# SE Assignment - 2

## Problem statement chosen : Scenario #2, XYZ  a leading global mobile application service-provider for utility payment.

# A.

Before diving into planning the test strategy, one must ensure the context, project/product is well understood and any discrepancies between the testing teams should be settled. The project understanding involves reviewing the possible use case scenarios of the product, discussions with the upstream and downstream partners, playing around with the product,  performing deep down inspection and walk through of the product. Since it will be infeasible to test all the test case scenario combinations, one must prioritise the test cases based on customer requirement, schedule, budget, skills and talent of the test team. There should be a clear understanding of what is “in scope” and “out of scope” of the testing. Once this is done test Adequacy criteria should be planned to determine when to stop testing or consider testing to be completed. In the view of planning effective and efficient test strategy the following must be followed :

***1. The testing mindset or the test models that will be followed***
***Demonstration model/ mindset :***
To make sure the mobile application for payment runs well and doesn’t have any issues in the newer domains of banking and financial transactions, travel bookings and loans. Ensuring the application works for both iOS and Android platforms as well as for various kinds of mobiles and their configurations in Indian market by performing regression testing, and making sure the testing will be done to find out any bugs or issues which cannot be easily detected.

***Evaluation model/ mindset :***
To find faults throughout the application development life cycle and making a log of the error measures and metric throughout. Since cost of fixing errors goes up as we move down the phases in the lifecycle, any faults in requirements and design phase should be detected early by analysis and review techniques.

***Destruction model/ mindset:***
Try to make the application fail many times, to find as many faults since good test cases are those which finds faults in the product and decide when to stop testing based on a measure like number of faults or issues left in the system or fault density.

***Preventive model/ mindset :***
Fixing the issues which happens in the early phases of the application lifecycle can drive the cost down, by reviewing and following an effective test driven development.

2. ***Testing types which will be followed :***

***Unit Testing :*** Test for coding/ construction errors before the application goes for quality assurance test. Tests the smallest individually executable code units done by the programmers. The test cases could be for code logic/algorithms, boundary conditions, interfaces, error handling etc. It brings out any issues in the application which might have happened during the implementation phase.

***Integration Testing :*** Test to verify the interfaces and connectivity between various components in the application, can find problems like latency issues and resource contention problems which are not detectable by unit testing. Can be done either by top down or bottom up approach by integrating the unit tested components until the application works as a system. It bring out any issues which might have happened in the detailed design phase of the lifecycle.

***System Testing*** : Tests a completely integrated system to verify that its following all the specified requirements. It involves testing the end-to-end flow of an application or the software as a user, for validating if the end features work fine, with the test focused towards the whole application.It bring out any issues which might have happened in the requirements and architecture phase. Some of the system testing which could be done are the following -

- ✓ Smoke and Sanity testing - To make sure the most important functionality works, and to decide if the application is fit for further testing.

- ✓ Regression testing - To make sure all kinds of mobiles and operating systems in the Indian market and overseas are compatible with the application. These test ensure any changes made to the application doesn’t have any side effects.

- ✓ Functional & Non functional testing - to test the features/functionality of the application, covering all the scenarios including failure paths and boundary

cases. Non-functional testing looks to verify the attributes of the application such as performance or robustness of the application etc.

- ✓ Usability Testing - To determine the ease of use or to evaluate if the application is easily usable from an end-user’s perspective. How easily the application can be used by end users, level of Skill required to learn/use the application, time required to get used to in using the software.

- ✓ Localisation Testing - Performed to verify the quality of the application

- localisation for a particular target culture/locale.

- ✓ Platform or Cross Platform tests - To make sure the application runs on various operating system like Android or iOS and various mobile hardware configurations.

- ✓ Load Testing - carried out to determine the behaviour of the application or to check the robustness of the application under varying loads.

- ✓ Security Testing - Testing to uncover vulnerabilities of the application and determine that its data and resources are protected from possible intruders.

- ✓ Compliance Testing - Testing to determine the compliance of the application with internal or external standards.

- ✓ Scalability Testing - carried out to check the performance of the application in terms of its capability to scale up or scale down the number of user request load.

***Acceptance Testing :*** involves running a suite of tests on the whole system. These are high-level tests to verify the completeness of the user story and requirements. The objective is to provide confidence that the delivered system meets the business requirements of both sponsors and users. It's essential that these tests include both business logic tests as well as UI validation elements. The acceptance phase may also act as the final quality gateway, where any quality defects not previously detected may be uncovered. It usually bring out any issues which might have happened by not fulfilling a particular requirement and also showcases if any requirements were left out during the feasibility phase.

