Project Sample

Test Plan

DOCUMENT DATE

August 2023

PREPARED FOR

Client A

PREPARED BY

Neil Casaandra Sudharmono
Quality Assurance
Addresss
22 Example Street North Sydney NSW 2060
ABN: XX XXX XXX XXX

Table of Content

Document Information

Details	Version Approved
Title	Project Sample
Author	Neil Casaandra
Draft Distribution	
Final Distribution	
Status	Draft

Document Revision

Version	Date	Modified By	Comments
0.1	26/09/2023	Neil Casaandra Sudharmono	Initial Draft

Document Approval

Name	Version Approved	Position	Date
Reviewer 1	1.0	Technical Director	
Reviewer 2	1.0	Business Director	
Reviewer 3	1.0	CEO	

0. Introduction

Company A is an Australian not-for-profit organisation that provides legal education and training to law graduates and admitted practitioners which is headquartered in Sydney, with operations in Brisbane, Melbourne, Perth, Adelaide, Auckland, Singapore and the UK. As the leading provider of Admission to Practise programs, we teach over 10,000 students per year and employ over 300 staff.

Our approach to teaching and learning differs from universities in three key respects:

- Our offerings focus on developing practical, workplace-ready skills and knowledge for career success. Our teachers are not academics but practising lawyers, and our learning activities are designed to simulate tasks and develop innovative solutions to challenges encountered in practice.
- Content authoring and course creations are managed centrally.
- Our offerings are flexible to fit in with our students' work commitments, with frequent start dates, choice of study modes and formats (primarily blended or online).

The purpose of this document is to gather all information necessary to plan and control the test effort for the Project App and Admin portal. This document describes the approach to test the software, and is the top-level plan generated and used by managers to direct the test effort.

This document supports the following objectives:

- Identify the items that should be targeted by the tests.
- Outline the testing approach that will be used.
- Specify the deliverable elements of the test project.

1. Testing Scope

This section will describe the types of testing that QA will run and features that will be included in the release phase.

1.1 Test Type Coverage

Types of tests executed for this project are mentioned in the table below.

Type	Description
Unit Testing	● To validate the smallest possible unit of code, such as a function or a method by developer
API Testing	● To test the API as early as possible to identify issues that might have occurred within it.
Integration Testing	● To ensure the web application can interact with other parties such as Optimizely, Mulesoft, Salesforce, CRM, etc.
Compatibility Testing	● To ensure the web application is compatible with different types of device and browser, which are Android and iOs based phones as per design.
E2E Functional Testing	● To ensure all the functionality of the UI is comply with the requirement and design.
CMS Testing	● To ensure content management capabilities of the web application.
Load Testing	● To ensure system could handle the expected load with the expected number of users
Stress Testing	● To find how a system behaves in extreme conditions such as extremely many amount of user accessing a page in the same time

Regression Testing	● To ensure existing and new features are working as expected before release
User Acceptance Testing	● A phase of in which the web is tested by the intended audience or business representative
Production Verification Testing	● To ensure the release is successfully deployed to production environment

1.2 Scope Requirements

Functional requirements of the system that is approved, finalised, and expected to be developed are described in

- Requirements: LINK
- Design: LINK
- Test Scenarios: LINK

In high level, the features and CMS testing that are included within this release are:

Testing Types	**Scope**
Functional Testing	● B2C Customers ○ My Account Page ○ Add to cart for single course ○ Add to cart for bundle course ○ Add to cart for subscription ○ Checkout as new user ○ Checkout as existing user ○ Refund enrollment ○ Exchange course ● B2B Customers ○ My Organisation Page ○ Add to cart for single course ○ Add to cart for bundle course ○ Add to cart for subscription ○ Order on behalf as new user ○ Order on behalf as existing user ○ Refund enrollment ○ Exchange course ● General ○ Sign In

