MU Connect – Test Plan Document

Prepared by: MU Connect team

Version History

Version	Date	Description	Author
1.0	April 10, 2025	Initial Draft	MU Connect Team
2.0	April 15, 2025	Refined the Scope & Schedule	MU Connect Team
3.0	April 18, 2025	Finalized Schedule	MU Connect Team

1. Introduction

This test plan outlines how we will verify and validate the MU Connect mobile application. MU Connect is a student-alumni networking platform tailored for Mahindra University. Our goal is to ensure the app runs smoothly, feels intuitive to users, and meets all the expected functional, security, and performance standards.

2. Objectives and Tasks

Our main objectives are:

- ➔ Confirm that each feature performs as it should
- ➔ Detect and fix bugs early
- ➔ Ensure a smooth experience across multiple devices based on the platforms we are using
- ➔ Test security, responsiveness, and usability

Key tasks include writing test cases, executing tests, fixing bugs, and documenting results.

3. Test Scope

We will test the full set of core features, including:

- ➔ User login and registration
- ➔ Profile management
- ➔ Messaging
- ➔ Job and event posting
- ➔ Forum participation
- ➔ Admin moderation tools

We will not be testing future integrations like LinkedIn or multilingual support at this stage.

4. Test Strategy

4.1 Entry and Exit Criteria

To make sure our testing process starts and ends smoothly, we've defined clear checkpoints:

Entry Criteria (When we're ready to begin testing):

- ➔ All major features of the app have been built and are available in the testing environment.
- ➔ The test environment is set up, stable, and mirrors the real usage conditions.
- ➔ Our test cases are fully written, reviewed, and good to go.
- ➔ We've gathered all the test data and tools we'll need.

Exit Criteria (When we're confident testing is complete):

- ➔ Every planned test case has been executed.
- ➔ All high-priority bugs have been either fixed or acknowledged with approved solutions.
- ➔ Our reports—test summaries, screenshots, and bug logs—are complete and accurate.
- ➔ The QA team and project leads have reviewed and signed off on everything, giving the green light.

4.2 Bug Severity and Priority Handling

To stay focused and fix the right things at the right time, we'll handle bugs based on how serious and urgent they are:

High Severity, High Priority

- ➔ Example: App crashes during login
- ➔ Action: Fix immediately before moving on

Medium Severity, High Priority

- ➔ Example: A button or feature doesn't respond
- ➔ Action: Fix soon—it affects how people use the app

Low Severity, Medium Priority

- ➔ Example: Text alignment looks off
- ➔ Action: Fix if we have time—it's not urgent, but nice to clean up

Low Severity, Low Priority

- ➔ Example: Slight color mismatch or outdated icon
- ➔ Action: Won't block anything—can fix in a later update

This way, we ensure major problems get fixed fast, while small issues are noted and handled thoughtfully.

4.3 Types of Testing

We'll apply the following types of testing:

- ➔ Unit Testing: Focused on individual features like login or post creation
- ➔ Integration Testing: Making sure components (e.g., chat + profile) work well together
- ➔ System Testing: Full app testing across devices
- ➔ User Testing: Real users (students/alumni) try the app and share feedback
- ➔ Performance Testing: Checking load speed and responsiveness
- ➔ Security Testing: Verifying login protection, role access, and encryption

5. Test Deliverables

We'll provide:

- ➔ This test plan
- ➔ A full list of test cases
- ➔ A defect log with screenshots
- ➔ Summary report of test outcomes
- ➔ Final sign-off sheet

6. Test Environment

- ➔ Hardware: iOS devices
- ➔ Software: Xcode (IDE), Swift (language), SwiftUI (UI framework), PhotosUI (for image picking), UserDefaults (for local storage), Core Data (for structured local storage)

7. Roles and Responsibilities

Each team member has a clear role:

- ➔ Niyathi: Test planning and oversight
- ➔ Divya & Sadhika: Write and run test cases
- ➔ Varshini & Sathvick: Handle bug fixes and unit testing

Dr. Vijay Rao Duddu & Nartkannai K: Approves and reviews

8. Schedule

Here's our approximate testing timeline:

- ➔ April 18–19: Finalize all written test cases for each module (Login, Profile, Chat, Events, Jobs, Forums, Admin tools)
- ➔ April 20–21: Conduct unit testing for individual features (e.g., registration, post creation, messaging)
- ➔ April 22–23: Perform integration testing to ensure smooth coordination between modules (e.g., Login + Profile, Chat + Notifications)
- ➔ April 24–25: Execute system testing across different iOS devices
- ➔ April 26–27: Run user testing with actual MU students and alumni to get usability and experience feedback
- ➔ April 28–29: Fix reported bugs and perform regression testing to ensure nothing broke after updates
- ➔ April 30: Perform final cleanup – double-check test logs, complete any pending cases, and validate all reports
- ➔ May 1–2: Prepare and finalize test summary report, screenshots, bug logs, and approval sheets
- ➔ May 3: Ready to submit final test plan, test summary report, and app for evaluation

9. Risk Management

- ➔ Device differences might cause UI issues – we'll use responsive design and test widely
- ➔ Tight timeline – buffer days added for fixes
- ➔ Internet/API issues – mock servers will be used

10. Approval and Sign-off

Once all tests are passed and reviewed, this document will be signed off by the instructor and team leads to confirm that MU Connect is ready for release.

- Dr. Vijay Rao Duddu & Nartkannai K: Approves and reviews