SocialSpot

Testing Documentation

Unit test inventory, test case descriptions, and execution instructions for the SocialSpot iOS application.

Author: Sawyer Kent

Framework: XCTest | Build Tool: xcodebuild | Platform: iOS Simulator

Components and Services to Test

This section identifies the key modules and service layers targeted by the SocialSpot test suite.

Models

- **User behavior:** age computation from birthday, distance(from:), and sampleUsers data integrity.
- **SocialMediaAccount and SocialMediaPlatform:** metadata validation including icon names, colors, and display names.

Image URL Resolution

- **ImageURLResolver.resolve(_:):** handles empty values, SF symbols, full HTTP(S) URLs (appends cache-busting timestamp), and storage paths (Supabase storage public URL branch).

Nearby Listing and Formatting

- **NearbyUsersList logic:** sortedUsers ordering by distance, nearbyCount threshold (500 ft / 152.4 m), and behavior when userLocation is nil.
- **NearbyUserCard.formatDistance(_:):** formatting output in feet when under 500 ft and miles otherwise.

Integration Points

- **SupabaseManager / Supabase storage:** used by ImageURLResolver when building public URLs. This requires dependency injection or mocking to test the storage-URL branch safely in unit tests.

Significant Test Case Descriptions

The following subsections describe the most important test scenarios, their inputs, expected outputs, and relevant edge cases.

User Age Calculation

Aspect	Detail
Input	Profile birthday string in yyyy-MM-dd format (e.g., 2000-01-01)
Expected Result	Computed age equals the number of full years between birthday and Date() using Calendar.current
Edge Cases	Invalid or missing birthday string defaults age to 0

Social Accounts Parsing

Aspect	Detail
Input	Social links with or without a leading @ symbol
Expected Result	Usernames are parsed consistently and stored on User.socialAccounts

Image URL Resolution

Scenario	Expected Behavior
Empty or whitespace-only string	Returns nil
SF symbol-like name (no slashes, not an image extension)	Returns nil — UI should use system images instead
Full HTTP(S) URL	Returns a URL string with a t= cache-buster query parameter appended
Storage path (contains / or image extension)	Returns public Supabase URL (requires mocking Supabase client)

Nearby Users Sorting and Counting

Scenario	Expected Behavior
Reference location provided	sortedUsers orders results by distance ascending
nearbyCount threshold	Returns the count of users within 152.4 meters (500 ft)
userLocation is nil	Original user array order is preserved

Test Scripts and Commands

Tests can be executed through the Xcode GUI or from the command line using xcodebuild. This section provides step-by-step instructions for both approaches.

Running Tests from Xcode

1. Open SocialSpot/SocialSpot.xcodeproj in Xcode.
2. Choose the SocialSpot scheme.
3. Run Product → Test (Cmd+U) to execute all tests.

Running Tests from the Terminal

Full Test Suite

```
cd /path/to/SocialSpot
xcodebuild test \
  -project SocialSpot/SocialSpot.xcodeproj \
  -scheme SocialSpot \
  -destination 'platform=iOS Simulator,name=iPhone 16'
```

Single Test Class

```
xcodebuild test \
  -project SocialSpot/SocialSpot.xcodeproj \
  -scheme SocialSpot \
  -destination 'platform=iOS Simulator,name=iPhone 16' \
  -only-testing:SocialSpotTests/NearbyUsersListTests
```

Single Test Method

```
xcodebuild test \
  -project SocialSpot/SocialSpot.xcodeproj \
  -scheme SocialSpot \
  -destination 'platform=iOS Simulator,name=iPhone 16' \
  -only-testing:SocialSpotTests/ImageURLResolverTests/testEmptyStringReturnsNil
```

Tips and Notes

- List available simulators with xcrun simctl list devices and pick a device name exactly as shown.
- If the simulator is not running, open it with open -a Simulator or boot it with xcrun simctl boot 'iPhone 16'.
- swift test is for Swift Package Manager projects. This project uses Xcode, so use xcodebuild or the Xcode GUI.

Test Files and Coverage

The table below maps each test file to the functionality it validates.

Test File	Coverage
ModelsTests.swift	User model behavior: age calculation, social account parsing, distance computation between coordinates
ImageURLResolverTests.swift	Image URL resolution branches: empty string, SF symbol detection, HTTP URL cache-buster appending
NearbyUsersListTests.swift	sortedUsers ordering, nearbyCount threshold filtering, nil-location fallback behavior
SocialMediaPlatformTests.swift	SF Symbol icon name validation, color hex string accuracy, displayName variants (e.g., Twitter → X)
SocialSpotUITestsLaunchTests.swift	Basic app launch UI test with launch screenshot capture