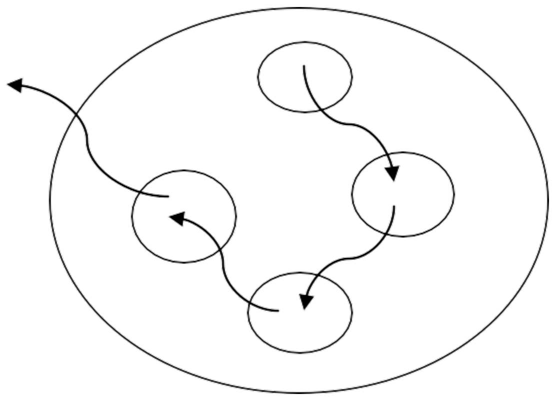
3) SYSTEM TESTING

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

End – to – end testing



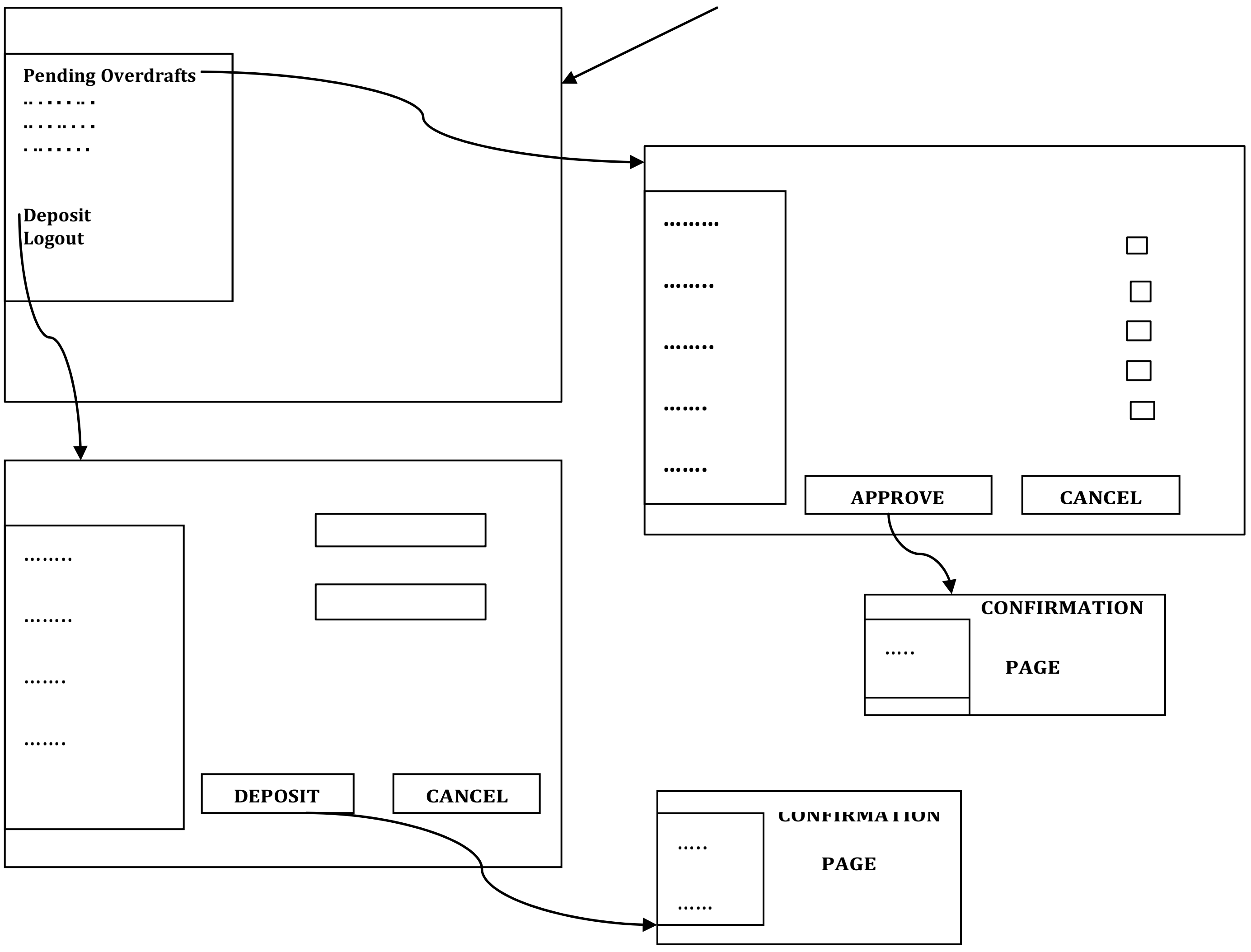
Here, we navigate through all the features of the software and test if the end business / end feature works. We just test the end feature and don‟t check for data flow or do functional testing and all.

