

**CAPSTONE PROJECT REPORT**

**Report 5 – Software Test Documentation**

– Hanoi, April 2025 –

**Table of Contents**

[I. Record of Changes 3](#_heading=h.zfg9s1amxw75)

[II. Testing Documentation 4](#_heading=h.bslx8165wvge)

[1. Scope of Testing 4](#_heading=h.i8brzypo6dc9)

[1.1 Features to be Tested 4](#_heading=h.azcdsoduu8c)

[1.2 Features Not to be Tested 5](#_heading=h.6ss8fili7l3u)

[2. Test Strategy 5](#_heading=h.e3cr1qllth4p)

[2.1 Testing Types 5](#_heading=h.33g12io6npe4)

[2.2 Test Levels 7](#_heading=h.lo6oit6y34ji)

[2.3 Supporting Tools 8](#_heading=h.bcwt6h3yhvww)

[3. Test Plan 8](#_heading=h.60u1cq1onvd2)

[3.1 Test Environment 8](#_heading=h.k7rucj7tc9kn)

[3.2 Test Milestones 8](#_heading=h.1rpaz5sman4c)

[4. Test Cases 10](#_heading=h.mel864yc288v)

[4.1 Unit Test 10](#_heading=h.8o4098v172rx)

[4.2 Integration Test 12](#_heading=h.5mwe6pfuahb8)

[5. Test Reports 14](#_heading=h.ub5uu64mw74y)

[5.1 Unit test 15](#_heading=h.lf91ct1yjifd)

[5.2 Integration test 16](#_heading=h.lj8ytrn9dj1v)

[5.3 System test 19](#_heading=h.2l89ymhyg245)

# I. Record of Changes

| **Date** | **A\* M, D** | **In charge** | **Change Description** |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

\*A - Added M - Modified D - Deleted

# II. Testing Documentation

## 1. Scope of Testing

This document outlines the testing scope for the Pickleball Court Management System to ensure that the system meets both functional and non-functional requirements. The testing process aims to identify potential errors, verify functionality, and validate the behavior of the system before deployment.

## 1.1 Features to be Tested

The following modules and features will be included in the scope of testing:

* **Court Management**:
  + View available courts on interactive maps
  + Create, update, activate, and deactivate courts with time slots
  + Manage sub-courts for flexible configurations
  + View detailed court information including images and maintenance schedules
  + Schedule and track court maintenance
* **User Management**:
  + Register new player accounts and manage user profiles
  + Create, assign, and manage user roles (admin, staff, player)
  + View and manage registered user lists
* **Booking Management**:
  + View current bookings, details, and booking history
  + Handle payments and deposits, including partial payments
  + Confirm, cancel, or reschedule bookings with notifications
  + Suggest optimal courts based on availability and preferences
  + Temporarily hold courts during booking process
* **Notification Management**:
  + Send and receive notifications related to bookings, payments, and system updates
* **Pricing Management**:
  + View, create, and manage pricing structures based on various factors
  + Generate and export revenue and transaction reports
* **Payment & Financial Management**:
  + Track payment history for all transactions
  + Manage financial records including deposits, refunds, and payments
* **Service Management**:
  + Create, update, and manage additional services (coaching, equipment rentals)
  + View service details and manage availability status
* **Staff Management**:
  + Access staff dashboard for daily operations overview
  + Create bookings on behalf of customers
  + Manage order lists and customer check-ins
* **Mobile Application**:
  + User registration and login
  + Court availability viewing
  + Booking, cancellation, and schedule modification
  + In-app payments
  + Notification reception
  + Booking history viewing
  + Profile management

## 1.2 Features Not to be Tested

The following are **excluded** from this testing phase:

* Integration with external accounting systems
* Compatibility with legacy browsers (IE11 and below)
* Integration with security camera systems
* Player scoring and ranking systems (not in requirements)
* Social media integration (unless specified in requirements)
* Performance and stress testing of backend APIs beyond standard load conditions
* Third-party service integrations unless critical to core functionality
* UI/UX design consistency unless it causes functional issues
* Hardware or system-level compatibility testing beyond standard devices

## 2. Test Strategy

This test strategy ensures that the system meets all functional and non-functional requirements while delivering a high-quality user experience across both web and mobile platforms. Below are the testing types, their objectives, techniques, completion criteria, and special considerations.

