# Manual part- 2

* **Tester work** **company** 🡪 **TCD** (Test scenario & Test cases), **Test Case review**, **TCE**, **Defect raised/ create**, **Different** **Report**, etc

# Test Documentation –

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

* Test Document hierarchy **Client (Project) – Automation (Java, C#, Python, etc)**

Quality control Peoples/

Testing Head -TH

Company Level document

**Test Methodology**

**Test Strategy**

**Test Policy**

Test Strategist/ Test manager – TS & PM

Project manager – PM

**TRM**

**Test Plane**

Team Lead/ Test lead – TL

**Test Scenario/ Case**

**Test Procedure/Design & Test cases review**

Team Lead – TL

Tester – Tester

Project Level document

**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 Testing | No | No | No | Yes | Yes |
| Performance Testing | No | No | No | Yes | Yes |
| GUI/ UI | No | Yes | Yes | Yes | Yes |
| Automation testing | No | No | No | 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? OR What is STLC?**

Test Scenario

Test Plan

Defect sent to developer

Test Case Execution- Test Proof

Test Case Design

Requirement Analysis

Test Summery Report/ Test Closer

Test Initiation- **TRM**

**Test Initiation -**

* In test initiation stage, PM will work
* PM will prepped a TRM documents
* TRM – Defines mapping between development stage & testing factor
* After compilation of TRM, PM will sent these TRM to team lead/ test lead

**Test plan-**

* In Test plan stage, Test lead/ Team lead will work
* Test plan will contain What to testing, Who will test When it will test, etc
* Test plan will be prepare for to deiced testing start & end date

**Test scenario & Test cases-**

* Tester will identify the test scenario & write the test cases against the test scenario
* After Test cases design, Test cases should be review

**Test cases execution –**

* When developer will sent the build then tester will do TCE
* In TCE, if we found then tester will raised/ create the defect into JIAR/ Azure develops tool & inform to developer throw mail
* Developer will fix these defect & Sent the modified build then tester will do re-testing & Regression

**Test summary & Test closer report –**

* At the end of sprint (2 week) test lead will prepared the Test summary report
* At the end of module (4 to 6 month) test lead will prepared the Test closer report

# Testing Process-

**BRS**

Testing

Development

**SRS/FRS/CRS**

**Design- HDL Test Initiation Stage- TRM**

**Coding - LLD Test Plan – against Sprint**

**STLC**

**Unit Testing Test scenario & Test Case Design – against US**

**SDLC**

**Integration Testing Test Case Execution & Test proof**

**(Install Build)**

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

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

**(**If found **defect** sent to developer) (Create into JIRA/Azure)

**Level 2 Re-testing & Regression testing (on Modified built)**

**Level 3 Final Regression Testing (**execute regression suit)

* **I have worked in all level of testing** (Level 0, Level 1, Level 2, Level 3)

# Test Plan\*-

* Test plan will be **prepared by Test leas/ Team lead**
* Test plan will be **prepared against a Sprint**
* Test plan will contains, **Who will test, What to test, When to test, How will test**
* Main purpose of Test plan, **to decide start and end date of testing activity**

**Test Plan-**

* **Field present in Test plane documents**
  1. Test plan ID
  2. Task
  3. Feature to test
  4. Feature not to test
  5. Pass / Fail criteria
  6. Test Environments
  7. Defect life cycle
  8. Testing Risk
  9. Test deliverables (Entrance Criteria & ExitCriteria)
  10. Roles & Responsibility
  11. Signature & Approvals

**Unit Testing 🡪 Integration Testing 🡪 BBT (System & functional testing) 🡪 UAT**

* **Entrance Criteria BBT –** Completion of integration testing is called Entrance Criteria BBT
* **Exit Criteria BBT -** Completion of BBT testing is called Exit Criteria BBT OR Entry Criteria of UAT
* **In my Project,** I have **helped my Test lead for preparation test plan**

# Test Scenario-

* Use cases derived from SRS/ FRS
* US derived from Use case
* Test scenario written against the US
* Test cases derived from the Test scenario

**Difference between SRS & Use Case-**

**SRS-**

* SRS defines software requirement specification
* SRS also called as FRS
* SRS derived from BRS
* SRS will be prepared by BA
* SRS will contains

1. Functional requirement
2. Functional flow diagram
3. Use Case
4. Screenshot

**Use Case-**

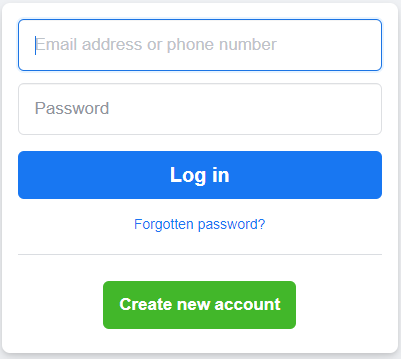
* Use Cases defines single specific requirement
* Use Cases also called as functional requirements
* Use Cases derived from SRS
* Use Cases will be prepared by BA
* Use Cases will contains

1. Description
2. Acceptance criteria

**Difference between Test scenario & Test cases\*\* -**

**Test scenario –**

* Test scenarios derived/ prepared against US
* Test scenarios prepared by Tester
* Test scenarios defines **Ways to test functionality/ What to test**
* Test scenarios defines **always in +ve ways**
* Ex.

****

**Test scenario-**

* + 1. Verify the facebook login page by passing mobile number
    2. Verify the facebook login page by passing email id
    3. Verify the facebook login page by passing password
    4. Verify log in button functionality by passing values into username & password text box