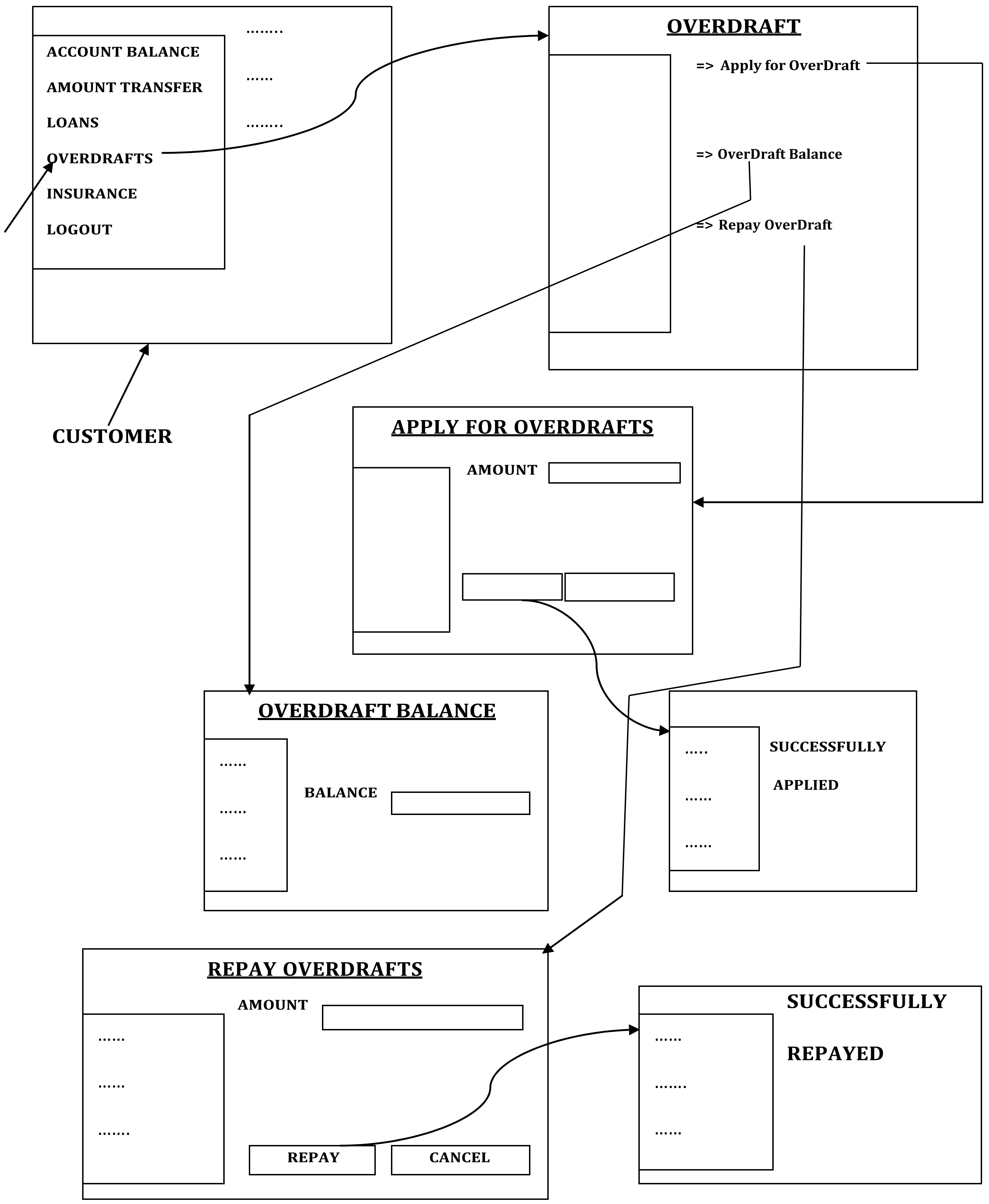
Let us consider an example to explain System Testing.

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).




....
ACCOUNT BALANCE
......
AMOUNT TRANSFER
....
LOANS
OVERDRAFTS
INSURANCE
LOGOUT
CUSTOMER
OVERDRAFT
=> Apply for OverDraft
=> OverDraft Balance
=> Repay OverDraft
APPLY FOR OVERDRAFTS
AMOUNT
OVERDRAFT BALANCE
BALANCE
SUCCESSFULLY
APPLIED
REPAY OVERDRAFTS
AMOUNT
REPAY
CANCEL
SUCCESSFULLY
REPAYED

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.

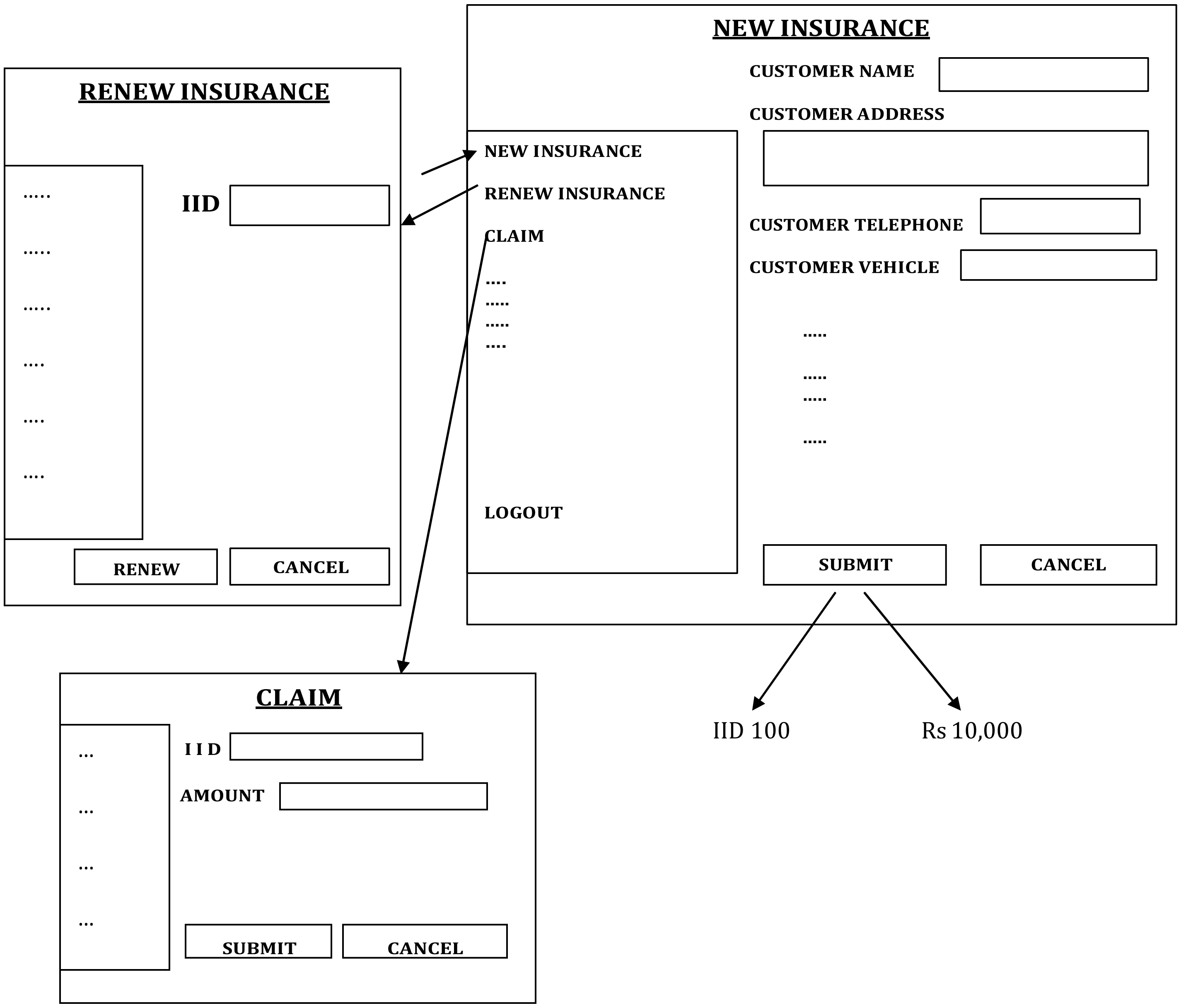
But, then we get a requirement conflict – Suppose the customer has applied for Rs 20000 as OD with the existing Activation Fee for Rs 250. Before the manager is yet to approve it, the bank lowers the activation fee to Rs 125. Now we have to test what Activation Fee is charged for the OD of the Pending 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.