***Alpha & Beta Testing :*** Before rolling out the application for the end users the application must be tested amongst a small group of potential users either in-house (alpha) or for external users (Beta).

3. ***Test environment that will need to be available for supporting the strategy :***

It mentions the minimum hardware requirements that will be used to test the Application. Setting up a right test environment ensures success of software testing else it could result in delay, cost escalations and incorrect conclusions. For the given mobile application test bed could include :

- ✓ Test cases which include simulations of payments from newer domains like banking and financial transactions, travel bookings and loans.

- Servers having mobile Operating systems like Android and iOS simulators and emulators to test the mobile application.
- The servers in use should have  at least descent minimum configurations in terms of hardware and software to run the application and to handle varying stress and load tests.
- Actual physical mobile hardware in case a particular hardware or software mobile operating system could not be emulated or simulated in order to carry out the testing.
- Testing the front end UI  to determine the ease of use of the application and to ensure there will be no problems in payments happening from the front end.
- Testing the backend routes,  API calls to test how secure is the application so that the payment gateway is secure and not exposed to intruders/hackers.
- Testing the ACID (Atomicity + Consistency + Isolation + Durability) properties of the database to ensure all the payments are done in a consistent way and making sure there is no discrepancies in payments made.
- Testing the ROLLBACK feature if the payments has happened from the payment gateway from the frontend but backend could not verify the payment or some component in the application crashed so the payment made is refunded to the correct user.
- Testing how much network bandwidth is needed for the payments to be completed and test cases for the application under varying network conditions to test coverage of all types of users and situations with high speed/ low speed network connectivity.

## 4. ***Automation Strategy :***

Since periodic testing and running is to be done, planning a good automation strategy is of utmost importance to the company. The mobile industry is changing rapidly with new hardware and software configurations coming out to the market frequently ,so an automation framework should be set up to perform regression testing and platform cross-platform testing so that the application runs on the newer software and hardware versions without any issues. Since manual testing can be slow and tedious and hard to get test coverage, the newer test cases should be generated and run without human assistance so that a wide range of functionalities and non-functionalities can be tested  in shorter amount of time with newer versions of the application and newer versions of the hardware and software mobile configurations. Selecting such an automation tool/ framework will be of critical importance.

The application should be compatible with the tool selected and the testing team should be comfortable to use it or else full effective use of the tool will not happen. Tools chosen needs to be balanced in terms of the features offered, ability to generate reports/data needed for different stakeholders and the ease of use. Cross platform support is an expectation as automated tests would/may need to run on different platforms. Acceptability/Popularity/prevalence of the tool in the Industry is an indication of availability of support, quality documentation, technical forums and availability of trained personnel. Using open source or proprietary tools is left to the company, but proprietary tools usually offer more features.

Tools like Selenium, Capybara, QF-Test, Robot Framework etc. can be used as the automation framework. Since these frameworks are fast and repeatable, the company can make sure they are not behind their competitors by quickly developing new test cases and adapting to newer versions of configurations introduced in the market. If any issues found during testing should be immediately logged and actions should be taken to remove the issue or mitigate the issues in future. Maintaining the automation framework and test suite and keeping it up to date will ensure effective and efficient automated testing.

### 5. ***Risk identification for the strategy, analysis, contingency planning and trigger for the risk***

Risk is the probability of an unwanted incident during or towards testing. Any changes in the application logic or plans, evolving technology or competitors direction should be done in a controlled manner so no unwanted bugs or defects is introduced in the application. Changes made towards improving the quality of the product should be taken care of. If some of the test models or some type of testing could not be carried out, alternate options should be looked and used to mitigate the issues in the application. Any issues in the automation framework or the test environment should be handled by trained personnel. If any issues are found by the end user or during the testing the first step will be to find out from which component is the defect being generated, next careful assessment and mitigation planning needs to be done in collaboration with the developers and other upstream and downstream partners, once the risk assessment plan is made, triggers should be identified so that in future if similar types of issues pop up, the testing team will know where in the application is the issue happening. Once all this is done the new mitigation plan developed is executed and monitored.

