

# PlaywrightWithAI Test Suite Documentation

## Overview

This directory contains comprehensive test documentation for the PlaywrightWithAI project, which implements both traditional C# Playwright tests and modern BDD/SpecFlow test scenarios.

## Test Suite Architecture

### Traditional C# Tests

Located in the main PlaywrightTests directory, these tests provide direct, developer-focused validation:

- **Test1.cs:** User Registration and Duplicate Prevention
- **Test2.cs:** Record Management (CRUD Operations)
- **Test3.cs:** Multiple Record Operations

### BDD/SpecFlow Tests

Modern business-readable tests using Gherkin syntax:

- **SearchFunctionality.feature:** 6 comprehensive search scenarios with wildcard support

## Test Documentation Files

### SearchFunctionality\_Steps.md

#### Comprehensive BDD Test Documentation

- **Scenarios:** 6 complete test cases covering all search patterns
- **Wildcard Testing:** Single/multi-character patterns, multi-field combinations
- **Performance:** Optimized browser lifecycle with single-instance reuse
- **Validation:** Record counts, field verification, specific ID validation
- **Execution Time:** ~40 seconds for complete suite

#### Key Test Scenarios:

1. Exact search: Dorothy Burch lookup
2. Gender filtering: 45 Female records
3. Single wildcard: A\* patterns (5 results)
4. Multi-char wildcard: AI\* patterns (3 results)
5. Multi-field: D\* + I\* combination (2 results)
6. Year-based: 1941 with specific IDs (4 results)

### Test1\_Steps.md

#### User Registration Flow

- User authentication and form validation

- Record creation with required fields
- Duplicate entry prevention testing
- Navigation and UI element validation
- User session management (login/logout)

## Test2\_Steps.md

### Record Management Operations

- Data grid navigation and sorting
- CRUD operations (Create, Read, Update, Delete)
- Record editing and data persistence
- Delete confirmation workflows
- Grid sorting and data validation

## Test3\_Steps.md

### Multiple Record Operations

- Bulk record creation (4 test records)
- Multiple delete functionality
- Checkbox selection interface
- Batch operations confirmation
- Data grid management

## Technical Framework Details

### BDD Implementation

- **Framework:** SpecFlow 3.9.74 with step tracing
- **Browser:** Playwright with optimized lifecycle management
- **Assertions:** MSTest integration with detailed logging
- **Performance:** Single browser instance across scenarios

### Traditional Tests

- **Framework:** Playwright + MSTest
- **Browser:** Direct browser automation
- **Validation:** Traditional assert-based testing
- **Coverage:** Complete application workflow testing

## Execution Methods

### BDD Tests Only

```
cd PlaywrightTests
dotnet test --filter "FullyQualifiedNames~SearchFunctionality" --logger:"console;verbosity=detailed"
```

### Traditional Tests Only

```
04/09/2025, 15:24
cd PlaywrightTests
dotnet test --filter "FullyQualifiedName!~StepDefinitions" --logger:"console;verbosity=normal"
```













All Tests

```
cd PlaywrightTests
dotnet test --logger:"console;verbosity=normal"
```

Script Execution

```
.\Scripts\run-bdd-tests.ps1           # BDD only
.\Scripts\run-bdd-tests.ps1 -AllTests  # All tests
.\Scripts\run-bdd-tests.ps1 -Traditional # Traditional only
```

Test Coverage Matrix

Feature	Traditional Tests	BDD Tests	Status
User Authentication	 Test1, Test2, Test3	 Background steps	Complete
Record Creation	 Test1	 Not covered	Partial
Record Management	 Test2	 Not covered	Partial
Search Functionality	 Not covered	 6 scenarios	Complete
Multiple Operations	 Test3	 Not covered	Partial
Navigation/UI	 All tests	 Background steps	Complete

Best Practices Demonstrated

BDD Tests






- **Business Readability:** Gherkin scenarios readable by stakeholders
- **Step Reusability:** Common steps shared across scenarios
- **Browser Optimization:** Performance-conscious single-instance pattern
- **Detailed Logging:** Step timing and execution trace

Traditional Test Features

- **Direct Validation:** Fast, developer-focused test execution
- **Complete Workflows:** End-to-end user journey testing
- **UI Interaction:** Comprehensive element validation

Recent Updates

September 3, 2025

-  Complete 6-scenario BDD search functionality implementation
-  Browser lifecycle optimization for performance
-  Comprehensive wildcard pattern testing
-  Specific record ID validation
-  Documentation updates with technical details

# Future Considerations

## Test Expansion Opportunities

- **BDD Coverage:** Add scenarios for record creation, editing, deletion
- **Integration Tests:** Cross-feature workflow testing
- **Performance Tests:** Load testing with larger datasets
- **Error Handling:** Negative test case scenarios

## Framework Enhancements

- **Parallel Execution:** Multi-browser test execution
- **Reporting:** Enhanced HTML/Visual reporting
- **CI/CD Integration:** Automated pipeline testing
- **Cross-browser:** Testing across multiple browser engines

---

## Organized File Structure

### Test Steps Directory Layout

```
Test Steps/
├── Word/
│   ├── README.docx           # Microsoft Word versions (.docx)
│   ├── SearchFunctionality_Steps.docx # Test suite overview
│   ├── Test1_Steps.docx      # BDD scenarios documentation
│   ├── Test2_Steps.docx      # User registration flow
│   └── Test3_Steps.docx      # CRUD operations flow
├── PDF/
│   ├── README.pdf           # Multiple operations flow
│   ├── SearchFunctionality_Steps.pdf # Portable Document Format versions (.pdf)
│   ├── Test1_Steps.pdf      # Test suite overview
│   ├── Test2_Steps.pdf      # BDD scenarios documentation
│   └── Test3_Steps.pdf      # User registration flow
├── README.md                # CRUD operations flow
├── SearchFunctionality_Steps.md # Multiple operations flow
├── Test1_Steps.md            # Test suite overview (markdown source)
├── Test2_Steps.md            # BDD scenarios (markdown source)
└── Test3_Steps.md            # User registration (markdown source)
                                # CRUD operations (markdown source)
                                # Multiple operations (markdown source)
```

## File Format Benefits

- **Markdown (.md):** Source files for editing and version control
- **Word (.docx):** Business-friendly format for stakeholders and documentation reviews
- **PDF (.pdf):** Universal format for archival, sharing, and printing

## Conversion Process

The convert-teststeps.ps1 script automatically generates Word and PDF versions from markdown sources:

1. **Word Generation:** Pandoc converts markdown directly to .docx format
2. **PDF Generation:** Chrome headless converts HTML (from markdown) to PDF
3. **Organization:** Files are automatically placed in appropriate subfolders

*This test suite demonstrates both traditional and modern testing approaches, providing comprehensive application validation through multiple testing paradigms.*