<TheAvengers>

<Test Plan>

Version <1.3>

Revisions

Date	Version	Description	Author
<18/12/2024>	<1.0>	Features not to be tested, item pass/fail criteria, schedule and risks and contingencies	Agustín Prieto Páez
<04/01/2025>	<1.1>	Test items, test deliverables, responsibilities and staffing and training needs	Javier García Tercero
<05/01/2025>	<1.2>	Introduction, suspension criteria and resumption requirements, environmental needs and approvals	Carmen María Noblejas
<0-/01/2025>	<1.3>	Test plan identifier, features to be tested, approach and testing tasks	Lucian Andrei Negoita

Index

Test plan

To prescribe the scope, approach, resources, and schedule of the testing activities. To identify the items being tested, the features to be tested, the testing tasks to be performed, the personnel responsible for each task, and the risks associated with this plan.

1. Test plan identifier

AppForDevices-TestPlan-v1.3

2. Introduction

The purpose of this document is to define the testing strategy, scope, objectives, schedule and resources required for our software system AppForDevices.

AppForDevices enables users to purchase or rent electronic devices as well as leave reviews and keep track of the maintenance activities.

Testing is essential to ensure the functionality, performance and security of the software guaranteeing a good, seamless and satisfactory user experience for customers.

2.1. Objectives of Testing

The goals of testing are as follows:

- Ensuring that the users can filter, rent, leave reviews, maintain and purchase devices successfully.
- Verify the accuracy of rental period calculations and device availability.

- Ensure that the payment processing is secure.
- Confirm that the stock and inventory management and order work properly.

In conclusion, testing will mitigate risks, identify potential defects and verify compliance with security and accessibility standards.

2.2. Background

This year, the team TheAvengers was asked to develop a software system for managing the rental, purchase, maintenance and reviews of electronic devices. The team collaborated to define the requirements and design the different use cases to meet the business needs.

The main goal of the system is to streamline processes for renting, purchasing, maintening and reviewing electronic devices. AppForDevices offers flexible solutions for individuals and businesses, enhancing accessibility to electronic devices.

This release introduces enhanced security features, expanded payment options, and an improved User Interface based on customer feedback. Testing is needed to validate these updates and ensure backward compatibility with previous versions.

2.3. Scope of Testing

This test plan covers a full system testing of the AppForDevices. Testing includes:

- User interface testing: verify user interactions and workflows
- API processes testing: validate backend processes and data exchanges.
- End-to-end functionality testing: ensure complete system functionality across all the modules.
- Performance testing: Under various load conditions to evaluate system stability and responsiveness.

The objective is to validate all systems interfaces, user procedures and critical functionalities. This ensures that the application operates as expected under different scenarios.

2.4. References

The following documents were used as sources of information for the test plan:

- Project Authorization - [DeviceRent-Purchase-Authorization-2025]
- Project Plan - [DeviceRent-Plan-v1.2]
- Quality Assurance Plan - [QA-Plan-Devices-v1.0]
- Configuration Management Plan - [Config-Manage-Devices]
- Relevant Policies and Standards:

- Data Protection Policy - [DPP-2024]
- PCI DSS Compliance - [PCI-DSS-Guide-2025]
- ISO 27001 Security Guidelines - [ISO-27001-Devices]
- WCAG 2.1 Accessibility Compliance - [Accessibility-Guide-2024]

3. Test items

The following test items are identified for evaluation, along with their version/revision level and transmittal media characteristics:

3.1 Test Application:

Name: **App For Devices**

Version: **1.1**

Transmittal Media: Deployed on a **remote staging environment** accessible at https://dev.azure.com/SE2425TheAvengers/SEIIProject. Test credentials are provided by the development team.

Hardware Requirements: Compatible with Selenium tests executed on:

- Google Chrome (v114.0) with ChromeDriver.
- Microsoft Edge (v115.0) with EdgeDriver.
- Test environment requires at least 8 GB RAM and Intel i5 processor or equivalent for local testing.

Pre-Testing Transformation: No transformations are required; the application is deployed and ready for automated testing.

