 **---------------------------------------------------------------------**

**Manual Testing by Vaibhav Sir**

**---------------------------------------------------------------------**

**Contents Page No.**

1. Syllabus 02
2. Team 03
3. SQA 03
4. SDLC 04
5. Fish Module 07
6. Waterfall Model 09
7. V-Model 10
8. Agile Model 13
9. Agile Meetings 14
10. Technology used in Project 18
11. Environment 19
12. Unit Testing and Integration Testing 20
13. Sanity/Smoke Testing 22
14. System and Functionality Testing 23
15. Functional Testing 24
16. Non Functional Testing
17. Security and Performance Testing
18. Retesting and Regression Testing
19. User Acceptance Testing
20. Monkey Testing, Exploratory Testing, Adhoc Testing.
21. Priority and Severity

**Syllabus**

**Manual Testing – I.**

* Software Development Life Cycle.
* Waterfall Model.
* V-Model.
* **Agile Model/Agile Methodology (90-95% Used in Project)**

**Types of Testing**

* Sanity / Smoke Testing
* System and Functionality Testing
* Usability Testing
* Retesting and Regression Testing

**Manual Testing – II.**

**Real Time Part –** Test Case Design, Test Case Execution, Test Case Review, Defect Logging, User Story Understanding.

**Database Testing.**

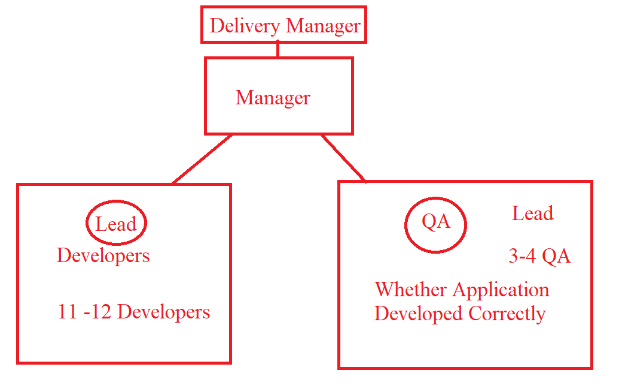
**SQL Language | Application – Oracle Live etc**

**API / Web Service Testing.**

SOAP and REST Services  
Tool – Postman

**JIRA – Project Management Tool.**

**2 Live Projects.**

**Teams**

**Project Team.**

* **Delivery Manager (1)** – Project Deliver to Client.
* **Project Manager (1)** – Manage the Project (Handle the Dev and Testing Team)
* **Business Analyst (1)** – Interact with Client related to Clients Requirement.
* **Designer / Design Architect (1)** – Lead of the Dev Team. – Project Design
* **Development Team (10/11/12 Dev**) – Project Coding / Dev the Application.
* **Testing Team (3/4 QA)** – Test the Application.

**Support Team.**

* **Support Team will work on the existing issue /Tickets raised by the client.**
* **Project Team(1) –** will manage the project work
* **Developer Team (2-3) –** Manage Coding Part
* **Testing Team (1) –** Test the Application.

**What is Your Team Size? 🡪 18 People.**

**In which team you have worked? 🡪 Project Team.**

**Software Quality Assurance (SQA)**

* SQA is a process used to **measure and monitor** the development and testing of the application.
* SQA is done by **BA.**

**SQA Considers**

1. **Client’s/Customer’s Requirements:** The requirement fill full to the client e.g Project is Banking / e-Commerece / Healthcare etc domain.
2. **Client’s/Customer’s Expectations :** Suppose Team is developing Banking ATM related Project – ( Privacy : Personal Data should be kept private/Secured)
3. **Cost / Budget of the Project :**

Estimation /Cost required for completing the Client’s Application

E.g (12 Dev + 3 QA \* 2000)

1. **Delivery / Deployment Time (Duration).**

In How many Days/Weeks/Months/Years will team complete the Application?

1. **Risk in the Project.**

What if our company fail to deliver project within given time.  
Government MPSC Exam – Online – 10 Oct – Deadline 07 Oct Product Application should be ready. IF Application is not ready till 10 Oct-🡪 Exam Cancelled 🡪 That Risk factor discussed in this phase.

1. **Maintenance.**

(1-2 Month)

What is SQA?

Explain the SQA.

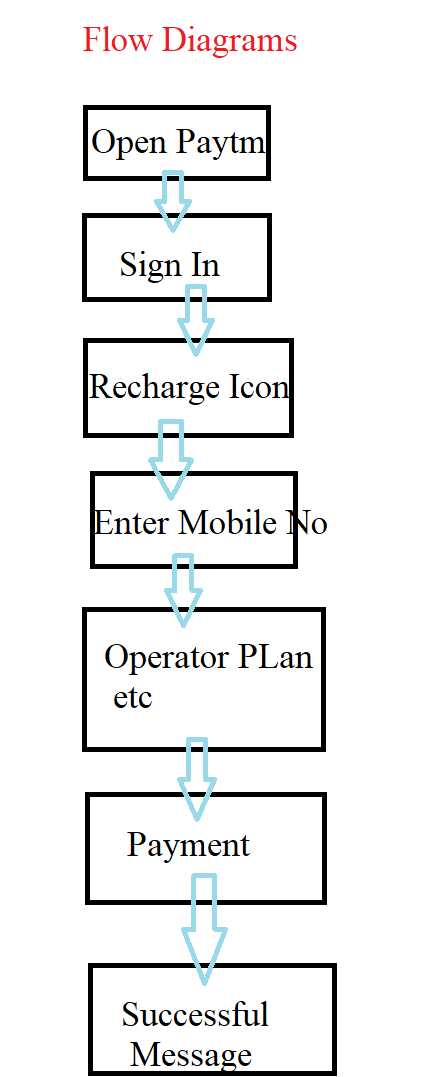
**Software Development Life Cycle.**

1. **Information Gathering.**
2. **Analysis.**
3. **Design(High Level / Low Level)**
4. **Coding**
5. **Testing**
6. **Support / Maintenance**
7. **Information Gathering (BA) (BRS)**

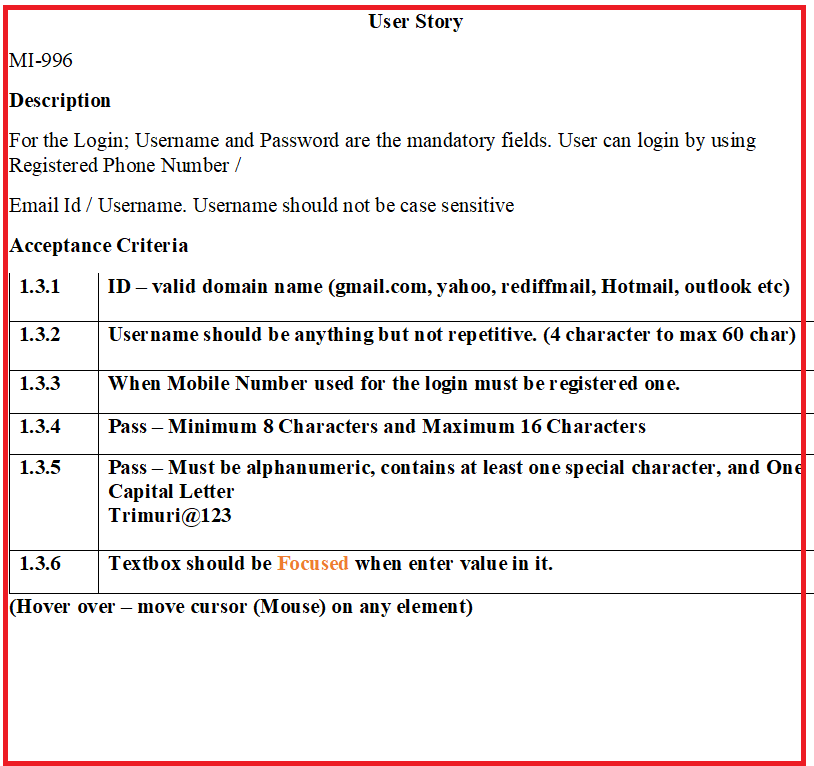
* In the information gathering is done by BA.
* In the information gathering stage, BA collects all the requirements related to the **client business.**
* After collecting all the Business related requirements BA will prepare **BRS (Business Requirement Specification) document.**
* BRS is a Company level document, Dev and QA don’t have access of it.
* E.g Hotstar App – Business 🡪 Collect Subscription Amount from the End Users.