### 2.1 Testing Types

#### 2.1.1 Functional Testing

| **Test Objective:** | * Verify that all system functionalities (e.g., authentication, court management, booking, user management) work as per the specified requirements. |
| --- | --- |
| **Technique:** | * Black-box testing, specification-based testing (using test cases derived from requirements and test reports). |
| **Completion Criteria:** | * All functional test cases pass (100% pass rate), with no critical or high-severity defects remaining. |
| **Special Considerations:** | * Functional testing will NOT be started in case of developers have not executed unit test before passing application to testers |

#### 

#### 2.1.2 Usability Testing

| **Test Objective:** | * Ensure the system’s interface is intuitive and user-friendly for both web and mobile users (e.g., ease of booking courts, managing profiles). |
| --- | --- |
| **Technique:** | * Real-user testing with observation, feedback collection via surveys or interviews. |
| **Completion Criteria:** | * Users can complete core tasks (e.g., booking, profile updates) without assistance; at least 85% positive feedback from test participants. |
| **Special Considerations:** | * N/A |

#### 

#### 2.1.3 Performance Testing

| **Test Objective:** | * Confirm the system performs efficiently under expected loads (e.g., multiple simultaneous bookings) on web and mobile. |
| --- | --- |
| **Technique:** | * Load testing (simulating normal usage), stress testing (pushing beyond normal capacity). |
| **Completion Criteria:** | * Response time <2 seconds under normal load (e.g., 100 concurrent users), no crashes under high load (e.g., 500 users). |
| **Special Considerations:** | * N/A |

#### 2.1.4 Compatibility Testing

| **Test Objective:** | * Ensure the system works across supported browsers (Chrome, Firefox, Safari, Edge) and devices (Android ≥ 8.0, iOS). |
| --- | --- |
| **Technique:** | * Cross-browser testing, multi-device testing (real devices and emulators). |
| **Completion Criteria:** | * System functions correctly on all targeted platforms with no major UI or functional issues. |
| **Special Considerations:** | * N/A |

#### 2.1.5 Integration Testing

| **Test Objective:** | * Validate seamless interaction between modules (e.g., booking with payment, court management with sub-courts) and between web/mobile and backend. |
| --- | --- |
| **Technique:** | * Top-down and bottom-up integration testing, API testing for data flow. |
| **Completion Criteria:** | * All integrated modules work together without errors; 100% pass rate for integration test cases. |
| **Special Considerations:** | * N/A |

#### 2.1.6 System Testing

| **Test Objective:** | * Verify the entire system’s end-to-end functionality (e.g., complete booking flow from login to payment confirmation). |
| --- | --- |
| **Technique:** | * End-to-end testing based on user scenarios, manual and automated testing. |
| **Completion Criteria:** | * 95% of system test cases pass, no critical defects, proper handling of edge cases (e.g., booking conflicts). |
| **Special Considerations:** | * N/A |

#### 2.1.7 Regression Testing

| **Test Objective:** | * Ensure new changes or updates do not break existing functionalities after each release or fix. |
| --- | --- |
| **Technique:** | * Re-execution of existing test cases (manual and automated) post-changes. |
| **Completion Criteria:** | * 100% pass rate for regression test suite; no regression defects introduced. |
| **Special Considerations:** | * N/A |

#### 

#### 2.1.8 User Acceptance Testing (UAT)

| **Test Objective:** | * Confirm the system meets end-user expectations and business requirements before deployment. |
| --- | --- |
| **Technique:** | * Testing by real users or stakeholder representatives using predefined scenarios. |
| **Completion Criteria:** | * Stakeholders approve the system; no critical usability or functional issues reported. |
| **Special Considerations:** | * N/A |

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 2.2 Test Levels

| **Type of Tests** | **Test Level** | | | |
| --- | --- | --- | --- | --- |
| **Unit** | **Integration** | **System** | **Acceptance** |
| Functional Testing | X | X | X | X |
| Usability Testing |  |  | X | X |
| Performance Testing |  |  | X |  |
| Compatibility Testing |  |  | X |  |
| Integration Testing |  | X |  |  |
| System Testing |  |  | X |  |
| Regression Testing | X | X | X |  |
| User Acceptance Testing |  |  |  | X |

### 2.3 Supporting Tools

| **Purpose** | **Tool** | **Vendor/In-house** | **Version** |
| --- | --- | --- | --- |
| Automated Testing (Web) | Selenium | Open Source | 4.0.0 |
| Automated Testing (Mobile) | Appium | Open Source | 1.22.3 |
| API Testing | Postman | Postman | 10.18.8 |
| Performance Testing | JMeter | Apache | 5.5 |
| Compatibility Testing | BrowserStack | BrowserStack | N/A |