3.2 Test Framework:

Name: **Selenium WebDriver**

Revision Level: Latest NuGet package version installed in Visual Studio 2022.

Transmittal Media: Included in the project's NuGet dependencies within the test automation solution.

Pre-Testing Transformation: Ensure all NuGet dependencies are restored before running tests.

3.3 Test Scripts:

Name: **Automated Test Scripts**

Version: **1.0.0**

Revision Level: Commit hash `abc1234` in the **Azure DevOps Repository**.

Transmittal Media: Cloned from the repository to the local test environment.

Pre-Testing Transformation: The scripts require configuration of environment variables (Preconditions, username, password) that could be found in dbo.Users head before execution.

3.4 Excluded Items

The following items are explicitly excluded from testing:

Third-Party Integrations:

Name: "Payment Gateway Integration"

Justification: Payment functionality is handled by an external vendor, and integration testing is out of scope for this phase.

Mobile Application Testing:

Justification: Mobile app testing will be handled in a separate project.

4. Features to be tested

Device Purchase

Test Cases:

Filtering devices by name, brand, model, and color.

Adding a device to the purchase cart.

Removing a device from the purchase cart.

Completing the purchase process.

Associated Design Specification: Use Case: Purchase Device.

Device Rental

Test Cases:

Filtering devices available for rental.

Adding devices to the rental cart.

Removing devices from the rental cart.

Validating rental period calculations

Associated Design Specification: Use Case: Device Rental.

Device Maintenance

Test Cases:

Filtering repairs by name and scale.

Adding repairs to the maintenance list.

Removing repairs from the maintenance list.

Associated Design Specification: Use Case: Maintain Device.

Leaving Reviews

Test Cases:

Filtering devices to leave reviews on.

Adding comments and ratings.

Validating review data submission.

Associated Design Specification: Use Case: Leave Review.

General Application Features

Test Cases:

Verifying UI compatibility on Microsoft Edge.

Validating API connectivity and functionality before running the app.

Confirming that the application initializes correctly in a Windows 11 environment.

Technical Prerequisites

Before testing the application:

1. Ensure the API is running and accessible.
2. Use Visual Studio 2022 for running and debugging the test scripts.
3. Hardware requirements: Windows 11, Intel Core i7 12800H, 16GB RAM, 1TB SSD.
4. Web Browser: Microsoft Edge.

5. Features not to be tested

Identify all features and significant combinations of features that will not be tested and the reasons.

The input of the payment method when trying to create a receipt will not be tested as a value is given by default. Here we are working with an InputSelect.

The nullability on the items when creating a receipt, rental or purchase will not be tested as in the previous step, if the user has not selected any items to be added to the shopping cart, the continue option will not be available. An example of this will be the Alternative Flow 3 on the UC Maintain Device. When there are no repairs in the shopping cart, the shopping cart does not appear. Therefore, you can’t continue if you haven’t added any repairs on the shopping cart because there is no button.

We didn’t test the functionality of the dialogs either.

6. Approach

6.1 Main testing activities

The test environment must be configured:

- Configure the testing environment on Windows 11 with Visual Studio 2022.
- Validate that the API is operational before initiating tests.
- Ensure the application’s web view is accessible via Microsoft Edge.

The feature testing must be done for:

- **Device Purchase**: Validate filtering, cart management, and checkout workflows.
- **Device Rental**: Ensure proper rental flow, including filtering and rental period calculations.
- **Device Maintenance**: Test filtering and management of repair tasks.
- **Leaving Reviews**: Verify data input validation, submission, and storage.
- **General Application Features**: Assess UI compatibility, application initialization, and API connectivity.

The testing must be automated and manual :

- Use Selenium WebDriver for automated functional testing.
- Perform manual exploratory testing for edge cases and scenarios that automation may not cover.

Error identification and reporting will be conducted as follow:

- Log all identified defects, specifying steps to reproduce, expected versus actual behavior, and severity.
- Critical errors (e.g., crashes, data corruption) will be prioritized for resolution.