1. **Analysis (BA) (SRS)**

* Analysis is done by BA.
* In the Analysis phase BA Collects are the information / requirement related to the **Client’s Application (Software Specification).**
* Based on the software requirements BA prepares the **SRS (Software Requirement Specification) Document.**
* SRS is also known as FRS – Functional Requirement Specification / CRS – Customer Requirement Specification.
* Once SRS document is prepared BA will send that Document to Project Manager Later PM will share the Document with Testing and Development Team.
* **SRS is a Project level Document. (i.e QA Team and Dev Team has access of it)**

**SRS Documents**

* **Functional Flow Diagram.**
* It consist of the All the requirement Flow diagrams of the application/Functionality
* Flow diagram reprints the Flow of the Application9 The steps in which Application navigates)
* **Functional Requirements.**
* All the requirements related to the Application / Software.
* E.g – Facebook Logo, TagLine.
* Email ID and Password Textbox for the Login.
* Login Button Color.
* Create New Account.
* **Forgot Password Button.**

****

* **User Case. (User Story)**
  + **Description**

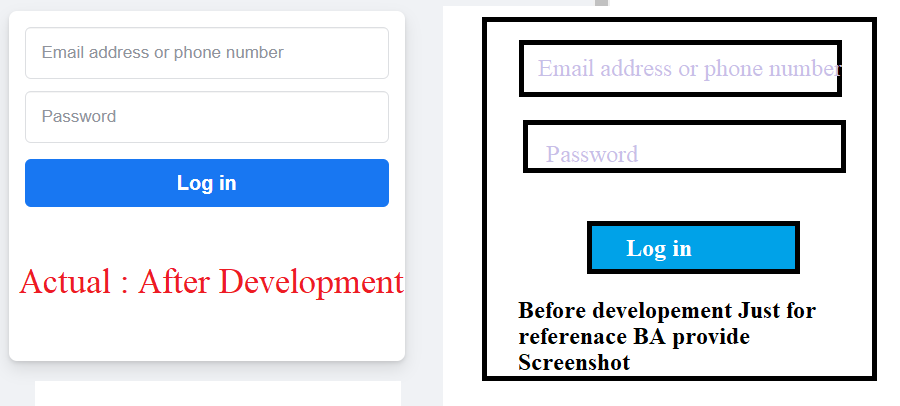
Complete detailed information about every single requirement. E.g Forgot Password button should be displayed below the Log in button. Forgot Password? Text should be displayed in blue color and consist of hyperlink.

* + **Acceptance Criteria**

In the Acceptances criteria It consist of Do’s and Don’ts about the user story.

**PASSWORD**

|  |  |
| --- | --- |
| **Accept** | **Reject** |
| 8 – 16 Digit | Less than 8  More than 16 |
| Special Character 1 |  |
| 1 Capital Letter  1 Small Letter  1 Number |  |

****

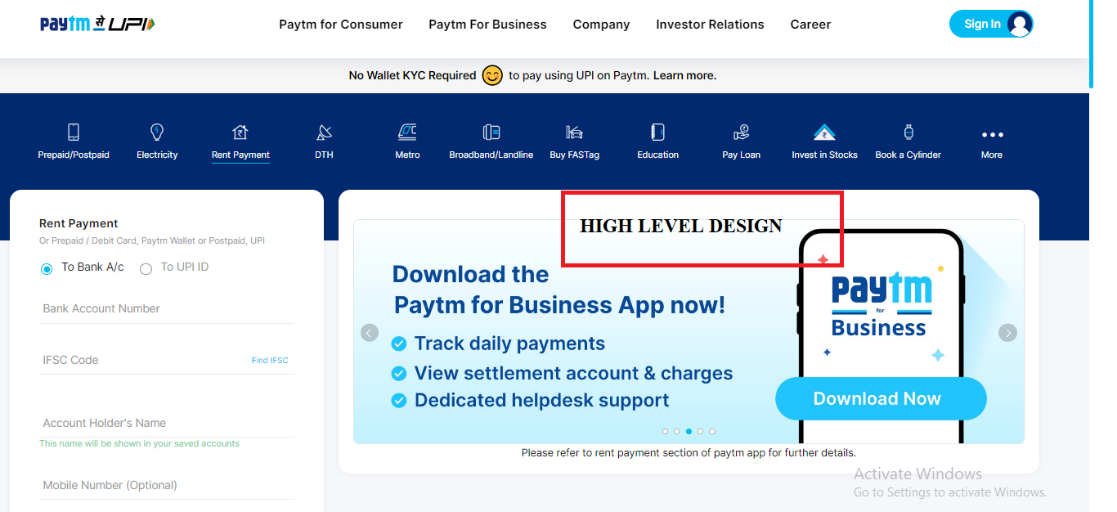
* **Screenshot / Prototype**

Screenshot is a document prepared by BA before the development process so that Developer/Designer can understand the requirement more clearly.

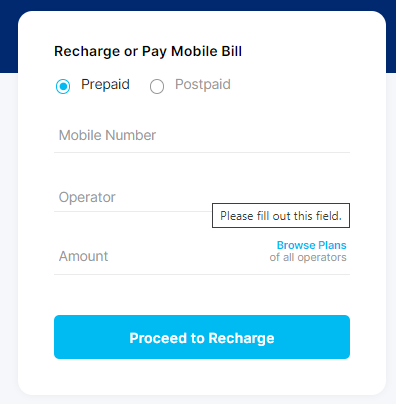
1. **Design**

In the Design phase Designer / Design Architect works.

* 1. High Level Design



* 1. Low Level Design

****

1. **Coding**

* In the Coding phase developers are developing the code against the requirement.
* Logic to be implemented to work the functionality.
* When developers finish the coding Part then Developer will provide a **Build** to the QA.

1. **Testing**

* In the Testing Phase QA performs the different testing on the build(Application) when QA receives the build by dev.
* QA has to Design (Write) Test Cases, Execute the Test Cases and While Testing IF found Bug/Defect then QA has to report a Bug. (Defect Logging)

1. **Maintenance.**

* Work on the existing issue in build/application

**FISH Module**

Information Analysis Design Coding Testing Support/Maint.

Gathering (BRS) (SRS) (HLD, LLD) (LLD) (BBT)

Review Review Review WBT BBT

Static Testing/ **Verification** Dynamic Testing/ **Validation**/

Quality control Quality assurance

**Review: Review is the Process where documentation (Activities) is to be checked.**

* When **BRS** document is prepared by the **BA**, **Client will check** the prepared **BRS** document.
* In the Analysis when **SRS** document is prepared by **BA**, **Client will check** the prepared **SRS** document.
* In the Design Phase, **Designer** will create a **Design**;🡪 **BA, PM and Client -🡪 will review the design.**
* In the Coding Phase 🡪 Developer will write the Code 🡪 **Designer will review** the Code of Developer. + Developer will check the Code and Run his code in Development Env (White Box Testing) 🡪 developer will check the Code.

|  |  |
| --- | --- |
| **White Box Testing** | **Black Box Testing** |
| Perform by developer | Perform by QA |
| In the WBT, Dev will check the Logic, loops, conditions, etc. code part. | In the BBT, QA will perform the different Testing. |
| 1. Unit Testing. 2. Integration Testing. | 1. Smoke / Sanity Testing. 2. Functional and Non Functional Testing. 3. Retesting and Regression Testing. |
| WBT is also known as Code Level Testing. | BBT is also known as System and Functionality Testing. |

|  |  |
| --- | --- |
| **Static Testing** | **Dynamic Testing** |
| In the Static Testing BA will confirm the BRS and SRS Document from the Client. | QA will test the Application using system and functionality Testing |
| Static Testing is also known Verification / Quality Control / In progress Testing | Dynamic Testing also known as Validation/ Quality Assurance/ End Progress Testing |

**Verification**

* Verification is the process which confirms whether the Application/build meet the specification.
* It includes checking the document, design programs etc.
* It consist of method like review, walkthrough/inspections, White box testing.

**Validation.**

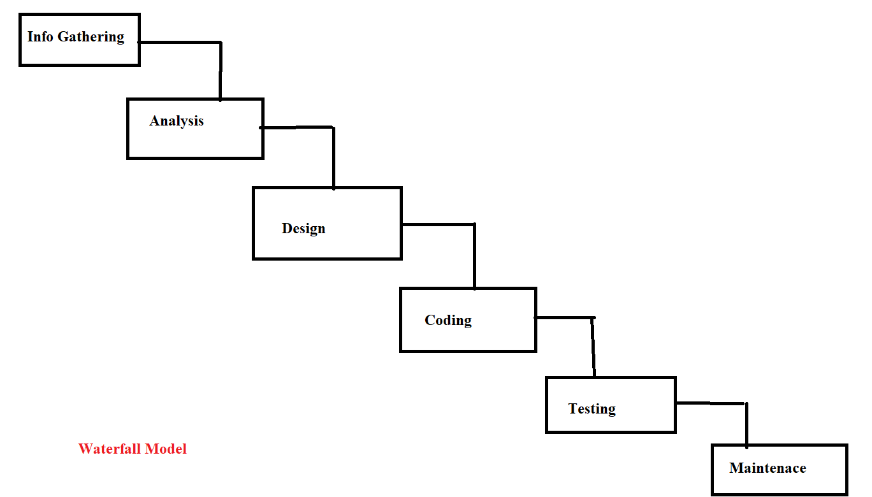
* Validation is the process which confirms whether the application /software meet the requirement.
* In the Validation part QA check/confirms whether the requirement is fulfill according to the product/application need.
* QA performs Black Box Testing for the Validation.

------------------------------------------------------------------------------------------------------------------

Interview Questions

1. What is SDLC
2. What is Difference between SDLC and STLC(Manual Part – 2)
3. When QA will start Testing?
4. What is difference between WBT and BBT?
5. What is difference between Static Testing and Dynamic Testing?
6. What is difference between Verification and Validation?
7. What is your Team size?
8. How many developers are working in your team?