### 6. ***Process Improvement suggested***

Creating a detailed test schedule with work breakdown schedule and estimating the efforts of tasks in terms of man hours and other metrics can drive the test team towards building an effective test strategy with a list of deliverables which includes test specifications for each of the modules of the application and test cases for different conditions planned. Proper planning and allocation of resources to ensure that the schedule factors in the characteristics of the planned resources. Once the resources are identified and their capacities, skills are identified, the schedule can be reworked. Risks for completion of each of the task from a schedule and quality perspective should be identified ,project milestones or checkpoints should be identified considering the deliverables expected, the schedule and commitments if any to the end users. These milestones are used to track or monitor progress and control overruns. These milestones are also used to identify any risk triggers and to help kick-in the mitigation plans. Effective communication between the upstream and downstream partners can also boost the project development and testing since everyone involved in the project will be well aware of the application and its use cases being developed. Further testing team should be organised with roles and responsibilities assigned to each member. The roles can be test director, test automation manager, test analyst, test development engineers, and software test engineers.

## C.

The measures and metrics which are planned for use

The measures which can be used are :
Fault Density - The ratio of number of faults found to the size of the programs(bugs/LOC). Higher density represents lower quality.
Defect Leakage - Indicates test efficiency  (Total Number of Defects Found in application after testing / Total Number of Defects Found Before  testing) x 100
MTBF - Mean time between failure measured in hours. Based on statistical analysis that indicate the probability of failure

The metrics which can be used are :
Requirement Compliance Factor - Using the traceability matrix, the RCF measures the coverage provided by a test case to one or set of requirements.
Defect Discovery Rate - Number of defects found per line of code (LOC).
Cost of Quality - Total cost of prevention, appraisal, rework/failure, to the total cost of the project.

## B.

Two test cases :

Test case #1
Test case Id : TC1
Title : Successful payments on Android versions 8 and above
Description : User should be able to use the application and make payments successfully on any mobile android version 8 and above. There should not be an incompatibility issues and all the features in the application should work in the intended way. UI should be the same on all compatible android versions and not tailored and designed for a particular android version.

Precondition : The user must be verified in the application by preset criteria's, must have a legitimate source of payment to initiate the payments and legitimate bank account or other forms of verified money wallets to receive the payments from the application.
Assumption : User mobile operating system is running android version 8 and above with minimum hardware requirements and good connectivity to internet.
Test Steps :

- ✓ Check the current running android version in mobile device or android simulator under test.
- ✓ If android version >= 8, check the connectivity and run the application.
- ✓ If any issues found in the above step log the issue and take actions to mitigate or prevent such issues.
- ✓ Make sure the payment sources for sending and receiving are verified sources and legitimate sources.
- ✓ Next add a source of payment and a source to receive the payments which have been verified.
- ✓ Make real or simulated payments either small or big from the portals of newer domains like banking and financial transactions, travel bookings and loans.
- ✓ Verify the payments from the start to end till the payments are received in the correct source for receiving payments.

Expected results :
Show "Successful Transaction" message with other details like the transaction ID, sender and receiver name, account numbers or wallet id of sender and receiver with an option to download the transaction page for future references.
Comments: Additional tests can be performed to verify if the payments reach within a preset time interval threshold like 2-5 seconds.

**Similar test can be made to verify for iOS version 10 and above.**

Test case #2
Test case Id : TC2
Title: Successful payments on mobile devices with network bandwidth 500kbps and above
Description : Since a user cannot have good network connectivity in all times and situations, test designed to make sure he will be able to make a successful payment transaction when he has a minimum of 500Kbps connectivity speeds. He must not face any kind of latency issues if he has the minimum network speed specified.
Precondition: The verified user must be connected to the internet either by cellular network or any type of local area networks.
Assumption : User mobile is running android version >=8 and iOS version >=10 with a minimum network connectivity speed of 500kbps.
Test Steps:

- ✓ Check the network connectivity speeds on the mobile device or any

- simulator or emulator and verify if there is any firewall enabled which prevents from connecting to some of the portals needed for payment transactions.
- If network connectivity speed is less than 500kbps alert the user saying transactions cannot be made, if network connectivity is absent then display a message stored in the cache of the application saying to connect to the internet, any firewall related issues should be alerted to the user and other issues should be logged and taken care of.
- Once source of payment and destination of payment are verified either simulate the payments or make real payments either small or big from portals of newer domains like banking and financial transactions, travel bookings and loans.
- Test to make sure the roll back feature of the transactions work without any kinds of issues in case the network connectivity drops below the minimum speed or if the connectivity gets lost.
- Simulate network congestion and if its taking longer to send and receive the payments then roll back the payment else continue with the transaction.
- Test if the payments are sent and received at correct specified sourced within a time interval threshold without latency.

Expected results :
Show “Successful Transaction” message with all the other relevant details like transaction ID, sender and receiver names and account numbers or wallet id, with an option to download the transaction page for future reference.
Comments : Additional test cases to test how much longer will the transaction take to complete if network speeds are less than 500kbps and can it be in acceptable limits.