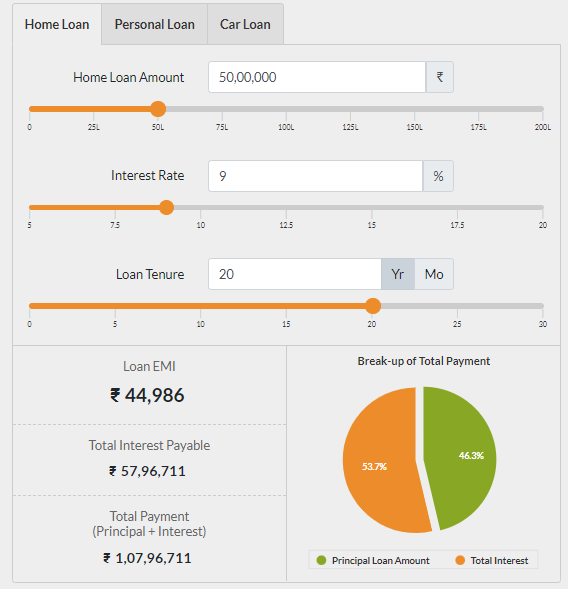
**Test cases-**

* Test cases derived against Test scenario
* Test cases prepared by Tester
* Test cases defines **How to test functionality**
* Test cases defines **always in +ve ways & -ve ways**
* **Ex. Test scenario-**Verify the facebook login page by passing mobile number
* **Test cases-**
  + - 1. Verify the facebook login page by passing starting Airtel mobile number
      2. Verify the facebook login page by passing JIO mobile number
      3. Verify the facebook login page by passing VI mobile number
      4. Verify the facebook login page by passing BSNL mobile number
      5. Verify the facebook login page by passing MTNL mobile number
      6. Verify the facebook login page by passing invalid mobile number (0000000000)
      7. Verify the facebook login page by passing mobile number in chart format (Nine nine two)
      8. Verify the facebook login page by passing null/black mobile number
* **Ex. Test scenario-**Verify the facebook login page by passing email id
* **Test cases-**
  + - 1. Verify the facebook login page by passing Gmail email id
      2. Verify the facebook login page by passing Yahoo email id
      3. Verify the facebook login page by passing Outlook email id
      4. Verify the facebook login page by passing Refdamil email id
      5. Verify the facebook login page by passing invalid Gmail email id (amol.reddy@gmail.outllok.com)
      6. Verify the facebook login page by passing invalid Gmail email id (amol.reddy.amol.reddy.amol@gmail.com)
      7. Verify the facebook login page by passing null/ blank values
* **Test scenario** - Verify log in button functionality by passing values into username & password text box
* **Test cases-**
  1. Verify the button functionality present facebook login by passing valid username (email id) & valid password
  2. Verify the button functionality present facebook login by passing valid username (mobile no) & valid password
  3. Verify the button functionality present facebook login by passing invalid username & valid password
  4. Verify the button functionality present facebook login by passing valid username & invalid password
  5. Verify the button functionality present facebook login by passing invalid username & invalid password
  6. Verify the button functionality present facebook login by passing null/blank username & null/blank password

# Test cases-

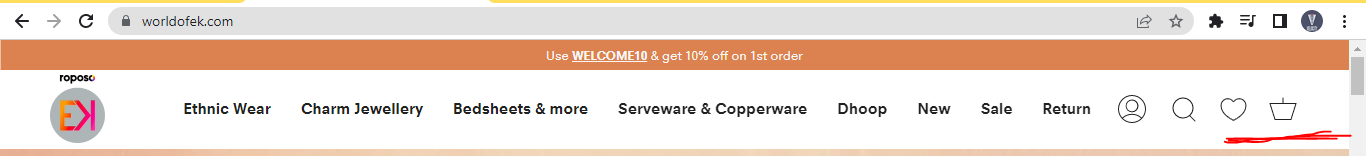
* Test cases will be written into **Project management tool/ Bug tracking tool (JIRA/ Azure DevOps tool)**
* While writing Test cases, we will consider
  + 1. Test cases should cover client business
    2. Test cases should cover functionality mentioned in US
    3. Test cases should cover GUI/ UI
* Written down Test cases for below –
* **Web site-** [**https://emicalculator.net/**](https://emicalculator.net/)**,** [**https://worldofek.com/**](https://worldofek.com/)**,** [**https://www.heenakochhar.com/**](https://www.heenakochhar.com/)**,** [**https://strfix.com/#!/**](https://strfix.com/#!/)
* **Object-** Pen, Chair, Fan, ATM, Left, Bottle, etc
* **Daily life- Whatup, Instgram, Facebook, Flipkart, etc**
* **Scenario based -**
* **Real project US –**
* **Real project-**

## Test cases Web site- <https://emicalculator.net>



* **Test scenario-**
  + - * Verify the functionality of home loan amount text box and slider
      * Verify the functionality of interest rate text box and slider
      * Verify the functionality of Loan tenure text box and slider
      * Verify the calculation and PI-chart present in home loan tab
      * **Test scenario-** Verify the functionality of home loan amount text box and slider
* **Test cases –**
  + - * 1. Verify the bypassing integer values into home loan amount text box
        2. Verify the bypassing 100/ 1000/ 10000 integer values into home loan amount text box
        3. Verify the min integer values pass into home loan amount text box
        4. Verify the max integer values pass into home loan amount text box
        5. Verify by passing 5000/ 70000/ 700000 integer values pass into home loan amount text box
        6. Verify by passing charter/ symbol into home loan amount text box
        7. Verify by passing decimal/ fractional into home loan amount text box
        8. Verify unit present near to home loan amount text box
        9. Verify values separation in home loan amount text box
        10. Verify forward slider moment present below the home loan amount text box
        11. Verify backward slider moment present below the home loan amount text box
        12. Verify range present in a slider
        13. Verify range difference present in a slider
        14. Verify color present in a slider
        15. Verify the slider moment by passing values = 0 to 10,00,000 into home loan amount text box
        16. Verify the slider moment by passing values = 50,00,000 to 5,00,000 into home loan amount text box
        17. Verify the values present in home loan amount text box by moving slider from 50,000 to 125,00,000
        18. Verify the values present in home loan amount text box by moving slider from 150,00,000 to 1,00,000
      * **Test scenario-** Verify the functionality of interest rate text box and slider
* **Test cases –**
  + - * **Test scenario-** Verify the calculation and PI-chart present in home loan tab
* **Test cases –**
  + - * 1. Verify the calculation by passing Home loan = 10L, Tenure = 5Y & interest rate = 7%
        2. Verify the calculation by passing Home loan = 100L, Tenure = 7Y & interest rate = 6%
        3. Verify the calculation by passing Home loan = 28L, Tenure = 3Y & interest rate = 6.6%
        4. Verify the calculation by passing Home loan = 122L, Tenure = 20Y & interest rate = 6.3%
        5. Verify the calculation by passing Home loan = 4Cr, Tenure = 4Y & interest rate = 11.50%
        6. Verify the calculation by passing Home loan = 1L, Tenure = 20Y & interest rate = 7.2%
        7. Verify the calculation by passing Home loan = 122L, Tenure = 24month & interest rate = 6.3%
        8. Verify the calculation by passing Home loan = 4Cr, Tenure = 48month & interest rate = 11.50%
        9. Verify the calculation by passing Home loan = 1L, Tenure = 240month & interest rate = 7.2%
        10. Verify PI-Chart session present in a home page tab 🡪 EC = Principal Loan Amount & Total Interest session should be present
        11. Verify total sum of number present in PI-Chart in a home page tab 🡪 EC = 100% total sum will shows
        12. Verify color present for Principal Loan Amount session in PI-Chart 🡪 EC = Principal Loan Amount color should be green
        13. Verify color present for Total Interest session in PI-Chart 🡪 EC = Total Interest color should be orange
        14. Verify after mouse click/ Over action on session present in PI-Chart 🡪 EC = Mouse click Session will shows with pop and other session will disappear
        15. Verify the Pi-chart calculation by passing Home loan = 122L, Tenure = 24month & interest rate = 6.3%
        16. Verify the Pi-chart calculation by passing Home loan = 25L, Tenure = 10Yera & interest rate = 6.3%
        17. Verify the Pi-chart calculation by passing Home loan = 125L, Tenure = 7Yera & interest rate = 8%

## Test cases Web site- <https://emicalculator.net>



**Test cases for Wishlist-**

**Test scenario-**

* + - * 1. Verify the Wishlist icon present in a home page
        2. Verify the Wishlist icon present in Wishlist tab

**Test scenario-** Verify the Wishlist icon present in a home page

**Test Cases-**

* + - * 1. Verify the pop message after mouse over on Wishlist icon
        2. Verify the default values & icon of Wishlist present in a home page
  1. Verify the values present in Wishlist by adding 1 product into Wishlist
  2. Verify the values present in Wishlist by adding 2 product into Wishlist
  3. Verify the values present in Wishlist by adding 4 product into Wishlist
  4. Verify the max values present in Wishlist by adding max product into Wishlist
  5. Verify the values present in Wishlist by adding & removing 1 product into Wishlist