Let us consider another **example – insurance domain**

When we buy a car, we have to obtain an insurance on that. Let us consider that Bajaj Allianz Insurance company is the one which provides this car insurance. The insurance policy works in this way – for the 1st year when the car is bought, the insurance to be paid is Rs10000. For the 2nd year, the insurance must be renewed at Rs10000 again. For the 3rd year, if no claim has been made, then that customer is offered a discount of Rs1500 and the he must have to renew the insurance at Rs8500 only. If an insurance claim has been made, then the insurance must be renewed at Rs10000 only. Bajaj Allianz Insurance wants the s/w to be developed which works as above. Thus it gives Wipro the CRS of above, to develop the s/w.

The development team develops the s/w as shown below,



The above s/w works in this way. When the insurance agent logs in to the home page, he clicks on New Insurance and creates a new insurance policy for the new customer and fills up all the details and the new customer is assigned a IID (Insurance ID) 100. He then pays the insurance amount Rs10,000. After 1year, when the time has come for renewal, then the agent logs in and clicks on Renew Insurance and enters the IID and renews insurance for Rs 10,000. The different test scenarios for the above are,

Scenario 1 :
1) Login as Agent – click on New Insurance – Create IID 100 and Amount Rs10000
2) Change server date by 1year
3) Login as agent – Click on Renew Insurance – and pay amount Rs 10,000
4) Change server date to 1year
5) Login as Agent – Renew Insurance – IID 100 – Insurance Amount must be Rs 8500 since no claim has been made – thus the test is pass

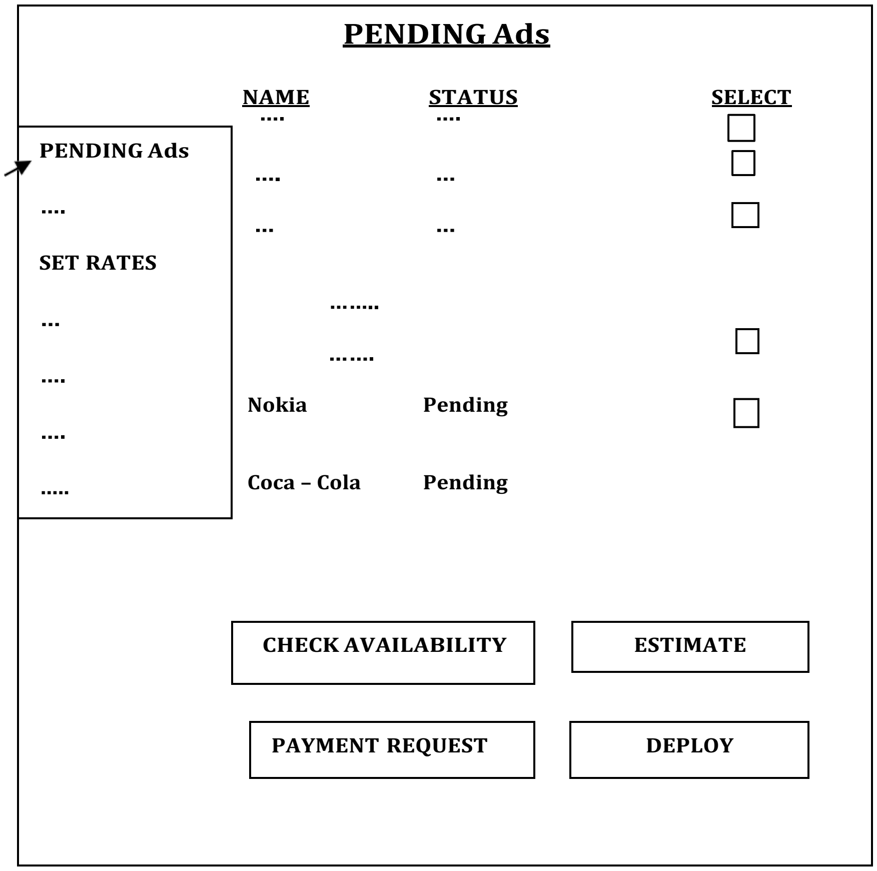
Scenario 2 :
Same as 1) , 2) , 3)
4) Before you renew for the 3rd year, Claim Insurance – Change server date to 1year
5) Login as Agent – Renew IID 100 – Rs 10000 – No discount should be made because of the claim made above.

Scenario 3 :
Same as 1), 2), 3) and 4) of 1st scenario
5) Before you renew 3rd time, - Click on Claim IID 100 , Rs 15,000 – Try to claim the amount – We shouldn‟t be able to claim this because the insurance has expired.

Let us consider another **example – Advertisements on the Internet**

When we open a website, say **www.yahoo.com** , we see an ad posted on the above – top of the homepage – it remains there for a few seconds before it disappears – this management of ads is done by something known as AMS – Advertisement Management System. Now we do s/w testing for this domain.

