TEST PLAN

CARD PAYMENT

Table of Contents

1

Introduction

This document describes the scope, objective, resources and schedule of all integration-testing activities of card payment functions that will be used by the ABC team. The test plan is intended for project managers, Business Analyst, product developers and QA engineers.

1.1 Objectives

The Primary objective of the project is to ensure the card payment work properly if the conditions are met. The objective of this testing activity is to check the integrations points and layers of a card payment system for web browsers (Chrome, Firefox, Edge, Safari) and Android and iOS devices.

The secondary objective of testing application systems will be to: identify and expose all issues and associated risks, communicate all known issues to the project team, and ensure that all issues are addressed in an appropriate matter before release. Must have functionality is considered more important than the delivery date in this project.

1.2 Team Members

Resource Name	Role

2 Scope

2.1 In Scope

Integration layer testing is in the scope of this test plan. Systems under test includes Commbank app, Netbank and any other system interacts with this layer. The following components/functions would be tested:

1. Card payment request from a merchant with transaction information to Integration layer through message Queue
2. Enrichment data input from DW to Integration Layer
3. Message enrichment in Integration Layer
4. Transferring the enriched message to Fraud Decisioning engine
5. Response back to the payment system through Integration layer

2.2 Out of Scope

1. Testing of transactions in money withdrawal using ATM, Mobile payment, Direct Debit and Schedule payments.

3 Assumptions / Risks

3.1 Assumptions

This section lists assumptions that are made specific to this project.

1. Transactions from the Merchants are manipulated or simulated from internal team to avoid dependency throughout the project.

3.2 Risks

The following risks have been identified and the appropriate action identified to mitigate their impact on the project. The impact (or severity) of the risk is based on how the project would be affected if the risk was triggered. The trigger is what milestone or event would cause the risk to become an issue to be dealt with.

#	Risk	Impact	Trigger	Mitigation Plan
1	Dependency with the Merchant systems	High	Change in functionality from Merchants	Each iteration, functionality will be closely monitored. Priorities will be set and discussed by stakeholders. Since the driver is functionality and not time, constant collaboration with Merchant is mandatory.
2	Changes to the functionality may negate the tests already written and we may loose test cases already written	High – to schedule and quality	Loss of all test cases	Export data prior to any upgrade as necessary and re-import after upgrade.
3	Risk of Resource retention or constraint in project budget	Medium	Product did not get delivered on schedule due to cost cutting	

4 Test Strategy

The project is using an agile approach, with bi-weekly iterations. At the end of every 2nd week, the requirements identified for that iteration will be delivered to the team and team will develop and test.

4.1 Test Automation

Automated tests are part of the iteration process. For every new component/api introduced/changed, backend automation needs to be performed as a part of iteration story itself.

4.2 System Test

As soon as the iteration/sprint stories are delivered, individual components are tested as per the requirements. During the repeated cycles of identifying bugs and taking receipt of new builds (containing bug fix code changes), there are several processes which are common to this phase across all projects. Both positive scenarios and error scenarios should be captured as a part of System Testing.

4.3 Integration Test

Once all the components are delivered, there should be integration testing cycles which involves testing end to end functionalities. The following paths are to be tested

a. Card payment request from Merchant to Integration Layer
b. Enrichment data from DW to Integration Layer
c. Enrichment data from Integration Layer to Fraud Decisioning Engine
d. Response from decision engine to Integration Layer and then to Card payment
e. Full payment cycle from Merchant back to Card payment successful message

4.4 Performance Test

Performance Test is run for every individual component and after integration, it should be run for whole system. Load testing should be done with Jmeter tool by providing intense load of users accessing the system to find the break point of the system.

4.5 Security Test

Since this functionality involves card payment (Fund related), complete security test should be carried out at every stage of the testing. Suitable Hashing should be used to hash the values of Customer DOB, Customer post code and Transaction amount.

