Testing

Development Environment

1. Unit testing
2. Integration testing

SIT Environment

1. Sanity testing/Smoke testing
2. BBT- System & Functional testing
3. Retesting / Regression testing

UAT Environment

1. Alpha testing
2. Beta testing

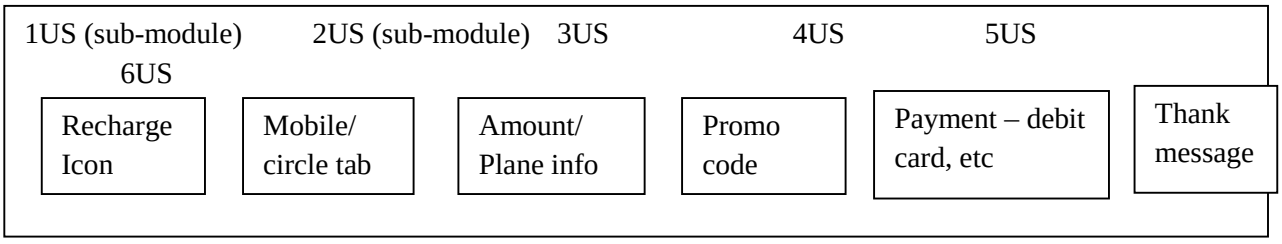
Prod / Live Environment- end user will use

Development Environment

- In Dev environment, developer is working or involved
- When developer will done the coding, then developer will perform testing like unit testing & integration testing

1. **Unit Testing**

- Unit testing will perform by developer on every user story after coding
- Every user story against, developer perform the unit testing
- For every user story check the coding is correct or not in the positive way only
- Unit testing contains- documents, step to execution, flow of application for testing (screenshots) table name, UN & PW, URL etc.
- Unit testing, it will performed in sub module
- Ex. Paytm- Recharge module
- Recharge Module- **Main module**



2. Integration testing

- Integration testing perform by developer
- This testing is carried out after the unit testing
- Integration means combining all sub-module & prepare main module
- In integration- all the dependent modules are added/integrated together to form one application/product
- For integration testing, developer should have knowledge about functionalities, dependencies on other modules, relation because output of one module acts as a input to the other module
- **Integration testing** is the process to check completeness & correctness of the flow of functionality whenever integration of module performed or when modules are integrated
- Integration testing there are 2 type
    1. **Front end integration**

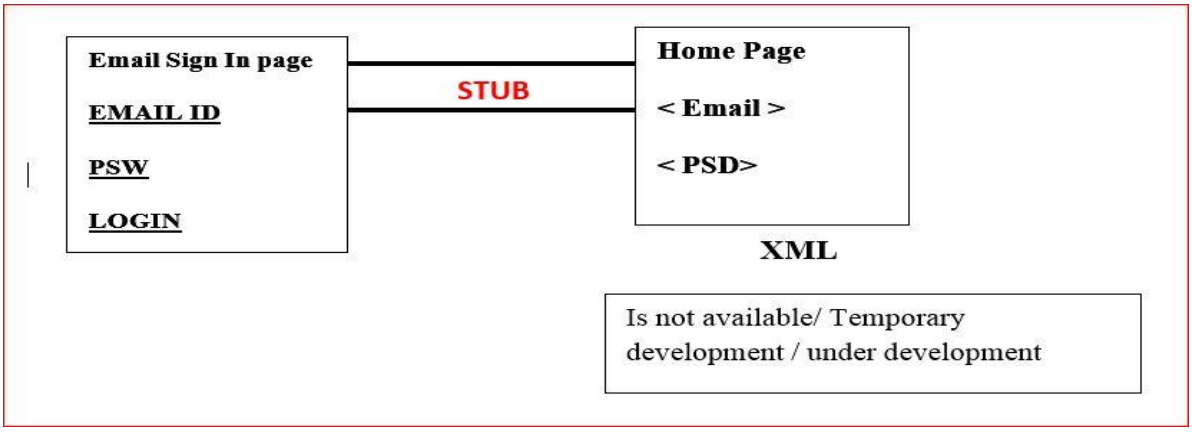
       Front end integration developer combine/connects all the dependent modules by using call function
    2. **Backend integration**

       Back end integration developer combines or connects all the table together in database by using join function
- Different approaches for integration testing
    1. Top – down approach
    2. Bottom – up approach
    3. Bidirectional / sandwich approach

1. Top – down approach

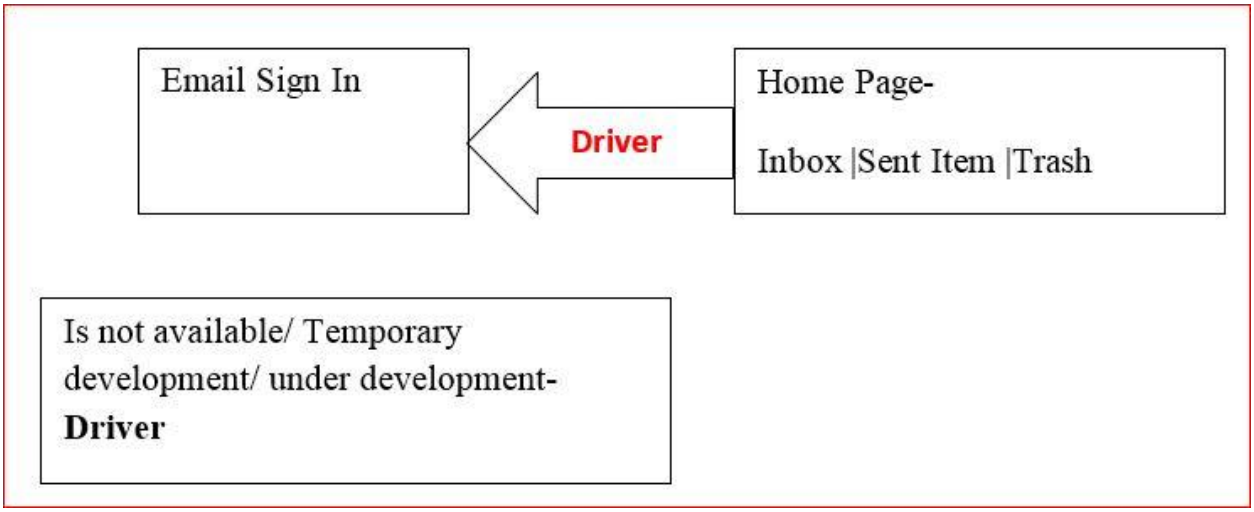
- In this approach, main module with functionality is available for testing but sub module are not available for testing
- Sub module are not available because
    1. It can be under development
    2. Sub modules have defects

- So, to test this main module, developer creates dummy module i.e. developer creates dummy program in XML language is known as stub program
- XML is code language, which is used to communicate between two applications
- XML language has request & response



2. Bottom – up approach

- In this approach, sub module with functionality are available but main module is not available for testing
- To test sub module, developer creates dummy main module, i.e. developer creates dummy program in XML languages is known as driver program



3. Bidirectional approach (Sandwich approach)

- It’s a combination of top down & bottom up approach
- If developer wants to check functionality of the main module & he does not have sub module then he uses stub program
- If developer wants to check functionality of the sub module & he does not have main module in such case he uses the driver program