

# Typing Trainer

---



Lorenzo Zulli



Università Politecnica delle Marche  
Facoltà di Ingegneria Informatica e dell'Automazione  
Anno Accademico 2023 - 2024

# Indice

<b>1</b>	<b>Descrizione in linguaggio naturale</b>	<b>3</b>
1.1	Cos'è Typing Trainer . . . . .	3
1.2	A chi è rivolto Typing Trainer . . . . .	3
1.3	Struttura del software . . . . .	3
1.4	Glossario dei termini . . . . .	4
<b>2</b>	<b>Analisi dei requisiti</b>	<b>5</b>
2.1	Requisiti funzionali . . . . .	6
2.2	Requisiti non funzionali . . . . .	8
2.3	Diagrammi dei casi d'uso . . . . .	9
2.3.1	Gestione Autenticazione . . . . .	10
2.3.2	Gestione Utenti . . . . .	14
2.3.3	Gestione Test . . . . .	19
2.3.4	Gestione Backup . . . . .	28
2.4	Matrice di Mapping . . . . .	29
<b>3</b>	<b>Diagrammi di Analisi</b>	<b>30</b>
3.1	Diagrammi delle Classi di Analisi . . . . .	30
3.2	Diagrammi di Sequenza . . . . .	31
3.2.1	Log In . . . . .	31
3.2.2	Registrazione . . . . .	32
3.2.3	Aggiungi Test . . . . .	33
3.2.4	Elimina Utente . . . . .	34
3.2.5	Inizia Test . . . . .	35
3.3	Diagrammi di Attività . . . . .	36
3.3.1	Log In . . . . .	36
3.3.2	Registrazione . . . . .	37
3.3.3	Aggiungi Test . . . . .	38
3.3.4	Elimina Utente . . . . .	39
3.3.5	Inizia Test . . . . .	40
<b>4</b>	<b>Diagrammi di Progettazione</b>	<b>41</b>
4.1	Diagramma delle Classi di Progettazione . . . . .	42
4.2	Diagrammi delle Macchine a Stati . . . . .	44
4.2.1	Macchina a stati: Utilizzatore . . . . .	44
4.2.2	Macchina a stati: Test . . . . .	45
4.2.3	Macchina a stati: Statistiche . . . . .	46
4.3	Diagramma dei Componenti . . . . .	47
<b>5</b>	<b>Implementazione</b>	<b>48</b>
5.1	Diagramma di Deployment . . . . .	48
5.2	Mockup . . . . .	49
5.2.1	LogIn . . . . .	49
5.2.2	Registrazione . . . . .	50
5.2.3	Recupera Password . . . . .	51
5.2.4	Landing Page Utente . . . . .	52
5.2.5	Inizia Test . . . . .	53
5.2.6	Visualizza Statistiche . . . . .	54
5.2.7	Visualizza Profilo . . . . .	55
5.2.8	Modifica Profilo . . . . .	56
5.2.9	Landing Page Admin . . . . .	57
5.2.10	Gestione Test . . . . .	58
5.2.11	Aggiungi Test . . . . .	59
5.2.12	Modifica Test . . . . .	60
5.2.13	Gestione Utenti . . . . .	61

5.2.14	Modifica Utente . . . . .	62
5.3	Descrizione Unit Tests . . . . .	63
5.3.1	Test Accesso . . . . .	63
5.3.2	Test Administrator . . . . .	64
5.3.3	Test Utente . . . . .	65

# 1 Descrizione in linguaggio naturale

## 1.1 Cos'è Typing Trainer

Typing Trainer è un software che ha l'obiettivo di far imparare, tecniche di dattilografia al computer, e permette di testare le abilità imparate tramite dei test, inoltre tramite la visualizzazione di statistiche nel profilo permette la visualizzazione dei miglioramenti nel tempo.

## 1.2 A chi è rivolto Typing Trainer

Typing Trainer è rivolto ai neofiti che vogliono imparare la dattilografia al computer da zero, ma anche a utenti che sono già abili permettendo test di battitura a tempo.

## 1.3 Struttura del software

All'apertura il software propone una schermata di Log In dove una persona può entrare nel sistema tramite l'inserimento delle proprie credenziali di accesso (Username e Password). Se non si hanno credenziali, queste possono essere generate premendo nella schermata di Log In il tasto "Registrazione": si verrà allora reindirizzati verso la pagina di registrazione dove, completando i campi Username (che deve essere almeno lungo 8 caratteri), e-mail (dove si dovrà inserire un indirizzo appropriato) e password (di almeno 8 caratteri e con almeno un numero). Se si è dimenticata la password, essa potrà essere resettata tramite la schermata di "Recupera Password" al quale possiamo accedere tramite un tasto posto nella schermata di Log In, qui verrà chiesto il proprio identificatore univoco e la nuova password da inserire. Una volta effettuata la procedura di Log In, la schermata home è costituita da due elementi principali:

- Vai a Profilo: Quando si interagisce con questo tasto si viene reindirizzati verso la schermata dell'utente, qui vengono visualizzate le sue credenziali (tranne la password) e vengono mostrate le sue statistiche, sono di rilievo: la media delle parole digitate per minuto, la precisione di battitura espressa in valore percentuale di tutti i test eseguiti, la media degli errori effettuati per test e infine il numero di test completati. Inoltre viene riportato un tasto per la modifica delle proprie credenziali che, permette di modificare facilmente Username, e-mail e Password.
- Lista dei Test: Organizzati all'interno di una tabella, riporta tutti i test disponibili per sfidarsi, di fianco ad ogni test è presente il tasto "seleziona" che permette di andare ad effettuare la sfida.

Il software è strutturato su due livelli di utenza, oltre all'utente possiamo, tramite la stessa schermata di Log In accedere in modalità Admin, inserendo le credenziali corrette (queste credenziali speciali sono create insieme al progetto). Una volta effettuata la procedura di Log In con le credenziali privilegiate, la schermata home sarà provvista di tre nuovi tasti:

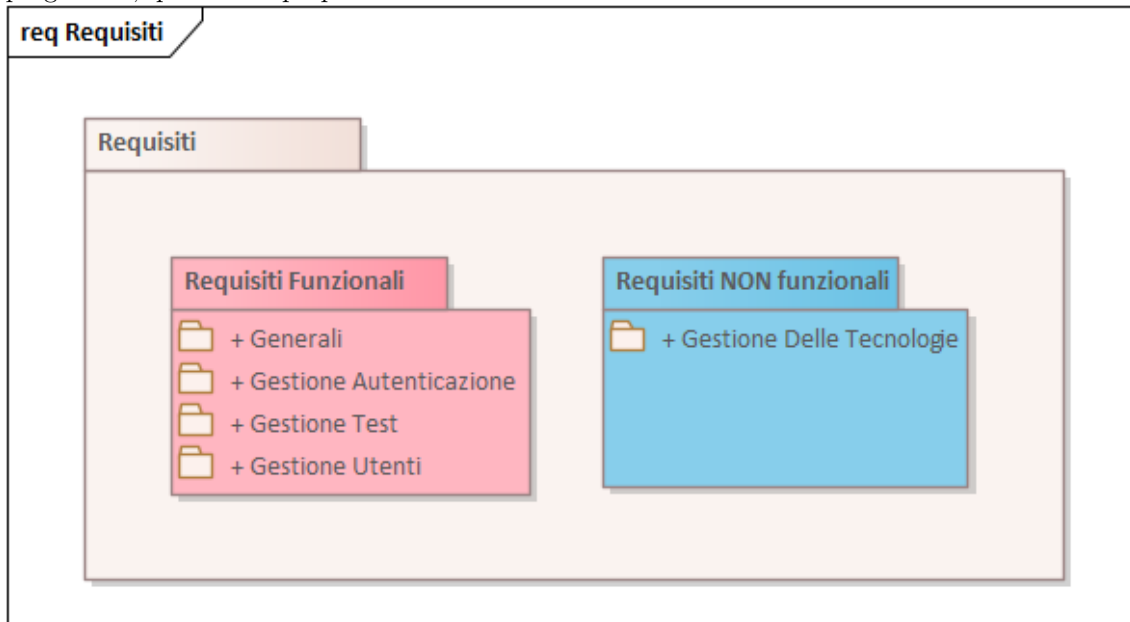
- Gestione Utenti: Quando si interagisce con questo tasto si viene reindirizzati verso una tabella che mostra ID, Username e e-mail di ogni utente registrato, con di fianco, un tasto con un'etichetta di "Modifica" e un altro con un'etichetta di "Elimina"
  - Modifica: Quando si interagisce con questo tasto si apre una schermata che permette la modifica esclusivamente dell'Username e della e-mail dell'utente selezionato.
  - Elimina: Quando si interagisce con questo tasto avviene la cancellazione dell'utente dalla tabella.
- Gestione Test: Anche qui verrà visualizzata una tabella, questa volta però con ID e nome di tutti i Test, si avranno i tasti di Modifica ed Elimina, inoltre se si cliccherà il nome di un Test verrà riportata la lista di tutte le parole di cui è composto. Sarà presente infine un tasto che permette la creazione di un nuovo Test tramite un'apposita schermata, al completamento di questa procedura si verrà riportati nella schermata di Gestione Test con il nuovo Test aggiunto in tabella.
- Backup: Tramite l'interazione con questo bottone si va ad effettuare un Backup della lista di tutti gli utilizzatori e di tutti i test che sono stati generati.