4.6 Usability Test

The purpose of usability testing is to ensure that the new components and features will function in a manner that is acceptable to the customer. UX/UI team should be involved in the design and UI testing should be carried out based on the UX wireframes.

4.7 User Acceptance Test (Alpha/Beta)

Alpha testing should be co-ordinate within the organization. All stakeholders and other team members should be invited to participate in the Alpha testing.
Beta Testing should be arranged with external stakeholders like Merchants who are going to interact with the system, identified users and business teams.

5 Test Criteria

5.1 Suspension Criteria

If the test team report that there are 40% of test cases failed, suspend testing until the development team fixes all the failed cases.

Exit Criteria
- Attempted 100% of planned test cases unless a clear reason is given.
- No outstanding severity 1 or 2 defects (unless agreed with business)
- A resolution plan for any outstanding severity 3, 4 or 5 defects
- Pass rate is 80%, achieving the pass rate is mandatory.

5.2 Resource planning

5.1 System resource

No	Resources	Description
1	Server	MySql server (database) server Apache (Web) server
2	Project Backlog Management	JIRA/Confluence
3	Browsers	Chrome, Firefox, Edge, Safari
4	Devices	Android, iOS
5	Build	Microsoft Azure
6	Test Case tool	qTest or HP QC or JIRA
7	Defect Management	JIRA

5.2 Human resource

No	Member	Tasks
1	Test Manager	Manage the whole project Define project directions Acquire appropriate resources
2	Test Lead	Identifying appropriate test techniques/tools Verify and assess the Test Approach Create test summary report
3	Test Analyst	Test case creation and execution, log defects
4	Test Environment Admin	Ensure test environments are available for testers

6 Test Resources and Environment

6.1 Test Tools

HP QC or qTest can be used as a tool for test case management. Alternatively for cost effective approach, Jira can be used as a tool for test case management.

6.2 Test Environment

a. SIT
b. UAT
c. Azure Blob Storage

6.3 Bug Severity and Priority

Severity ID	Severity	Severity Description
1	Critical	The module/product crashes or the bug causes non-recoverable conditions. System crashes, GP Faults, or database or file corruption, or potential data loss, program hangs requiring reboot are all examples of a Sev. 1 bug.
2	High	Major system component unusable due to failure or incorrect

		functionality. Sev. 2 bugs cause serious problems such as a lack of functionality, or insufficient or unclear error messages that can have a major impact to the user, prevents other areas of the app from being tested, etc. Sev. 2 bugs can have a work around, but the work around is inconvenient or difficult.
3	Medium	Incorrect functionality of component or process. There is a simple work around for the bug if it is Sev. 3.
4	Minor	Documentation errors or signed off severity 3 bugs.

6.4 Bug Report/Defect Management

Testing team recognizes that the bug reporting process is a critical communication tool within the testing process. Without effective communication of bug information and other issues, the development and release process will be negatively impacted.

Testing's standard bug reporting tools and processes will be used. BugDB or Bugzilla can be used for bug trakcing. Since the project is run with Jira/Confluence, bugs can be managed in Jira itself.

7 Milestones / Deliverables

7.1 Test Schedule

The initial test schedule follows

Task Name	Start	Finish	Effort	Comments
Test Planning				
Review Requirements documents				
Create initial test estimates				
Staff and train new test resources				
First deploy to QA test environment				
Functional testing – Iteration 1				
Iteration 2 deploy to QA test environment				
Functional testing – Iteration 2				
System testing				
Regression testing				
UAT				
Resolution of final defects and final build testing				
Deploy to Staging environment				
Performance testing				
Release to Production				

7.2 Deliverables

Deliverable	For	Date / Milestone
Test Plan	Project Manager; QA Manager; Test Team	
Traceability Matrix	Project Manager; QA	

	Manager	
Test Results	Project Manager	
Test Status report	QA Manager, QA Director	
Metrics	All team members	