---------------------------------------------------------------------------------------------------------------------

**Waterfall Model**

* Waterfall model is **sequential model** for software development and testing.
* i.e after completion of one phase then another phase/stage will start.
* If we found any defect while testing at that time QA will not send back to dev.

**Disadvantages.**

* Deployment Time / **Delivery time is not fixed** in the waterfall model. (min 3 months)
* Back track is not possible.

**Advantage.**

* When the requirement is **small**, **clear** and **constant** (Requirement is not continuously changing) at that time we can use waterfall model.
* When the Budget of the Project is less / Project – Short.
* Waterfall model is used when your **application is not complex.**

---------------------------------------------------------------------------------------------------------------------

**V-Model**

**LCD (Life cycle development) LCT (Life cycle Testing)**

**Verification Validation**

Information gathering (**BRS**) Assessment of development plan.

Analysis (**SRS**) **pdf format**. Prepare test plan.

Requirement Testing/ Understand.

Design (**HLD**, **LLD**) Design phase testing (Review)

Coding (**LLD**- dev. team) Program phase testing (**WBT**)

Test case design (**TCD**)

Install **Build**/ Application System & Function testing (**BBT**) – (**TCE**) -> Pune

User acceptance testing (**UAT**) -> USA (Client side)

Knowledge transfer (**KT**)

**Maintains**/ Support **CR** (Change request)/ **Post Mortem testing**

**DRE** (defect removal efficiency).

* V-Model stands for Verification and Validation.
* The drawback of the waterfall model **Deployment/Delivery Time is not fixed and Backtrack** is not possible are overcome in the in the V Model.
* In the V-Model Verification stages are mapped with the Validation stages.
* Verification and Validation running in parallel.
* In the V-Model Delivery or Deployment **Time is fixed (3 Months.)**
* When one stage is completed and another stage running if there is change in Completed stage then we can accept the change i.e Backtrack is possible in the V-Model.
* Company will charge extra for the Change Request.
* V-Model is **PLAN** driven model.

**Dis-advantage.**

* **Time is fixed (3 Months.)**

---------------------------------------------------------------------------------------------------------------------

**Assessment of development plan.**

* In the Assessment of development plan, Strategy for the Testing is decided.
* E.g Testing – Manual /Automation
* TRM (Test Responsibility Matrix) document will create in this stage.

**Test Plan Preparation**

* In the Test Plan preparation – Planning required for the Testing (Test Plan) decides.
* E.g No of QA (4QA – 1 Automation Expert : Only Automation, 1 Manual QA: Perform only Manual, 2 QA – Automation + Manual)
* Distribute the work to team.

**Requirement Understanding.**

* In this stage Requirements in the SRS documents will understand.
* E.g Google Meet Module – 200 People can Join, Mute Unmute Functionality, Hand Raise Functionality.

-------------------------------

**Design phase testing (Review)**

* After the Design (HLD and LLD) Design will review by the Client/BA.
* In this testing Design conforms correct or not.
* Designer will have to present Design in meeting (Meeting Consist of Client, BA, Manager and Designer)

**Program phase testing (WBT)**

* When developer finish coding part then Code need to be checked by the Developer.
* This phase consist of Code Testing.
* Developers are involve in this phase.

**Test Case Design**

* When Developers are/start coding; at the Time QA will start Design the Test Cases.
* QA involves in this stage.

---------------------------------

BBT - Black Box Testing – QA Performs Testing. (System and Functionality Testing)

UAT – User Acceptance Testing (QA Testing).

KT: Knowledge Transfer: - Training provided to the QA.

----------------------------------

**CR (Change request)**

E.g : Recharge Successful 🡪 “Recharge Successful”

“Recharge Successful” changed to 🡪 “Recharge Successful, Thanks for Using Paytm”

Defect Removal Efficiency (DRE)

* DRE is used to checked how thourouly your application is tested.
* DRE is calculated using the Formula :

DRE = (A) / (A+B) = 5/6 = 100%

Where

A = No of Bug find in SIT

B =No of Bug find in UAT

|  |  |  |  |
| --- | --- | --- | --- |
| DRE | 0.8 to 1 | 0.6 to 0.8 | Below 0.6 |
| Remark | Good Testing | Average Testing | Poor Testing |

--------------------------------------------------------------------------

**Agile Methodology.**

* Agile method defines it **is Continuous and Iterative** process of software development and Testing.
* Agile follow **incremental approach.**
* Agile is a **Value driven process** i.e Complete priority (Importance) is given to the client
* In the Agile Methodology **requirements are continuously changing**.
* If **CR** is given by the Client then we will **surely accept the change request** but we will **analyze** **the impact of the CR** on current development and testing process.
* If the Impact of CR is higher then we will inform that Impact to the Client
* If the Impact of CR is less then we will accept the change and proceed further.

**Advantages of Agile**

1. Deployment time in the Agile is fixed (2/3/4 Weeks) 🡪 **2 WEEKS**
2. It is a **Continuous** process and follows **Incremental** approach.
3. Agile has Different Meetings.

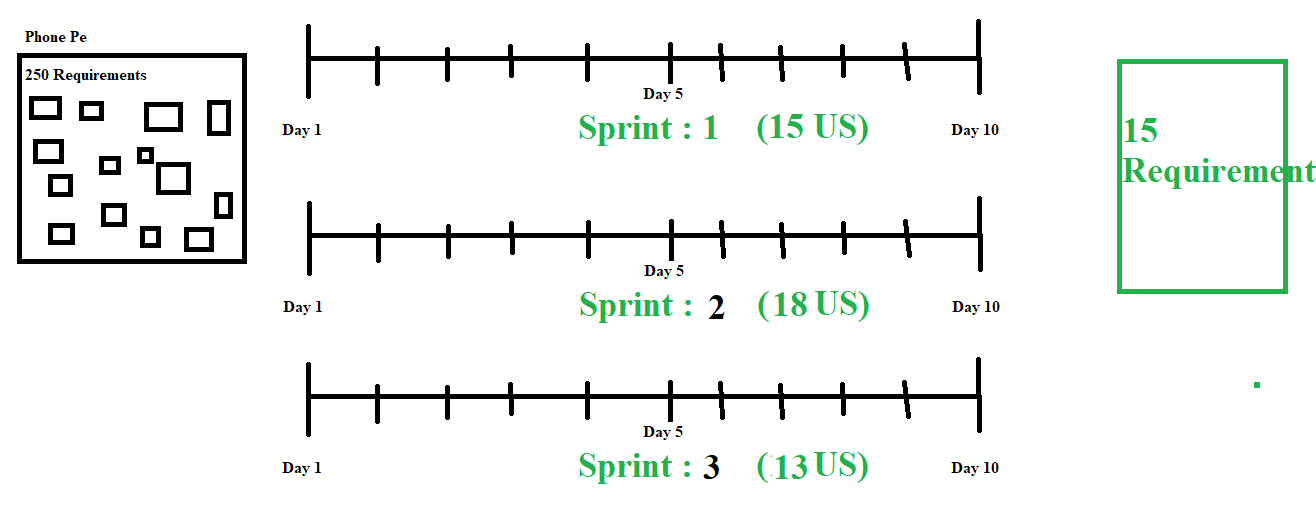
* Grooming Meeting. – US Understand and Doubts cleared
* Sprint Planning Meeting. – Sprint successfully execute planning
* Daily Stand Up Meeting. – Work done by team monitored daily.
* Demo Meeting. – Client/Stake holders give **feedback** on the work done.
* Sprint Retrospective Meeting. – What good happened, What bad happened, feedback, appreciation, (Opportunity to improve)

1. Checkpoints are available.

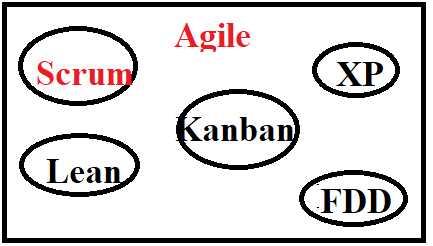
**Disadvantages.**

* Agile Time duration in fixed and also less and suppose your project concepts are not clear then it will become difficult to deliver story on time.
* If your one module (Project) dependent on other module and that other module not responding then also difficult to deliver stories on time.