The below s/w works in the following way – let‟s say that Nokia wants to showcase an ad on Feb 14th at exactly 9:00AM on Yahoomail‟s homepage for the whole of Asia region. The sales manager logs into the website and makes a request for an ad dated for the above day. He attaches a file (preferably a video or image file of the Ad) and submits the application. The following day AMS manager (Advertisement Management System) of Yahoomail logs on and checks the pending ad requests. He checks the Nokia ad and checks if the space is available for the concerned day and time. If yes, then he estimates the cost of putting up the ad – at 10$ per second, the ad cost estimates upto 100$ for 10seconds. He then clicks on payment request and sends the estimate cost along with the request for payment to the Nokia manager. The Nokia manager logs onto Ad status – and checks the payment request and he makes the payment as per all the details and clicks on Submit and Pay. As soon as Yahoo‟s AMS manager receives the payment, he deploys the Ad at the concerned date and time on Yahoomail‟s Homepage.

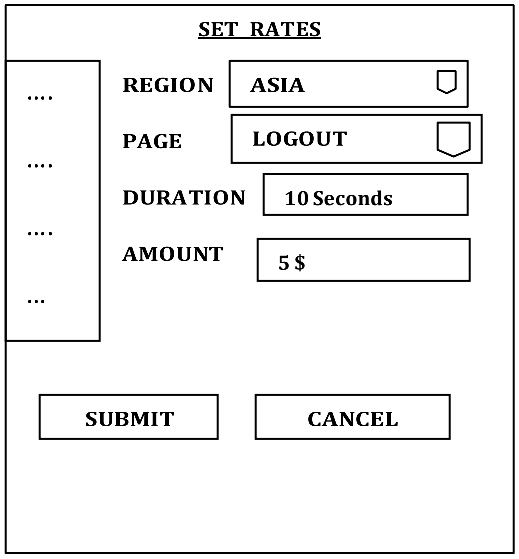


PENDING Ads

PENDING Ads	NAME	STATUS	SELECT
....	....	....	☐
SET RATES	....	...	☐
...	...	...	☐
....	....		
....	....		☐
.....	Nokia	Pending	☐
	Coca – Cola	Pending	

CHECK AVAILABILITY | ESTIMATE

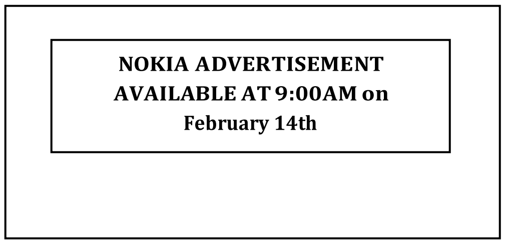
PAYMENT REQUEST | DEPLOY



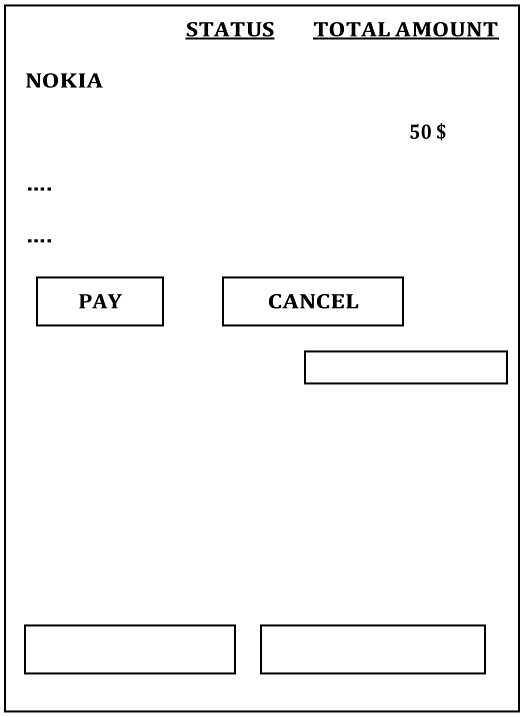
SET RATES

REGION	ASIA
PAGE	LOGOUT
DURATION	10 Seconds
AMOUNT	5 $

SUBMIT | CANCEL

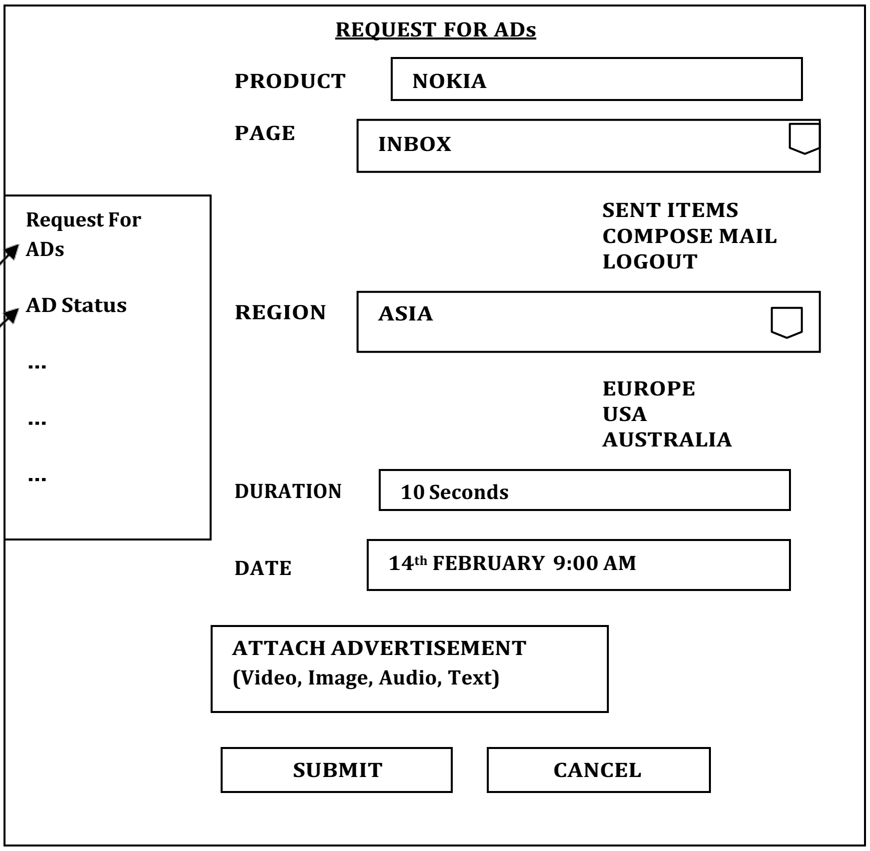


NOKIA ADVERTISEMENT AVAILABLE AT 9:00AM on February 14th



	STATUS	TOTAL AMOUNT
NOKIA		
		50 $
....		
....		