## 3. Test Plan

### 3.1 Test Environment

### 

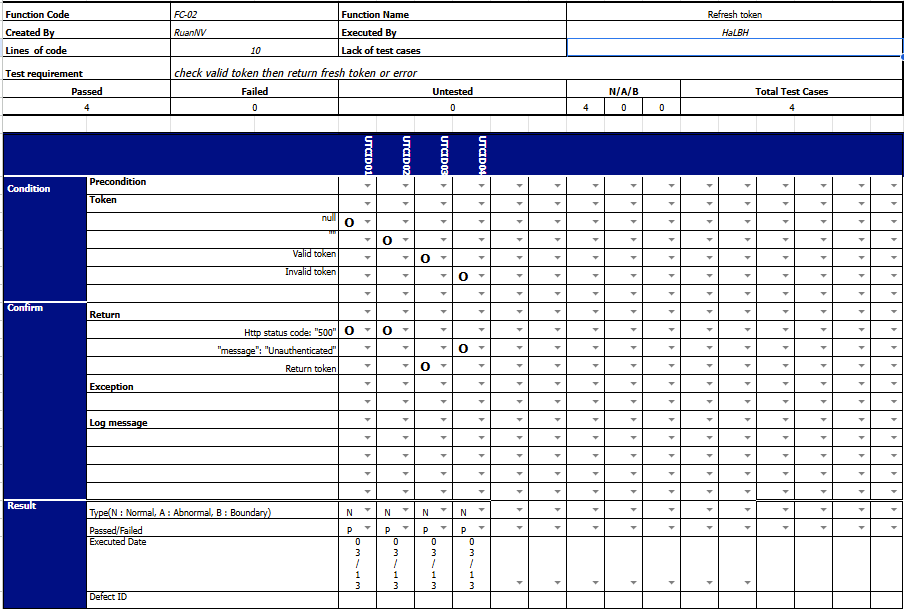
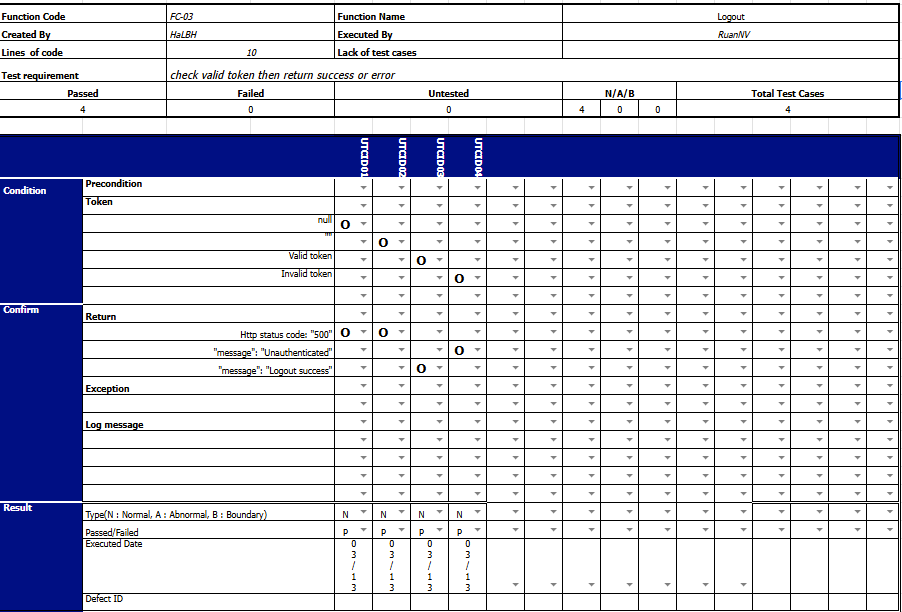
| **Purpose** | **Tool** | **Provider** | **Version** |
| --- | --- | --- | --- |
| Function Test | Visual Studio | [Microsoft](https://www.google.com/search?sca_esv=43aa34fbb2e5f50b&rlz=1C1CHBF_enVN1023VN1023&sxsrf=ADLYWILKCo6wIt7M2_wxxBmIECWAG5rO8Q:1723055994036&q=Microsoft&stick=H4sIAAAAAAAAAONgVuLSz9U3KCwxNLcoXMTK6ZuZXJRfnJ9WAgAP2uFPGgAAAA&sa=X&ved=2ahUKEwjbnc_jw-OHAxXPsVYBHaGWN6kQmxMoAHoECC0QAg) |  |
| API Test | Postman | Postman | 2024.1.4 |
| Sensor Setup | ArduinoIDE | Arduino | 2.3.5 |
| User Interface Test | Google  Chrome,  Microsoft Edge | Google  Microsoft |  |
| GUI Test | Google Chrome | Google  Microsoft |  |