**Test scenario-** Verify the Wishlist icon present in Wishlist tab

**Test Cases-**

* + - * 1. Verify product details / message if no product present in Wishlist tab
        2. Verify the product details present in Wishlist tab for 1 product present in a Wishlist
        3. Verify the product details present in Wishlist tab for 2 product present in a Wishlist
        4. Verify the product details present in Wishlist tab for 3 product present in a Wishlist
        5. Verify the product details present in Wishlist tab for 4 product present in a Wishlist
        6. Verify the product details present in Wishlist tab for max product present in a Wishlist
        7. Verify button present when product (if product available) seen in Wishlist tab
        8. Verify button present when product (if not product available) seen in Wishlist tab
        9. Verify product / values if product will be deselected/ removing from Wishlist tab

**Test scenario-** Verify the add to cart icon & values present in home page

**Test Cases-**

1. Verify pop message after mouse over on add to cart icon
2. Verify default values present in a add to cart icon
3. Verify values present in add to cart icon after adding 1 product in add to cart
4. Verify values present in add to cart icon after adding 2 product in add to cart
5. Verify values present in add to cart icon after adding 3 product in add to cart
6. Verify values present in add to cart icon after adding 4 product in add to cart
7. Verify values present in add to cart icon after adding max product in add to cart
8. Verify values present in add to cart icon after adding & removing product in add to cart

**Test scenario-** Verify the product details present in add to cart tab/ page

**Test Cases-**

1. Verify add of cart functionality for user without login
2. Verify add of cart functionality for user with login
3. Verify product details (name, image, amount) present in add o cart tab/ page for 1 single product added in add to cart
4. Verify product details (name, image, amount) present in add o cart tab/ page for 2 single product added in add to cart
5. Verify product details (name, image, amount) present in add o cart tab/ page for 3 single product added in add to cart
6. Verify product details (name, image, amount) present in add o cart tab/ page for 5 single product added in add to cart
7. Verify product details (name, image, amount) present in add o cart tab/ page for max single product added in add to cart
8. Verify by removing product from add o cart tab/ page
9. Verify by adding 2 quantity of product from add o cart tab/ page
10. Verify by adding 3 quantity of product from add o cart tab/ page
11. Verify by adding 5 quantity of product from add o cart tab/ page
12. Verify by adding max quantity of product from add o cart tab/ page

## Test cases Object – Pen

* **Test scenario-**
  1. Verity the metrical used in the Pen
  2. Verity the cap used in the Pen
  3. Verity the refill/ ink used in the Pen
  4. Verity the text present in a Pen
  5. Verity the writing capacity of a Pen
  6. Verify the different holding present in pen
* **Test scenario-**Verity the metrical used in the Pen
* **Test Cases-**
  + 1. Verify the pen made of a fiber
    2. Verify the pen made of a wood
    3. Verify the pen made of a metal
    4. Verify the pen made of a glass
    5. Verify the pen made of a gold/ sliver
* **Test scenario-**Verity the different cap used in the Pen
* **Test Cases-**
  + - 1. Verify the shape of the cap present in a pen
      2. Verify the tick top cap present in a pen
      3. Verify the holder/ hanger present in cap
      4. Verify the cap present for rotating cap
      5. Verify the text/ logo present in a cap in a cap
      6. Verify the locking feature present in a cap for cap
      7. Verify the different color present in a cap for cap
* **Test scenario-**Verity the different refill/ ink used in the Pen
* **Test Cases-**
  + - * 1. Verify the liquid present in ink pen
        2. Verify the different refill present in jell pen
        3. Verify the different refill present in bull pen
        4. Verify the 1/2 refill present in jell pen
        5. Verify the 3/4 refill present in jell pen
        6. Verify the 1/2 refill present in bull pen
        7. Verify the 3/4 refill present in bull pen
        8. Verify the ink flow present in pen
        9. Verify the leakage ink flow present in pen
* **Test scenario-**Verity the different writing capability of a Pen
* **Test Cases-**

Verify the pen writing capability on paper

Verify the pen writing capability on wood

Verify the pen writing capability on steel/ metal

Verify the pen writing capability on cloth

Verify the pen writing capability on glass

Verify the pen writing capability on whiteboard

Verify thewriting capability of pen with small tip/ nib

Verify thewriting capability of pen with larger tip/ nib

Verify the pen writing capability on water

Verify the pen writing capability on cold temperature at -400c

Verify the pen writing capability on cold temperature at 1000c

Verify the pen writing capability on moon temperature

## Test cases Object – Bottle

* Test scenario-

Verify the different material used for bottle

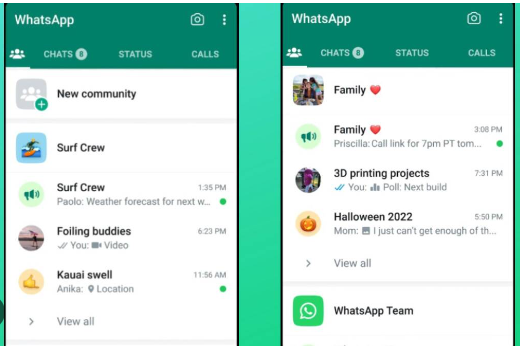
Verify the shapes, size, weight for bottle

Verify the storage capacity of a bottle

Verify different cap present in bottle

Verify different text present in bottle

## Test cases Daily life –



**Test scenario-**

* 1. Verify the community feature/ icon present in what up
  2. Verify the community creation functionality/ feature present in what up
  3. Verify the community text while creating community & group of community

**Test scenario-** Verify the community feature/ icon present in what up

**Test cases-**

* 1. Verify the new community feature/ icon for normal whatup user 🡪 EC = Community feature/ icon should be present for normal whatup user
  2. Verify the new community feature/ icon for business whatup user 🡪 EC = Community feature/ icon should not be present for business whatup user
  3. Verify the new community feature/ icon position present in whatup 🡪 EC = Community feature/ icon should present in Left side of whatup home page
  4. Verify the default text present after click on community icon 🡪 EC = Default text should be “New community”

**Test scenario** - Verify the community creation functionality/ feature present in what up

**Test Cases-**

* 1. Verify suggestion text present while creating a community group 🡪 EC = Default message shows “Max 5000 members will a present “
  2. Verify text/icon present while creating a community group 🡪 EC = It shows “Icon picture, community name only 24 charter, community description”
  3. Verify by clicking on uploading picture/ image while creating community group 🡪 EC = It shows option – Upload image, Take image, Upload Emoji, etc
  4. Verify by uploading picture/ image of 10kb for community group 🡪 EC = It should be accepted
  5. Verify by uploading picture/ image of 100kb/ 10Mb for community group 🡪 EC = It should be accepted
  6. Verify by uploading picture/ image of 100MB for community group 🡪 EC = It should be accepted
  7. Verify by uploading picture/ image of 1GB for community group 🡪 EC = It should not be accepted
  8. Verify the icons “Drag the image to adjust” feature present while uploading the community image 🡪 EC = “Drag the image to adjust” feature shows icon as Closed, upload, Back, Zoom in and Zoom out button, etc
  9. Verify the click on upload button icons present in a “Drag the image to adjust” pop 🡪 EC = Image will be uploaded in community group
  10. Verify the click on back & upload icons present in a “Drag the image to adjust” pop 🡪 EC = Previous Image tab will be display
  11. Verify by click on cancel icons present in a “Drag the image to adjust” pop 🡪 EC = community group home page should be display
  12. Verify community creation functionality while creating one community 🡪 EC = community should be created only with one community and one group

