## EXPLORATORY TESTING

Understanding the application, identifying all possible scenarios, documenting the scenarios and testing the application by referring the document is called as Exploratory Testing

(OR)

Exploring the application, understanding how each and every feature works and testing the application as per your understanding is called as Exploratory Testing

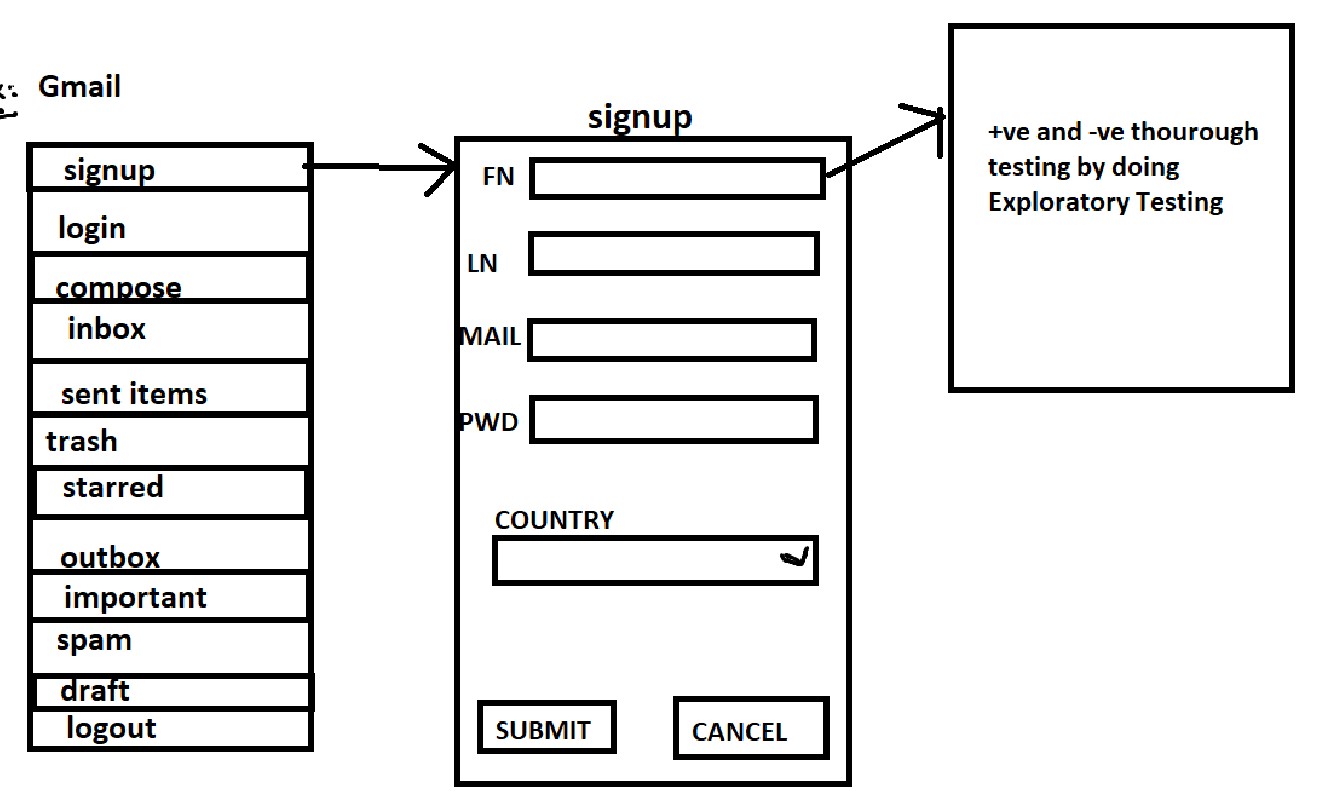
**When to do Exploratory Testing?**

* When there are no requirements
* Requirements are present but TE doesn't have time to understand the requirement (Some companies they hire TE in the mid of the project and they will not give sufficient time to understand the req.)
* There is requirement, but it is not understandable (it means it's very complex)

**Why requirements will be missing?**

* Chances are there if project is very old and big project, for the old features requirements might be missing.
* Chances are there in the product-based companies, for some products requirements might be missing. (some startup companies might not document the requirements)

