



2- Test Report Document

2.1 Reporting the status of test cases related to Checkout screen to colleagues:-

- After running several test cases on the checkout screen application page, you cannot move forward without filling in all the required information such as your address which is asked in separate fields for more accuracy. After this you have to choose a payment method which is set to cash by default but you can easily choose to pay using your debit/credit card providing all the required information. Additional information such as promo code is not required. This functional is running successfully.

2.2 What information will be provided if a bug is found during manual testing, how will you make sure that the tester has reported the required information?

- ⇒ After a bug is found during manual testing, a sequence of steps has to be taken from the Tester's end to ensure that the issue is well acknowledged and understandable for the developer to solve. These steps include determining the kind of failure that happened whether it's a functionality error, validation error or a GUI issue.
- ⇒ The tester should know and report the exact steps that should be taken to replicate that issue to know the exact scenarios that execute this error and how often this error occurs.
- ⇒ The bug report provided by the tester should contain the following main points and follow this structure that makes sure that the bug is well documented and clear for developers to take action based on his report. The main points are concluded in the following :-
 - **Description of the Bug:** - briefly explaining the issue, identify when the issue has been encountered.



- **Replicate the Bug:** - providing the developer with direct steps to replicate the bug instead of writing paragraphs. For example :-
 - 1- Open the app
 - 2- Enter the credentials
 - 3- press login button
- **Provide Examples:** -cite examples if a certain test data or inputs are required to replicate the bug. For example, entering these characters “&ao*!” on the phone number text field.
- **Identify the severity and priority** – Determine the severity of the bug: For example, if it is a showstopper bug wherein you can't proceed to testing, then assign it as “Urgent” and “High” priority. Being able to understand the severity of bug shows that you relate the software to what customer needs and shows you understand the nature of the bug you are reporting.
- **State the expected result** – Specify the correct output logically and based on the requirements provided, and if appropriate, explain why.

3-Non-Functional Testing

What non-functional testing you would be considering during Mobile application testing. Give at least 10 cases.

⇒ Nonfunctional testing includes but is not limited to:-

- Performance Testing
- Usability Testing
- Compatibility Testing
- Battery Usage testing
- Security Testing

⇒ Examples on the previous nonfunctional testing items includes but is not limited to:-



- Can the app handle projected load volumes?
- What are the various mobile app and infrastructure bottlenecks preventing the app from performing as expected?
- Is the response time as expected?
- Are battery consumption, memory leaks, GPS, and camera performance within required guidelines?
- Is the current network coverage able to support the app at peak, average, and minimum user levels?
- Are there any performance issues if the network changes to/from WIFI and 2G/3G/4G?
- How does the app perform during intermittent phases of connectivity?
- Are existing client-server configurations providing the optimum performance level?
- Battery and data usage leaks
- New features and updates are not introducing new battery and data usage
- Buttons are a user-friendly size.
- Buttons location, style are consistent within the app.
- Icons are consistent within the application.
- Does the app's user interface (UI) remain consistent, visible, and accessible on various screen sizes?
- Is the text readable for all users?
- Does the app permit an attacker to access sensitive content or functionality without proper authentication? This includes making sure the communications with the backend are properly secured.
- Protect the application and the network from denial of service attacks.

4-Automation

4.1:-

1. Is Automation testing Black Box testing or White Box testing?

- Automation testing can be both black and white box type of testing depending on the scenarios in which automation is performed. For example, it is black-box testing as tester usually tests the application without knowing the low-level design or code of the application. But sometimes, automated test scripts need access to the database details that are used in the application thus it can be a type of white-box testing as well.

2. How do you decide which tests to automate? Which tests don't you automate, and why?

A test case should be automated if:

- The task is going to be repeated.
- It's going to save time.



- The requirements, the test, or the task are low risk, stable, and unlikely to change often.
- The test is subject to human error.
- The test is time consuming.
- The test has significant downtime between steps.
- The test is repetitive.

The tests that should not be automated include but are not limited to:-

- ❖ User Experience Testing
- ❖ Complex Functions
- ❖ Quality Control

While running the mentioned tests, it would be more accurate if done by a human as the automated test would not be able to think about good/bad paths to invoke each function as a real human would think.

3. What are the different types of locators?

⇒ ID – NAME – LINKTEXT – PARTIAL LINKTEXT – TAG NAME – CLASS NAME – CSS – XPATH

4. Is Automation testing in agile methodology useful? Why?

⇒ Yes, because it helps covers many different aspects such as :-

- **Incremental development:** The first and foremost factor that necessitates automation in agile testing is the short development cycle. Agile teams have only a few weeks to get a grasp of the requirement, make the code changes and test the changes. If all testing were to be done manually, the time required would surpass the actual development time. Alternatively, testing would have to be hurried, thus compromising on quality.
- **Continuous testing:** Agility demands early and continuous testing. Test coverage extends to not only the newly added code but also the code from previous iterations. This is to ensure previous functionality is not broken due to the newly added functionality. This puts a lot of pressure on the testers and can seriously affect the quality of the product. Automating some of the testing means testers have more time in hand for exploratory testing.
- **Exhaustive testing:** With automation, testing can be repeated as many times allowing detailed and exhaustive examination of the code. This is highly useful in ensuring code quality when working in a limited testing window.



5. What are the Automation Testing Challenges?

- a.** Effective Communication and collaboration between team members.
- b.** Selecting the best tool for the automation testing.
- c.** Having skilled software testers to perform the automation as not all manual testers could cover the requirements of automation testing and vice versa.
- d.** Selecting the best test approach
- e.** The huge money it costs upfront.