Sprint



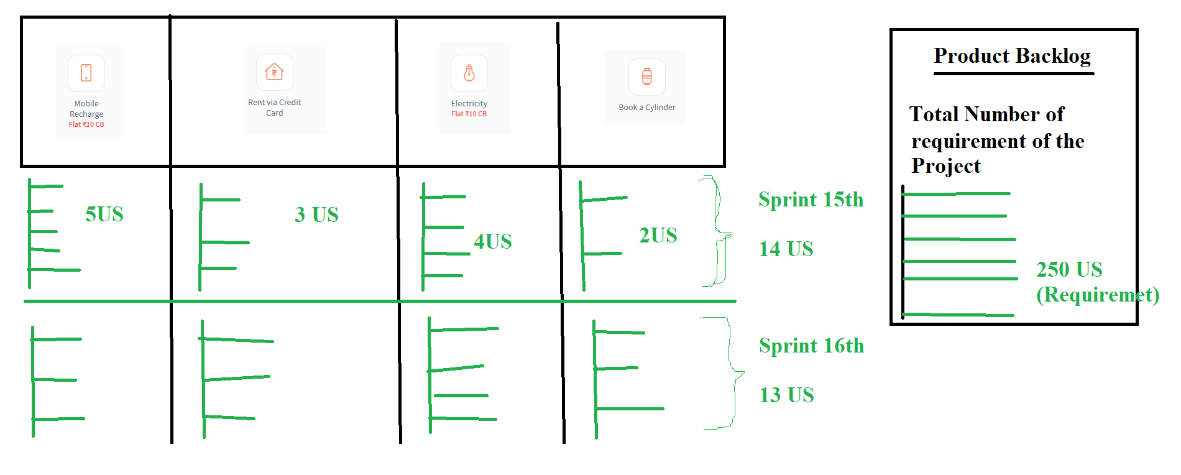
**Type of Agile Method**



1. Kanban – Support Team.
2. Lean – Support team.
3. XP – Xtreme Programming
4. FDD – Future Driven Development.

In my organization **Agile –Scrum methodology follows**

**SCRUM: Scrum includes working in the Sprints.**

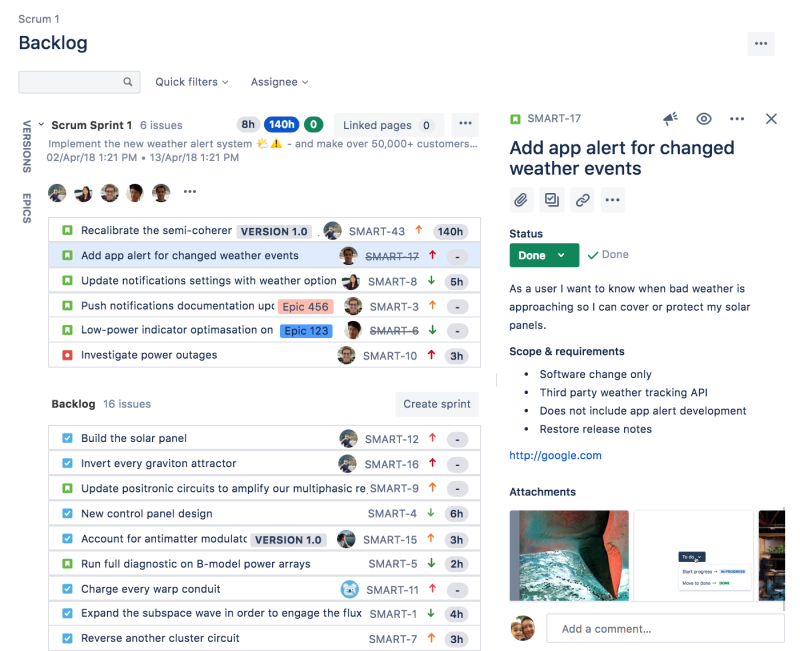


**Agile Architecture.**

|  |  |  |
| --- | --- | --- |
| **SDLC** |  | **Agile** |
| Information Gathering 🡪 BA 🡪 BRS | **Product Backlog** |
| Analysis 🡪 BA 🡪 SRS | **Sprint Backlog ( Sprint 1 🡪 14 US)** |
| Use Case   1. Description 2. Acceptance Criteria 3. Screenshots | User Story   1. Description 2. Acceptance Criteria 3. Screenshots |
| Design(HLD, LLD) and Coding | Design(HLD, LLD) and Coding |
| Testing (TCD, TCE) | Testing (TCD, TCE) |
| Support and Maintenance | Support and Maintenance |

**Agile Terminologies**

|  |  |
| --- | --- |
| **SDLC** | **Agile** |
| Client | **Stake Holders** |
| Delivery Manager | **Solution Master** |
| Project Manager | **Scrum Master** |
| Business Analyst | **Product Owner** |
| Designer | Designer |
| Developer | Developer |
| QA | QA |



**Agile Meeting.**

* **Grooming Meeting.**
* **Sprint Planning Meeting.**
* **Scrum Meeting / Daily Stand Up Meeting / Daily Sync Up / Scrum Call.**
* **Sprint Review Meeting/ Demo Meeting.**
* **Sprint Retrospective Meeting.**

|  |  |  |
| --- | --- | --- |
| **Meeting** | **Purpose** | **Members Involved** |
| Grooming Meeting (Before or At of Start of the Sprint) (Sometimes in the Middle of Sprint) | If Dev and QA have doubt in the US then Doubts are cleared in this meeting. | 30 min **1 Hour – BA,** PM, Designer, QA and Dev Team  BA is Chair Person |
| Sprint Planning Meeting  (At Start of the Sprint) | 1. Current Sprint – Total 16 US 2. Estimation (Time) –   (10 Hr Dev + 6 Hr Testing)   1. **Assigned to Me – 04 US.** | 30 min **1 Hour –** BA**(**optional**), PM**, Designer, QA and Dev Team  PM is Chair Person |
| Daily Stand Up / Scrum Meeting  (Everyday 12.30 PM) | 1. What did you yesterday? 2. What you will do today? 3. Roadblock. | 15 – 30 Min / Scrum Master, Dev and Testing Team. |
| Sprint Review Meeting. / **Demo Meeting** (Last Day of Sprint) | QA has to give Demo to Client of Every User Story to Team in front of Client. | **1** – 2 Hr : **Client**, PM, BA, Dev and Testing Team. |
| Sprint Retrospective Meeting  (Last Day , After the Demo Meeting) | Feedback of overall sprint.  What Good and Bad/Disaster happened in current sprint.  Suggestion. | 15 – 30 Min. (PM, QA Dev Team and BA) |

**Agile Day wise Planning. (03 to 14 Oct)**

**Day 01: Monday 03 Oct – Sprint Start Day.**

* Grooming Meeting (30 Min – **1 Hour**) – cleared the Doubts
* Sprint Planning Meeting - (30 Min – **1 Hour**)
  + Current Sprint 1 – 15 US Pulled from backlog.
  + Estimation (Burn Hour Timesheet)
  + Scrum Meeting.
  + 4 US – Assigned to me.
  + **1st US – Understand the User Story, Test case design, Test case Review [8 Hour]**
  + **(Dev – 1st User Story Understand and Develop)**

**Day 02: Tuesday 04 Oct:**

* Daily Stand Up Meeting.
* What did I yesterday? - User Story Understand and Design Test cases for US-101
* What I will do Today? **If dev provide build then I will test US – 101** / Otherwise I will design Test cases for other User story [ 6 Hours]
* Roadblock – No. ( Dev provided Build)

**Day 03: Wednesday 05 Oct:**

* Daily Stand up Meeting.
* What did I yesterday? : I have executed the Designed test cases i.e Tested the User Story – 101
* What I will do Today? User Story Understand and Design Test cases for US-102
* Roadblock – No [ 7 Hour]
* Dev – Today : US – 102 Development complete AT THE END OF DAY]

**Day 04 : Thursday 06 Oct :**

* Daily Stand up Meeting.
* What did I yesterday? : I have Designed test cases for US -102. Test Cases Reviewed by the Senior QA.
* What I will do Today? Test the User Story – 102
* Roadblock – No [ 8 Hour]

**Day 05 : Friday 07 Oct :**

* Daily Stand up Meeting.
* What did I yesterday? Finished the Testing part , No Bug found.
* What I will do Today? I have Designed test cases for US -103. Test Cases Reviewed by the Senior QA.
* Roadblock – No [ 6 Hour]

**Day 06 : Monday 10 Oct :**

* Daily Stand up Meeting.
* What did I yesterday? I have Designed test cases for US -103 Test Cases Reviewed by the Senior QA.
* What I will do Today? Test the US-103.
* Roadblock - No

(While Testing you found 3 Bug—Log 3 bugs – according to dev 2 bug Valid – 1 Bug Invalid According to Dev.) [8 Hours]

**Day 07: Tuesday 11 Oct:**

* Daily Stand up Meeting.
* What did I yesterday? Testing US-103, US not completed: 3 Bugs Logged. US103 Assigned to Developer.
* What I will do today? I will Design Test cases for US-104.
* **Roadblock – Conflict: One bug is not accepted by Developer.**

**Day 08 : Wednesday 12 Oct:**

* Daily Stand up Meeting.
* What did I yesterday? Logged Bugs Dev fixed -- assigned to Me ; I have tested the Bug o check Bug fixed or not (Testing) US- 103 Completed Testing - [ Test cases Designed – Partially Completed]
* What I will do today? Designed Test cases for the US – 104.
* Roadblock – No.