## 1.4 Glossario dei termini

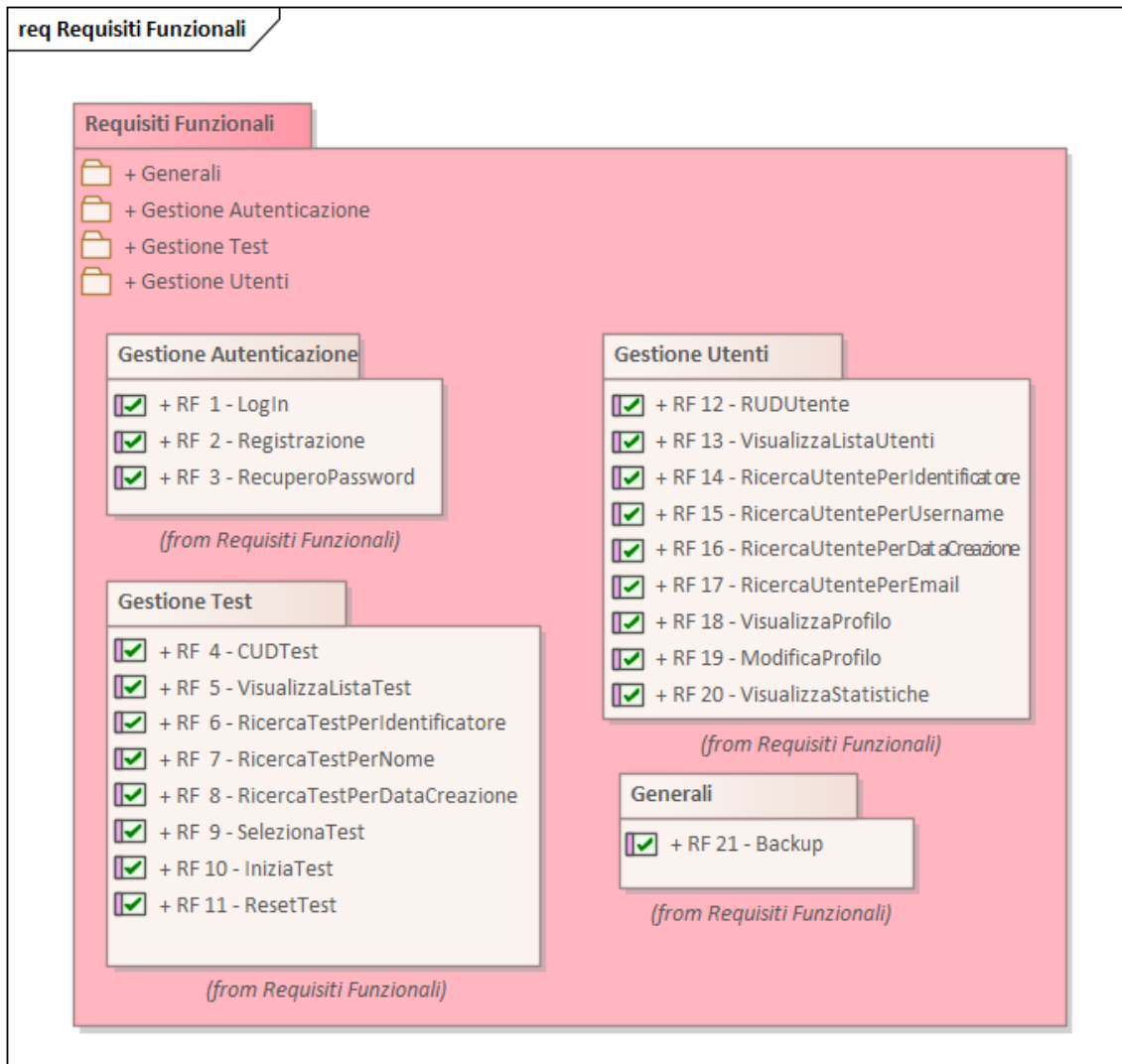
TERMINE	DEFINIZIONE	TIPOLOGIA	SINONIMI
Utilizzatore	Individuo umano non ancora loggato nel sistema	Tecnico	Nessuno
Credenziali	Dati con il quale si accede al sistema	Tecnico	Nessuno
Utente	Individuo umano loggato nel sistema	Tecnico	Nessuno
Admin	Individuo umano loggato nel sistema con privilegiato	Tecnico	Nessuno
Backup	Sistema di riserva e copia dei dati	Tecnico	Nessuno
Username	Parola con il quale un utente può essere riconosciuto all'interno del sistema	Tecnico	Nessuno
Password	Parola segreta che permette di entrare nel sistema	Tecnico	Nessuno
E-mail	Indirizzo di posta elettronica	Tecnico	Nessuno
Test	Una prova dattilografica con il quale l'utente può interagire	Business	Nessuno
Identifier	Identificatore universale associato ad Utenti e Test per permettere di essere trovati in modo semplice	Tecnico	Nessuno
Statistiche	Insieme di dati che si accumulando effettuando test (es: average wpm, accuracy,...)	Business	Nessuno
wpm	(Words Per Minute) parole digitate per minuto	Business	Nessuno

## 2 Analisi dei requisiti

I requisiti del sistema sono stati divisi in due macrocategorie: I requisiti funzionali riguardano le funzionalità richieste per il corretto funzionamento del software, quelli non funzionali sono tutti i vincoli progettuali, quindi non propriamente relativi al software.



## 2.1 Requisiti funzionali



### Area: Gestione Autenticazione

- RF 1 - **LogIn**: Il sistema deve permettere alla persona di accedere all'interno del sistema solo tramite l'inserimento delle credenziali corretto.
- RF 2 - **Registrazione**: Il sistema deve permettere ad una persona sprovvista di credenziali di crearne di nuove per accedere all'interno del software.
- RF 3 - **RecuperoPassword**: Il sistema deve permettere il recupero della Password tramite il reset della precedente qualora provvisti di Identificatore univoco.

### Area: Gestione Dei Test

- RF 4 - **CUDTest**: Il sistema dovrà permettere all'admin di effettuare le operazioni di CUD sui test.
- RF 5 - **VisualizzaListaTest**: Il sistema dovrà permettere all'admin di visualizzare la lista di tutti i test.
- RF 6 - **RicercaTestPerIdentificatore**: Il sistema dovrà permetterere all'admin di cercare un test per ID.
- RF 7 - **RicercaTestPerNome**: Il sistema dovrà permettere all'admin di cercare un test grazie al suo nome.

- RF 8 - **RicercaTestPerDataCreazione**: Il sistema dovrà permettere all'admin di cercare un test grazie alla sua data di creazione.
- RF 9 - **SelezionaTest**: Il sistema dovrà permettere all'utente di selezionare il test che vuole effettuare.
- RF 10 - **IniziaTest**: Il sistema dovrà permettere all'utente di iniziare un test.
- RF 11 - **ResetTest**: Il sistema dovrà permettere all'utente di ripristinare un test mentre è in svolgimento ritornando alla situazione di principio.

#### **Area: Gestione Utenti**

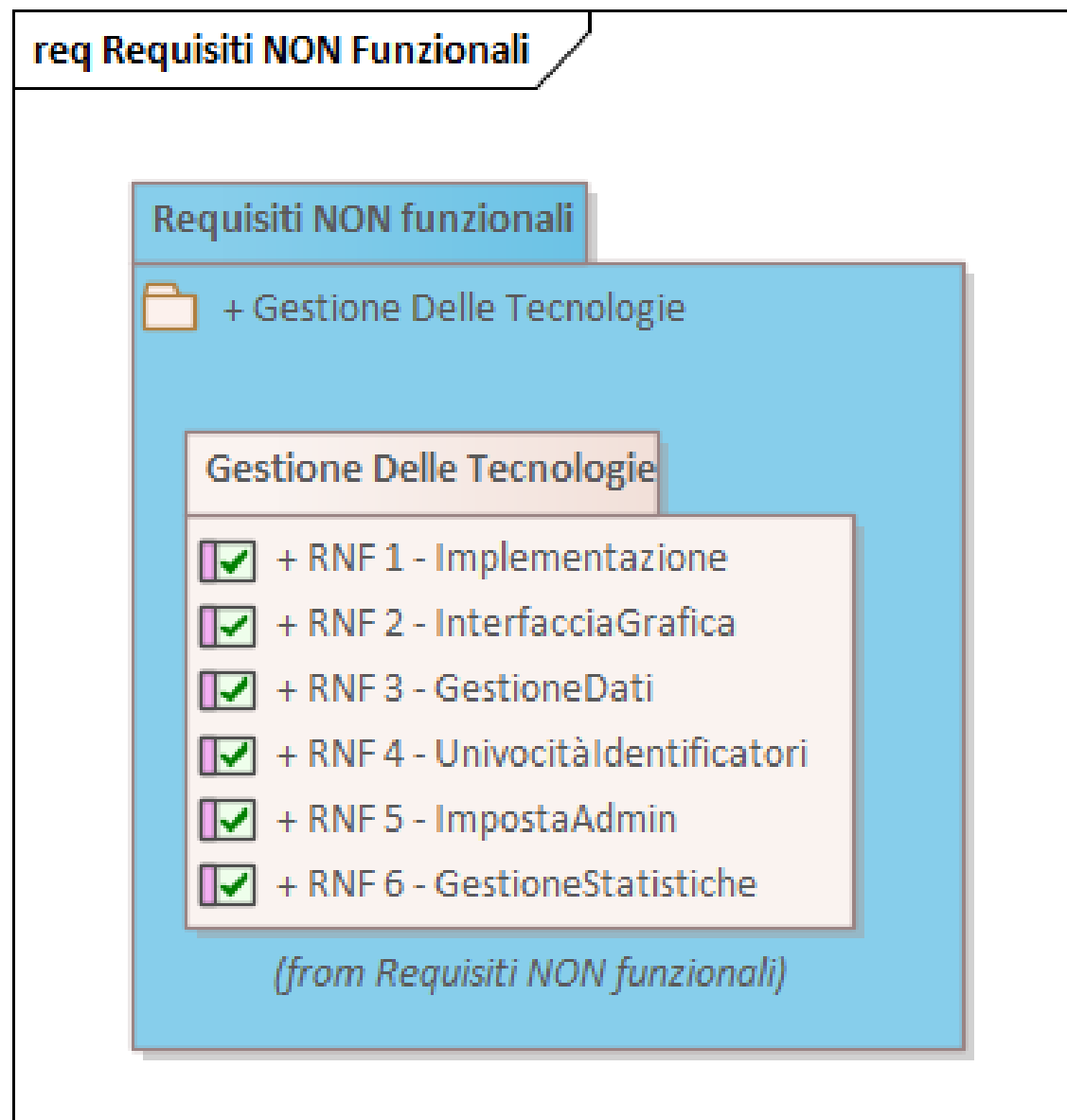
- RF 12 - **RUDUtente**: Il sistema dovrà permettere all'admin di effettuare operazioni di RUD sugli utenti.
- RF 13 - **VisualizzaListaUtenti**: Il sistema dovrà permettere all'admin di visualizzare la lista di tutti gli utenti.
- RF 14 - **RicercaUtentePerIdentificatore**: Il sistema dovrà permettere all'admin di cercare un utente per ID
- RF 15 - **RicercaUtentePerUsername**: Il sistema dovrà permettere all'admin di cercare un utente per Username.
- RF 16 - **RicercaUtentePerDataCreazione**: Il sistema dovrà permettere all'admin di cercare un utente grazie alla sua data di creazione.
- RF 17 - **RicercaUtentePerEmail**: Il sistema dovrà permettere all'admin di cercare un utente per e-mail.
- RF 18 - **VisualizzaProfilo**: Il sistema dovrà permettere all'utente di visualizzare il proprio profilo.
- RF 19 - **ModificaProfilo**: Il sistema dovrà permettere all'utente di modificare le credenziali di accesso del proprio profilo.
- RF 20 - **VisualizzaStatistiche**: Il sistema dovrà permettere all'utente di visualizzare le proprie statistiche relative ai test.