	○ Sign Out ○ Search & Filter single course ○ Search & Filter bundle course ○ Search & Filter subscription ● Payment ○ Valid payment method ○ Decline payment method ○ Expire payment method ○ Insufficient balance payment method
CMS Testing	● Home Page ● All Course Catalogue Page ● Course Catalogue Page ● One Column Content Page ● Cart Page ● User Detail Page ● Checkout Page ● Order on behalf Page ● Search Page ● Error Pages ● Email templates ● Cookie Notification Pop Up ○ ● Blocks ○ Accordion ○ Text ○ Container Two Columns ○ Course Catalogue ○ CTA Banner ○ CTA Card Container ○ FAQ Block ○ Feature Cards ○ Image Container ○ Partner ○ Practice Area Container ○ Presenter Biography ○ Testimony ○ Product Recs Subscription ○ Product Tecs Bundle ○ Testimonials
Compatibility Testing	● Devices that we are going to test are ○ Desktop - MacOS (admin Portal) ○ IPhone 11 - iOs mobile ○ IPad 9 Pro - iOs tablet ○ Samsung Galaxy Note 8 - Android mobile ● Browsers that we are going to test are ○ Desktop - Chrome, Edge, Firefox, Safari ○ iOs - Safari ○ Android - Chrome

	● We use actual devices to test
API & Integration Testing	● Colpass ID generation ● Colpass ID De-dup check ● Colpass ID Regions (AU, NZ, UK) ● B2B customer flow ● B2C customer flow ● Refund flow ● Subscription renewal flow ● Course creation integration ● Customer portal integration
Load Testing	● Perform Load testing to ensure normal load ○ Maximum concurrent users load (peak time) – 1000 users ○ Response Time – less than 5 seconds ○ Throughput value – 20 TPS ○ Duration 1 hour
Stress Testing	● Perform Stress testing to ensure extreme load ○ Maximum concurrent users load (peak time) – 2000 users ○ Response Time – less than 5 seconds ○ Throughput value – 20 TPS ○ Duration 2 hour

1.4 Out of Scope

The QA Team will not execute tests to verify the following features:

- Security Testing
- CRM Functionality Testing

1.5 Risks

Based on the business needs of the system, the following risks are identified. The identified risk will be tested by the QA team, as a mitigation plan to minimise risk of the product. Risks are mentioned in the table below.

Category	Quality Risk	Mitigation
Product	Availability of devices for testing may impact the testing process	To assess and provide the required test devices
Product	Load testing will be performed locally which might resulting using the same IP for all virtual users, hence not reflecting the actual condition	Development team will research tool that can accommodate this, until then, we will still run the load test locally
Product	Actual content of the site are not fully ready after each sprint is finished	Development team to run a content that are similar to the figma design which reflects on the actual content
Product	Purchasing can't use actual payment method	We use stripe's testing card for payment scenarios

2. Test Strategy

2.1 Test Design

- In this project, we will identify test scenarios instead of detailed step by step test cases. By this approach we could fast track the test case writing and achieve more test coverage within a shorter time period.
- The created test cases then will be given to the product owner for review. Here is an example of it:

Scenario ID	Feature	Scenario
TC001	Sign Up	Verify user can successfully sign up with valid first name, username, password, and checked checkboxes

2.2 Test Approach

This section includes detailed information about the various types of tests conducted by the QA team. Also, this section explains the following with respect to each testing type.

- Entry Criteria – The pre-conditions that needs to be satisfied to start the given testing type
- Exit Criteria – The conditions to conclude the given testing type

Type	Entry Criteria	Exit Criteria
Functional Testing	● Feature are developed, deployed, and ready to be tested in test environment ● If previous testing failed, the related feature / bugs has been fixed and deployed to the test environment	● All requirement and validations are covered within the test ● If failed, testing activities will be stopped and QA will raise the defect to the team
CMS Testing	● Front-end and Back-end are integrated ● Feature are developed, deployed, and ready to be tested in test environment ● If previous testing failed, the related feature / bugs has been fixed and deployed to the test environment	● All requirement and validations are covered within the test ● If failed, testing activities will be stopped and QA will raise the defect to the team
Compatibility Testing	● Front-end and Back-end are integrated ● Devices are available ● If previous testing failed, the related feature / bugs has been fixed and deployed to the test environment	● Feature is consistently appeared as per design for each agreed browsers and devices ● If failed, testing activities will be stopped and QA will raise the defect to the team