Day 09: **Thursday 13 Oct ;**

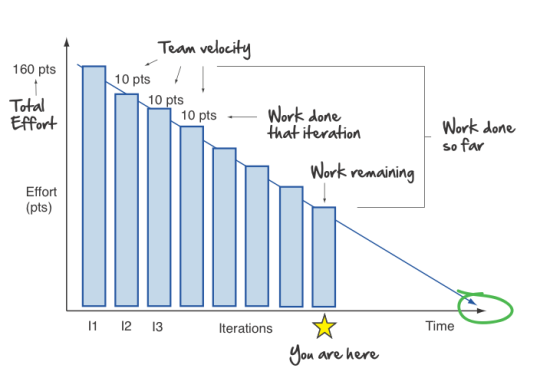
* Daily Stand up Meeting.
* What did I yesterday? Designed Test cases for the US – 104.
* What I will do today? US – 104 will test if got the build.
* Roadblock – No
* ( Dev provided build at 3 PM - ) (Testing Start but not finished)

**Day 10: Friday 14 Oct :**

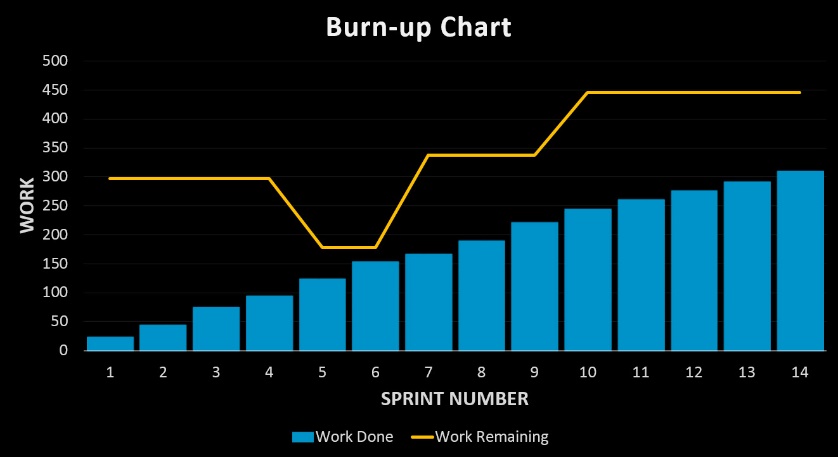
* Daily Stand up Meeting.
* What did I yesterday? Demo Preparation as got the build at 3 Pm; Started the Testing Partially Done.
* What I will do today? Will complete Testing the US – 104; and Be ready for the Demo. Present the Demo in Demo Meeting
* Road block – No.
* 6 PM to 7 Pm – Demo Meeting Attained - Present the Demo.
* 7 to 7.30 PM – Sprint Retrospective Meeting.

**-------------------------------------------------------------------------------------------------------------------**

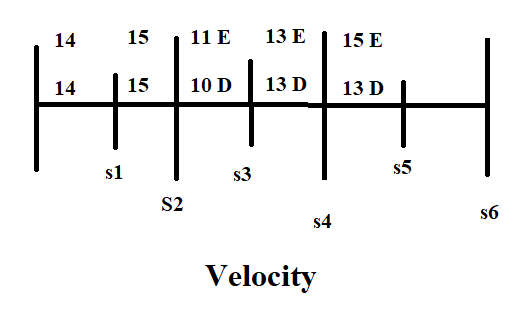
**Agile Terms.**



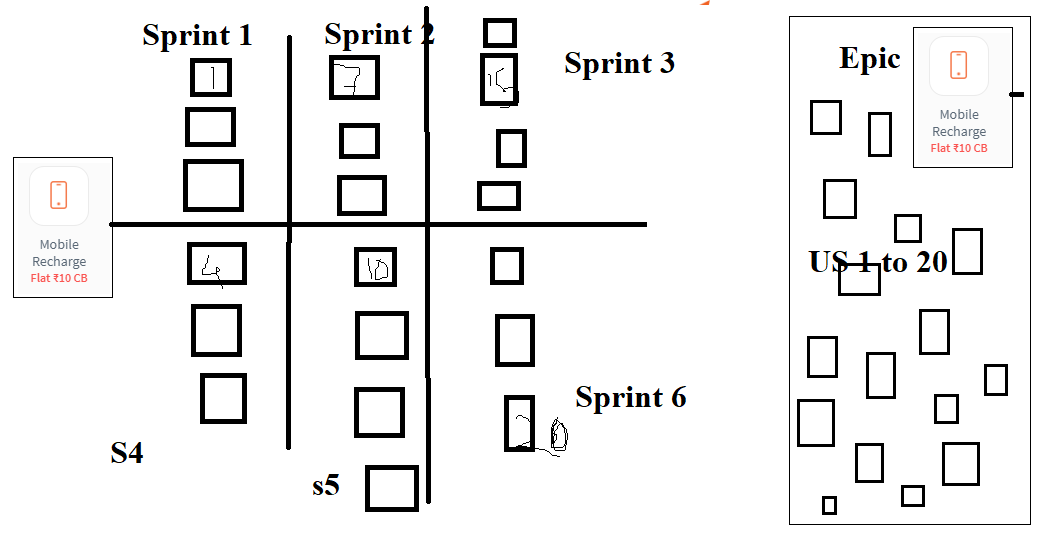
* **Burn down Chart.**
  + How much work is pending with respect to days/time in a Sprint?
  + Graphical Representation of work left to do w.r.t time.



* **Burn Up chart**
  + How much work is completed with respect to days/time in a Sprint?
  + Graphical Representation of work completed to do w.r.t time.

****

* **Sprint Velocity**
  + **How many user stories are deploying per sprint to the Client.**
* **Epic**
  + **Epic is Main module which consist of multiple user stories**
  + **Epic is bunch of user stories of the same module.**

****

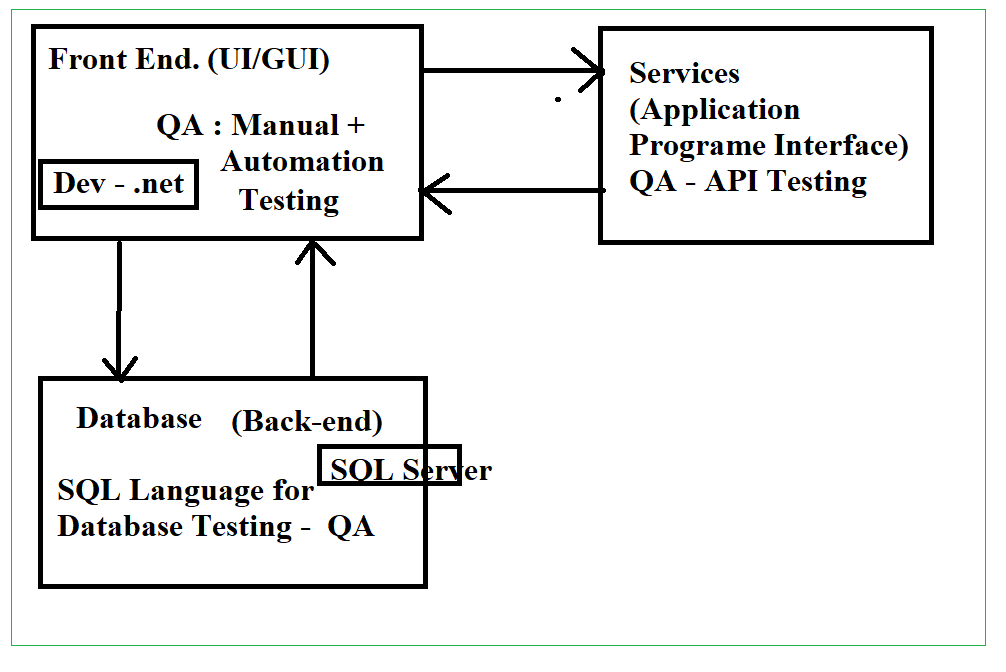
* **Estimation / Hours (Burn Hour) Story point ( 1 Point 2 Point 3 Points)**