## Test cases for Scenario based –

* Lift will stop in odd floor in 8 floor building

**Test cases-**

* + 1. Verify the lift moment after pressing floor number 3 from floor number 1 🡪 EC = Lift will move in upward direction and stop in floor number 3
    2. Verify the lift moment after pressing floor number 5/7 from floor number 1 🡪 EC = Lift will move in upward direction and stop in floor number 5/7
    3. Verify the lift moment after pressing floor number 2/4/6/8 from floor number 1 🡪 EC = Lift will not move in upward direction and stop in floor number 1 only
    4. Verify the lift moment after pressing floor number 5/7 from floor number 3 🡪 EC = Lift will move in upward direction and stop in floor number 5/7
    5. Verify the lift moment after pressing floor number 2/4/6/8 from floor number 3 🡪 EC = Lift will not move in upward direction and stop in floor number 3 only
    6. Verify the lift moment after pressing floor number 7 from floor number 5 🡪 EC = Lift will move in upward direction and stop in floor number 7
    7. Verify the lift moment after pressing floor number 2/4/6/8 from floor number 5 🡪 EC = Lift will not move in upward direction and stop in floor number 5 only
    8. Verify the lift moment after pressing floor number 1/3/5 from floor number 7 🡪 EC = Lift will move in downward direction and stop in floor number 1/3/5
    9. Verify the lift moment after pressing floor number 2/4/6/8 from floor number 7 🡪 EC = Lift will not move in upward direction and stop in floor number 7 only
    10. Verify the lift moment after pressing floor number 1/3 from floor number 5 🡪 EC = Lift will move in downward direction and stop in floor number 1/3
    11. Verify the lift moment after pressing floor number 2/4/6/8 from floor number 5 🡪 EC = Lift will not move in upward direction and stop in floor number 5 only

## Filed present while writing Test cases-

* Test cases ID
* Priority
* Reference/ pre-request condition
* Test scenario
* Test cases
* Test data
* Test Step
* Expected result
* Comments & Suggestion
* Actual result
* Pass/ Fail

**Task 1-**

GROUP 1- Vendor Configuration

GROUP 2- Manage Vendor – DASHBOARD

GROUP 3- Manage Vendor – ADD VENDOR

GROUP 4- Manage Vendor – WEGDES OF COPY/ EXCEL/PDF/COLUMN

GROUP 5- Manage Governor– DASHBOARD

GROUP 6- Manage Governor– ADD Manage Governor

GROUP 7- Manage Governor– WEGDES OF COPY/ EXCEL/PDF/COLUMN

GROUP 8- Manage Area– DASHBOARD

GROUP 9- Manage Area – ADD Manage Area

GROUP 10- Manage Area – WEGDES OF COPY/ EXCEL/PDF/COLUMN

GROUP 11- Manage Category – DASHBOARD

GROUP 12- Manage Category – ADD Manage Category

GROUP 13- Manage Category – WEGDES OF COPY/ EXCEL/PDF/COLUMN

GROUP 14- Gold Rate Settings

GROUP 15- Splash Screen – DASHBOARD

GROUP 16- Splash Screen – ADD Splash Screen

GROUP 17- Splash Screen – WEGDES OF COPY/ EXCEL/PDF/COLUMN

GROUP 18- Contact Quires – DASHBOARD

GROUP 19- Contact Quires – ADD Contact Quires

GROUP 20- Contact Quires – WEGDES OF COPY/ EXCEL/PDF/COLUMN

GROUP 21- General Notification

GROUP 22- Vendor Product - DASHBOARD

**Task 2-**

GROUP 1- Vendor Product - DASHBOARD

GROUP 2- Vendor Product - ADD

GROUP 3- Vendor Product - WEGDES OF COPY/ EXCEL/PDF/COLUMN

GROUP 4- All Customer – DASHBOARD

GROUP 7- All Customer – ADD

GROUP 6- All Customer – WEGDES OF COPY/ EXCEL/PDF/COLUMN

GROUP 5- Invoice– WEGDES OF COPY/ EXCEL/PDF/COLUMN

GROUP 10- Invoice – DASHBOARD

GROUP 9- Slider – DASHBOARD

GROUP 8- Slider – WEGDES OF COPY/ EXCEL/PDF/COLUMN

GROUP 11- Slider – ADD

GROUP 13- Splash Screen – DASHBOARD

GROUP 12- Splash Screen – WEGDES OF COPY/ EXCEL/PDF/COLUMN

GROUP 14- Splash Screen - ADD

GROUP 17- Vendor Product - DASHBOARD

GROUP 16- Vendor Product - ADD

GROUP 15- Vendor Product - WEGDES OF COPY/ EXCEL/PDF/COLUMN

GROUP 19- All Customer – DASHBOARD

GROUP 20- All Customer – ADD

GROUP 18- All Customer – WEGDES OF COPY/ EXCEL/PDF/COLUMN

GROUP 21- Slider – DASHBOARD

GROUP 22- Slider – WEGDES OF COPY/ EXCEL/PDF/COLUMN

GROUP 23- Slider – ADD

# Test cases Review-

* Review – To check correctness and completeness of your own documents
* 4 types test cases review
  + - * **Self review**
      * **Peer review**
      * **Internal review**
      * **External review**
      * **Self review-** When tester will write test cases into project management tool (JIRA/ Azure DevOps tool), then **tester will review their own test cases**
      * **Peer review/ colleagues review –** In peer review test cases will be **review by senior tester/ colleagues/ Test lead.**
      * When tester will write test cases into project management tool (JIRA/ Azure DevOps tool), then tester will inform to **senior tester/ colleagues/ Test lead throw mail**.
      * **Internal review –** In internal review test cases will be **review by BA.**
      * When tester will write test cases into project management tool (JIRA/ Azure DevOps tool), then tester will inform to **BA throw mail.**
      * **External review -** In external review test cases will be **review by Client/ UAT team.**
      * When tester will write test cases into project management tool (JIRA/ Azure DevOps tool), then tester will inform to **Test lead.**
      * Test lead will set up one meeting with **Client/ UAT team**
      * In meeting, Test leas will start the meeting, individual Tester will review their test cases form client/ UAT team
      * **In my project**, we are following **External review.**
      * In cases review, If we got comments/ suggestion Client/ UAT team/ BA then we will accept these **comments/ suggestion and modify the test cases**.
      * **In Test cases review, Client/UAT team will check**
        1. Test cases should be client business
        2. Test cases should be functionality in US
        3. Test cases should be GUI/ UI
        4. Test cases should be simple & Understandable
        5. Test cases should be proper format
        6. Test cases should grammatically correct.