6.2 Techniques and tools

API Testing: Validate API endpoints for functionality, manually test endpoints using Swagger for exploratory and quick validation of edge cases, automate API tests using unit tests (XUnit with Moq for mocking dependencies) to verify the correctness of responses.
Functional Testing: Will be automated test written in Visual Studio using Selenium WebDriver. Manual testing for API, Database creation and update and UI edge cases.
Non-Functional Testing: API validation to ensure connectivity and performance and compatibility testing to verify smooth operation on Microsoft Edge.
Coverage Assessment: Achieve at least 90% coverage using automated test suites and perform additional tests to cover high-risk scenarios identified during exploratory testing.
Traceability: Link test cases to use cases and requirements documented in the specification and use Azure DevOps for tracking test execution and requirements coverage.

6.3 Completion

Testing can be considered complete when:

1. All functional test cases pass with no critical or major defects.
2. Automated tests achieve at least 90% code coverage.
3. Manual tests confirm the resolution of identified high-priority defects.
4. Defect error frequency falls below 5% for major workflows.

6.4 Constraints

The resources are limited to the hardware (Windows 11, Intel i7, 16GB RAM, 1TB SSD) and the testing must be conducted using Microsoft Edge as the designated browser.
For the dependencies, the API must be fully operational before the web view testing begins.

6.5 Deadlines

Tests must align with project sprints:

1. Sprint 1: 16/09/2024 - 18/10/2024
2. Sprint 2: 21/10/2024 - 22/11/2024
3. Sprint 3: 25/11/2024 - 10/01/2025

7. Item pass/fail criteria

Specify the criteria to be used to determine whether each test item has passed or failed testing.

If you execute all the tests on the Test Explorer on the IDE Visual Studio and they all pass, then the tests have passed. The test must be coherent. We are not going to accept trivial tests like [Fact] public void test{Assert.True(true)}.

8. Suspension criteria and resumption requirements (Carmen)

8.1. Suspension criteria

The system will be suspended if any of the following conditions occurs:

- Critical Defects Detected: Any defect that causes the system to crash, leads to data corruption or blocks core functionalities such as:
    1. Device purchase, rental, maintenance or leaving review cannot be completed.
    2. Payment processing fails.
    3. User authentication or authorization prevents access for multiple users.
- Environment Instability: Testing environments experience outages, performance degradation, or lack of necessary resources.
- Incomplete Test Data or Scripts: Missing critical test data or failure in automated test scripts that prevents meaningful test execution.

8.2. Resumption requirements

When a new version of the system is transmitted to the test group after a suspension of testing has occurred, a regression test will be run.

9. Test deliverables

The following documents will be generated by the system test group and will be delivered

to the configuration management group after test completion:

Test documentation:

- ❖ System Test Plan
- ❖ System Test Case Specification 1
- ❖ System Test Case Rental Specification
- ❖ System Test Case Purchase Specification

❖ System Leave Review Test Case Specification

Test data:

(1) Copies of all data entry and inquiry screens and the reply screens are to be attached to the related test case document.

(2) Copies of the input and output test files should be delivered to the configuration management group.

10. Testing tasks

10.1 Preparation Tasks

Setup

1. Configure the development and testing environments:
   - Install Visual Studio 2022.
   - Verify the operating system (Windows 11) and hardware specifications (Intel i7, 16GB RAM, 1TB SSD).
   - Ensure the Microsoft Edge browser is updated to the latest version.
2. Set up the API:
   - Deploy the API on a staging server.
   - Validate connectivity using Swagger.
   - Configure web view access to the API.

Data preparation

1. Populate the database with test data for devices, models, and reviews.
2. Create specific data sets to cover edge cases (e.g., missing fields, invalid input, and boundary conditions).

Configuration

1. Install and configure tools for automated testing:
   - Selenium WebDriver for UI tests.
   - XUnit and Moq for API tests.
2. Ensure integration with Azure DevOps for test tracking and execution.

10.2 Execution Tasks

Unit Testing

1. Execute unit tests to validate API endpoints
2. Ensure all test cases pass with expected outputs.

Integration Testing

Test interactions between the API and the web view:

1. Validate data consistency.
2. Verify proper error handling for failed API calls.

Functional Testing

Execute functional tests for:

1. Filtering devices by parameters (e.g., brand, year, and name).
2. Purchase, rental, and maintenance workflows.
3. Review creation and submission.