PAY | CANCEL



REQUEST FOR ADs

Request For ADs

AD Status

...

...

...

PRODUCT	NOKIA
PAGE	INBOX
	SENT ITEMS COMPOSE MAIL LOGOUT
REGION	ASIA
	EUROPE USA AUSTRALIA
DURATION	10 Seconds
DATE	14th FEBRUARY 9:00 AM

ATTACH ADVERTISEMENT (Video, Image, Audio, Text)

SUBMIT | CANCEL

The various **test scenarios** are,

Scenario 1 : the first case is the normal scenario as explained above. The test engineer does end-to-end testing for the normal scenario. Wherein the Nokia manager makes the request for the Ad and the Ad is deployed at the concerned date and time.

Scenario 2 : let us consider a scenario where-in the Nokia manager feels that the Ad space is too costly and cancels the request. At the same time, Coca-Cola makes a request for the Ad space on Feb 14th at 9AM. Since the request of Nokia has been cancelled, thus Coca-Cola‟s Ad must be deployed On Feb 14th at 9AM after all the request and the payment has been made. Now, if there is a change in heart from Nokia and they feel that they are ready to make the payment for Feb 14th at 9AM – that slot should not be given because Coca-Cola has already utilized that space. Thus , an alternative calendar must open up for Nokia to make their booking.

Scenario 3 : Login as AMS manager – Click on Set Rates – and set the rate for Ad space on logout page to 5$ per second. Login as Nokia manager – and choose the date and time to put up an Ad on the logout page. The payment should be 50$ for 10seconds for an Ad on Yahoomail‟s Logout page.

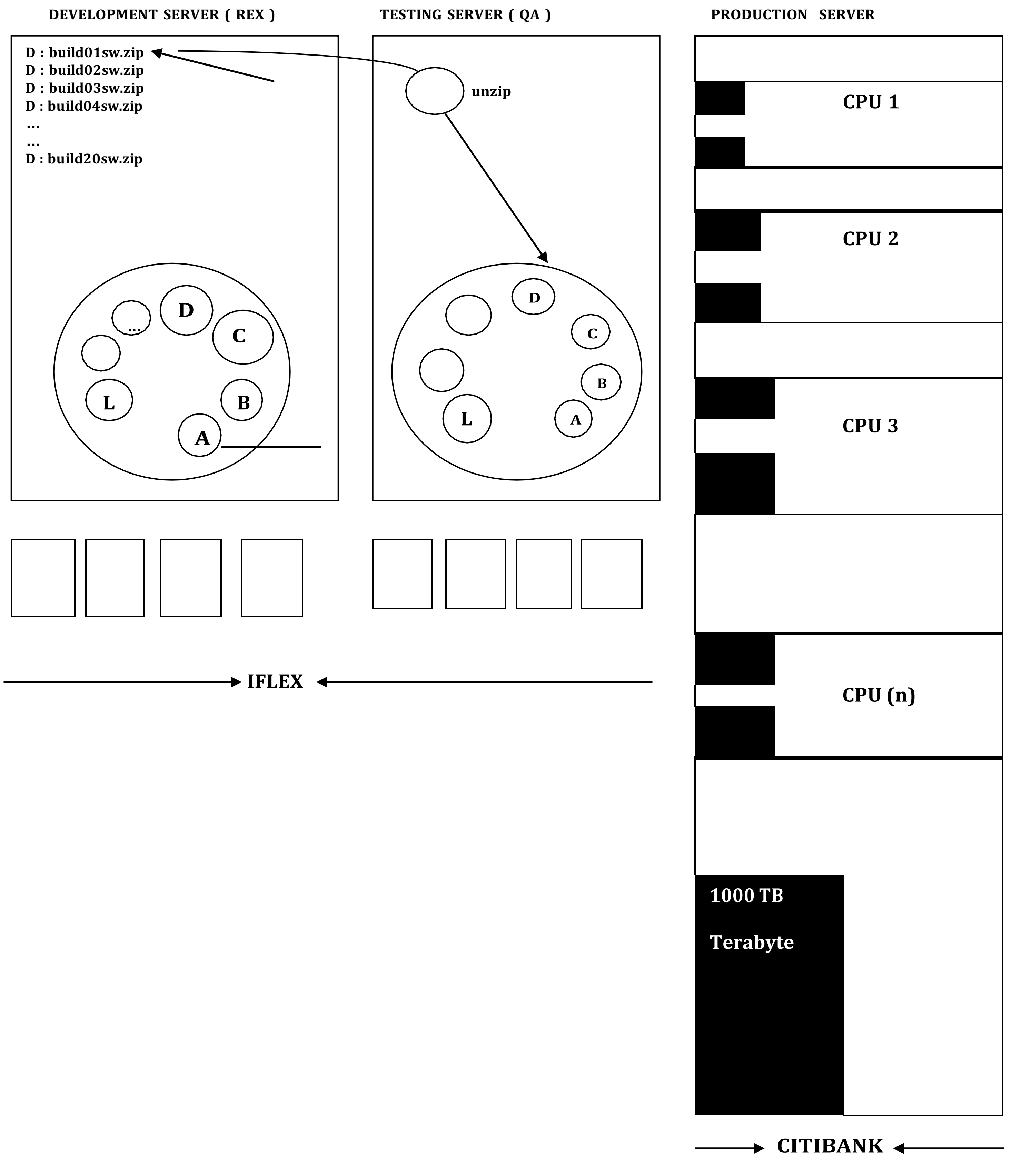
Thus, the actual definition of End-to-End testing can be given as – Take all possible end-to-end business flows and check whether in the software, all the scenarios are working or not. If it is working, then the product is ready to be launched.

Testing Environment and Why it should be similar to Production Environment ?

After the requirements have been collected and the design of the s/w has been developed, the CRS is then given to the development team for coding and building of the modules and the s/w. the development team stores all the modules and the code it builds in a development server which they name it REX (any name can be given to the server).