#### **Area: Generali**

- RF 21 - **Backup**: Il sistema dovrà permettere all'admin di effettuare un backup dei dati tramite l'interazione con un bottone nell'apposita schermata



## 2.2 Requisiti non funzionali

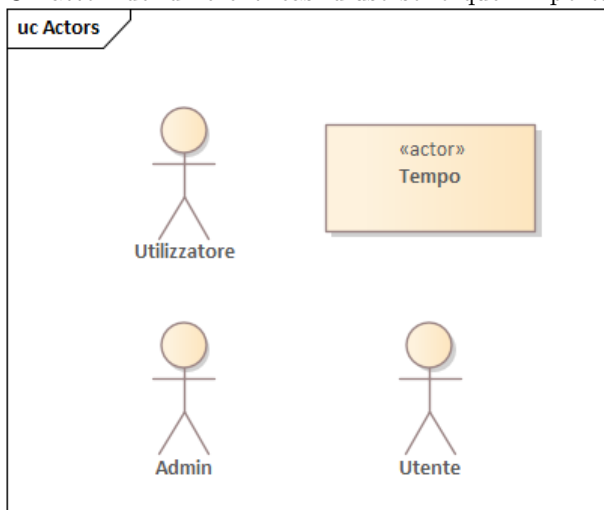


### Area: Gestione delle tecnologie

- RNF 1 - **Implementazione**: Il sistema dovrà essere scritto in python 3.
- RNF 2 - **InterfacciaGrafica**: Il sistema dovrà essere dotato di interfaccia grafica.
- RNF 3 - **GestioneDati**: Il sistema dovrà usare dei file PICKLE per la gestione dei dati.
- RNF 4 - **UnivocitàIdentificatori**: Il sistema dovrà essere dotato di identificatori univoci per gli utenti e per i test.
- RNF 5 - **ImpostaAdmin**: Il sistema dovrà essere dotato in fase di prima configurazione di un account predefinito di tipo Admin.
- RNF 6 - **GestioneStatistiche**: Il sistema dovrà automaticamente calcolare e aggiornare tutte le statistiche.

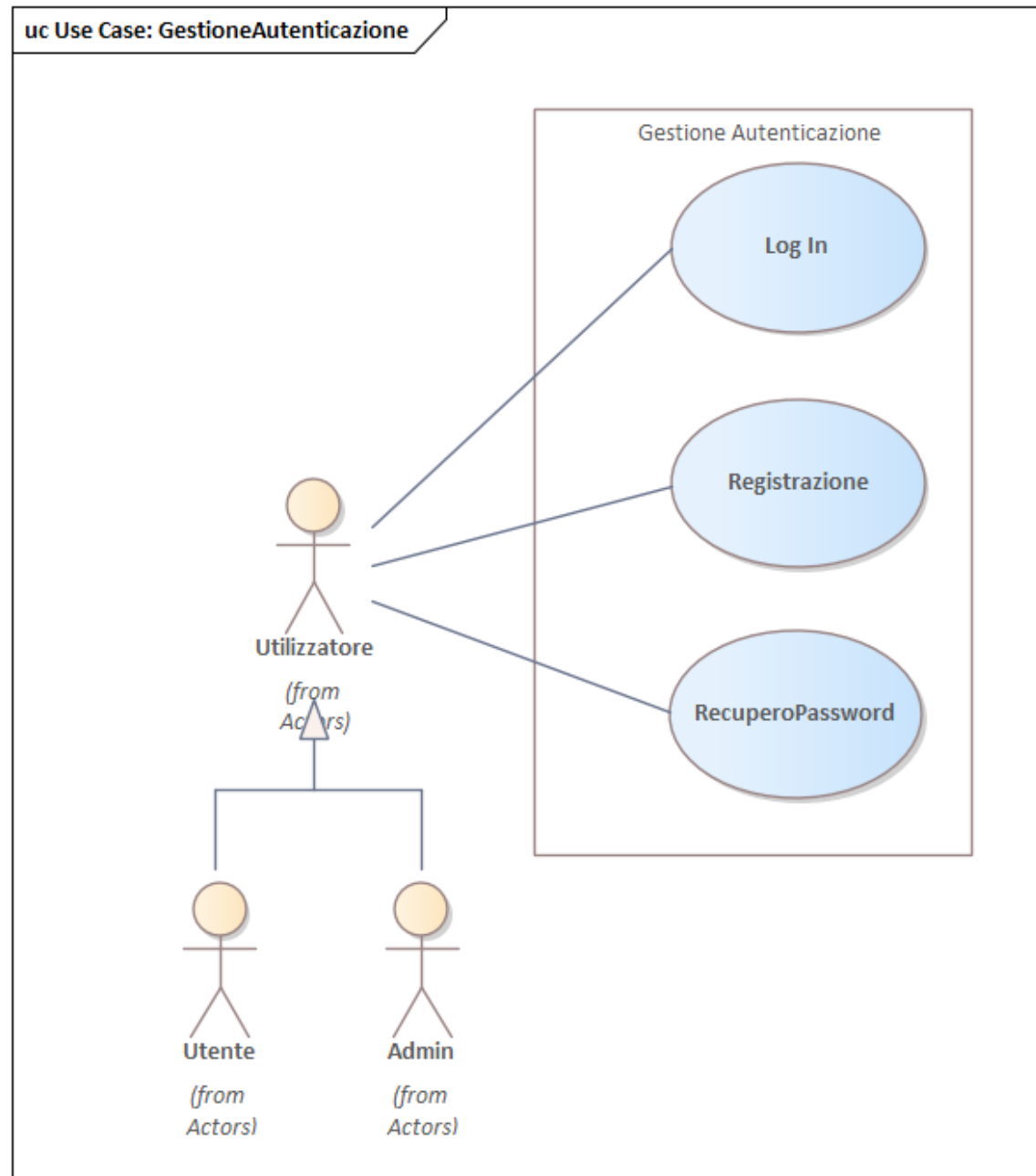
## 2.3 Diagrammi dei casi d'uso

Gli attori dei differenti casi d'uso sono quelli riportati nella seguente figura:



Un utilizzatore è un'entità generica del sistema, dopo essersi autenticato esso, a seconda delle credenziali inserite può essere riconosciuto come Utente oppure Admin. L'ultimo attore è il tempo che non viene identificato come una persona vera e propria, ma interviene in tutte quelle situazioni in cui abbiamo bisogno del controllo temporale (gestione del tempo in un test e backup dei dati).

### 2.3.1 Gestione Autenticazione



## Log In

- Type: Use Case
- Package: UseCase\_GestioneAutenticazione
- BREVE DESCRIZIONE  
Il sistema permette all'utilizzatore di entrare all'interno del sistema tramite l'inserimento delle credenziali corretto.
- CONSTRAINTS
  - Precondizioni: Nessuna
  - Postcondizioni: Accesso alla Home Page
- SCENARIOS
  - Sequenza degli eventi principale
    1. Il caso d'uso inizia quando un utilizzatore vuole accedere al sistema.
    2. Il sistema chiede all'utilizzatore di inserire nei campi appositi Username e Password di accesso
    3. L'utilizzatore inserisce all'interno dei campi corrispondenti il proprio Username e Password.
    4. **If** Username non esiste all'interno della lista utenti
      - (a) Il sistema restituisce un errore.
    5. **else if** Password non è corretta.
      - (a) Il sistema restituisce un errore.
    6. **else if** L'utilizzatore ha usato delle credenziali Utente.
      - (a) Il sistema carica la Home Page versione Utente.
    7. **else if** L'utilizzatore ha usato delle credenziali Admin.
      - (a) Il sistema carica la Home Page versione Admin.
  - Sequenza degli eventi alternativa
    1. Nessuna

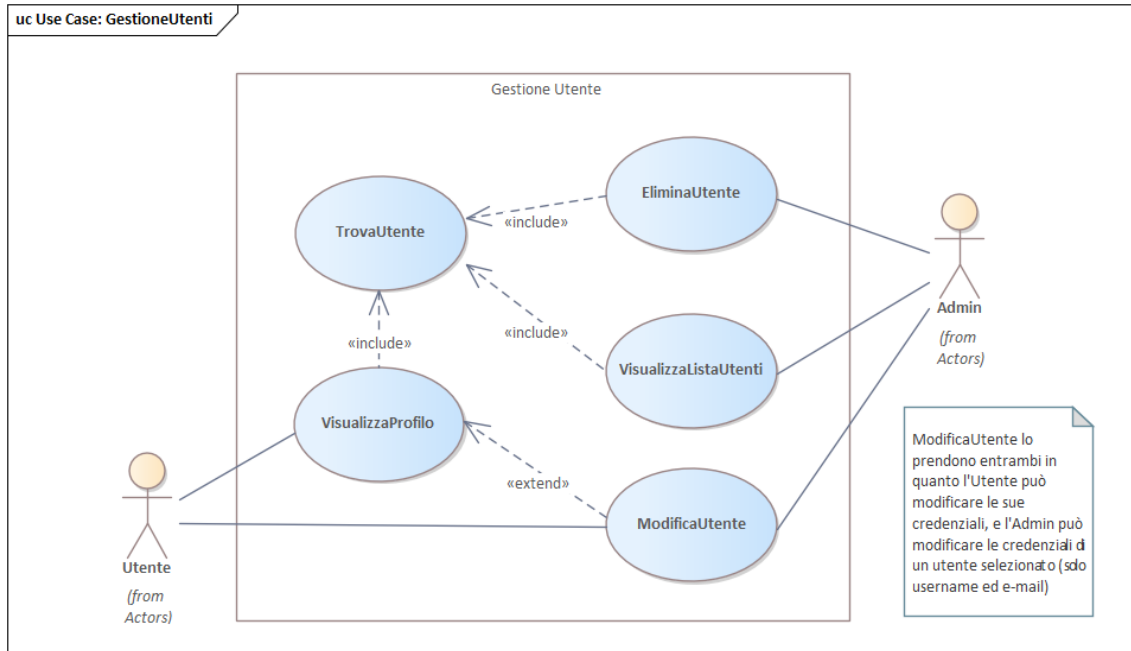
## Registrazione

- Type: Use Case
- Package: UseCase\_GestioneAutenticazione
- BREVE DESCRIZIONE  
Il sistema permette ad un utilizzatore che non ha credenziali di crearne di nuove
- CONSTRAINTS
  - Precondizioni: L'utilizzatore non ha credenziali
  - Postcondizioni: Nessuna
- SCENARIOS
  - Sequenza degli eventi principale
    1. Il caso d'uso inizia quando un utilizzatore non è provvisto di credenziali appropriate per l'accesso al sistema
    2. Il sistema chiede all'utilizzatore di inserire l'username, l'indirizzo e-mail e la password che vuole utilizzare.
    3. If Username possiede meno di 8 caratteri **or** più di 16 caratteri.
      - (a) Il sistema restituisce un errore.
    4. **else if** indirizzo email non possiede il carattere @.
      - (a) Il sistema restituisce un errore.
    5. **else if** password non possiede un carattere tra i numeri che vanno da 0 a 9
      - (a) Il sistema restituisce un errore.
    6. **else**
      - (a) Il sistema salva le credenziali nella lista di tutti gli Utenti
  - Sequenza degli eventi alternativa
    1. Nessuna.

