Torrecamps Marketing

Test Strategy

Prepared By:

Aloya, Jayson

Langcauon, John Christopher

Prion, John Gabriel

Sajul, Marc Julian

Zamora, Marc

Table of Contents

1. Scope

This document will cover the general testing strategy for Concave's Torrecamps Marketing: Inventory system and ordering system. This document will give a detailed summary of the approach, tools, schedules, and procedures. All of which are to give proper feedback on Concave's project, to help them deliver a top-quality product for their clients.

This document will be reviewed by the testing team (Roles and Responsibility section), and the Quality instructor Mr. Jose Eugenio L. Quesada. Approval will come from Optimum Five's Quality Assurance Manager, Marc Julian Sajul, and the Concave team.

2. Test Approach

Process of Testing

- Planning and Control

  When preparing for testing, the team needs to keep a few key things in mind. The team must fully understand what they are testing and why it is tested. Additionally, a test plan is not complete without a risk plan. These general keys are needed to create a comprehensive test plan.

  After understanding the goals, objectives, and the risks. When the team's plan is in place, it will move forward to the execution phase. If executed, the test control comes into place. Test control is crucial because at this stage the team must monitor the progress of the tests and make any necessary adjustments along the way. During the testing includes tracking defects and issues and ensuring that your testing stays on schedule.

  Effective test planning and control are critical to ensuring that the software that is tested is high quality and meets user needs. By adhering to testing best practices, the team can detect issues early on and prevent costly mistakes in the future.

- Analysis and Design

  When it comes to testing analysis and design, it is imperative to have a crystal-clear understanding of the project's objectives. Meticulous planning and control of the testing process are crucial.

  In the Analysis and Design lies the test basis and the determining and identifying the test conditions, analyzing the software and its behavior. It

reviews the product requirements, the system architecture, and back and front design specifications like interfaces. It includes the devices and the software itself tested.

By thoroughly analyzing its data and designing effective testing strategies, the team can guarantee that the software or application is performing as intended, which is based on the documented use cases. With scrupulous attention to detail and a steadfast focus on precision, you can create a testing plan that will lead to triumphant outcomes.

- Implementation and Execution

  In the implementation and execution lies the actions that should be done during the testing progress of the project system. It starts when the team completes the planning and analysis of the project and proceeds to execute the tests. Things can get tricky because many different types of tests can be run. Some common types include unit tests, integration tests, system tests, end-to-end tests, and acceptance tests.

  What should be done when implementing and executing the tests is that no matter what type of test is currently running, it's important to be documented. Doing this will help the team keep track of what was tested, what the results were, and what changes need to be made.

  When the progress is documented, it is easier for the teams to record, update logs, and thus change and keep track of the changes that were recorded in the testing phase.

  Overall, testing the implementation and execution is one of the crucial parts of any development process. By being thorough and methodical in testing, the team can catch bugs and other issues before they become bigger problems down the line.

- Evaluation

The evaluation stage will look further and outsee a variety of factors, including the validity and reliability of the test, as well as the appropriateness of the test for the population being evaluated.

Additionally, it is needed to measure the results of the test executions that are done which some of the tests may be against the objectives. In the evaluation stage, the team may go back and double test if there are left or more tests that need to be done. Once everything is all set, the team will fully document the testing progress with a summary report.

It's important to interpret the test results in a meaningful way and use them to inform any necessary interventions or support. Overall, a thorough and thoughtful approach to testing evaluation can provide valuable insights into an individual's abilities and needs.

- Test Closure

  The final stage of the process of testing is the Test Closure. In this phase, the team must perform the checking of all the deliverables that were accepted and signed off, archive the test cases that were done, close the environment that the team was in when testing the project, and analyze the lessons that were related to testing.

  The closure is important, cause with the test closure, not only the tested project will be improved and developed than it was before the progress of testing, but the testers may also apply themselves to testing and the ability to provide features, wants and needs of their future to be developed project systems.

Levels of Testing

- Unit Test

  The Unit test will look at the parts of the software and will study it thoroughly. It checks the component's functionalities.

- Integration Test

  The Integration test will check the data flows or components that revolve around the project system.

- System Test

  The System test is the final test that verifies that the system meets a specific specification either from the software or hardware. It also evaluates both of the system's functionality needed for the testing.

- Acceptance Test

  The acceptance test is technically outside the testing area of the testers or the developers of the project. It is mainly done by the user who uses the system or its customers.

Roles and Responsibility

QA Manager	Marc Julian Sajul
QA	John Christopher Langcauon
QA	Jayson Aloya
Tester	Marc Zamora
Tester	John Gabriel Prion

Types of Testing

1. Functional Testing
    i. System Testing

        Functional testing is a method of testing that foresees and validates the system against the users, customers, stakeholders, requirements, and specifications.