The development team builds module A of the s/w – does WBT – installs the s/w at http://qa.citibank.com - zips the code of module A and stores it in REX – the team lead of the development team then emails the zip file of module A to the test lead and tells him that the module A has been built and WBT has been performed and that they can start testing the module A – the test lead first unzips the module A and installs it in the testing team server named QA - the test lead then calls in the test engineers in his team and assigns them different parts of the module A for testing – this is the **first cycle** – the testing team do functional testing on A – let‟s say the testing team finds 100bugs in module A – for each bug found, the testing team prepares a report on the bug in a Word document file and each bug is assigned a number – like this, the testing team finds 100bugs in the s/w – each test engineer when he finds a bug, he immediately emails bug report to the development team for defect repair – the testing team take 5days to test module A.

The developers are reading the defect reports, goes through the code, fixes the problem – when testing team is testing s/w, the developers are fixing defects and also preparing another module and also doing WBT for the repaired program – now the developers fix majority of the defects(say 70) and also build module B – now the team lead of the development team installs the s/w at the above website, zips the code of the module B and sends a mail to the test lead containing the code – the test lead first uninstalls the old s/w and installs the new one containing module B and also repaired module A – and sends a mail to the test engineers in his team containing the new module and repaired module –


DEVELOPMENT SERVER (REX)
TESTING SERVER (QA)
PRODUCTION SERVER
D : build01sw.zip
D : build02sw.zip
D : build03sw.zip
D : build04sw.zip
...
...
D : build20sw.zip
unzip
D
C
B
A
L
...
D
C
B
A
L
CPU 1
CPU 2
CPU 3
CPU (n)
1000 TB
Terabyte
IFLEX
CITIBANK

Whenever a new build comes in, the testing team concentrates on testing the new feature first – because the probability of finding the bugs is more, we expect more number of bugs in the new feature – as soon as new build comes in,
a) test new features b) Do integration testing
c) retest all the fixed defects d) test unchanged(old) feature to make sure that it is not broken
e) in the new build, we retest only fixed defects f) each test engineer retests only his bugs which are fixed, he is not responsible for other bugs found by other test engineers.
We find new bugs in old feature because – a) fixing the bugs may lead to other bugs
b) adding new features (modules) c) might have missed it in the earlier test cycle
In the **second cycle** – we do both functional and integration testing for A and B – we find 80 bugs – each bug is sent in a report of Word format – the developers repair about 40bugs and also repair 5bugs of the remaining 30bugs in the first test cycle.
Like this we carry on, and do about 20cycles and reach a stage wherein the developers are developing the 20th build, say module L – now the testing team gets a server which is similar to the production server (real –time server on which the s/w will run at the client‟s place) – and install the s/w there – and they start off with system testing.
We start System Testing – a) when the minimum number of features are ready b) basic functionality of all the modules must be working c) testing environment should be similar to production environment
We say that the product is ready for release when,
a) all the features requested by customer are ready
b) when all the functionality, integration and end-to-end scenarios are working fine
c) when there are no critical bugs
d) bugs are there, but all are minor and less number of bugs
e) by this time, we would have met the deadline or release date is very near.
The entire period right from collecting requirements to delivering the s/w to the client is known as release. Each time a new module is built and old module is tested is known as Build – testing. Each build takes about 5days or more or less.

[redacted]

<u>The testing environment should be similar to production environment means,</u>

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

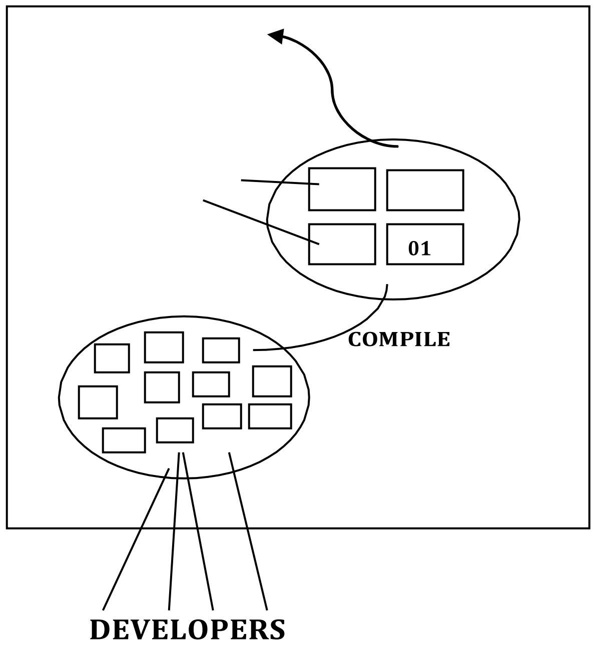
3) Data should be similar to production – a) We should create data similar to production **b)** We should create a script to create a dummy data which is similar to production environment .

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 entire and thus can be used for testing.

In Testing environment, who is involved in installing the software ?

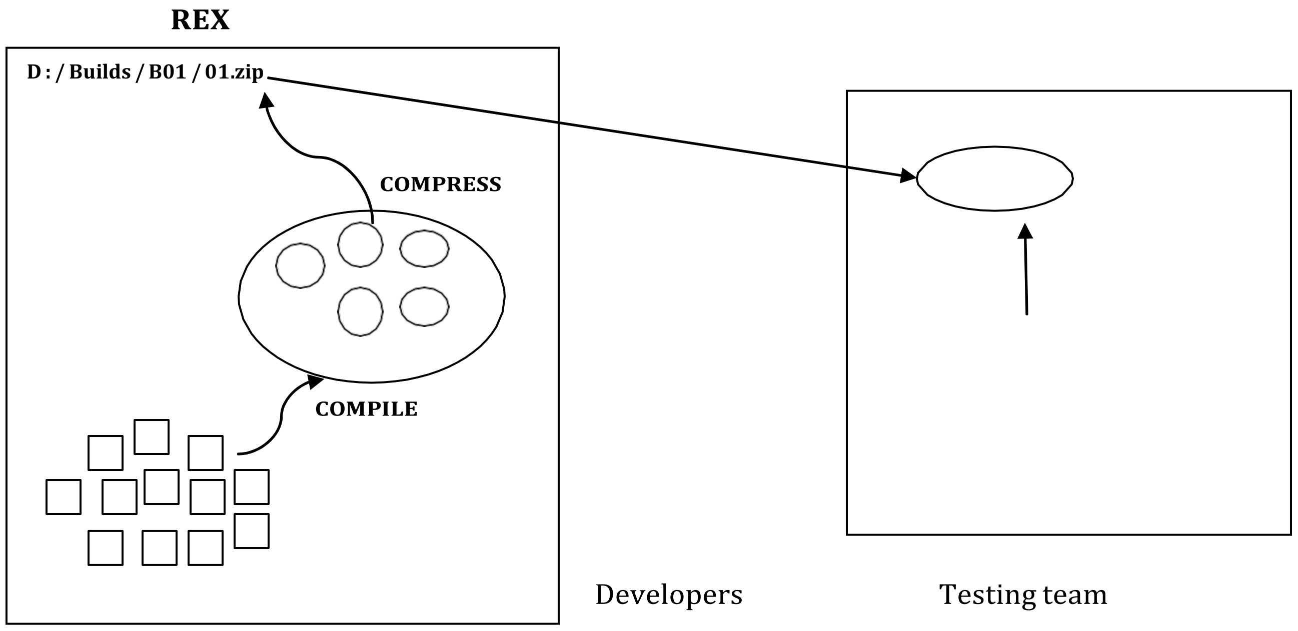
- Test engineer (anybody from testing team)
- Anybody from development team
- Release engineer / Build engineer

