# Flutter Application Test Report

Date: 2024-01-18

## Overview

This report documents the test results for the Training Calendar Flutter application. The tests cover both unit testing of the TrainingService and widget testing of the main application.

## Test Environment

- Framework: Flutter Test
- Testing Libraries:
  - flutter_test
  - mockito
  - build_runner

## Unit Tests Results

### TrainingService Tests

Location: `test/training_service_test.dart`

#### 1. Search Functionality Test 

**Description**: Tests the normal search functionality for trainings
**Status**: PASSED
**Verified**:

- Correct API endpoint called
- Proper data transformation
- Expected response structure

#### 2. Error Handling Test 

**Description**: Tests error handling for API failures
**Status**: PASSED
**Verified**:

- Exception thrown on 500 status code
- Proper error message propagation

#### 3. Empty Response Test 

**Description**: Tests handling of empty search results
**Status**: PASSED
**Verified**:

- Empty list returned
- No exceptions thrown

#### 4. Malformed Response Test 

**Description**: Tests handling of invalid JSON responses
**Status**: PASSED
**Verified**:

- Exception thrown on invalid JSON
- Proper error handling

#### 5. Data Formatting Test 

**Description**: Tests null value handling and data formatting
**Status**: PASSED
**Verified**:

- Default values applied for null fields
- Proper data transformation

## Widget Tests

### App Initialization Test 

Location: `test/widget_test.dart`

**Status**: PASSED
**Resolution**:

- Added missing providers in widget tests
- Fixed layout overflow issues in welcome screen

## Code Coverage

- Unit Tests: 5/5 passing (100%)
- Widget Tests: 1/1 passing (100%)

## Recommendations

1. Add more widget tests for other screens
2. Consider adding integration tests

## Next Steps

1. Implement UI component testing
2. Add performance testing

## Conclusion

All tests have passed successfully. The core functionality of the TrainingService and the widget initialization are working as expected. Further improvements can be made by adding more comprehensive tests for other components.