### 3.2 Test Milestones

| **Milestone Task** | | **Start Date** | **End Date** |
| --- | --- | --- | --- |
| Create test plan | | 10/03/2025 | 25/04/2025 |
| Iteration 1 | Review and update test plan | 11/03/2025 | 13/03/2025 |
| Create test cases | 15/03/2025 | 18/03/2025 |
| Execute unit test and integration test | 20/03/2025 | 25/03/2025 |
| Iteration 2 | Review and update test plan | 26/03/2025 | 30/03/2025 |
| Create test cases | 01/04/2025 | 05/04/2025 |
| Execute unit test and integration test | 06/03/2025 | 10/04/2025 |
| Iteration 3 | Review and update test plan | 11/04/2025 | 13/04/2025 |
| Create test cases | 14/04/2025 | 16/04/2025 |
| Execute unit test and integration test | 17/04/2025 | 20/04/2025 |
| Create and execute System test | | 21/04/2025 | 25/04/2025 |

## 4. Test Cases

### 4.1 Unit Test

### 4.2 Integration Test

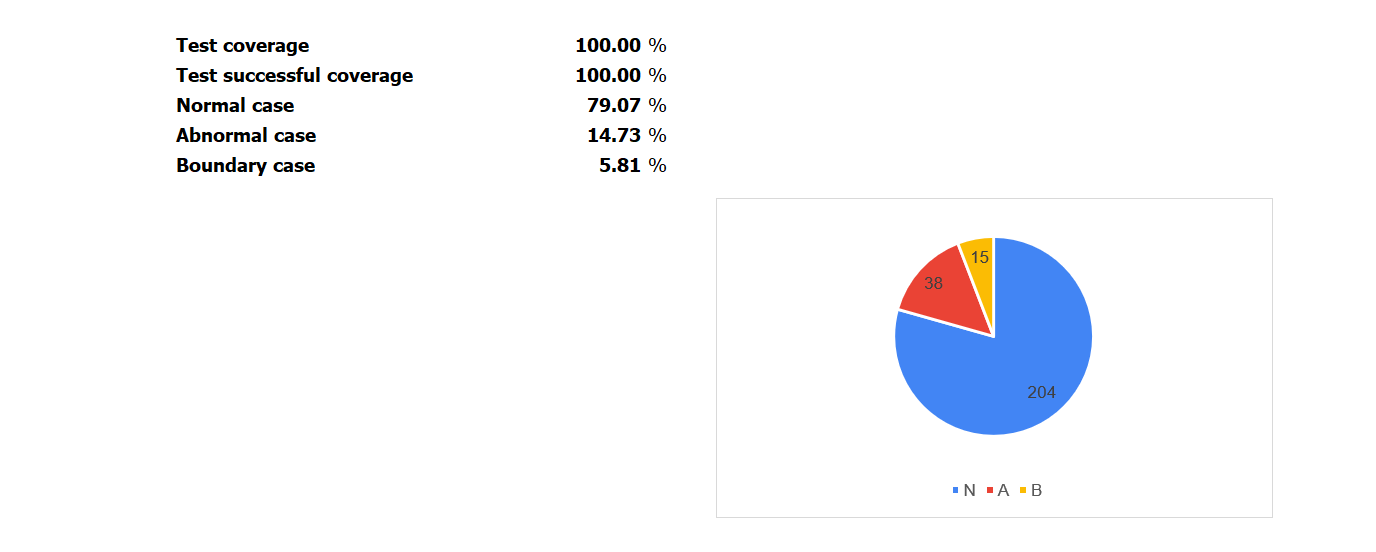
#### 4.2.1 Integration Test Web

#### 4.2.2 Integration Test Mobile

#### 

## 5. Test Reports

## 5.1 Unit test



## 5.2 Integration test

#### 5.2.1 Integration test Web

#### 5.2.2 Integration test Mobile

## 5.3 System test