Build – Build is a piece of software which is copied, unzipped and installed at the testing server.



All the programs will be compiled and then compressed (compressed file should be in zip, rar, war, gunzip, tar, jar) format and compressed file is called build which is copied and pasted in the test environment, installed and we start testing the software.

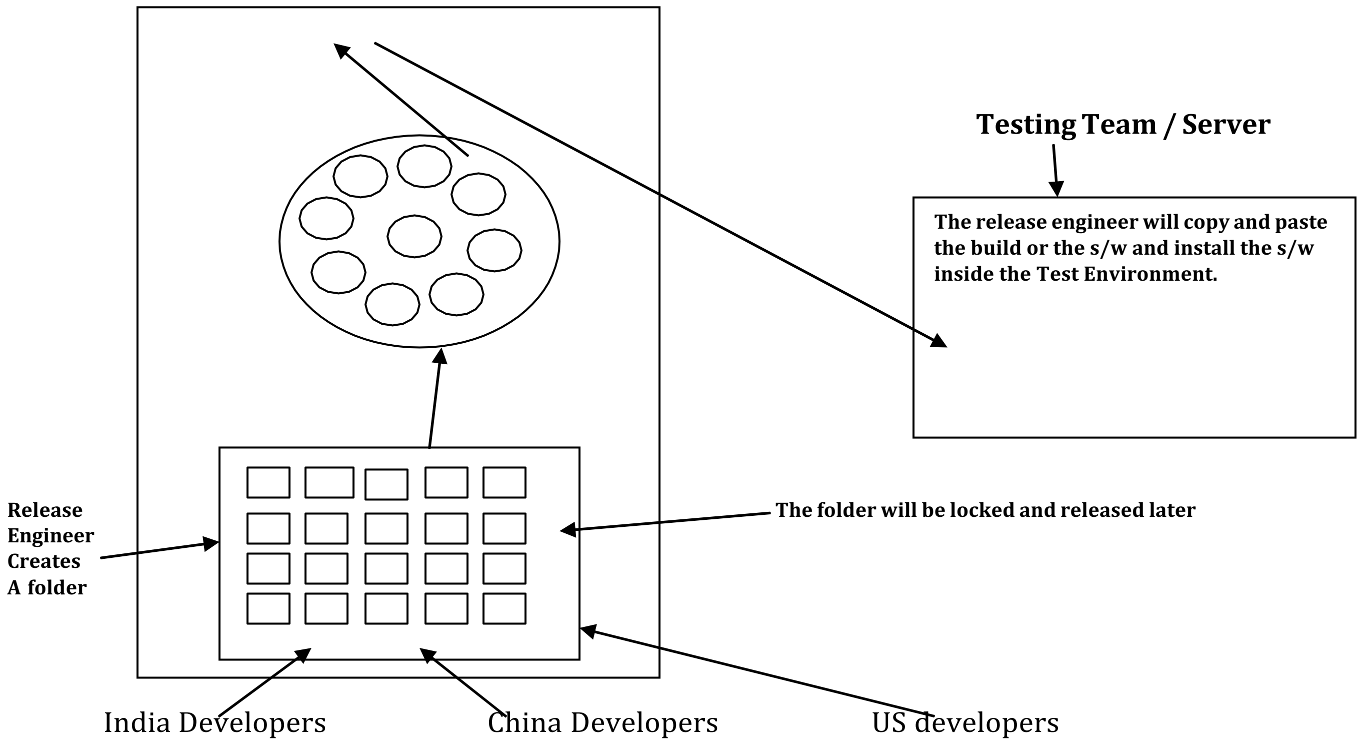
CASE 1 : **Test Engineer is installing the s/w**



Here, the developers once they get the requirements, they start developing the software. As each feature is built, the code is compiled, compressed and stored in a compressed format. In the above example, the first build is shown. As soon as the first build is ready, the development lead sends a mail to the test lead saying that the first build is ready and they can start testing it. He also gives the name of the server and the file in which the first build is stored. The test engineer then goes to the development server named Rex, and copies the file to the testing server. He then unzips the file, installs the s/w in the testing team server and allots the various features of the first build to be tested to the testing team.

CASE 2 : <u>Develop is installing the software</u>
When the developer is installing the s/w in the testing environment, and test engineer team should open the browser, copy the URL that is sent by the developer and paste it in the browser. If the developer installs, they‟ll give username and password. The test lead then logs in using the given the username and password, and creates his own username and password through which the testing team will log in and start testing the software.

CASE 3 : <u>Release engineer / Build Engineer</u>
Release engineer is the one who manages the source code. If the developers are working at different location, the release engineer first installs a VCT (Version Control Tool) like CVS, VSS, Clear Case and creates a folder in the tool. The developers should copy and paste the programs into that folder. Once the folder is locked, the developers can‟t send their programs when the folder is locked. Now, release engineer will compile, compress and then build a s/w. Now he will only install the s/w inside the test environment (testing server) and sends a mail to the testing team. The testing team then start testing the s/w. If they find bugs, they report it to the developers. Once the developers fix these bugs and creates a new module simultaneously, the folder will be unlocked and once again the developers send the programs into that folder which is locked after a new build has been developed.


