

TP1 Test Cases (v2025-09-29)

1) Email – Valid basic

ID: E-001

Preconditions: N/A

Steps: evaluateEmail("alice@example.com")

Expected: ""

2) Email – Valid with subdomains

ID: E-002

Steps: evaluateEmail("bob.smith@dept.cs.school.edu")

Expected: ""

3) Email – Missing '@'

ID: E-003

Steps: evaluateEmail("alice.example.com")

Expected: "Invalid Email: Must be in the form example@email.com"

4) Email – Missing dot in domain

ID: E-004

Steps: evaluateEmail("alice@localhost")

Expected: "Invalid Email: ..."

5) Email – Space present

ID: E-005

Steps: evaluateEmail("al ice@example.com")

Expected: "Invalid Email: ..."

6) Email – Local starts/ends with special

ID: E-006

Steps: evaluateEmail(".alice@example.com"), evaluateEmail("alice.@example.com")

Expected: "Invalid Email: ..." (for both)

7) Email – Double dot in local/domain

ID: E-007

Steps: evaluateEmail("al..ice@example.com"), evaluateEmail("alice@ex..ample.com")

Expected: "Invalid Email: ..." (for both)

8) Email – Non-alnum in domain label

ID: E-008

Steps: evaluateEmail("alice@exa-mple.com")

Expected: "Invalid Email: ..."

9) Name – Empty

ID: N-001

Steps: evaluateName("")

Expected: "Input Empty"

10) Name – Too long (>255)

ID: N-002

Steps: evaluateName("a" * 256)

Expected: "Input Too Long"

11) Name – Invalid dashes

ID: N-003

Steps: evaluateName("-Anna"), evaluateName("Anna-"), evaluateName("Anna--Marie")

Expected: "Invalid Use of Dashes"

12) Name – Non-alpha chars

ID: N-004

Steps: evaluateName("Ann4"), evaluateName("Ann_")

Expected: "Contains Non-Alphabetical Characters"

13) Name – Valid with dash and formatting

ID: N-005

Steps: evaluateName("anna-marie"), formatName("anna-marie")

Expected: "", "Anna-Marie"

14) Password – Empty

ID: P-001

Steps: evaluatePassword("")

Expected: "The password is empty!"

15) Password – Invalid character

ID: P-002

Steps: evaluatePassword("Abcdef1!")

Expected: "An invalid character has been found!"

16) Password – Too short (7)

ID: P-003

Steps: evaluatePassword("Abc1def")

Expected: contains "Between 8-32 Chars"

17) Password – Too long (>32)

ID: P-004

Steps: evaluatePassword("A" + "b"*31 + "1")

Expected: contains "Between 8-32 Chars"

18) Password – Missing category flags

ID: P-005

Steps: evaluatePassword("lowercase1"), evaluatePassword("UPPERCASE1"),

evaluatePassword("NoDigitsHereA")

Expected: contain "Upper case", "Lower case", "Numeric digit" respectively

19) Password – Valid

ID: P-006

Steps: evaluatePassword("StrongPass1")

Expected: ""

20) Username – Empty

ID: U-001

Steps: checkForValidUserName("")

Expected: "The user input is empty!"

21) Username – Must start with letter

ID: U-002

Steps: checkForValidUserName("1abc")

Expected: contains "must start"

22) Username – Min length (message says ≥ 4)

ID: U-003

Steps: checkForValidUserName("Abc")

Expected: contains "at least 4"

23) Username – Length > 16

ID: U-004

Steps: checkForValidUserName("Abcdefghijklmnop")

Expected: contains "no more than 16"

24) Username – Valid letters/digits and underscore

ID: U-005

Steps: checkForValidUserName("User_123")

Expected: ""

25) User entity – Constructor & getters

ID: UE-001

Steps: new User(...); call getters

Expected: values match inputs; getNumRoles accurate

26) Database – Connect & empty check (optional if H2 present)

ID: DB-001

Preconditions: H2 driver on classpath

Steps: connectToDatabase(); getNumberOfUsers()

Expected: ≥ 0 without exception

27) Database – Register & login admin

ID: DB-002

Preconditions: H2 present

Steps: register(User admin); loginAdmin(admin)

Expected: true

28) Database – Invitation lifecycle

ID: DB-003

Preconditions: H2 present

Steps: generateInvitationCode(); getRoleGivenAnInvitationCode(); getEmailAddressUsingCode();

removeInvitationAfterUse()

Expected: correct role/email; removed returns empty