**Time span required to complete the US. US-101**

**Dev – Estimation - 10 Hrs.**

**QA – Estimation – 6 Hrs.**

**Technologies used in the Project**

****

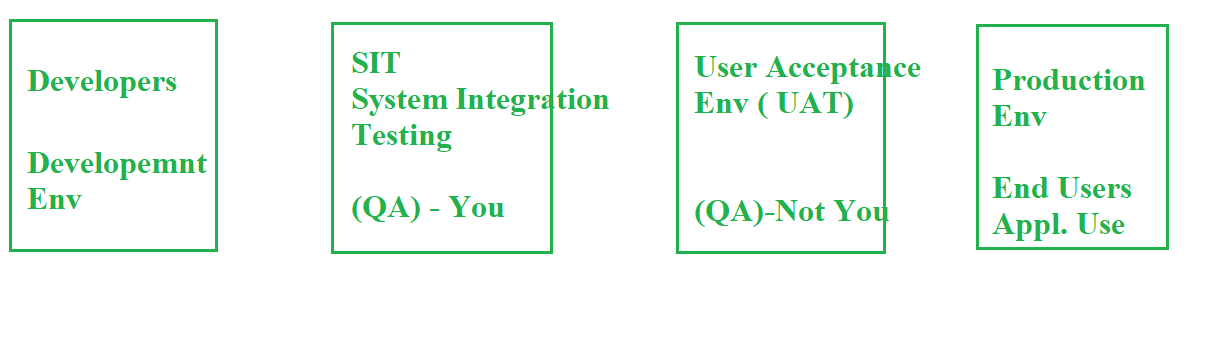
**Front end 🡪 .net language**

**Backend/Database 🡪 SQL server**

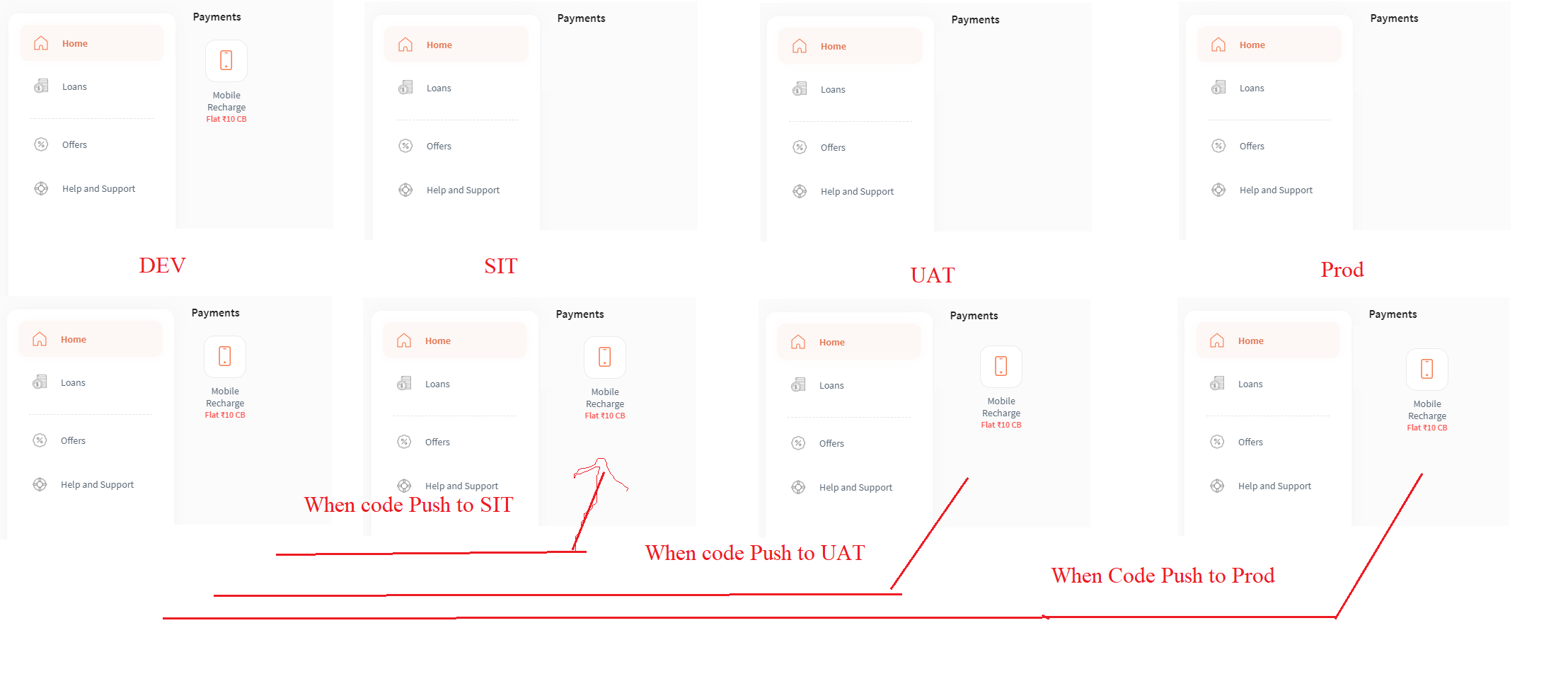
**API / Web Services 🡪 Java Language using Postman.**

**---------------------------------------------------------------------------------------------------------------------**

**Types of Environment**

****

|  |  |  |  |
| --- | --- | --- | --- |
| **https://www.DEVfreecharge.in/** | **https://www.SITfreecharge.in/** | **https://www.UATfreecharge.in/** | **https://www.freecharge.in/** |

****

**Build Version:**

**Dev sent a build to QA (SIT) 🡪 Build Version No – 1.2.6 (QA finds a Bug in SIT)**

**Dev will fix the Bug (Modify the Build);**

**Dev sent a build to QA (SIT) 🡪 Build Version No. – 1.2.7 (QA find no Defect)**

**Dev sent a build to QA (UAT) 🡪 Build Version No. – 1.2.7 (QA finds a Bug in UAT)**

**Dev will fix the Bug (Modify the Build);**

**Dev sent a build to QA (SIT) 🡪 Build Version No. – 1.2.8 (QA find no Defect)**

**Dev sent a build to QA (UAT) 🡪 Build Version No. – 1.2.8 (QA find no Defect)**

**----------------------------------------------------------------------------------------------------------------**

1. **Dev Env. / (Sandbox) – (Dev will work)**

* In the Dev Env developers will work on coding part.
* When build is completed then Dev will send a build to the QA (Build – Link for the Build) through the **JIRA** tool.
* Dev will send Build Link –
* In the mail (Comment in JIRA) consist of the Build Link, Unit Testing Document, Tables Name (Database).

------------------------------------------------------------------------------------------------------------

1. **SIT Env**. – System Integration Testing Env. ( QA – You)

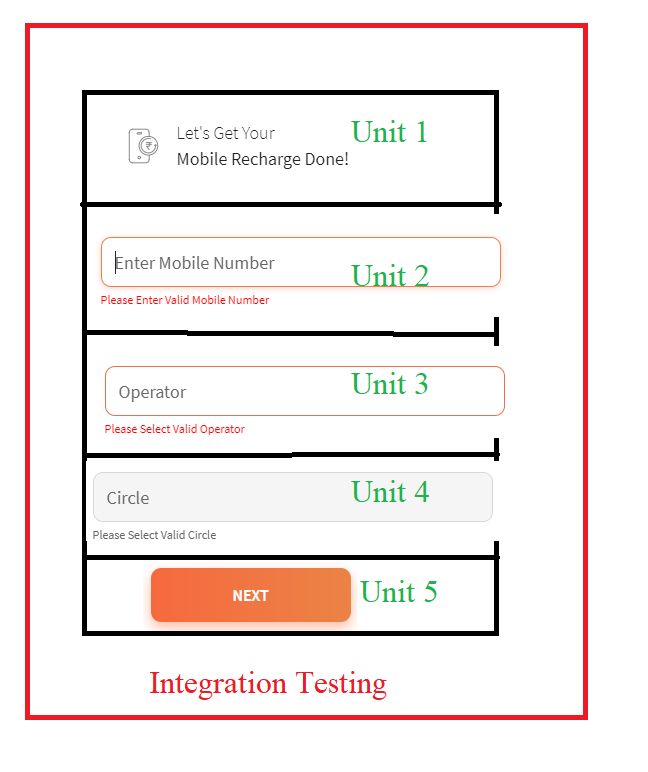
* In the SIT Env QA will perform – Design Test Cases, Review Test Cases and Execute the Test Cases.
* While executing the Test Cases if QA founds any bug then QA will logged a Bug.
* QA will log a Bug through the JIRA Tool (Project Management Tool)

|  |  |  |  |
| --- | --- | --- | --- |
| DIT | SIT | UAT | Prod |
| 1. Unit Testing 2. Integration Testing | 1. **Smoke Testing**/Sanity 2. Black Box Testing (Functionality Testing) 3. Re-testing 4. Regression | 1. Alpha Testing 2. Beta Testing | End user will use the application.  (If Defect is present then Defect will face by the User) |

**Q. How you receives the Build –**

**When Dev completes then Dev send me an Email through the JIRA Tool and mentioned the URL of the Build in that mail.**

Q. If you don’t have SIT Env in your organization then what you will do?

-----------------------------------------------------------------------------------------------------------------

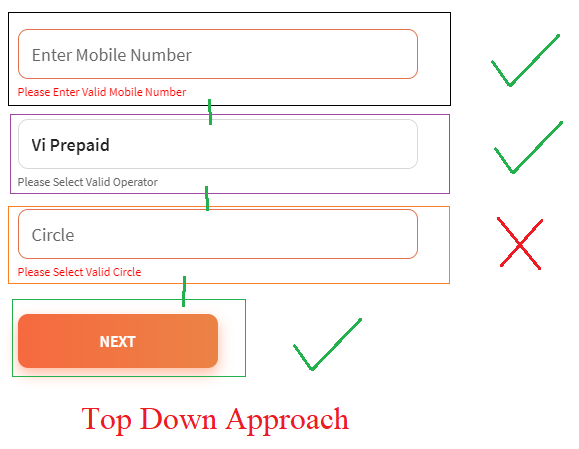
**Unit Testing. (Dev)**

* When developer write the code part against the user story (Simple few lines of code) then dev will perform the Unit Testing.
* Unit testing is done by the Developer.
* The purpose of the Unit testing is – To check the Code is correct or not.
* In the Unit testing it includes; Unit Testing Document, Steps to Executions, ID Passwords / URLs/ URI, Tables **(Database)**