Testing Team / Server
The release engineer will copy and paste the build or the s/w and install the s/w inside the Test Environment.
The folder will be locked and released later
Release Engineer Creates A folder
India Developers
China Developers
US developers

The release engineer performs 2 functions,

- Manages source code
- Creates a build and they install in the test environment

Whenever the Release Engineer compiles the program, if the program does not compile then he will send the program back to the developers and ask them to check (or) not to send that program at that particular build.

When do we find Release Engineer (ing) ?

- When the product is complex and big
- When more number of developers are there

Once the product is ready to release to the customer, then release engineer will only release the product to the customer because he will be knowing which piece of software is working fine.
The Automatic release of the Build can also be done by the release engineer. He should write a script while compiling the program. If the release engineer finds any bugs in the program, he will go and see who has written that program and sends that program to that developer (to fix the bug and send or not insert that program into that folder). **All the activities done by Release Engineer is called <u>Release Management</u> or <u>Release Engineering</u>.** For the entire project to manage resources, we use VCT. Whenever the developers want to take the program out from the VCT, we use **check out** option. Whenever the developer wants to insert a program into the VCT, we use **check in** option.

Test cycle – Time spent on testing a build or software completely / time taken by the test engineer to completely test one software.

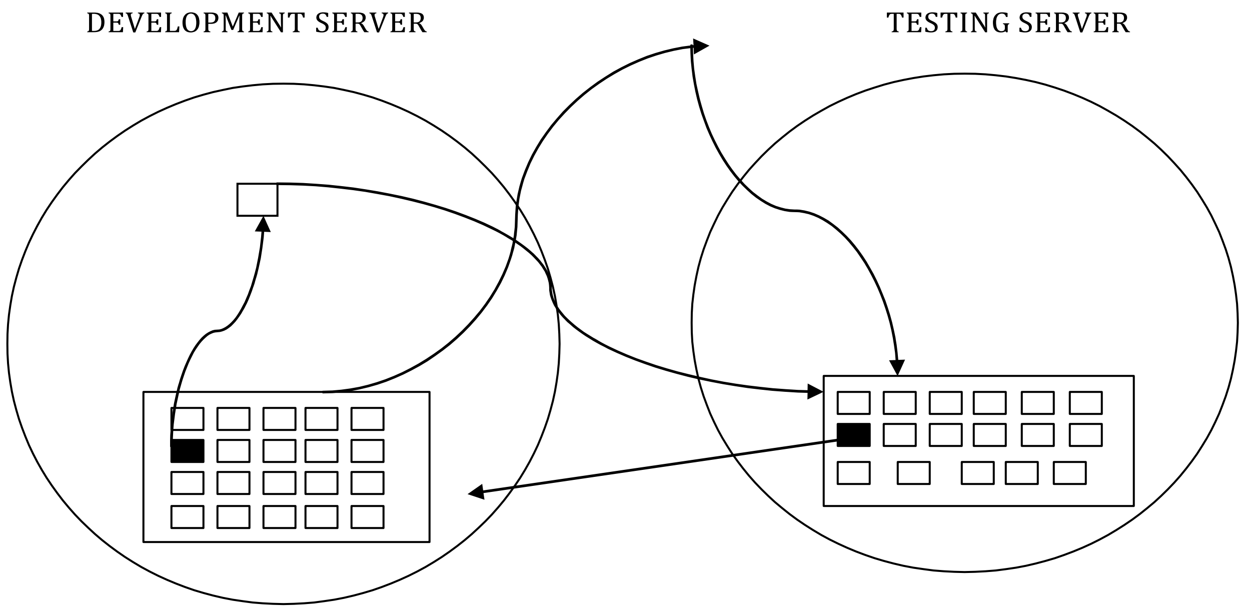
5 days – 1st test cycle	**5 days – 2nd test cycle**	**5 days – 3rd test cycle**

The developer writes a program and creates a module A and sends the module to the test engineer for testing. The test engineer tests the module and finds bugs and reports them to the developer. The developer fixes the bug and also creates a new module B. The developers only integrate module A & B and send it to test engineer. The test engineer will uninstall the old s/w and installs the new s/w and then does functional and integration testing for A & B. Whenever the new module is given for testing, if the test engineer catches bugs and reports it, the developers fixes the bugs. But, the developer does not send that particular module for testing, instead he sends the fixed module along with the new module for testing.

Whenever the new build comes, we should always uninstall the old build and install the new build (latest build). Whenever we uninstall, all the accounts and data created will be deleted. So, whenever we install new build – we must always login with manager Username and Password and create our own Username and Password.
When the first build comes in, immediately we find a bug and send it immediately to the development team. Thus, immediately we find a bug within a cycle. If another build comes within a cycle where the bug is fixed, then we call it **<u>respin.</u>**
We find respin – when the test engineer finds blocker defects / critical defects. **For ex,** if the login feature itself is not working in gmail, then the test engineer cannot continue with his testing and it has to be fixed immediately – thus respin comes into picture. Inbox, Compose mail, Sent items are not blocker defects.
More number of respin in a cycle means the developer has not built the product properly.

Whenever the developer writes so many programs and sends the module for testing. The test engineer finds bugs and reports it to the developers. Once the developer comes to know the bug, he will look into the source code, if the problem is with only 1 program and that too only a few lines – he will fix the bug. He (the developer) will take the modified program, compile it and compress it into installable (format) and sends a mail to the test engineer that a **patch file** has been sent. Once the test engineer starts installing this **patch file**, the program which had defect will be replaced by the corrected program.

Patch – is a piece of software which has only modified programs.



The above figure shows how a patch file is installed. Let us consider that the testing team have installed the build and started testing the s/w. they find bugs and report it to the development team. The block (shaded) is the defect program. The developer looks at the defect program and sees that it just needs a few minor changes. He makes the necessary changes, compiles it and compress it and creates an installable (patch) and goes to the testing server and just installs the patch file which has the modified program. The testing team need not have to uninstall and install the build again.

INTERVIEW Tips & Questions

1) What is Build ?
Ans) Answer is in the notes

2) In what format will you get builds ?
Ans) in compressed format like a zip file, tar file, rar file, jar file etc.