

## Test Procedure Document

### 1. Introduction

This test procedure ensures that the **Momento** app meets all functional and technical requirements. It combines manual acceptance testing steps and automated unit/integration test planning per the CMSI 4072 Unit Test Specification guidelines. The document outlines how we validate that the software behaves correctly and adheres to its design specifications.

### 2. Requirements & Verification Cross Reference

This section maps out major user requirements and ensures that corresponding test steps cover them:

Requirement	Verification Method
Users can capture and save daily photos	Step 2 + Unit test for saveEntry()
Users can check in with their mood	Step 3 + Unit test for mood association
Users can view past journal entries	Step 4 + Integration test for archive access
Users receive a daily quote	Step 5 + Unit test for fetchDailyQuote()
Users can view a Monthly Recap	Step 6 + Integration test for recap builder
Data is stored and synced via Firebase	Step 7 + Unit test for FirebaseService

### 3. Test Environment

- **Platform:** iOS 17+ on iPhone Simulator and physical devices
- **IDE:** Xcode 15+ with Swift 5.9
- **Frameworks:** SwiftUI, Firebase SDK, XCTest
- **Tools:** Firebase Emulator Suite (for isolated DB testing), optional mocking framework (e.g., Cuckoo)

### 4. Manual Functional Test Steps

#### Step 1: Start the Software

- Launch the Momento app
- **Expected Result:** App opens to the home screen without errors

#### Step 2: Capture and Save a Daily Photo

- Tap “Capture Photo” and take a picture
- Apply vintage filter (optional) and save the image
- **Expected Result:** Photo is saved and viewable in that day’s entry

#### Step 3: Check In with a Mood

- Tap “Mood Check-in,” select an icon, and save
- **Expected Result:** Mood is saved and linked to the day’s entry

#### Step 4: View a Past Journal Entry

- Navigate to archive and select a previous date
- **Expected Result:** Entry displays correct photo and mood

#### Step 5: View Daily Quote

- Open today’s entry and read the quote
- **Expected Result:** A motivational quote is displayed and changes daily

#### Step 6: Access Monthly Recap

- Tap “Monthly Recap” to view mood stats and images
- **Expected Result:** Summarized view of journal entries and moods for the month

#### Step 7: Test Data Persistence

- Close and reopen the app
- **Expected Result:** All saved data loads properly from Firebase

### 5. Unit Test Procedures (Automated using XCTest)

#### QuoteManager.swift

- testFetchDailyQuote\_returnsQuoteObject()
- testQuoteIncludesTextAndDate()

#### JournalEntryViewModel.swift

- testSaveEntry\_savesToFirebase()
- testDeleteEntry\_removesFromFirebase()
- testEditEntry\_updatesCorrectly()

#### MonthlyRecapBuilder.swift

- testBuildRecap\_correctEntryCount()
- testBuildRecap\_correctMoodStats()

#### FirestoreService.swift

- testFirestoreWrite\_persistsEntry()
- testFirestoreRead\_retrievesCorrectEntry()

#### AuthenticationViewModel.swift

- testLogin\_successfulWithValidCredentials()
- testLogin\_failsWithInvalidCredentials()

## 6. Integration Test Scenarios

### Scenario 1: Full Entry Creation

- Simulate photo capture + mood check-in + optional note
- Save and retrieve the entry
- **Expected Result:** Data is stored and accurately retrieved from Firestore

### Scenario 2: Monthly Recap Flow

- Insert multiple mock entries into Firestore
- Generate and display recap
- **Expected Result:** Summary and charts display correctly

### Scenario 3: Daily Quote Display

- On launch, fetch and render quote
- **Expected Result:** UI displays correct quote from QuoteManager

## 7. Test Execution Instructions

1. For **manual tests**, follow Section 4 steps in order on a test device.
2. For **unit/integration tests**, open the Momento project in Xcode.
3. Select the **MomentoTests** target.

4. Run all tests via Cmd + U.
5. Use Firebase Emulator when testing database interactions.

## 8. Success Criteria

- All manual and automated tests pass.
- All app features behave as expected under normal and edge conditions.
- Firebase reads/writes persist data across sessions.
- Daily quotes and monthly recaps update dynamically.

## 9. Known Limitations

- Gesture-based UI tests (e.g., long press) not yet automated.
- Firebase latency may vary in live vs. emulator environments.

This document follows CMSI 4072 unit testing guidelines and integrates robust feature validation for the continued development of Momento.