# Test cases execution –

* After completion of **coding against the US** then **developer** will **inform to tester through mail** (Throw JIRA/ Azure DevOps tool) & in mail sent unit testing documents.
* When we got build from developer then tester will start the testing/ TCE.
* SIT tester will open the application in SIT environments
* **In TCE**, Tester will prepare the **Test proof.**
* In test proof we will contains- Test cases and functionality screenshot

# Defect\*\* –

* **While doing TCE, if we found defect** then tester will **raised/ create the defect into JIRA/ Azure DevOps tool.**
* While creating/ raising the Defect, we have to pass data in or filed present

**Summary**\* – Short description about the defect

**Steps to re-product** – Defect found step add

**Status\***- New/ Open/ Fixed/ reject/ Closed/ Re-open/Differed

**Priority**- Very High, High, Medium & Low

**Severity**\*- Critical, High, Medium & Low

**Attachment/ screenshot** –

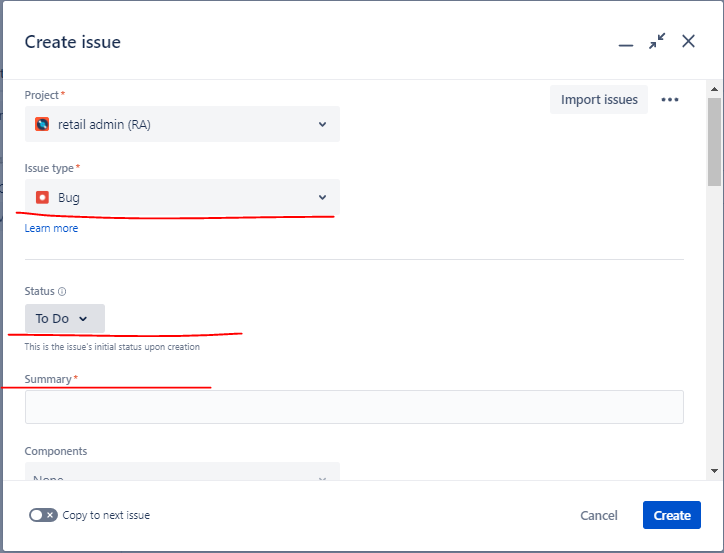
**Reported by\*** – Tester name

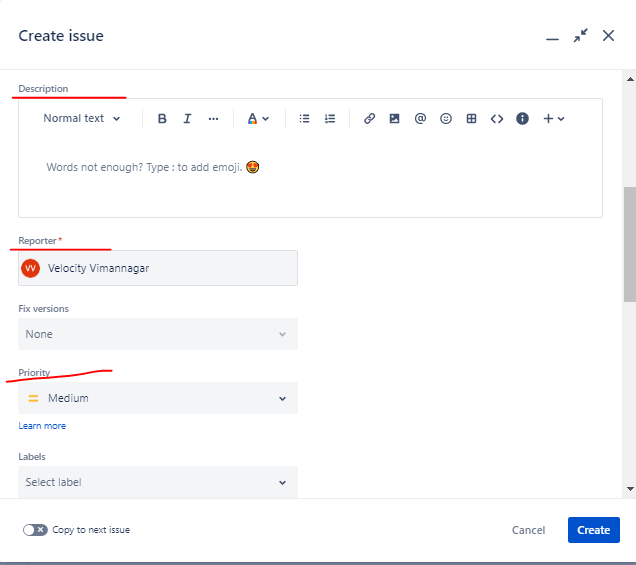
**Assign To\*** – Developer name

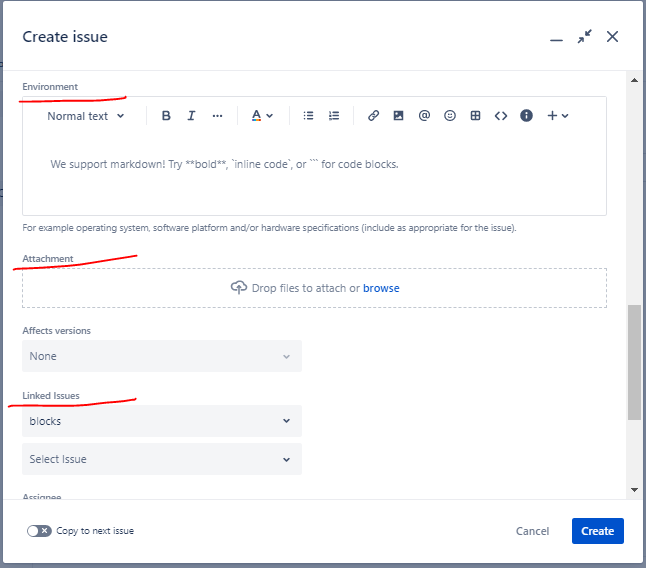
Created date – system date

**Defect link with US –**

**Environments –** SIT Environments







## Defect life cycle\*\*-

* Defect life cycle define **defect journey/ Life spam** from their opening status to closing state.
* Defect life cycle stage – New/ Open/ Fixed/ Reject/ Closed/ Re-open/Differed

**Closed**

**Fixed**

**Re-open**

**Open**

**New**

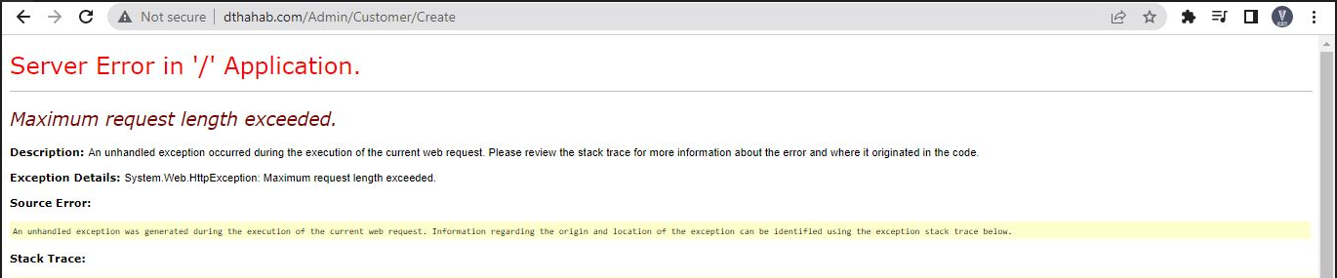
**Rejected**

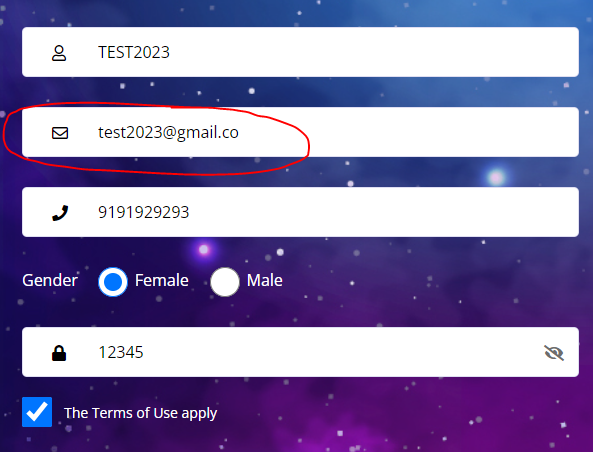
**Differed**

* **New-** When tester will found defect in TCE, then tester will raised defect in JIRA/ Azure DevOps tool. **Tester** will mark **defect status as “New”** in JIRA/ Azure DevOps tool. Tester will inform to developer throw mail
* **Open-** When developer is **analysis/ looking into the defect**, then developer will mark defect **status as “Open”** in JIRA/ Azure DevOps tool.
  1. **Fixed –** When developer found that **defect is valid** then **developer** will fix the defect **status as “Fixed”** in JIRA/ Azure DevOps tool. Developer will inform to tester throw mail.

