. It calculates from form **previous 1 month calculation** (coming from **Groww**- **API**)

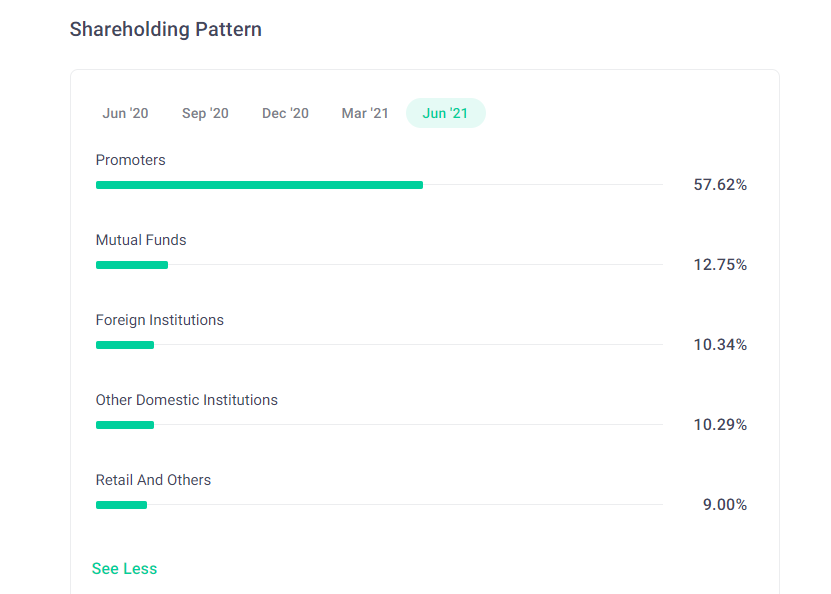


**US2- Stock- New Shareholding Pattern wedge in stock**

**Description-** Shareholding pattern will show in stock. It is used for user to see details holding of pattern. Select index/equity which have options available, set your trade preferences. Automatic selection of options where you just buy and sell. Account Summary of current day/last 150 days. Day wise summary and detailed report with P/L.

**Acceptance criteria-**

1. Shareholding pattern will show all values in “Promoters”, “foreign institutes”, “Retails & others”, “Other domestics institutes”, “Mutual fund”
2. Values in Shareholding pattern will show values in percentages only
3. Data in shareholding will shows for 5 months only
4. It will show, shows more & show less



# Test Case review-

* **Review-** Review defines to **check** **correctness and completeness of your documents**
* Test cases review 4 types

1. Self Review
2. Peer Review
3. Internal Review
4. External Review

**Self Review-**

1. In self review, Tester will do review their **own test case**

**Peer Review-**

* In Peer review, Test cases reviewed by **Senior tester/ Colleague / Test lead**
* When Tester has **completed test cases design**, Tester will **inform** to Senior tester/ Colleague / Test lead throw **Mail/ Meeting**

**Internal Review-**

* In internal review, Test cases review will **reviewed by BA**
* When Tester has **completed test cases design**, Tester will **inform** to BA throw **Mail**
* BA will set up **one meeting (Team Tool)** with Tester

**External Review-**

* In external review, Test cases review will **reviewed by Client/ UAT**
* When Tester has **completed test cases design**, Tester will **inform** to Test lead
* **Test lead** will set up **one meeting (Team Tool)** with **Client/ UAT**
* In these meeting, Every Tester will reviewed/ explain the test cases
* In my Project, we are following internal review OR **external review**
* In review, if we got the any suggestion/ feedback related test cases. Tester will **accepted these suggestion/ feedback** and Tester will modify/ change/ add test cases 🡪 **Directly modify/ change/ add test cases in the tool** OR In excel will modify/ change/ add into column “**Comments/suggestion“** present in excel sheet
* **While doing Test cases review check-**

1. Test cases should cover all functionality mentation in the US
2. Test cases should cover client business/ business logic related test cases
3. Test cases should not be duplicated
4. Test cases should cover standard format
5. Test cases should be simple / Understandable
6. Test cases should not grammatically mistake
7. Test cases should not spelling mistake

# Test Case execution-

* In TCE, Tester will **prepared the Test proof**
* In Test proof documents will contains Test cases, Screenshot of functionality related Test cases & Tables data stored
* In TCE, if we found defect then we will create/ raised these defect in JIRA/ Azure DevOps tool and inform to developer

# Defect life cycle-

* **Defect –** If tester found error in functionality then these error are called as defect
* if we found defect then we will **create/ raised these defect in JIRA/ Azure DevOps tool** and inform to developer
* **Defect life cycle –** The **journey of defect from there start to end stage**
* **Defect stage different**
* **Defect stage 🡪 New, Open, Fixed, Closed, re-open, rejected, differed**

**Closed**

**Fixed**

**Re-open**

**Reject**

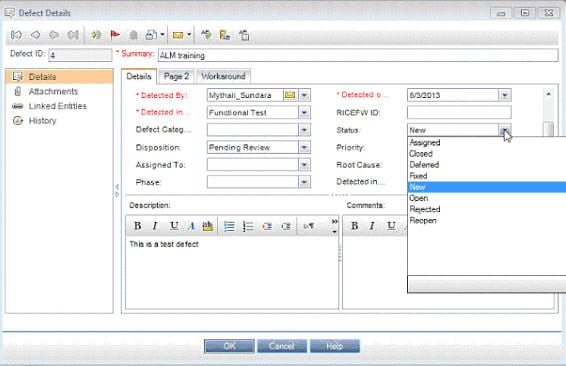
**Open**

**New**

**Differed**

**Tester Developer Tester**

* If we found defect then we will **create/ raised these defect in JIRA/ Azure DevOps tool.** Tester will mark different stage
* **NEW-** When **tester** will created/raised new defect then we will mark **defect status as New.** Tester will inform to developer throw mail
* **OPEN -** When **Developer** is **looking/analyze into defect** then he will mark **defect status as Open**
* **FIXED –** When developer found that it is **valid defect** then developer will fix these defect then he will mark **defect status as Fixed.** Developer will inform to tester throw mail
* **1. CLOSED-** When tester found that **defect has been fixed** then we will mark **defect status as Closed.** Tester will mark different stage
* **2. RE-OPEN-** When tester found that **defect has not been fixed** then we will mark **defect status as Re-open.** Tester will mark different stage
* **REJECT -** When developer found that it is **In**-**valid defect** then developer will mark **defect status as Rejected.** Developer will inform to tester throw mail
* **DIFFERED –** When Client has changed the priority of that US **OR** those defects has not been fixed in current sprint **OR** defect will take more time for fixing, then these defect has been fixed into next sprint then the **status of these defect will be mark as Differed** and these **decided by PM, Designer & BA.**



# Traceability matrix-

# Report –