Integration Testing	● 3rd party is integrated to the system ● If previous testing failed, the related feature / bugs has been fixed and deployed to the test environment	● All requirement and validations are covered within the test ● If failed, testing activities will be stopped and QA will raise the defect to the team
API Testing	● API specifications, documentation, and requirements should be finalised and available for reference. ● API is ready to test in the Postman environment	● All planned API test cases, including positive and negative scenarios, should be executed. ● If failed, testing activities will be stopped and QA will raise the defect to the team
Load Testing	● Functional testing is done ● System are already stable ● We have environment that have similar specs with the production ● If previous testing failed, the related feature / bugs has been fixed and deployed to the test environment or specs has been scaled up	● Test result is passed as per requirement ● If not, testing activities will be stopped and QA will raise the defect to the team
Stress Testing	● Functional testing is done ● System are already stable ● We have environment that have similar specs with the production ● If previous testing failed, the related feature / bugs has been fixed and deployed to the test environment or	● Test result is passed as per requirement ● If not, testing activities will be stopped and QA will raise the defect to the team

	specs has been scaled up	
Regression Testing	• Functional testing is done • Latest build before release is deployed to test environment • Must be conducted in E2E basis before release	• Existing feature is run without issues • If not, defects found during regression testing will be raised
User Acceptance Testing	• Regression testing is done • Latest build before release is deployed to test environment • Must be conducted in E2E basis before release	• Existing feature is run without issues • If not, defects found during regression testing will be raise
Production Verification Testing	• Functional testing is done • Latest build before release is deployed to production environment	• Existing and New feature is run without issues • If not, defects found during will be raised

3. Test Environments

The test environment outlined in this section is necessary for successful execution of test cases. Here are the environments and the allowed activity in each of the environments

Environment	Activity
DEV Environment	• DEV environment is developers playground to check and test the developed features • QA will occasionally check DEV environment for

	verification
STAGE Environment	- STAGE environment will be heavily used by QA and possibly Content Manager to test out the developed feature - Load and Stress Testing will also be used in this environment - STAGE environment can also be used for UAT process with client - STAGE environment should have the same specification with production environment
PROD Environment	- The Live site and only accessible to the Client - DEV and QA might have access to investigation purposes - QA will use this environment for Production Verification Testing

4. Test Execution

4.1 Test Metrics

The following test metrics will be used to monitor test progress and product quality. The metrics must be correlated to gain more understanding. These metrics will form part of project progress updates.

Metrics Name	**Description**
Test pass rate	- Test case pass rate is calculated by dividing the number of passed test cases with the total number of executed test cases. - Testing considered passed when test cases that passed is above 90% with no critical bugs
Test coverage	- Test coverage determines whether test cases are covering the application code and how much code is exercised when we run those test cases.
Load test pass rate	- Load Testing considered passed when error is below

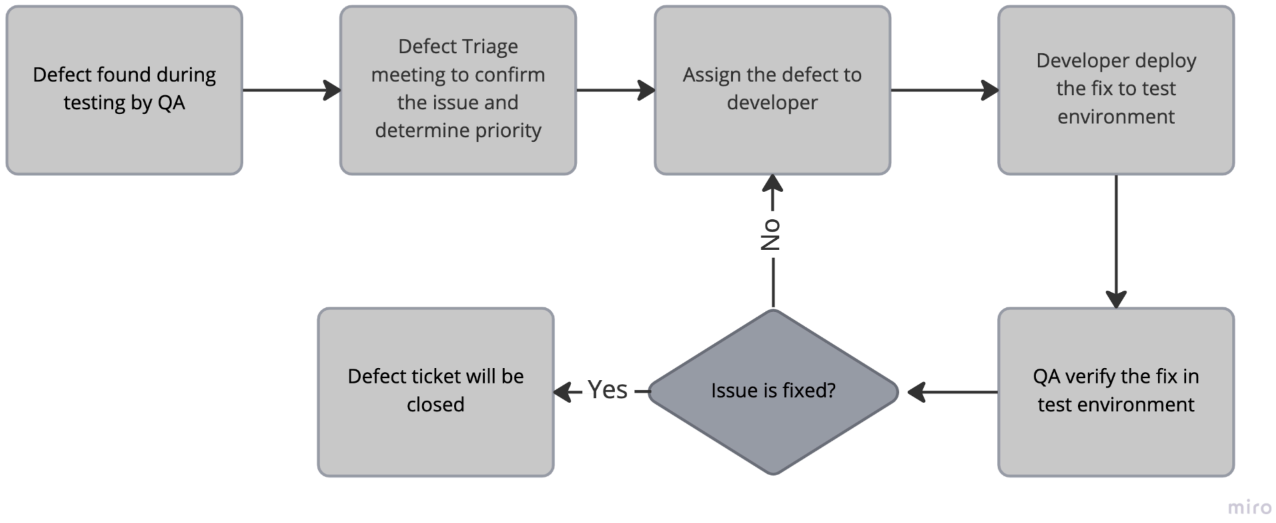
	3% with no critical bugs

4.2 Defect Management

This section contains information about all the defect management related activities in the project.