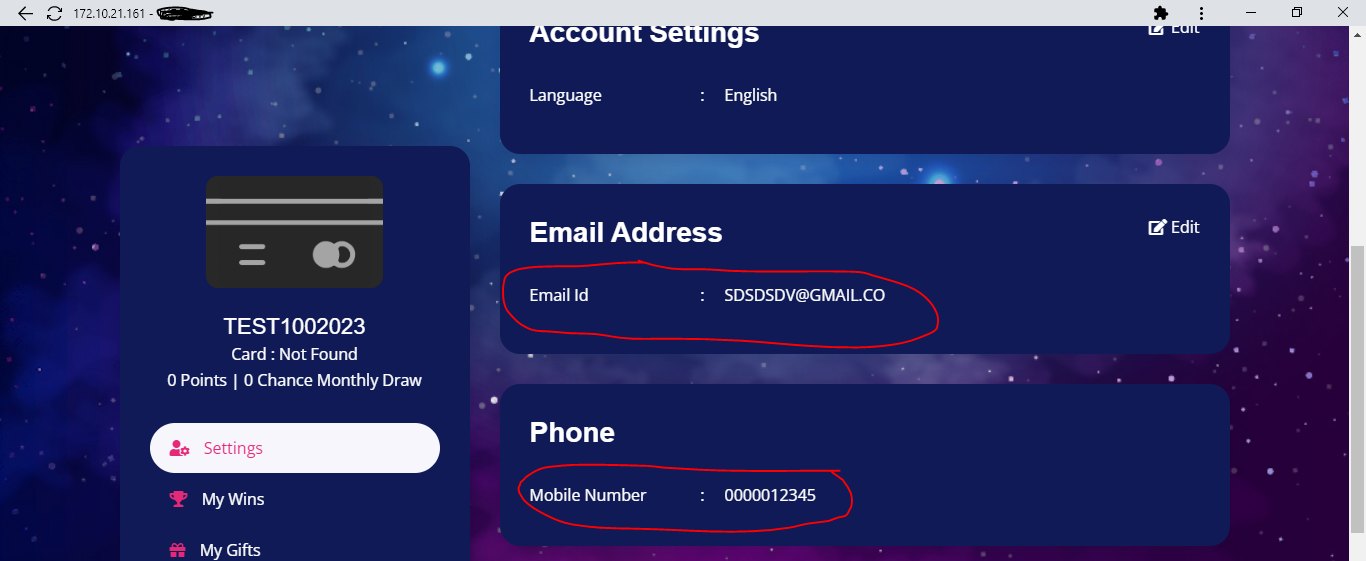
1. **Closed –** Tester will do re-testing & regression and if found that defect has been fixed/ working fine then **Tester** will mark **defect status as “Closed”** in JIRA/ Azure DevOps tool. Tester will inform to developer throw mail
2. **Re-open –** Tester will do re-testing & regression and if found that defect has not been fixed/ not working correctly then **Tester** will mark **defect status as “Re-open”** in JIRA/ Azure DevOps tool. Tester will inform to developer throw mail
   1. **Reject –** When developer found that **defect is In-valid** then **developer** will reject the defect and mark defect **status as “Rejected”** in JIRA/ Azure DevOps tool. Developer will inform to tester throw mail.
   2. **Differed –** If **client change the priority of the US** then the defect which is raised against that US, these all defect will be consider are differed defect **OR** If defect has not fixed in current sprint then due to these functionality we are covering in next sprint. These defects will be marked as differed defect. Differed defect will be decides by **PM,** Designer & BA.

## Defect type-

* **Blocker defect/ Show stopper defect –** Those defect which stop testing activity, these defect are called as show stopper / blocker defect.
* **Ex.** Core functionality/ Mani functionality is not working
* **Ex.** In my project, when I am uploading image of customer greater then 10MB then it shows error time error 
* **Re-producible defect – T**hose defects which is present in SIT environments and after fixing also still shows/ present. Becz these defect present in SUT but not occurred in production.
* **Ex.** Email id will accepted without gamil.com – these issue present in SIT becz SIT environments has some limitation (functionality not supported)



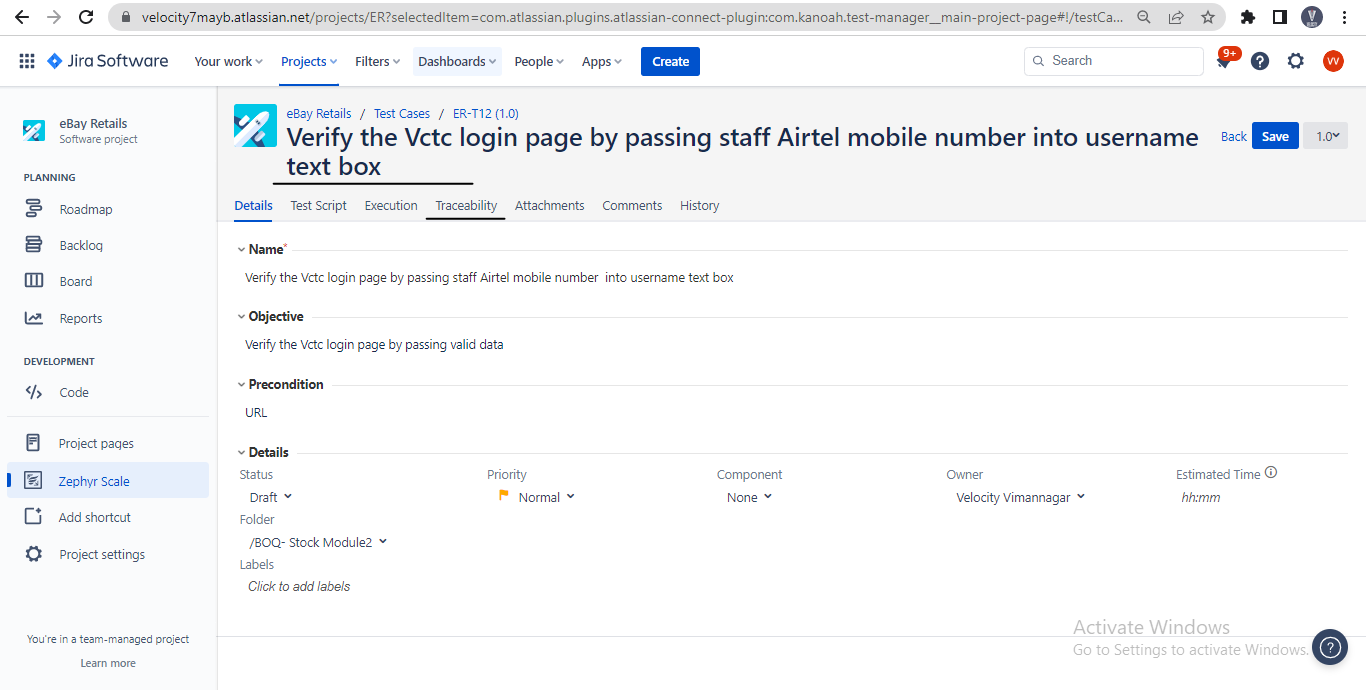
* **Conditional sing-off/ Bug released – T**hose defects which are present in SIT environments and after fixing also still shows/ present. Becz these defect present in SUT but not occurred in production.