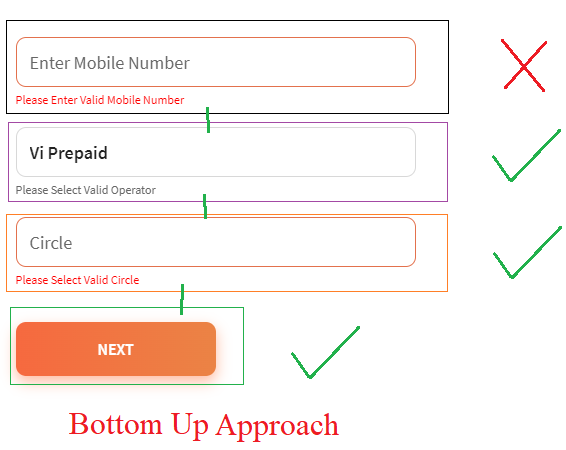
**Integration Testing (Dev)**

* Integration means combining all the Units / Sub modules and form one main module.
* In the Integration testing complete module (All Combined Units) functionality validates.
* There are Two Types of Integration

1. Front End Integration (UI) – CALL Function.
2. Back End Integration (Database) – Join Function.

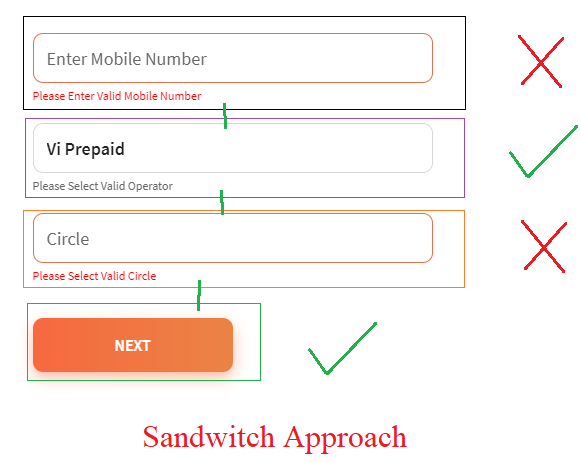
**Different Approach used for the Integration Testing**

1. **Top Down Approach**
2. **Bottom Up Approach**
3. **Sandwich Approach**
4. **Top Down Approach**

* In the Top down approach if the Starting module (Unit) is available but sub module (Middle Unit / End Unit) is not yet developed in such scenarios To Down approach is preferred.
* ****Developers creates a **STUB** for the Unit which is not yet developed.
* STUB is a temporary / Dummy unit developed using XML/HTML language.

1. **Bottom Up Approach**

* In the bottom up approach; If the Starting Unit is not developed yet but the Middle / End Units are developed the Dev will preferred Bottom Up approach.
* Developer creates a **Driver function** for the Starting Unit which is not developed yet.
* **Driver** is a temporary / Dummy unit developed using XML/HTML language.

****

1. **Sandwich Approach**

* In the Sandwich approach if starting as well as Middle/End Units is not yet developed but rest of the Units are developed at such scenarios Sandwich approach is preferred.
* In the Sandwich approach both Driver and STUB exists which will act as a Dummy Module.

Sanity Testing **/ Smoke Testing**

1. **Smoke Testing** is also known as **Tester or QA Acceptance / Buid 0 / Build Verification** Testing
2. When QA receives the build from the Developer so before proceed for the testing QA ensures (Check) whether the build is stable or not that process is called as Smoke Testing.
3. QA Performs the Smoke Testing before the Black Box (System and Functionality) Testing.
4. Smoke Test Includes

* Validate the Core functionality
* Validate the GUI /UI
* Validate the Link
* Validate the tab
* Validate the Navigation of the Application.

Core Functionality – Button, Icon and main functionality.

Tab Validation – Text box, tabs

Link Validation – Sequence of the Interlink pages (Flow of the Requirement)

Page Validation – Scroll Up / Scroll down.

UI Validation – Images / Videos / Icon should be displayed correctly and at correct position.

* While performing smoke test if any issue found the **QA reject** the build (1.1) and send build to the **developer**.
* Dev will fix that issue and will send a new build (Modifies – 1.2)

**Defect – Environment Issue, System Hangout, Runtime Error, Pop ups or Link is not working.**

**---------------------------------------------------------------------------------------------------------------------**

Q. Have you performed Sanity Testing?

Well In my organization we prefers/follows **Smoke testing.**

Q. When you performs smoke/ sanity Testing.

1. When you receives the build from the developer then QA performs the Smoke testing

2. When build is move from one Env on to another environment.

**Q. What is difference between Sanity and smoke testing?**

In the Smoke testing if QA founds a Bug then QA will reject the build but QA will provide the Root Cause of the Bug While in case of the Sanity Testing if there is a Bug then directly QA reject the build.

Smoke testing is the Extra sake of the Sanity Testing.

In sanity testing if there is an issue QA directly reject the build but in Smoke Qa also provide the Root cause of that issue.

---------------------------------------------------------------------------------------------------------------------

[**https://getsharex.com/**](https://getsharex.com/) **: Used to take Screenshots and Videos.**

**System and Functionality Testing.**

Once QA ensures that build is stable then QA performs System and Functional Testing.

Few Types of System and Functionality Testing.

1. **Usability Testing.**
2. **Functionality Testing.**
   1. **Functional Testing ( B I E B S C)**
   2. **Non Functional Testing ( R C C I I S G P)**
3. Security Testing.
4. Performance Testing. (J-Meter tool Used)
   1. LOAD Testing.
   2. Stress Testing.

---------------------------------------------------------------------------------------------------------------------

1. **Usability Testing**

* In the Usability testing, QA validates the friendliness of the screen/GUI/ UI.
  + 1. GUI /UI Testing
  + It includes the Validation of the **Appearance** (Look) and **feel** of the GUI / UI.
  + Ease of Use of the Application.
  + Speed of the Interface.
    1. Manual Support Testing.
  + Validation of the Manual Input Values.
  + Sensitiveness of the Screen.

1. Functionality Testing

* In the functional testing QA validates the internal and the external features of the application.

1. **Functional Testing ( B I E B S C)**

* QA validates the **internal features** of the application.

1. **Non Functional Testing ( R C C I I S G P)**

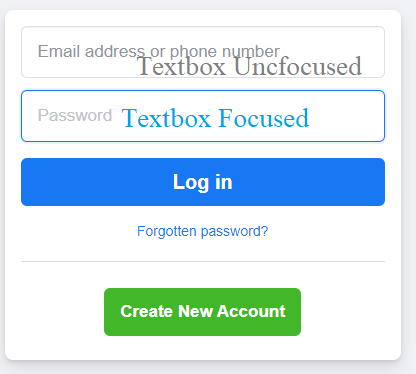
* QA validates the **external features** of the application.

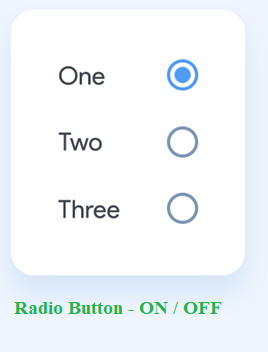
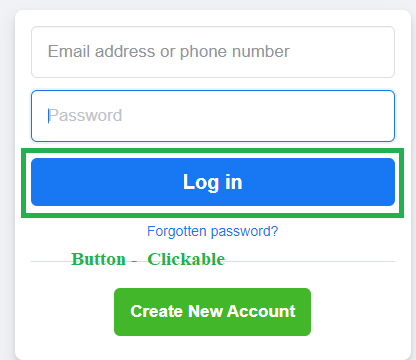
**Functional Testing (B I E B S C)**

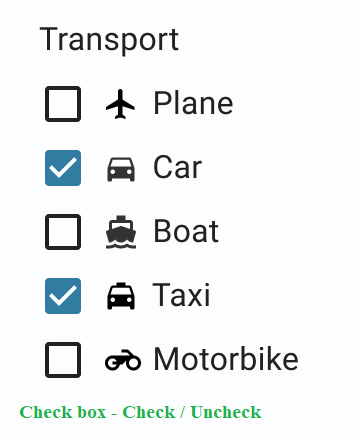
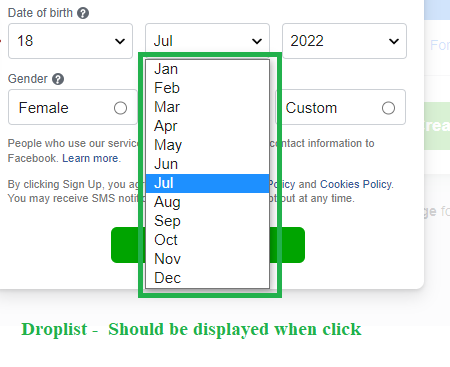
1. **B**ehavior Coverage Testing.
2. **I**nput Domain Coverage Testing.
3. **E**rror Handling Coverage Testing.
4. **B**ackend / Database Coverage Testing.
5. **S**ervice based Coverage Testing.
6. **C**alculation based Coverage Testing.
7. **B**ehavior Coverage Testing.