4.2.1 Internal Defect Tracking Process

In this section, we will provide the defect tracking process if the issue is found in our internal testing

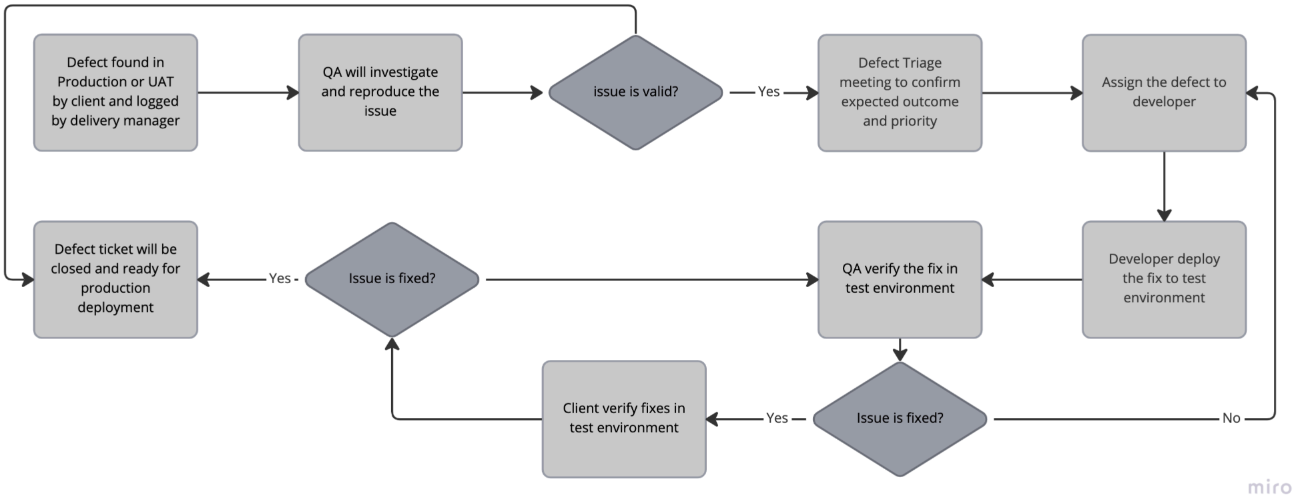

Defect found during testing by QA
Defect Triage meeting to confirm the issue and determine priority
Assign the defect to developer
Developer deploy the fix to test environment
QA verify the fix in test environment
Issue is fixed?
No
Yes
Defect ticket will be closed
miro

- QA engineers will log the defects in the bug tracking tool when the defects are found.
- Triage meetings will be held to discuss the identified defects and to clarify any concerns from the Development. Priority of the defects will be decided and set.
- A developer will be assigned to each defect to analyze and to find the resolution.
- Once the defects are fixed, they will be deploy the fix for QA to test
- Defects released to QA will be retested and closed if it is fixed, if not, QA will assign back to the developer for another fix
- If a defect is considered not to be a defect (unable to replicate/ existing requirement etc.) the QA Engineer who submitted the defect should withdraw it.

- If a defect identified is a requirement defect, it needs to be assigned to the PM or BA to update the requirements document. Once the requirements document is updated the defect should be closed by the QA Engineer who submitted the defect.

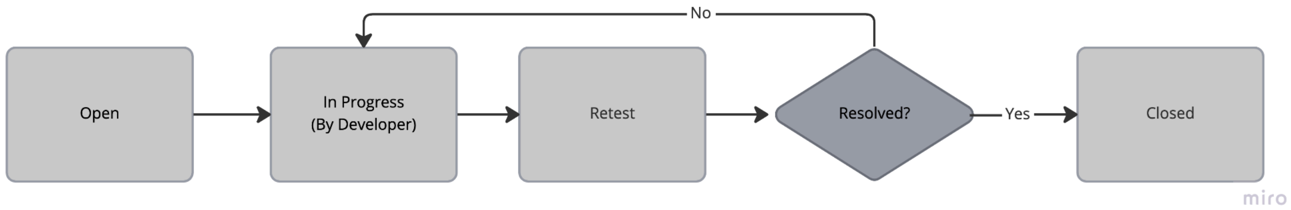
4.2.2 Production Defect Tracking Process

In this section, we will provide the defect tracking process if the issue is found in the production environment or by a client in the UAT environment.