Validate receipt/detail generation and error handling for mandatory fields.

Reporting

Log defects in Azure DevOps with detailed information, including:

1. Steps to reproduce.
2. Expected vs. actual behavior.
3. Severity and priority.

Then create a bug and resolve the issue that will be revised by the Scrum Master for approval.

10.3 Intertask Dependencies

API testing must be completed and validated before starting UI testing.
Automated test scripts require mock test data and environment setup.

10.4 Skill Required

1. API Testing:

   Familiarity in XUnit, Moq, and API debugging using Swagger.

2. UI Testing:

   Experience with Selenium WebDriver and test script development.

3. Test Management:

   Familiarity with Azure DevOps for tracking test cases, coverage, and defect reporting.

11. Environmental needs

11.1 Hardware

Web Server: Minimum 8-cores CPU, 32 GB RAM, 500GB SSD.
Application Server: 16-core CPU, 64GB RAM, scalable cloud-based storage.
Database Server: 16-core CPU, 128GB RAM, 1TB SSD (replicated for redundancy).

11.2. Software

11.2.1 Operating System

The production operating system will be used to execute these tests.
- Operating System: Ubuntu 20.04 LTS (for servers)
- Windows 11 (for desktop client tests)

11.2.2 Communications Software

All on-line programs will be tested under the control of the test communication software.

11.3 Security

Security will be limited to existing controls

11.4 Tools

The tools required for system testing include:

- **Functional Testing:** Selenium WebDriver
- **Performance Testing:** Apache JMeter
- **API Testing:** Postman

11.5 Publications

The following documents are required to support systems testing:

- Use Case - Buy Device
- Use Case - Device Rental
- Use Case - Leave Review
- Use Case - Maintain Device
- Test Case Purchase Specification
- Test Case Rental Specification
- Leave Review Test Case Specification
- Test Case - Maintain Devices

12. Responsibilities

The following groups have responsibility for segments of the testing.

12.1 System test group.

The group provides the overall management of the testing and the technical testing expertise and responds to the System Test Cases Reports.

12.2 Development project group.

This group transmits the system to be tested. This group does any program debugging that is required. It also supplies the database auditor.

13. Staffing and training needs

Specify test staffing needs by skill level. Identify training options for providing necessary skills.

13.1 Staffing.

The following staff is needed to carry out this testing project.

13.1.1 Test group.

Test Manager (Scrum master) 1

Junior Test Analyst and Technician 3

13.2 Training.

The Testing Department personnel must be trained to execute and document test cases effectively. The *ISTQB Software Testing Body of Knowledge (STBOK)* and the *IEEE 829 Standard for Software Test Documentation* will serve as the foundation for this training.

14. Schedule

Include test milestones identified in the software project schedule as well as all item transmittal events.

Define any additional test milestones needed. Estimate the time required to do each testing task. Specify the schedule for each testing task and test milestone. For each testing resource (i.e., facilities, tools, and staff), specify its periods of use.

We can appreciate three milestones, each having a period of more or less 4 weeks.

1. Sprint 1: 16/09/2023 - 18/10/2023
2. Sprint 2: 21/10/2024 - 22/11/2024
3. Sprint 3: 25/11/2024 - 10/1/2025

15. Risks and contingencies

Identify the high-risk assumptions of the test plan. Specify contingency plans for each (e.g., delayed delivery of test items might require increased night shift scheduling to meet the delivery date).

The detection of a bug in the latest stage of the iterative process might require the team to increase the number of meetings during the day.

The realization of a mistake in a previous stage will require the team to mitigate the consequences by increasing the night shifts.

Forgetting to submit the documents in time can make the team fail, forcing it to go for other extraordinary options.

16. Approvals

Test Manager: Carmen María Noblejas Carreto

Date:07/01/2025

Development Project Manager: Lucian Andrei Negoita

Date: 07/01/2025

Quality Assurance Manager: Javier García Tercero

Date: 07/01/2025