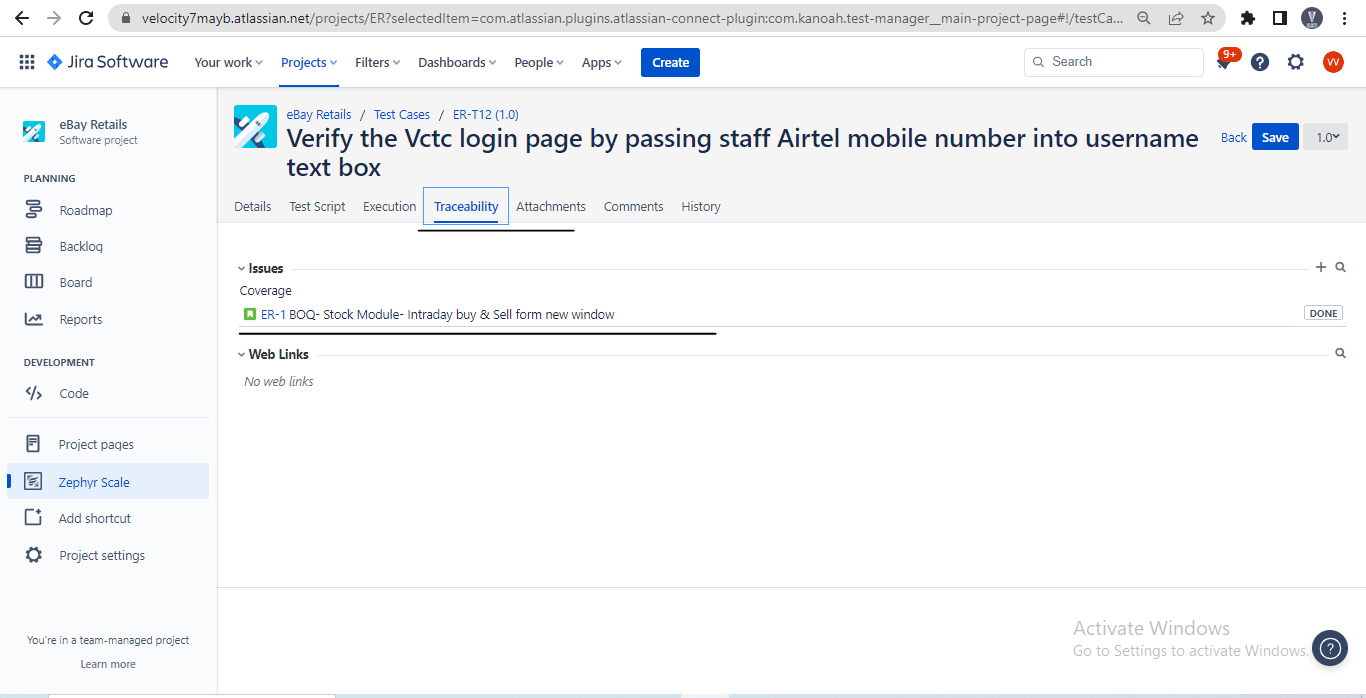


* **Duplicate defect-**Duplicate defect defines the defect which in **similar nature/ functionality**
* **Ex.** Paytm – Recharge module – Airtel recharge not happened – Defect ID = 342632 – Fixed
* **Ex**. Paytm – Recharge module – JIO recharge not happened – Defect ID = 342633 – Duplicate
* **Ex.** Paytm – Recharge module – VI recharge not happened – Defect ID = 342634- Duplicate
* **Defect density –** Number of defect found / Number of code written for functionality that ratio is called as defect density
* **Ex.** Paytm – Recharge module (Coding = 100 line) – tester found 10 defect –
* Defect density = 10/ 100 = 10%
* **Defect cluster –** The defect which is present near the some functionality.
* **Latent defect-** Those defect which will hide in another functionality/ another defect
* **Ex.** Paytm – Recharge module – Operator drop down is not working then hide functionary (Circle drop down) also not working
* **Test Bed/ Test Suite-** The similar test cases/ The bunch of test cases are collecting for execution
* **Released Note-** Whenever developer will merger code with existing code (For new development or bug fixes) then developer will add released note while pushing code in GitHub tool.

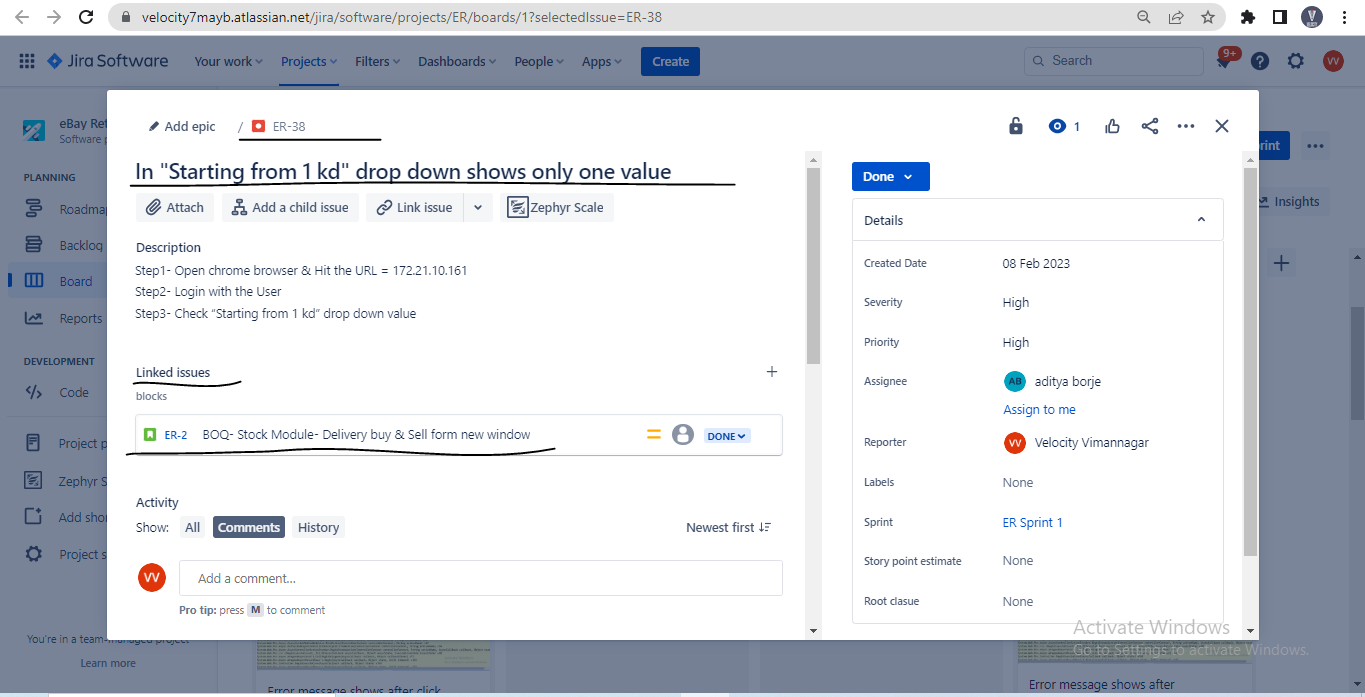
# Traceability Matrix-

* In Traceability matrix, **Test case and defect are linked/ mapped with US/ requirement**
* 2 types In Traceability matrix
  + 1. **Forward Traceability matrix** - In forward traceability matrix **Test case are linked/ mapped with US/ requirement**