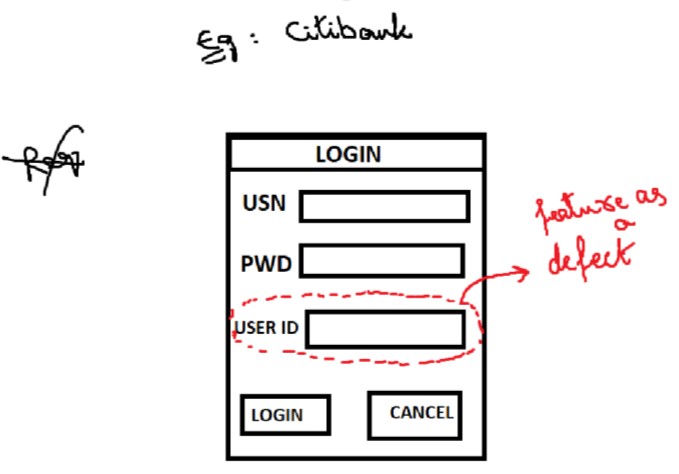
**How to do Exploratory Testing?**



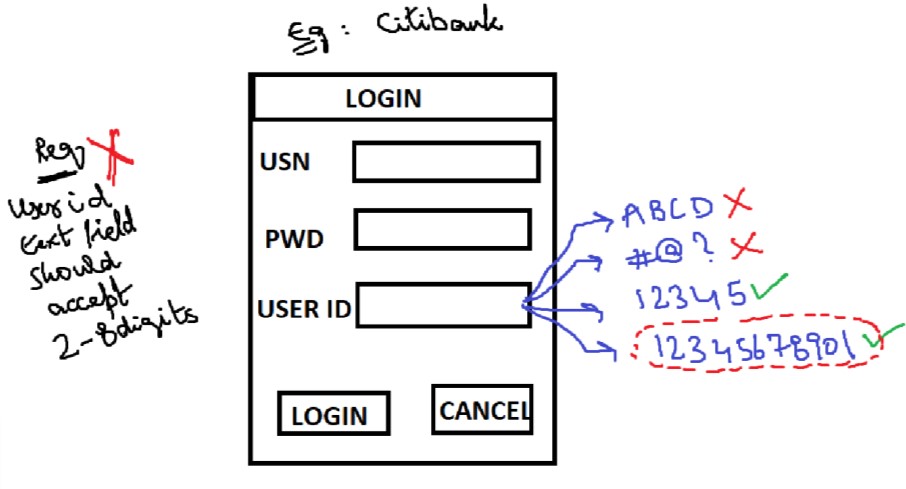
1. I will understand the application and I will try to enter all possible inputs to each and every component and I will do Exploratory Testing.
2. I will understand the application and check the dataflow b/w the modules by doing Exploratory Testing.
3. I will explore the application and I will understand how each and every feature works and I will try to test for all possible end to end scenarios by doing Exploratory Testing.

**Drawbacks of Exploratory Testing**

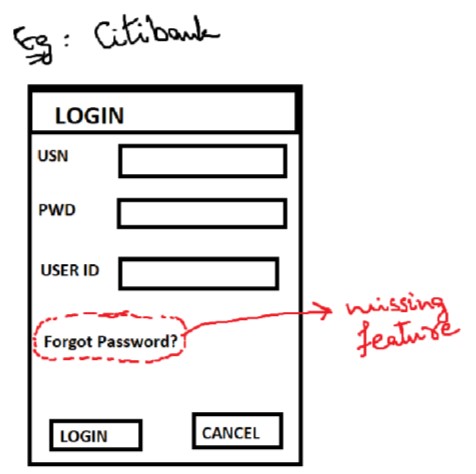
1. Chances are there we might misunderstand 'feature' as a defect.



1. Chances are there we might misunderstand 'defect' as a feature



1. If a feature is really missing, we will not get to know the feature is really missing.



1. It is more time consuming because we will spend more time in understanding the application.

**How to overcome these drawbacks?**

1. TE will interact with Sr. TE, Sr. Dev, BA, customer or talk to anybody who is having good knowledge on the application.
2. Based on product knowledge (If a person has worked on the same project for around 4-5 years and has good knowledge on that project then we can tell he is a product expert/product knowledge)
3. Based on domain knowledge (If a person has worked on multiple projects which belongs to same domain and he has got very good knowledge on the domain then we can tell he is domain expert/ domain knowledge)

Globalization Testing

1. By comparing with similar kind of application which is released in the market
2. By using some common sense, we can overcome the drawback.

***Interview questions***

* 1. Do you write Exploratory scenarios and test cases?

Yes

* 1. How to test the project when there is no requirement?

I will understand the application and I will do exploratory testing. Chances are there I might misunderstand feature as a defect or defect as a feature. So, in order to overcome this drawback, I will talk to my Sr. TE, Dev, BA, customer and based on my product knowledge, I will test the application.