* In the behavior coverage testing QA validates the behavior of the object / Property of the object

|  |  |
| --- | --- |
| Object | Property of Object |
| Text Box | Focused / Unfocused / User can add data |
| Radio Button | ON / OFF |
| Button | Clickable |
| Checkbox | Check / Uncheck |
| Dropdown | To show the hidden list if click.  (Dropdown list should be displayed) |
| Range Line |  |





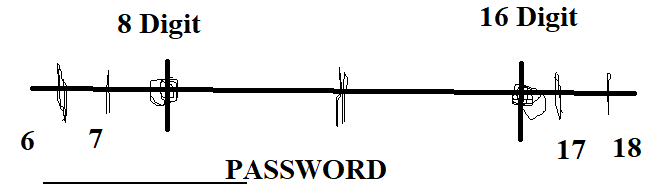
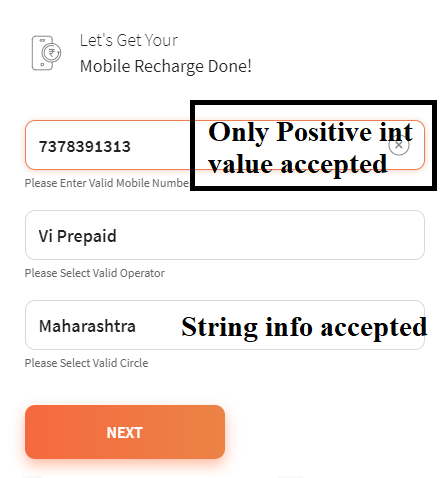


**II) Input Domain Coverage Testing**

* In the input domain coverage testing, QA validates the Size / Length of the input and the Data type of the input.

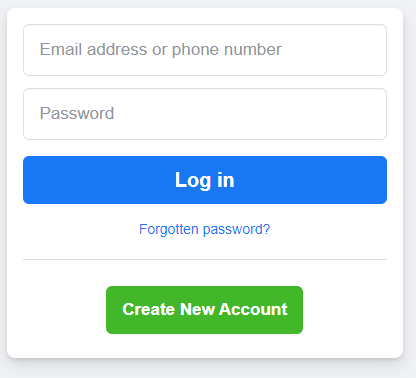
1. Boundary value Analysis (BVA).

* In the BVA, QA validates the **Input Size/ Length of the input.**



1. Equivalent Class Portioning (ECP).

* In the ECP, QA validates the **Data Type of the input.**

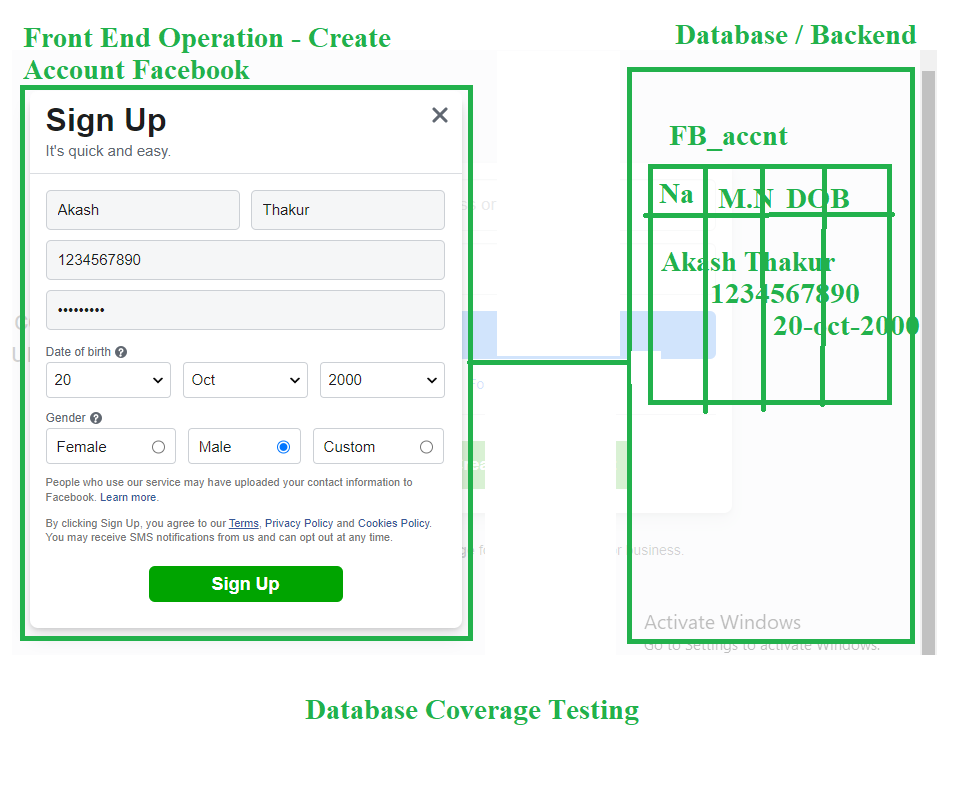
1. Decision Table Coverage Testing.

|  |  |  |
| --- | --- | --- |
| **ID** | **Password** | **Status** |
| Valid | Valid | Login |
| Valid | Invalid | Not Login |
| Invalid | Valid | Not Login |
| Invalid | Invalid | Not Login |
| Null | Valid | Not Login |

III) Error Handing Coverage Testing

* In the error handling coverage testing QA validates the different types of the error message generated in the application.
* When QA pass the Invalid input / performs invalid steps then QA validate the Error message.

1. Assigned Task – Take 10 Different examples for Error Handing Coverage Testing and **B**ehavior Coverage Testing.



**IV) B**ackend / Database Coverage Testing.

* Backend of the Application is **DATABASE**
* **In the Database/Backend testing** QA validates the Front end operation performed reflects on the backend i.e
* Whether the information filled/provided/submitted is stored in the database in correct tables.
* QA use SQL commands to test the Database.

1. **S**ervice based Coverage Testing.

* In the service based coverage testing QA validates the sequential flow of the application.
* Open Paytm -🡪 Click on Recharge Module -🡪 Add Mobile No 🡪 Operator 🡪 Circle 🡪 PLAN 🡪 Promo Code🡪Payment 🡪Successful /Error Message.

VI) **Calculation Based Coverage Testing**

* In the calculation coverage Testing QA validates the Arithmetic operations /Calculations on the Application.
* E.g 50% off on Myntra Application.
* If you want to Buy Shirt of MRP - 3000 Then After Offer Discount \_-- Shirt Price will be Rs 1500.

**Non Functional Testing ( R C C I I S G P)**

In the nonfunctional testing QA Validates the external features of the application.

1. Recovery Testing
2. Compatibility Testing
3. Configuration Testing
4. Installation Testing. (Not in SIT ENV)
5. Intersystem Testing.
6. Sanitization Testing.
7. Globalization Testing.
8. Parallelization Testing. (Not in SIT ENV)
9. **R**ecovery Testing / Reliability Testing.

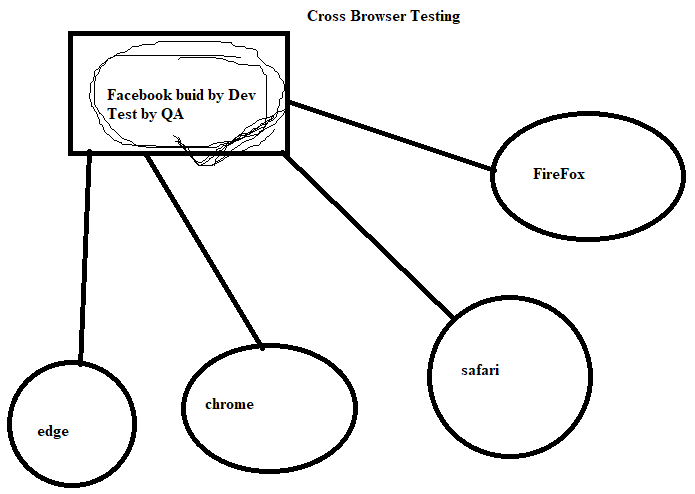
* In the Recovery testing QA validates – How build/application responds in the abnormal situation.
* E.g – Payment Page suddenly closed / No Internet Connect / Refresh the fish.

1. **C**ompatibility Testing

* In the **C**ompatibility Testing QA validates the build/application is compatible and Works in the client expected platform.

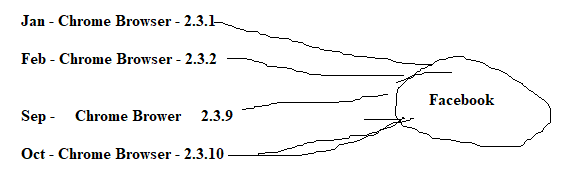
1. Forward Compatibility [Build – Ok But Issue in the OS] IT Team will handle the issues related to Forward Compatibility.
2. Backward Compatibility
   1. Cross Browser Testing.
   2. Version Control Browser Testing.
   3. Cross Browser Testing.

* In the cross browser testing QA Valides the **application support/**runs/works in different browsers **like chrome, Firefox, edge, opera, safari, Mozilla.**



* 1. Version Control browser Testing.
* In the version Control browser Testing, QA validates the build/application works/runs in the **same browser** but in **different versions**.

Virual Machine.



<https://www.w3schools.com/sql/>