

Epic #1 – Tester 1 Master Test Cases (Login & Registration)

Project: InCollege – Week 3 Deliverable (Log In, Part 1)

Role: Tester 1

Author: <Your Name>

Date: Sep 8, 2025

0) Scope & Assumptions

- **Scope:** Registration (up to 5 accounts), password validation, login, post-login menu, “under construction” features, skill list & back navigation, input-from-file, duplicate console+file output, and basic persistence of accounts across runs.
- **Assumptions (align with programmers):** 1) Main menu offers two choices in this order: 1) Log In 2) Create New Account then Enter your choice: 2) Post-login menu offers: 1) Search for a job 2) Find someone you know 3) Learn a new skill then Enter your choice: 3) **Learn a New Skill** submenu: 1..5 skills + 6) Go Back . 4) **Input format:** The program reads **each response** (menu choice, username, password) from the input file **one line at a time**. No extra prompts or blank lines in the input file. 5) **Output duplication:** All text printed to console is **identical** in the output file. 6) **Persistence:** Accounts are saved in a sequential file and reloaded on next run. Filenames are implementation-specific; tests refer to a “clean start” vs. “existing accounts” state. 7) **Password rules:** Length 8–12 characters inclusive, ≥1 uppercase, ≥1 digit, ≥1 special character (any non-alphanumeric, e.g., !@#\$%^&*).

If any assumption differs from implementation, update the test inputs accordingly and log a **Bug or Clarification** ticket in Jira.

1) Test Data (Usernames & Passwords)

Use these values across cases: - **U1:** alice / **P1 (valid 8):** Abcdef1! - **U2:** bob / **P2 (valid 9):** StrongP9! - **U3:** carol / **P3 (valid 10):** C4rol!Pass - **U4:** dave / **P4 (valid 11):** D@vePass123 - **U5:** erin / **P5 (valid 12):** Erin2025!aa - **Bad variants:** - **P_short (7):** Abcd1! (too short) - **P_long (13):** Abcdefg1234!@ (too long) - **P_noUpper:** abcdef1! - **P_noDigit:** Abcdefg! - **P_noSpecial:** Abcdefg1 - **WrongPw:** WrongP1!

2) Input Scripting Conventions

- Each **test case** shows the **exact sequence** of inputs (each on its own line) that the input file should contain **for that test**.
- Where a case needs multiple flows (e.g., create 5 accounts), the steps are shown as a **block** of menu/credential inputs.
- If a case depends on the outcome of a previous case, it lists **Preconditions**.

3) Test Cases

T-001 Create Account – Happy Path

Type: Positive

Preconditions: Start with no accounts in persistence (clean file).

Input (lines):

```
2
alice
Abcdef1!
```

Expected key output assertions: - Shows main menu; on choice 2, prompts Please enter your username: then Please enter your password: - Accepts password meeting rules; confirms creation (explicit success message or returns to main menu per design). - Account count is now 1.

Covers: US-Create, US-Persistence, US-Tester-InputFile, US-OutputFile

T-002 Create 5th Account – Boundary Pass

Type: Positive/Edge

Preconditions: Clean start; none of U1..U5 exist.

Input (lines): create U1..U5 sequentially

```
2
alice
Abcdef1!
2
bob
StrongP9!
2
carol
C4rol!Pass
2
dave
D@vePass123
2
erin
Erin2025!aa
```

Expected: - All five accounts created successfully. - No error messages. **Covers:** US-Create, US-Persistence, boundary of 5

T-003 Create 6th Account – Limit Message

Type: Negative/Boundary

Preconditions: After T-002, 5 accounts exist.

Input:

```
2
frank
Abcdef1!
```

Expected: - Displays: All permitted accounts have been created, please come back later (exact text per spec). - No new account is saved. **Covers:** US-AccountLimit message

T-004 Password Too Short (7 chars)

Type: Negative (validation)

Preconditions: Clean start recommended.

Input:

```
2
alice
Abcd1!
```

Expected: - Registration rejected with clear reason (length < 8). Stays in registration flow or returns per design. - **No account** created. **Covers:** US-Create validation

T-005 Password Too Long (13 chars)

Type: Negative

Input:

```
2
alice
Abcdefg1234!@
```

Expected: - Registration rejected for length > 12; no account created.

T-006 Password Missing Uppercase

Type: Negative

Input:

```
2
alice
abcdef1!
```

Expected: - Registration rejected (no uppercase); no account created.

T-007 Password Missing Digit

Type: Negative

Input:

```
2
alice
Abcdefg!
```

Expected: - Registration rejected (no digit); no account created.

T-008 Password Missing Special Character

Type: Negative

Input:

```
2
alice
Abcdefg1
```

Expected: - Registration rejected (no special char); no account created.

T-009 Duplicate Username on Create

Type: Negative

Preconditions: `alice/Abcdef1!` already created.

Input:

```
2
alice
Abcdef1!
```

Expected: - Registration rejected (username must be unique); **message should be explicit.**

T-010 Login Success – Happy Path

Type: Positive

Preconditions: `alice/Abcdef1!` exists.

