# SearchFunctionality.feature Step-by-Step Test Flow

## Feature Overview

**Purpose**: Comprehensive testing of search functionality with wildcard patterns and field combinations **Type**: BDD/Gherkin Test Suite **Test Framework**: SpecFlow 3.9.74 with Playwright **Total Scenarios**: 6 comprehensive test cases **Expected Duration**: ~40 seconds (optimized with browser reuse)

## Background Steps (Common Setup)

All scenarios share these common setup steps:

1. **Authentication**: Launch browser and log in as “davred” with password “Reinhart2244”
2. **Navigation**: Assert main page loads successfully after login
3. **Search Access**: Click on the “Search” link in navigation menu
4. **Form Ready**: Assert Search page is loaded and ready for input

## Test Scenarios

### Scenario 1: Search for a record

**Purpose**: Test basic search functionality with specific family name

#### Test Steps for Scenario 1

1. **View Search Form**: Verify the Search form displays with Family Name input field
2. **Enter Search Criteria**: Type “Burch” into the Family Name field
3. **Execute Search**: Click the “Search” button
4. **Verify Results**: Assert exactly one row returned with:
   * **Forename**: “Dorothy”
   * **Family Name**: “Burch”
   * **Gender**: “Female”
   * **Year of Birth**: “1901”

### Scenario 2: Search for records using wildcards 1

**Purpose**: Test gender filtering functionality

#### Test Steps for Scenario 2

1. **View Search Form**: Verify search form is visible and ready
2. **Clear Form**: Click Clear button to reset all fields
3. **Verify Blank Form**: Assert search form is completely blank
4. **Select Gender**: Choose “Female” from Gender dropdown
5. **Execute Search**: Click Search button
6. **Verify Results**: Assert 45 records returned, all with Gender = “Female”

### Scenario 3: Search for records using wildcards 2

**Purpose**: Test single-character wildcard pattern (A\*)

#### Test Steps for Scenario 3

1. **View Search Form**: Verify search form is ready
2. **Enter Wildcard**: Type “A\*” into Family Name field
3. **Execute Search**: Click Search button
4. **Verify Results**: Assert 5 records returned, all family names starting with “A”

### Scenario 4: Search for records using wildcards 3

**Purpose**: Test multi-character wildcard pattern (Al\*)

#### Test Steps for Scenario 4

1. **View Search Form**: Verify search form is ready
2. **Enter Wildcard**: Type “Al\*” into Family Name field
3. **Execute Search**: Click Search button
4. **Verify Results**: Assert 3 records returned, all family names starting with “Al”

### Scenario 5: Search for records using wildcards 4

**Purpose**: Test multi-field wildcard combination

#### Test Steps for Scenario 5

1. **View Search Form**: Verify search form is ready
2. **Enter Family Wildcard**: Type “D\*” into Family Name field
3. **Enter Forename Wildcard**: Type “I\*” into Forename field
4. **Execute Search**: Click Search button
5. **Verify Results**: Assert 2 records returned matching both criteria

### Scenario 6: Search for records using wildcards 5

**Purpose**: Test year-based search with specific ID validation

#### Test Steps for Scenario 6

1. **View Search Form**: Verify search form is ready
2. **Enter Year**: Type “1941” into Year of Birth field
3. **Execute Search**: Click Search button
4. **Verify Results**: Assert 4 records returned with specific IDs: 82, 35, 30, and 47

## Technical Implementation

### Browser Optimization

* **Single Browser Instance**: Reused across all scenarios for performance
* **Fresh Page Context**: Each scenario gets clean page context for isolation
* **Lifecycle Management**: Handled by PlaywrightHooks.cs

### Step Definitions Coverage

Located in StepDefinitions/SearchFunctionalitySteps.cs:

* **Form Interaction**: Search form viewing, field input, clear functionality
* **Search Execution**: Button clicking, search operation
* **Result Validation**: Record counting, field verification, ID validation
* **Wildcard Support**: Pattern matching for various wildcard types
* **Multi-field Searches**: Combined criteria validation

### Assertions Framework

* **MSTest Integration**: Uses MSTest Assert methods
* **Detailed Logging**: Console output with step execution timing
* **Error Reporting**: Clear failure messages with expected vs actual values

## Expected Outcomes

### Performance Metrics

* **Total Execution Time**: ~40 seconds for all 6 scenarios
* **Browser Startup**: Single instance initialization
* **Scenario Isolation**: Fresh page context per test

### Search Capabilities Validated

1. ✅ **Exact Match Search**: Specific person lookup
2. ✅ **Gender Filtering**: Dropdown selection with count validation
3. ✅ **Single Wildcard**: A\* pattern matching (5 results)
4. ✅ **Multi-char Wildcard**: Al\* pattern matching (3 results)
5. ✅ **Multi-field Wildcards**: D\* + I\* combination (2 results)
6. ✅ **Year-based Search**: 1941 with specific ID validation (4 results)

## Dependencies

### System Requirements

* Web application running and accessible
* Login functionality operational
* Search functionality with wildcard support enabled

### Test Infrastructure

* SpecFlow 3.9.74 with step tracing enabled
* Playwright browser automation
* MSTest framework for assertions
* VS Code task integration for execution

## Execution Commands

### Run All BDD Scenarios

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

### Alternative Script Execution

.\Scripts\run-bdd-tests.ps1

## Test Results Summary

**Last Execution**: September 3, 2025

* **Total Tests**: 6 BDD scenarios
* **Status**: ✅ All PASSED
* **Execution Time**: 39.6 seconds
* **Browser Optimization**: Successful single-instance reuse

*This comprehensive BDD test suite validates the complete search functionality including basic lookup, gender filtering, single/multi-character wildcards, multi-field combinations, and year-based searching with specific result validation.*