## RecuperoPassword

- Type: Use Case
- Package: UseCase\_GestioneAutenticazione
- BREVE DESCRIZIONE  
Il sistema permette ad un utilizzatore che non ricorda la propria password di resettarla tramite l'inserimento corretto del proprio identificatore.
- CONSTRAINTS
  - Precondizioni: Nessuna
  - Postcondizioni: Aggiornamento della password
- SCENARIOS
  - Sequenza degli eventi principale
    1. Il caso d'uso inizia con l'utilizzatore che vuole recuperare la propria password di accesso.
    2. Il sistema chiede all'utente l'identificatore e la nuova password da inserire.
    3. If l'identificatore esiste
      - (a) **if** password non possiede un carattere tra i numeri che vanno da 0 a 9
        - i. Il sistema restituisce un errore.
      - (b) **else**
        - i. Il sistema aggiorna la password
    4. else
      - (a) Il sistema restituisce un errore.
  - Sequenza degli eventi alternativa
    1. Nessuna

### 2.3.2 Gestione Utenti



#### Trova Utente

- Type: Use Case
- Package: UseCase\_GestioneUtenti
- BREVE DESCRIZIONE  
Il sistema permette la ricerca di un utente inserendo parametri.
- CONSTRAINTS
  - Precondizioni: L'attore primario deve essere l'Admin.
  - Postcondizioni: Nessuna
- SCENARIOS
  - Sequenza degli eventi principale
    1. L'Admin inserisce Username, indirizzo e-mail oppure Identificatore dell'Utente che vuole cercare.
    2. If viene trovato uno o più Utenti che corrispondono
      - (a) Il sistema aggiorna la lista utenti evidenziando gli Utenti appropriati.
    3. else
      - (a) Il sistema restituisce un errore.
  - Sequenza degli eventi alternativa
    1. Il caso d'uso inizia quando il sistema vuole effettuare la ricerca di un utente.
    2. Il sistema inserisce Username, indirizzo e-mail oppure Identificatore dell'Utente che vuole cercare.
    3. If viene trovato uno o più Utenti che corrispondono
      - (a) Il sistema non restituisce un errore.
    4. else
      - (a) Il sistema restituisce un errore.

## Visualizza Lista Utenti

- Type: Use Case
- Package: UseCase\_GestioneUtenti
- BREVE DESCRIZIONE  
Il sistema permette all'Admin la visualizzazione di tutti gli Utenti in una lista.
- CONSTRAINTS
  - Precondizioni: L'attore primario deve essere l'Admin.
  - Postcondizioni: Nessuna
- SCENARIOS
  - Sequenza degli eventi principale
    1. Il caso d'uso inizia quando l'Admin vuole visualizzare tutti gli utenti registrati
    2. **for each** utente nella lista utenti
      - (a) **Include(TrovaUtente)**.
      - (b) Il sistema carica i dati in una tabella.
  - Sequenza degli eventi alternativa
    1. Nessuna



## Modifica Utente

- Type: UseCase
- Package: UseCase\_GestioneUtenti
- BREVE DESCRIZIONE  
Un Utente può modificare le proprie credenziali di accesso, un Admin può modificare username ed e-mail di un Utente.
- CONSTRAINTS
  - Precondizioni: L'attore primario deve essere un Utente oppure l'Admin.
  - Postcondizioni: Utente modificato.
- SCENARIOS
  - Sequenza degli eventi principale
    1. Il caso d'uso inizia quando l'utente vuole modificare le proprie credenziali oppure quando l'admin clicca sul tasto **modifica** posto di fianco ad ogni utente nella lista utenti.
    2. **Include(TrovaUtente)**
    3. Il sistema apre il pannello di modifica delle credenziali.
    4. **If** l'attore primario è un Utente.
      - (a) Il sistema fornisce i campi di Username, indirizzo e-mail e Password disponibili per l'aggiornamento.
    5. **else**
      - (a) Il sistema fornisce solamente i campi di Username e indirizzo e-mail disponibili per l'aggiornamento.
    6. **If** Username possiede meno di 8 caratteri **or** più di 16 caratteri.
      - (a) Il sistema restituisce un errore.
    7. **else if** indirizzo email non possiede il carattere @.
      - (a) Il sistema restituisce un errore.
    8. **else if** password non possiede un carattere tra i numeri che vanno da 0 a 9
      - (a) Il sistema restituisce un errore.
      - (b) Il sistema aggiorna i dati di dell'utente con le nuove credenziali.
    9. Sequenza degli eventi alternativa
      - (a) La sequenza inizia quando l'attore primario esce senza salvare le credenziali modificate.
      - (b) **If** l'Utente/Admin non modifica le credenziali nel pannello.
        - i. Il sistema non aggiorna i dati, ma mantiene quelli esistenti.

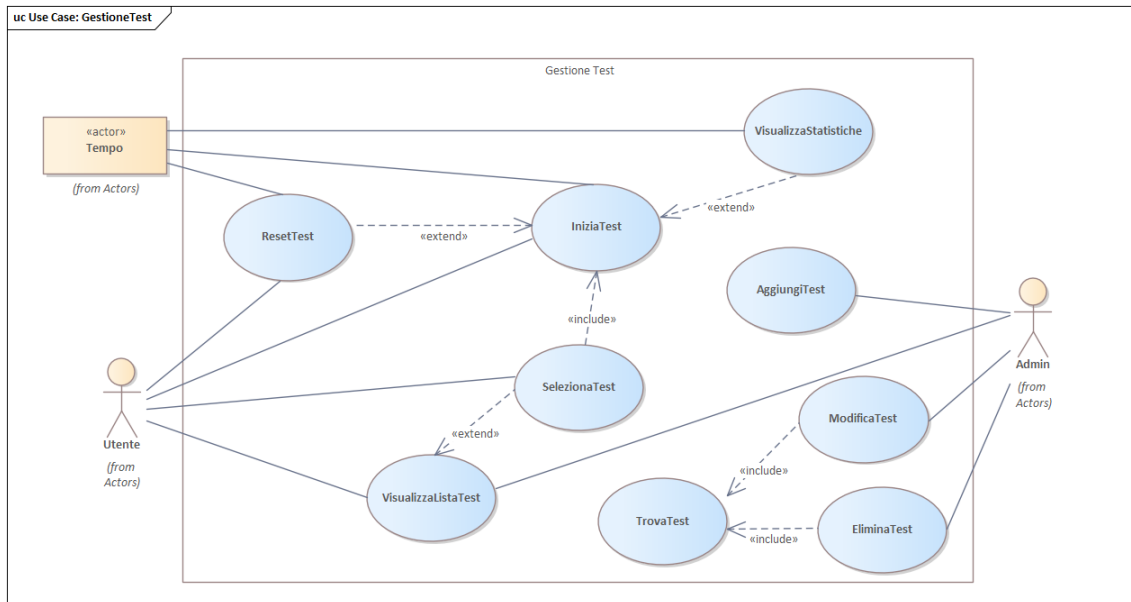
## **Elimina Utente**

- Type: Use Case
- Package: UseCase\_GestioneUtenti
- BREVE DESCRIZIONE  
Il sistema elimina l'utente selezionato dall'Admin
- CONSTRAINTS
  - Precondizioni: L'attore primario deve essere l'Admin
  - Postcondizioni: Utente Eliminato
- SCENARIOS
  - Sequenza degli eventi principale
    1. Il caso d'uso inizia quando l'Admin clicca il tasto **elimina** posto di fianco ad ogni utente nella lista utenti.
    2. **Include(TrovaUtente)**
    3. Il sistema cancella l'utente selezionato dalla lista di tutti gli utenti.
    4. Il sistema aggiorna la lista utenti.
  - Sequenza degli eventi alternativa
    1. Nessuna

## **Visualizza Profilo**

- Type: Use Case
- Package: UseCase\_GestioneUtenti
- BREVE DESCRIZIONE  
Il sistema permette ad un Utente di visualizzare il proprio profilo.
- CONSTRAINTS
  - Precondizioni: L'attore primario deve essere un Utente
  - Postcondizioni: Nessuno
- SCENARIOS
  - Sequenza degli eventi principale
    1. `Inlcude(TrovaUtente)`
    2. Il sistema carica dalla lista utenti tutti i dati dell'utente (tranne la password) corrispondente e li sistema nella pagina del profilo in modo ordinato.
  - Sequenza degli eventi alternativa
    1. Nessuna

### 2.3.3 Gestione Test



#### Trova Test

- Type: Use Case
- Package: UseCase\_GestioneTest
- BREVE DESCRIZIONE  
Il sistema permette ad un Admin oppure al sistema la ricerca di un test.
- CONSTRAINTS
  - Precondizioni: L'attore primario deve essere l'Admin oppure il sistema
  - Postcondizioni: Nessuna
- SCENARIOS
  - Sequenza degli eventi principale
    1. L'admin inserisce Nome, Identificatore o Data di creazione del Test che vuole cercare.
    2. If il sistema trova uno o più Test che corrispondono alle credenziali inserite
      - (a) Il sistema aggiorna la lista Test esclusivamente inserendo i Test appropriati.
    3. else
      - (a) Il sistema restituisce un errore.
  - Sequenza degli eventi alternativa
    1. Il caso d'uso inizia quando il sistema vuole effettuare la ricerca di un Test.
    2. Il sistema effettua la ricerca con i campi desiderati.
    3. If viene trovato uno o più Test che corrispondono
      - (a) Il sistema non restituisce un errore.
    4. else
      - (a) Il sistema restituisce un errore.

