**System Testing**:

It is **end-to-end testing** wherein **testing environment is similar to the production environment.**

**End – to – end testing**

Here, we navigate through all the features of the software and

test if the end business / end feature works. We just test the

end feature and don’t check for data flow or do functional

testing and all.

**Let us consider an example to explain System Testing.**

Let us consider Citibank wants a software for overdraft feature. It asks IFlex company to develop the software and it provides CRS to develop the feature. The CRS contains how the overdraft feature works,

The difference between personal loan and overdrafts is – personal loans, loans can be provided upto 20 times more than the monthly income and also takes a long time to approve by the manager for personal loan. Whereas, in Overdraft, the loan amount is twice the monthly income and takes hardly a day to be approved by the manager. For ex, if a customer of Citibank wants a overdraft loan of Rs 20,000 over his monthly income of Rs 10,000. The manager approves the loan. Let us say that the interest rate is 2% and the activation fee for the first time is Rs 250. When the customer repays the loan at the end of the month, then the total amount he pays is – 20,000 + (2% of 20,000) + Activation fee ( 250 ) = 20,000 + 400 + 250 = 20,650Rs.

Now, for the 2nd time if the same customer wants another overdraft loan – then no activation fee is taken. Now the customer applies for another overdraft loan of Rs20,000. This time the amount he has to repay is – 20,000 + ( 2% of 20,000 ) = 20,400Rs.

The development team develops the software which looks something like this, (Shown in the next page).

**MANAGER**

**Pending Overdrafts**

**.. . . . . .. .**

**.. . . .. . . .**

**. .. . . . . .**

**Deposit**

**Logout**

**CANCEL**

**DEPOSIT**

**ACCOUNT**

**AMOUNT**

**DEPOSIT**

**CANCEL**

**APPROVE**

**………**

**……..**

**……..**

**…….**

**…….**

…..

……

…..

……..

……..

…….

…….

**CONFIRMATION**

**PAGE**

**CONFIRMATION**

**PAGE**

**PENDING OD**

**Name Amount Select**

**…….. ………**

**…… ……..**

**……. ……..**

**…… ……..**

**A 20000**

……

…….

……

**SUCCESSFULLY**

**REPAYED**

**CANCEL**

**REPAY**

……

……

…….

**REPAY OVERDRAFTS**

**AMOUNT**

**OVERDRAFT BALANCE**

**BALANCE**

……

……

……

…..

……

……

**CANCEL**

**APPLY**

……

…….

……

……

…….

……

……

…..

**ACCOUNT BALANCE**

**AMOUNT TRANSFER**

**LOANS**

**OVERDRAFTS**

**INSURANCE**

**LOGOUT**

**SUCCESSFULLY**

**APPLIED**

**APPLY FOR OVERDRAFTS**

**AMOUNT**

**OVERDRAFT**

**=> Apply for OverDraft**

**=> OverDraft Balance**

**=> Repay OverDraft**

……..

……

……..

**CUSTOMER**

The development team develops the required software as shown above. The first figure represents the software that can be accessed by the manager only. The 2nd figure represents the software that can be accessed by the bank’s customers.

Let us consider system testing now. We test for interest calculation when the customer takes overdrafts for the 1st time and when he takes overdrafts for the 2nd time.

**Scenario 1**

1) Login as A – Apply for OD Rs 20000 – Click on Apply – Logout

2) Login as manager – Approve OD of A – Logout

3) Login as A – Check OD Balance – Rs 20000 should be deposited – Logout

4) Change the server date to next 30days

5) Login as A – Check OD Balance – 20000 + 400 + 250 = 20650 – Logout

6) Login as manager – click on Deposit – Deposit Rs 650 – Logout

7) Login as A – Repay OD amount – Check OD balance – Rs 0

8) Login as manager – Click on Deposit – Deposit Rs 20000 to A’s account - logout