* + 1. **Backward/ Reverse Traceability matrix** - In backward traceability matrix **Defect are linked/ mapped with US/ requirement**

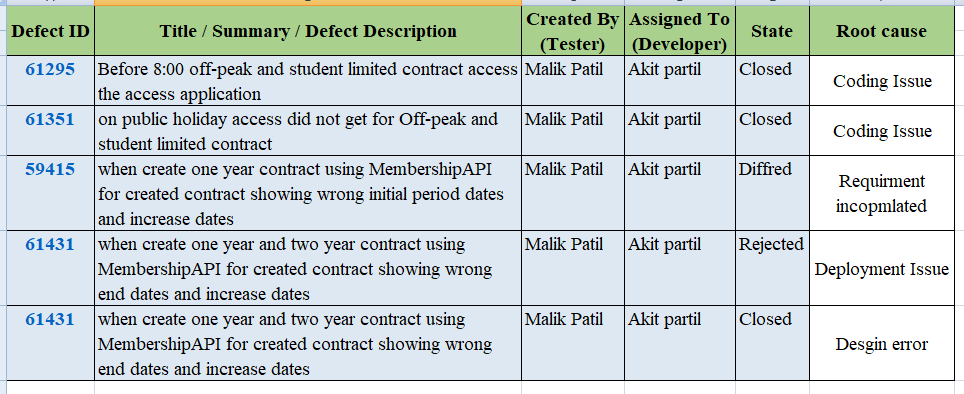


# Report-

* Test plan report 🡪 Test lead 🡪 against sprint
* Test cases design report 🡪 Tester 🡪 against US
* Test proof report 🡪 Tester 🡪 against US/ TCD
* Defect report 🡪 Tester 🡪 against US/ TCE
* Test summary report 🡪 Test lead 🡪 against sprint
* Test closer report 🡪 Test lead 🡪 against completed module

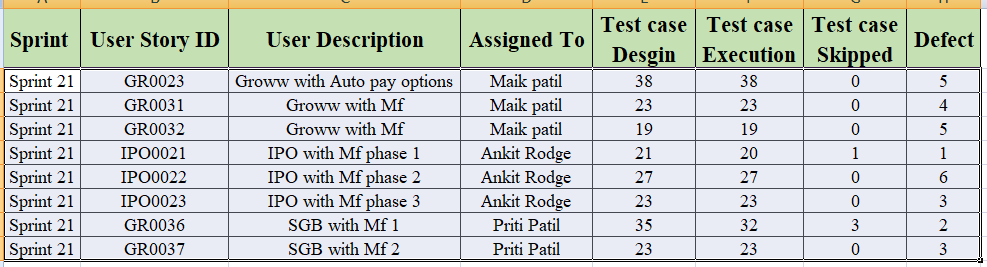
**Defect report-**

* Defect report will be **prepared by Tester against the US**



**Test Summary report-**

* Test summary report will be **prepared by Test lead against the Sprint**
* For preparation ofTest summary report, I have helped my Test lead



**Test Closer report-**

* Test closer report will be **prepared by Test lead against the completed module**

**Interview Question-**

* 1. **What are problem/challenges you have faced in your last project? OR in your testing carrier?**
* **Answer-** In my carrier I have faced problem/challenges
* If we don’t have domain knowledge / project knowledge
* If developer is not communicated properly (Developer not acting defect)
* If frequent change occurred in the requirement/ US (re-work)
* If we have less test data for testing
* If we lack of environments (Only SIT environments present)
* If we have less resources in testing (Tester, doc preparation, etc)
* If lack of budget
  1. **What is your roles & responsibility in your last project\*\*?**
* **Answer-**
* Requirement analysis/ Requirement understand / Understand SRS
* Conduct the Grooming meeting to clarification about the requirement
* Identify the test scenario against the US
* Writing the Test cases against the test scenario
* Test cases review – External review
* Test cases execution (Test proof)
* Defect report/ Defect raised & inform to developer
* Preparing the Traceability matrix
* Helping test lead for preparation of test summary report
* Attending daily stand up/ Scrum meeting
* Client interaction / Client review / demo (Sprint review meeting)
* Attending sprint retrospective
  1. **What are your approaches, if developer is not accepting the defect\*\*?**
* **Answer-** When we found defect then we will Crete these defeat into Bug tracking tool
* While creating defect, we will attached screenshot of the defect & inform to developer
* Still developer is not accepting, I will re-produce the defect again in SIT environments, if defect present then I will inform to developer
* Still developer is not accepting, We will take call with developer and share the screen and show cases the defect
* Still developer is not accepting, We will inform to Test lead
* Still developer is not accepting, We will inform to developer, Test lead, Designer & PM throw mail
* Still developer is not accepting, We will inform same in daily stand up meeting
  1. **When we know that we have to move into next US OR when you know that you have completed testing of a US?**
* **Answer-** All Test cases should be executed
* All defect has been re-tested
* When traceability matrix should be prepared / Test cases & defect are linked with US
  1. **What is ready & done state of a US?**
* **Answer-** Ready – When US is ready for deployment, then we will mark US status as Ready
* Done – When US is deployment then we will mark US status as Done
  1. **How many test cases you will design per US/ day and how many you will execute per day?**
* **Answer-**
* TCD it total depends on complexity of functionality. If complexity more the more test cases and if complexity is less then less test cases.
* **An average**, I will design **30 to 40 test cases per US / day**
* TCE, it depends how the complex & efforts the functionality
* **An average**, I will execute **20 to 25 test cases per day**
  1. **How many defect you will raised per US/ day**
* **Answer-**
* TCE it total depends on how the developer has done the coding against functionality. If coding done not properly the we will more get defect and If coding done properly the we will get less defect
* **An average**, I will found **4 to 5 defect will be found per US/ Day**
  1. **How many numbers of defects you will raise in last project/ last module?**
* **Answer-** In last project/ Module I have raised so many defects but difficult to tell you exact number.
* But I remember, one defect ID which I have raised that is Defect ID = 329090 (One these defect developer & me has done more efforts)
  1. **What last US you have testing in your project**
  2. **What last defect you have raised in the sprint/ last project**
  3. **What critical defect you have raised**