## Visualizza Lista Test

- Type: Use Case
- Package: UseCase\_GestioneTest
- BREVE DESCRIZIONE  
Il sistema permette ad un Utente oppure all'Admin, la visualizzazione di tutti i test in una lista
- CONSTRAINTS
  - Precondizioni: Nessuna
  - Postcondizioni: Nessuna
- SCENARIOS
  - Sequenza degli eventi principale
    1. Il caso d'uso inizia quando l'attore primario vuole visualizzare tutti i test.
    2. **for each** test nella lista test.
      - (a) **Include(TrovaTest)**.
      - (b) Il sistema carica i dati in una tabella.
  - Sequenza degli eventi alternativa
    1. Nessuna

## Modifica Test

- Type: Use Case
- Package: UseCase\_GestioneTest
- BREVE DESCRIZIONE  
testo di prova
- CONSTRAINTS
  - Precondizioni: L'attore primario deve essere l'Admin.
  - Postcondizioni: Test modificato.
- SCENARIOS
  - Sequenza degli eventi principale
    1. Il caso d'uso inizia quando l'Admin vuole modificare un Test
    2. **Include(TrovaTest)**
    3. Il sistema apre il pannello di modifica del test.
    4. **If** Il sistema trova un test con lo stesso nome di quello che sta provando a modificare
      - (a) Il sistema restituisce un errore
    5. **If** Il numero di parole inserite nel test è minore di 5
      - (a) Il sistema restituisce un errore
    6. **If** Il numero di caratteri totale inserito è maggiore di 350
      - (a) Il sistema restituisce un errore
    7. Il sistema aggiorna i dati del Test con quelli nuovi
  - Sequenza degli eventi alternativa
    1. La sequenza inizia quando l'Admin esce senza salvare le modifiche
    2. **If** l'Admin non modifica dati nel pannello.
      - (a) Il sistema non aggiorna i dati, ma mantiene quelli esistenti.

## **Elimina Test**

- Type: Use Case
- Package: UseCase\_GestioneTest
- BREVE DESCRIZIONE  
testo di prova
- CONSTRAINTS
  - Precondizioni: L'attore primario deve essere l'Admin
  - Postcondizioni: Test eliminato
- SCENARIOS
  - Sequenza degli eventi principale
    1. Il caso d'uso inizia quando l'Admin clicca il tasto **elimina** posto di fianco ad ogni Test nella lista Test.
    2. **Include(TrovaTest)**
    3. Il sistema cancella il test selezionato.
    4. Il sistema aggiorna la lista test.
  - Sequenza degli eventi alternativa
    1. Nessuna

## Aggiungi Test

- Type: Use Case
- Package: UseCase\_GestioneTest
- BREVE DESCRIZIONE  
L'admin aggiunge un test alla lista.
- CONSTRAINTS
  - Precondizioni: L'attore primario deve essere l'Admin
  - Postcondizioni: Test aggiunto
- SCENARIOS
  - Sequenza degli eventi principale
    1. Il caso d'uso inizia quando l'Admin vuole aggiungere un test.
    2. Il sistema chiede all'Admin di inserire i parametri che caratterizzano il test
    3. **foreach** parola nella lista di parole da inserire
      - (a) **If** la lunghezza della parola inserita è maggiore di 50
        - i. Il sistema restituisce un errore
    4. **If** Il numero di parole inserite nel test è minore di 5
      - (a) Il sistema restituisce un errore
    5. **If** Il numero di caratteri totale inserito è maggiore di 350
      - (a) Il sistema restituisce un errore
    6. Il sistema salva i parametri e riporta l'Admin nella pagina con tutti i test.
  - Sequenza degli eventi alternativa
    1. Nessuna



## Selezione Test

- Type: Use Case
- Package: UseCase\_GestioneTest
- BREVE DESCRIZIONE  
L'Utente seleziona il test che vuole effettuare
- CONSTRAINTS
  - Precondizioni: L'attore primario deve essere un Utente
  - Postcondizioni: Il test è pronto per essere svolto
- SCENARIOS
  - Sequenza degli eventi principale
    1. Il caso d'uso inizia con l'Utente che vuole effettuare un test
    2. `Include(VisualizzaListaTest)`
    3. L'utente può selezionare il test che preferisce dalla lista
  - Sequenza degli eventi alternativa
    1. Nessuna

## Inizia Test

- Type: Use Case
- Package: UseCase\_GestioneTest
- BREVE DESCRIZIONE  
Il sistema permette ad un Utente di iniziare il test selezionato
- CONSTRAINTS
  - Precondizioni: Test selezionato
  - Postcondizioni: Visualizzazione statistiche
- SCENARIOS
  - Sequenza degli eventi principale
    1. Il sistema avvia un cronometro
    2. **while** il testo da digitare è diverso dal testo digitato
      - (a) **for each** carattere digitato dall'utente
        - i. **If** il carattere è uguale a quello proposto dal test
          - A. Prosegui al carattere successivo.
        - ii. **else**
          - A. Il sistema dà un errore.
  - Sequenza degli eventi alternativa
    1. Nessuna

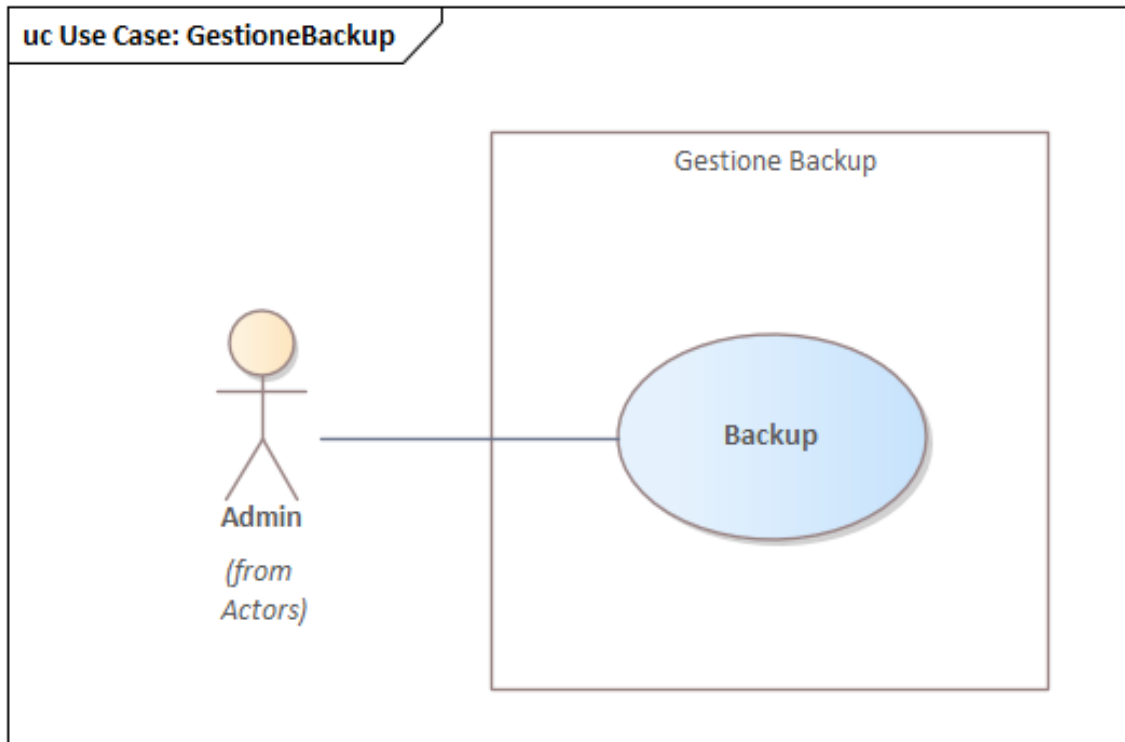
## Reset Test

- Type: Use Case
- Package: UseCase\_GestioneTest
- BREVE DESCRIZIONE  
Il sistema permette ad un utente di ripristinare il test in svolgimento alle condizioni iniziali
- CONSTRAINTS
  - Precondizioni: Inizia Test
  - Postcondizioni: Test reimpostato
- SCENARIOS
  - Sequenza degli eventi principale
    1. Il caso d'uso inizia con l'Utente che vuole riprovare il test.
    2. Il sistema riporta il conto alla rovescia all'inizio.
    3. Il sistema riporta tutti i parametri a quelli di default.
    4. Il sistema riporta il cursore per la battitura all'inizio del Test.
  - Sequenza degli eventi alternativa
    1. Nessuna

## Visualizza Statistiche

- Type: Use Case
- Package: UseCase\_GestioneTest
- BREVE DESCRIZIONE  
Il sistema alla fine di un test finito correttamente visualizza le statistiche.
- CONSTRAINTS
  - Precondizioni: Test finito
  - Postcondizioni: Visualizzazione Statistiche
- SCENARIOS
  - Sequenza degli eventi principale
    1. Il caso d'uso inizia quando un test termina con successo.
    2. Il sistema calcola la media del numero di parole digitate per minuto.
    3. Il sistema calcola la precisione percentuale media del test.
    4. Il sistema calcola il numero totale di errori eseguiti nel test.
    5. Il sistema incrementa il numero totale di test eseguiti di 1.
    6. Il sistema aggiorna la media totale di parole digitate al minuto.
    7. Il sistema aggiorna la media totale di precisione di battitura.
    8. Il sistema aggiorna la media totale degli errori.
    9. Il sistema carica i dati in una pagina apposita.
  - Sequenza degli eventi alternativa
    1. Nessuna

### 2.3.4 Gestione Backup



#### Backup

- Type: Use Case
- Package: UseCase\_GestioneBackup
- BREVE DESCRIZIONE  
Il sistema effettua un backup dei dati sotto richiesta dell'Admin
- CONSTRAINTS
  - Precondizioni: Nessuna
  - Postcondizioni: Nessuna
- SCENARIOS
  - Sequenza degli eventi principale
    1. Il caso d'uso inizia quando l'Admin vuole effettuare un backup dei dati.
    2. Il sistema preleva i dati degli utenti.
    3. Il sistema copia i dati degli utenti prelevati sul disco.
    4. Il sistema preleva i dati dei test.
    5. Il sistema copia i dati prelevati dei test sul disco.
  - Sequenza degli eventi alternativa
    1. Nessuna.