        It checks if the system is functioning properly through testing aligned with emulated scenarios.

    ii. Acceptance Testing

        Acceptance testing is a method of outside testing that checks the acceptability of the system and whether it reaches the standards from the needs or wants of the users, customers, stakeholders, requirements, and specifications. Here at stake is the delivery acceptance of the system.

        It is the last stage alongside the other testing and is mainly done after it is QA tested and before being released to the public services.

Testing Approach & Tools

The following are the software used alongside what the software is for:

GitHub	a code hosting platform for collaboration and version control.
GitBash	an application for Microsoft Windows environments that provides an emulation layer for a Git command line experience.
Laravel	is a free and open-source PHP web framework.
Xampp	is a cross-platform web server that is free and open-source. XAMPP is a short form for Cross-Platform, Apache, MySQL, PHP, and Perl.
Visual Studio Code	Visual Studio (VS) Code is an open-source code editor primarily used to correct and repair cloud and web applications coding errors.
Microsoft Team	is the ultimate messaging app for your organization—a workspace for real-time collaboration and communication, meetings, file and app sharing. *(Formal Communications)*
Discord	was made for online gamers as a free alternative to voice chat services like TeamSpeak, Mumble, and Ventrilo *(Informal Communications)*

3. Test Environment

The Quality Assurance team consists of 5 members which require equipment and setup good for Five people to be able to do the assigned task in testing the developed project.

The environment of testing is applicable on-premises and online and would require the following to able to progress the testing:

Location:

On-premises:	Asia Pacific College
Online:	Own residence

Device (Minimum):

CPU: Intel Core i5 13th Gen
GPU: Integrated Graphics
RAM: 8 GB DRR4 RAM
Storage: 250 GB SSD Storage

Operating System:

Not lower than Windows 7 32-bit

Microsoft Edge, Google Chrome, OperaGX, Mozilla Firefox

Connectivity:

3G/4G Connectivity

As for the requirements and data that will be used to progress the testing within the test environment. The QA team relies on the following:

GitHub	Resource and Source code sharing
Microsoft Team	Formal communication to the developers and advisers
Discord	Unformal communication used by the QA team

4. Testing Tools

To enable proper and seamless execution of the testing phase, a combination of both automation and test management tools will be needed. Shown below are the tools that will be used for the improvement on testing efficiency:

- **Jira**
  - Jira will be the test management tool for the entire testing procedure. Jira's features can be comparable to Selenium, as they both provide a seamless testing environment that is purpose is to improve the efficiency and precision of the testing procedure.
- **Test Link**
  - Test Link will be the documentation tool for the testing team to track their Test plan, use cases, and test cases. Test Link's enables testers to track their current progress on their testing phase and update them as the testing timeline progresses.

5. Release Control

RELEASES	WHEN	WHERE	WHO
Release 1	09/05/2023	Local PHP in GitHub	Concave
Release 2	10/02/2023	Local PHP in GitHub	Concave

6. Risk Analysis

Risk	Mitigation Plan:	Contingency Plan:
1. **Data Integrity Issues**	**Interface Testing:** Conduct thorough interface testing to ensure that the inventory and ordering systems integrate seamlessly. **Use of Standard Protocols:** Ensure that the systems use standardized protocols for data exchange, making integration easier.	If integration issues arise, have a backup plan for manual data entry or a temporary workaround to keep operations running smoothly while resolving the problem.
2. **Integration Challenges**	**Interface Testing:** Conduct thorough interface testing to ensure that the inventory and ordering systems integrate seamlessly. **Use of Standard Protocols:** Ensure that the systems use standardized protocols for data exchange, making integration easier.	If integration issues arise, have a backup plan for manual data entry or a temporary workaround to keep operations running smoothly while resolving the problem.
3. **Performance Bottlenecks**	**Performance Testing:** Identify and address performance bottlenecks in the systems. **Scalability Planning:** Ensure that the systems are designed to scale.	If performance issues affect operations, have a plan for temporarily reducing system load by manual processing or prioritizing critical tasks.
4. **Security Vulnerabilities**	**Security Testing:** Conduct security testing to identify vulnerabilities. **Access Controls:** Implement strict access controls and authentication mechanisms to protect sensitive data.	In case of a security breach, initiate an incident response that includes isolating affected areas, notifying stakeholders, and implementing security patches.

5. User Training	**User Training:** Provide comprehensive training for system users to ensure they understand how to use the inventory and ordering systems effectively. **User Documentation:** Develop user-friendly documentation for reference.	In case of user errors or misunderstandings, offer ongoing support and troubleshooting assistance from system developer.
6. Scope Creep	**Scope Documentation:** Clearly define the project scope in the initial project plan and the test strategy document.	In case of scope changes, assess their impact on the project and communicate these changes to stakeholders.

7. Review and Approvals

Name	Role	Signature	Date
Marcus Medina	Scrum Member		
Ken Angelo Carangan	Product Owner		
Jan Terrence Francisco	Scrum Master, Lead Programmer		