Input:

```
1
alice
Abcdef1!
```

Expected: - `You have successfully logged in.` then shows post-login menu. **Covers:** US-Login success

T-011 Login Fail – Wrong Password (Retry Allowed)

Type: Negative

Preconditions: `alice/Abcdef1!` exists.

Input:

```
1
alice
WrongP1!
alice
Abcdef1!
```

Expected: - First attempt: `Incorrect username/password, please try again.` - Second attempt (same session): success message. - Unlimited attempts are permitted. **Covers:** US-Login failure message & retry

T-012 Login Fail – Unknown Username

Type: Negative

Preconditions: No user `ghost`.

Input:

```
1
ghost
Abcdef1!
```

Expected: - `Incorrect username/password, please try again` and allows further attempts.

T-013 Post-Login: Job Search Under Construction

Type: Positive (navigation)

Preconditions: Logged in as .

Input:

```
1
alice
Abcdef1!
1
```

Expected: - After login, choose → shows and returns to post-login menu. **Covers:** US-PostLogin nav, UC messages

T-014 Post-Login: Find Someone Under Construction

Type: Positive (navigation)

Preconditions: Logged in as .

Input:

```
1
alice
Abcdef1!
2
```

Expected: - Shows then returns to post-login menu.

T-015 Post-Login: Learn Skill List & Select Skill

Type: Positive (navigation)

Preconditions: Logged in as .

Input:

```
1
alice
Abcdef1!
3
1
```

Expected: - Shows **Learn a New Skill** list with 5 skills. - Selecting any skill shows then returns to skill list.

T-016 Learn Skill: Go Back Returns to Post-Login Menu

Type: Positive (navigation)

Preconditions: Logged in as `alice`.

Input:

```
1
alice
Abcdef1!
3
6
```

Expected: - `Go Back` returns to post-login menu.

T-017 Persistence: Login After Program Restart

Type: Positive (persistence)

Preconditions: `alice/Abcdef1!` created in a **prior run**; program restarted with accounts file present.

Input:

```
1
alice
Abcdef1!
```

Expected: - Login succeeds without re-registration. - Confirms accounts are re-loaded on startup.

Covers: US-Persistence

T-018 Output File == Console (Golden Run)

Type: System/Comparative

Preconditions: Any successful flow (e.g., T-010).

Procedure: - Run program; capture console output (copy buffer or terminal capture). - Compare line-for-line with generated output file (tooling or manual diff). **Expected:** - **Every line** matches exactly (content and order). If not, log bug with mismatches. **Covers:** US-Tester-OutputFile

T-019 Create Multiple Valid Users then Login As Each

Type: Positive (batch)

Preconditions: Clean start.

Input:

```
2
alice
```

```
Abcdef1!  
2  
bob  
StrongP9!  
2  
carol  
C4rol!Pass  
2  
dave  
D@vePass123  
2  
erin  
Erin2025!aa  
1  
alice  
Abcdef1!  
1  
bob  
StrongP9!  
1  
carol  
C4rol!Pass  
1  
dave  
D@vePass123  
1  
erin  
Erin2025!aa
```

Expected: - All registrations succeed; each login succeeds.

T-020 Robustness: Extra Login Retries Until Success

Type: Negative→Positive

Preconditions: `alice/Abcdef1!` exists.

Input:

```
1  
alice  
WrongP1!  
alice  
WrongP1!  
alice  
Abcdef1!
```

Expected: - Two failures with the exact message each time; third attempt succeeds.

4) Mapping to User Stories

- **US-Create:** T-001, T-002, T-004...T-009, T-019
 - **US-Login:** T-010, T-011, T-012, T-020
 - **US-Persistence:** T-001 (save), T-017 (reload), T-019
 - **US-AccountLimit:** T-003
 - **US-PostLogin Nav & UC Messages:** T-013, T-014, T-015, T-016
 - **US-Tester Input/Output Files:** T-018 (+ all cases use file input)
-

5) Jira Artifacts (Templates)

Test Case Ticket Template - **Title:** [T-ID] <Short name> - **Description:** Preconditions, steps (input lines), expected results (assertions) - **Attachments:** Input snippet, expected output (if deterministic) - **Links:** Related US, Bugs

Bug Ticket Template - **Title:** [BUG] <Feature> - <Symptom> - **Environment:** Branch/commit, OS, compiler, input file used - **Steps to Reproduce:** Exact input lines - **Expected Result:** ... - **Actual Result:** ... (include console vs. file mismatch if applicable) - **Attachments:** Screenshots, diffs, output files

6) Deliverables You'll Produce

- `InCollege-Test.txt` - build from the **Input** blocks above (combine or run individually as needed)
- `InCollege-Output.txt` - captured actual output from the program
- `Jira.jpg` - updated board showing User Stories, test cases, and logged bugs

When programmers finalize prompt wording/menus, do a **quick pass** to synchronize the test input lines with the implemented prompts. Any mismatch becomes either a small test update or a bug (if it contradicts the spec's sample output).