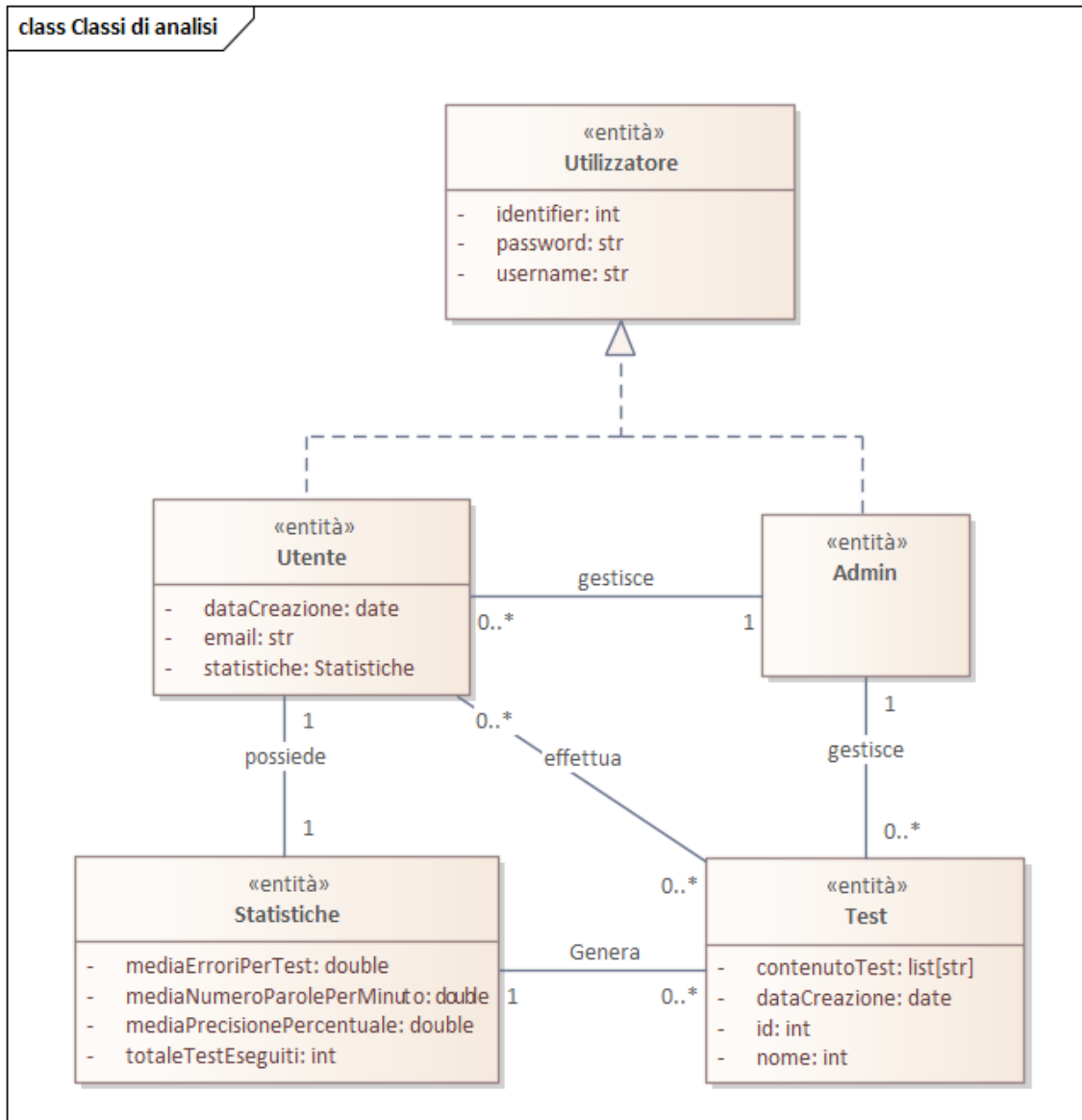
## 2.4 Matrice di Mapping

Target \ Source	RF 1 - Login	RF 2 - Registrazione	RF 3 - RecuperoPassword	RF 4 - CUDTest	RF 5 - VisualizzaListaTest	RF 6 - RicercaTestPerIdentificatore	RF 7 - RicercaTestPerNome	RF 8 - RicercaTestPerDataCreazione	RF 9 - SelezioneTest	RF 10 - IniziaTest	RF 11 - ResetTest	RF 12 - RUDUtente	RF 13 - VisualizzaListaUtenti	RF 14 - RicercaUtentePerIdentificatore	RF 15 - RicercaUtentePerUsername	RF 16 - RicercaUtentePerDataCreazione	RF 17 - RicercaUtentePerEmail	RF 18 - VisualizzaProfilo	RF 19 - ModificaProfilo	RF 20 - VisualizzaStatistiche	RF 21 - Backup
Backup																					↑
CreaUtente		↑																			
EliminaTest				↑																	
EliminaUtente												↑									
IniziaTest										↑											
Log In	↑																				
ModificaTest				↑																	
ModificaUtente												↑							↑		
RecuperoPassword			↑																		
Registrazione		↑																			
ResetTest										↑											
RicercaUtente														↑	↑		↑				
RiprovaTest																					
SelezioneTest									↑												
TrovaTest						↑	↑	↑													
TrovaUtente														↑	↑	↑	↑				
VisualizzaListaTest				↑																	
VisualizzaListaUtenti													↑								
VisualizzaProfilo												↑						↑		↑	
VisualizzaStatistiche																				↑	

### 3 Diagrammi di Analisi

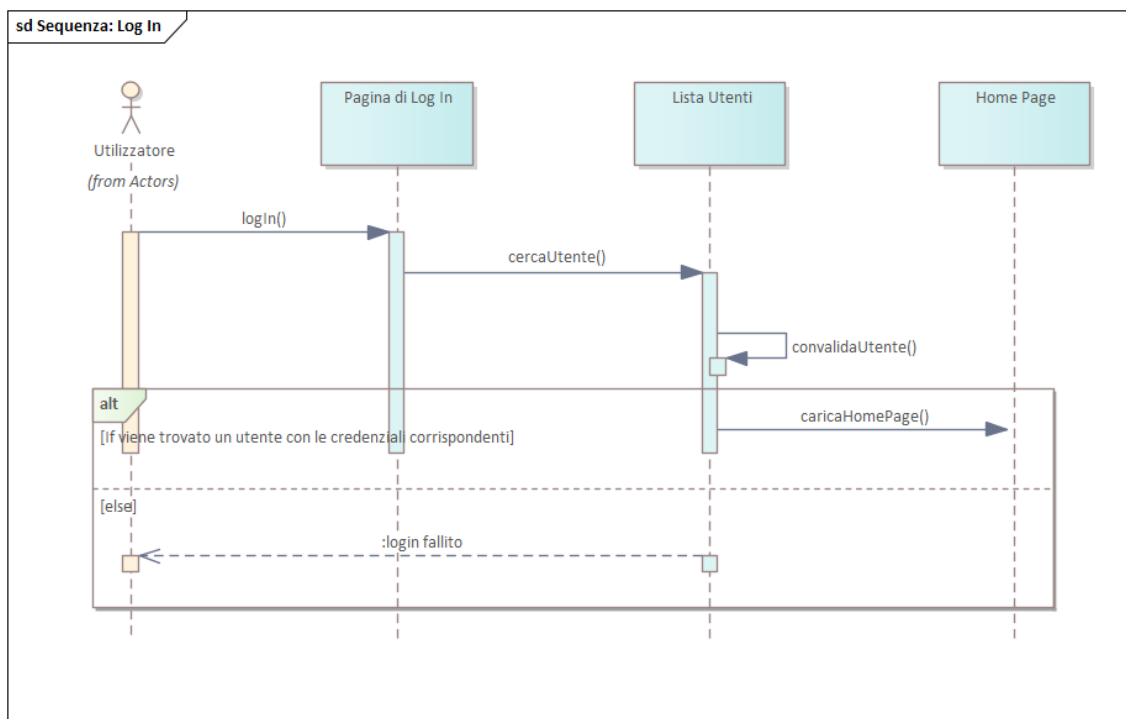
Dopo aver analizzato i casi d'uso sono stati estratti da essi le classi che andranno a comporre il sistema per poi andare a creare dei diagrammi di sequenza ed attività che rappresenteranno i comportamenti principali del software.