9) Login as A – Apply for OD Rs 20000 – Click on Apply – Logout

10) Login as manager – Approve OD of A – Logout

11) Login as A – Check OD Balance – Rs 20000 should be deposited – Logout

12) Change the server date to next 30days

13) Login as A – Check OD Balance – 20000 + 400 = 20400 – Logout

14) Login as manager – Deposit 400 – logout

15) Login as A – repay OD amount – Check OD balance – Rs 0

**Scenario 2 –** now we test another scenario where in – let us consider that the bank gives an offer that states that – a customer who takes Rs 50000 as OD for the first time will not be charged activation fee and activation fee will not be refunded when he takes another OD for the 3rd time – we have to test for 3test scenarios – wherein we have to take OD of Rs 50000 for the first time and check for OD Repay Balance after applying for another OD for 3rd time.

**Scenario 3 –** now we take in other scenario – let us consider that the software is being used normally by all customers – suddenly Citibank decides to lower the Activation fee to Rs 125 for new customers – we have to test OD for new customers and see if its accepting only Rs 125.

But, then we get a requirement conflict – Suppose the customer has applied for Rs 20000 as OD with the existing Activation Fee for Rs 250. Before the manager is yet to approve it, the bank lowers the activation fee to Rs 125. Now we have to test what Activation Fee is charged for the OD of the Pending customer – In this case, the testing team cannot assume anything – they have to contact the Business Analyst or the Client and find out what they want in such a case.

Thus, if they ( Client ) give 1 set of requirement, we must come up with maximum possible scenarios.

**The testing environment should be similar to production environment means,**

**1) The hardware should be similar to production – a)** The make ( manufactured by ) should be similar to production server ( **for ex,** if the production server is HP, then test server should also be HP server ) **b)** configuration and make must be similar, but different capacities i.e, number of CPUs ).

**2) The software should be similar to production – a)** The OS should be similar **b)** Application server should be similar **c)** Web server should be similar **d)** Database server should be similar

**3) Data should be similar to production – a)** We should create data similar to production **b)** We should create a script to create a dummy data which is similar to production environment .

**ex – while 20000**

**create Username**

**create password into table customer**

**Run**

In real time environment, we may make lakhs of entries into database. But, while testing we can’t enter manually lakhs of entries, so we write a test script program which generates thousands of entries and thus can be used for testing.

**What is Test Cycle:** It’s a duration given to test the application.

**What is Re-spin:** Fixing/resolving the defect and verifying the fixed/resolved defect in a same test cycle is called Re-spin

**Acceptance Testing/User Acceptance Testing [UAT]/Business Acceptance Testing:**

There are different types, it depends upon company and project

Type 1: Here Few End Users Start Using the software in real time business for a particular period of time and to find out whether the application supports all the business scenarios or not and gives the feedback to dev team

Type 2: Here the end users are not well educated so IT team goes to end users or the IT team test the application and gives feedback if any defects found to dev team

**Beta testing:** It’s a part of acceptance testing. Here the end users start using the s/w in the production server and then gives the f/b

Eg: Google plus recently did a beta release where end users started uing the s/w in the production server , where the users can give suggestions directly to the Google

**Alpha Testing:** It’s a part of acceptance testing and it is being performed in the engineering premises (UAT servers)

Eg: The call center uses the s/w and give the f/b to the manager

**Releases:** Software release is nothing but a duration from the coding till the application is moved /deployed on to the production server

2 types:

1. Short Term release
2. Long Term release

**Smoke Testing/ Sanity Testing/ Dry run Testing/ Skim testing:** Testing the basic or critical feature in an application before doing thorough or rigorous testing in an application is called as smoke testing

NOTE: As soon as we get the build, we should always start with smoke testing

What is Build: Compiled and compressed version of the code is called build

We perform smoke testing to find out whether the application is testable or not

E.g: Login Page -give valid inputs (we carry out only positive scenarios) and check whether you can login to application or not

Advantages:

1.Delay can be avoided

2. Finding out the defects in the early stage will give developers sufficient time to fix the defects and hence delay can be avoided