Defect found in Production or UAT by client and logged by delivery manager
QA will investigate and reproduce the issue
issue is valid?
Yes
Defect Triage meeting to confirm expected outcome and priority
Assign the defect to developer
Developer deploy the fix to test environment
QA verify the fix in test environment
Issue is fixed?
No
Yes
Client verify fixes in test environment
Issue is fixed?
Yes
Defect ticket will be closed and ready for production deployment
miro

- If a Client found an issue in UAT or Prod, Delivery manager will log the defects in the bug tracking tool when the defects are found.
- Defect will be assigned to QA for investigation and if it is valid, we can discuss in detail in the defect triage meeting and to determine priority
- A developer will be assigned to each defect to analyse and to find the resolution.
- Once the defects are fixed, they will be deploy the fix for QA to test
- Defects released to QA will be retested
- If the issue is fixed, the client can verify it also in the test environment. if not, QA will assign back to the developer for another fix
- If the client is satisfied with the fix, we can mark the ticket as ready to be deployed, if not, ticket will assigned to QA for re-testing
- If a defect is considered not to be a defect (unable to replicate/ existing requirement etc.) the client side will close the defect status
- If a defect identified is a requirement defect, it needs to be raised as a change request and another prioritisation should be determined.

4.2.3 Defect Status Life Cycle



- Once a defect is raised it will be in open status.
- Open defect will be assigned to developer for fixing
- When the defect is fixed and deployed to test environment, it will be changed to retest status and released to QA
- QA will verify the defect fix and if it is fixed will change the status to closed else it will be reopen to In progress and the defect will be assigned to the relevant developer for another fixing
- The same cycle will continue until the defect is closed.

4.2.4 Defect Triage

A classification of defect (bug) to indicate the degree of urgency of it to be fixed. Usually the higher the priority, the higher the severity also.

Priority	Description	Example
• **Highest**	• There is no workaround for the issue. • Total failure of the product or unrecoverable data loss. • Security showstoppers are available. • Impossible for testing to proceed.	*The login page could not be directed to the Home page even providing valid & correct, username and password.* *Error 500 when visiting homepage*
• **High**	• Affects major functionality or major data. • Workaround available but it is not obvious and is difficult.	*The login page directs to the Home page and a top up can be added to the system entering the mobile number. But scanning the QR code does not read the mobile number.*

• **Medium**	• Minor functionality or non-critical data is impacted. • Easy and obvious workaround available.	*The login page directs to the Homepage and one of the menus is not working. But can go to the page where the menu directs from another link.*
• **Low**	• Does not affect functionality or data. • Does not need a workaround. • Defect causes minimal or unnoticeable problems. • Mostly cosmetic errors.	*The login page directs to the Homepage and one of the images in the home page is missing.*

5. Roles & Responsibilities

Phase	Test Lead	Tech Lead	PM	BA
Requirement Analysis	I	I	A&R	R
Test Case Development	A&R	C	C	C
Test Case Sign Off and Review	A	R	R	R
Test Execution	A&R	I	I	I
Test Report	A&R	I	I	I
QA Pre-Live Checklist	A&R	I	I	I

R – Responsibility   A – Accountability   C – Consulted   I – Informed

Responsible: Those who do the work to achieve the task. There is typically one role with a participation type of responsible, although others can be delegated to assist in the work required

Accountable: (also Approver or final Approving authority) those who are ultimately accountable for the correct and thorough completion of the deliverable or task, and the one to whom responsible is accountable. In other words, an Accountable must sign off (Approve) on work that the responsible provides. There must be only one Accountable specified for each task or deliverable.

Consulted: Those whose opinions are sought; and with whom there is two-way communication.

Informed: Those who are kept up to date on progress, often only on completion of the task or deliverable; and with whom there is just one-way communication