#### 3.1 Diagrammi delle Classi di Analisi



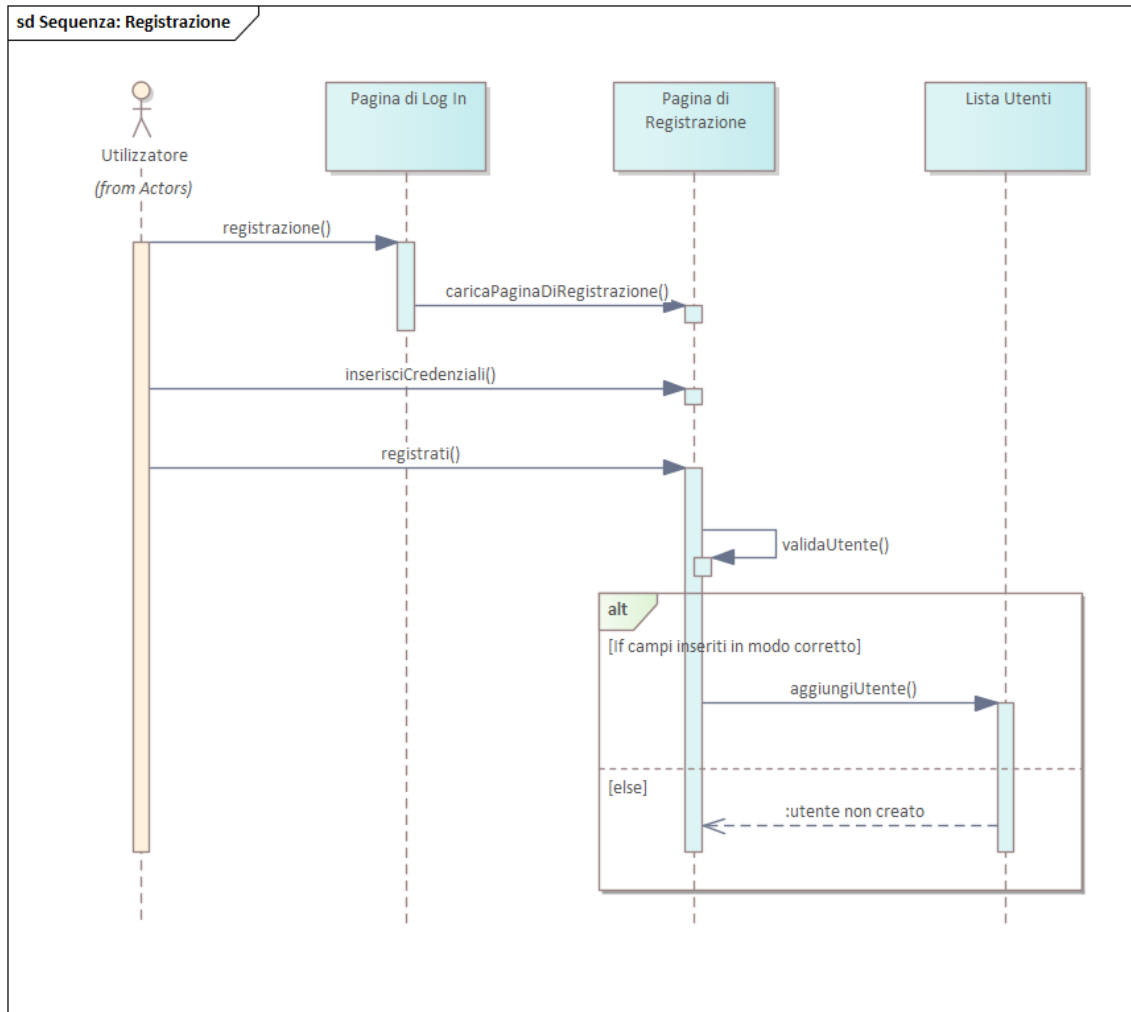
## 3.2 Diagrammi di Sequenza

### 3.2.1 Log In

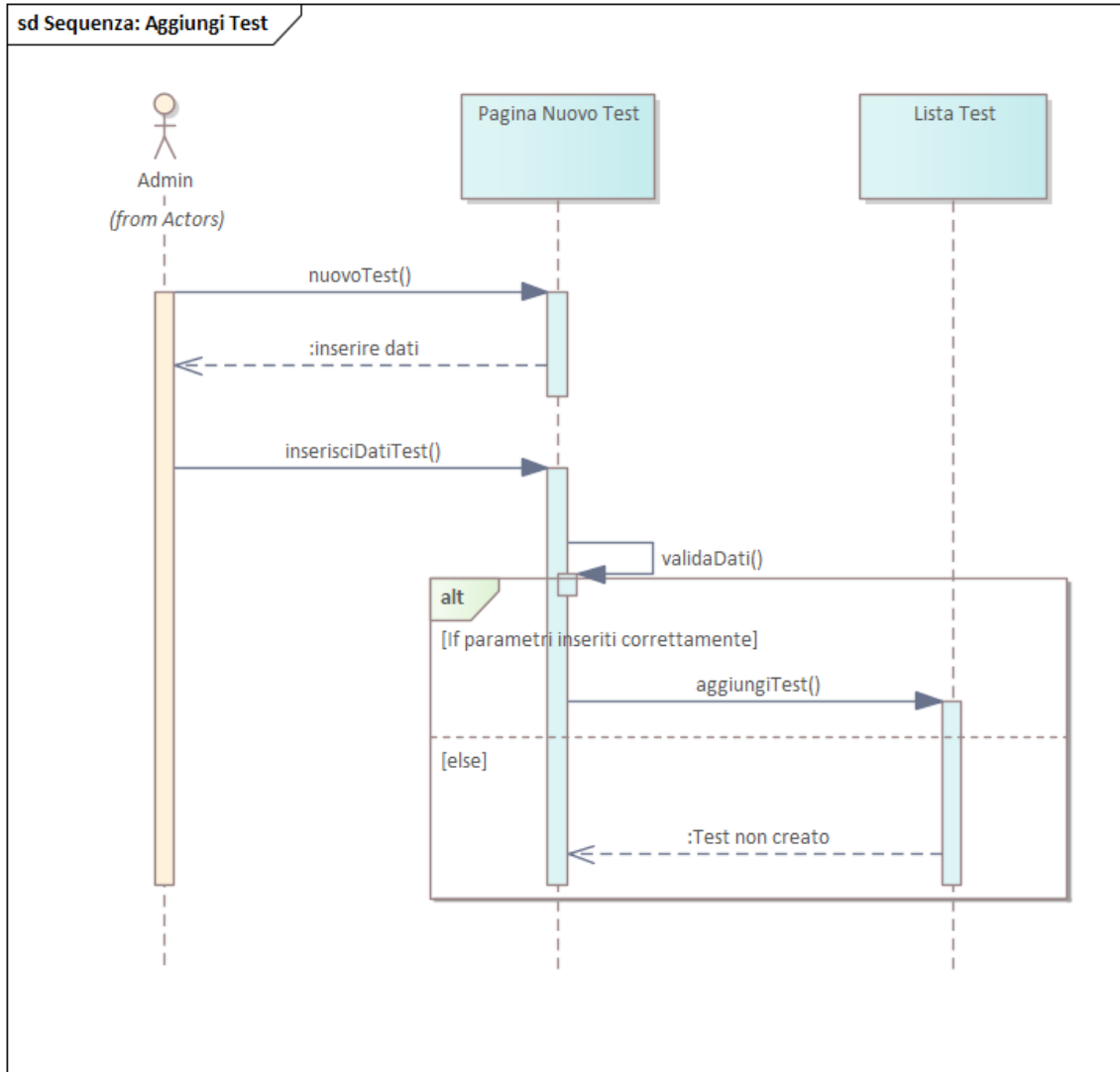




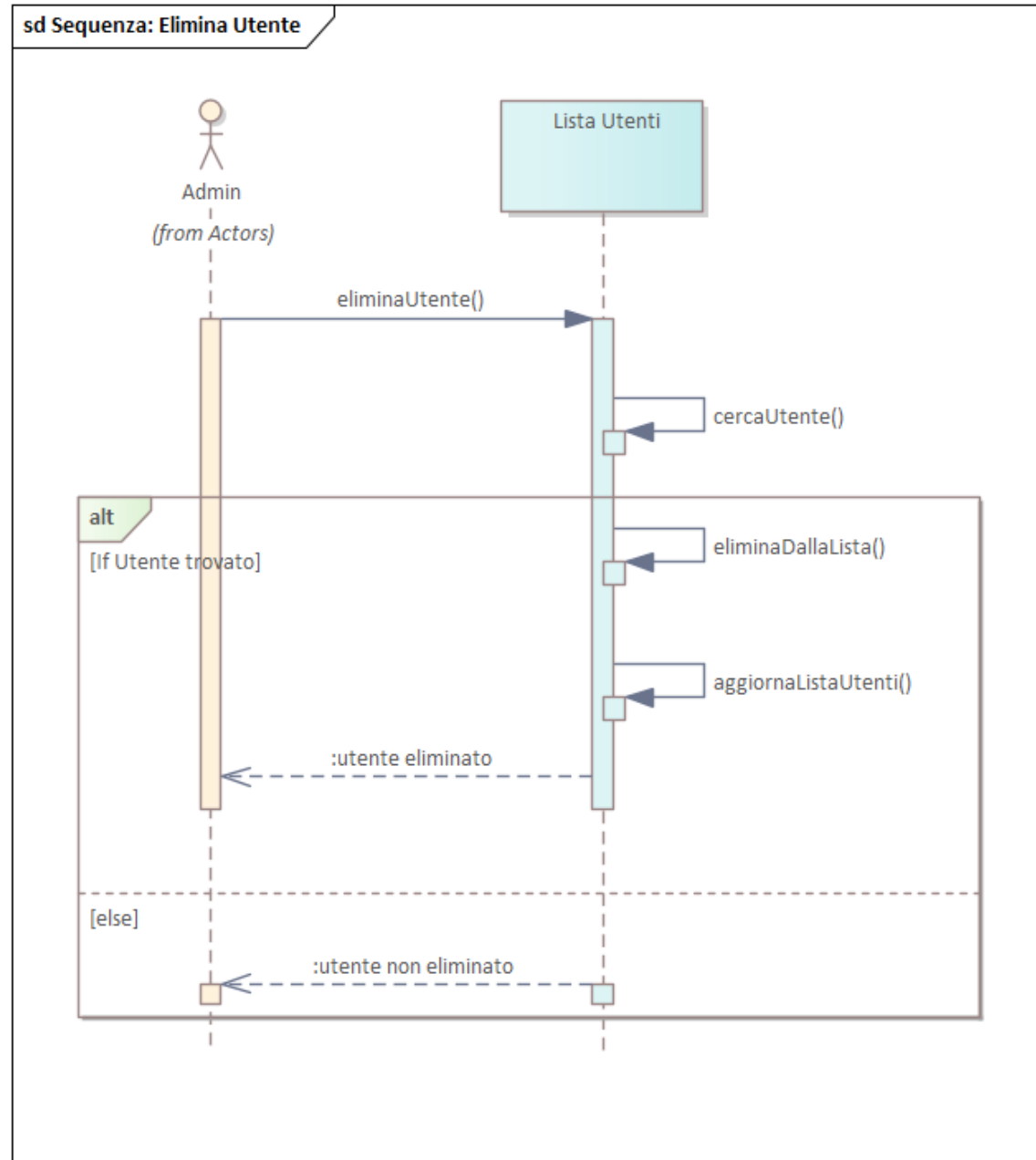
### 3.2.2 Registrazione



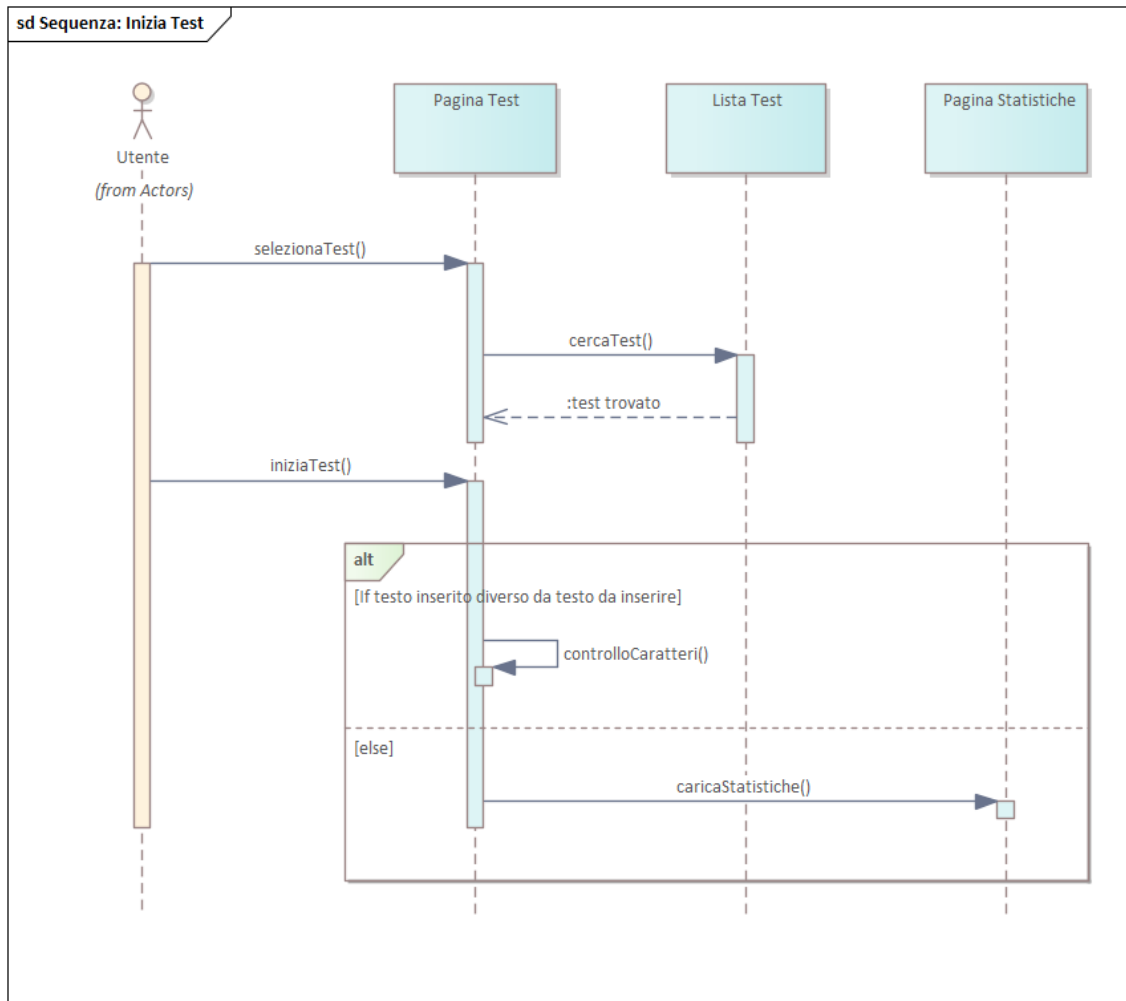
### 3.2.3 Aggiungi Test



### 3.2.4 Elimina Utente

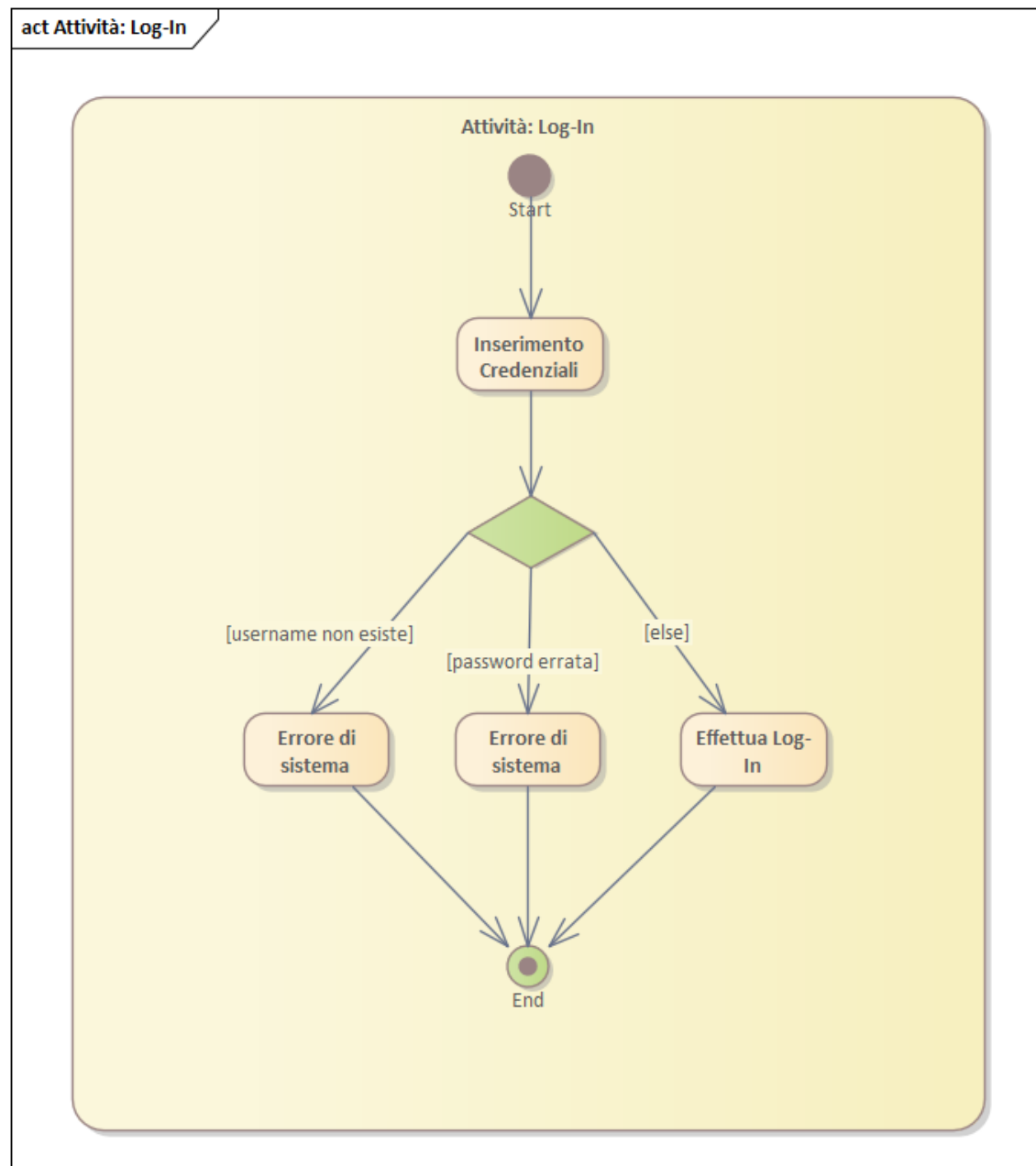


### 3.2.5 Inizia Test

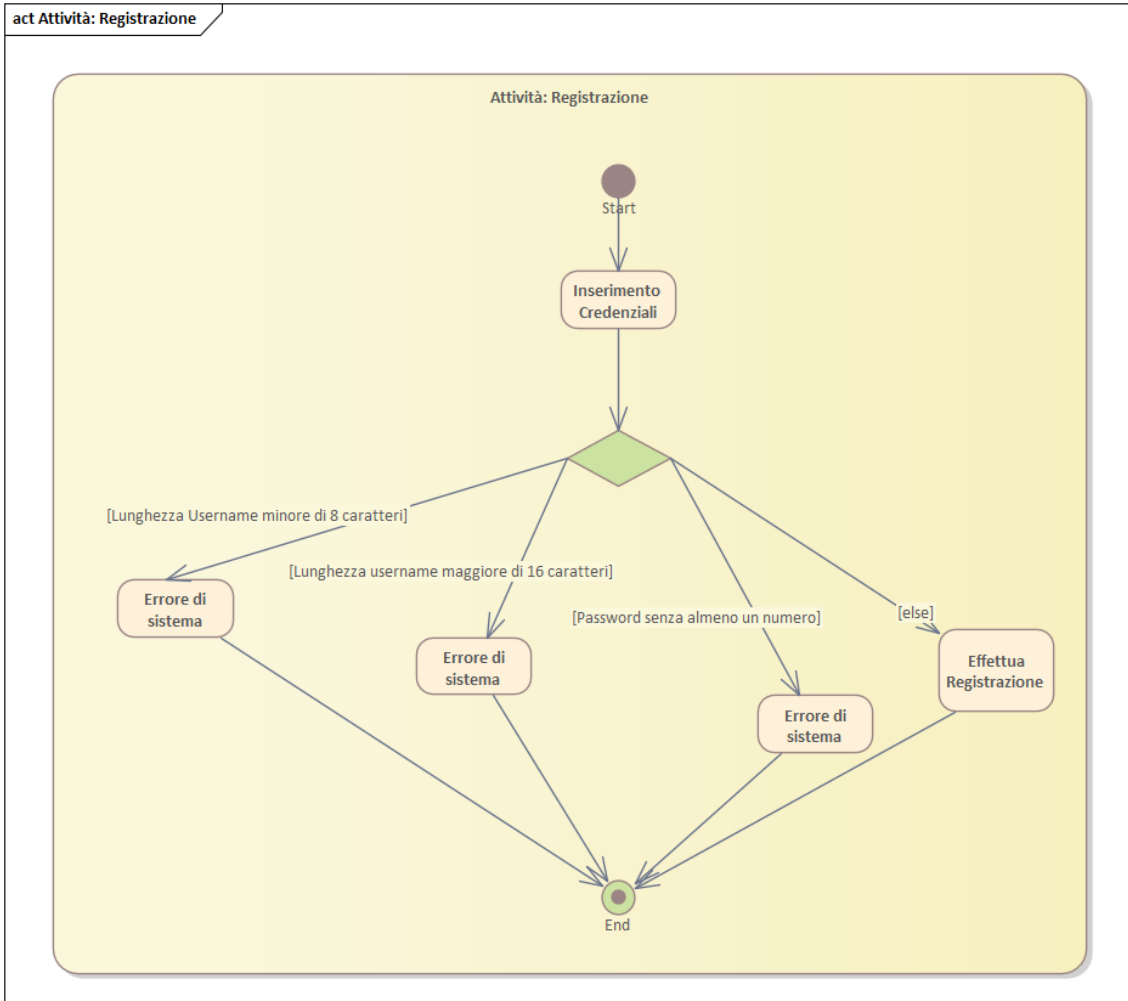


### 3.3 Diagrammi di Attività

#### 3.3.1 Log In

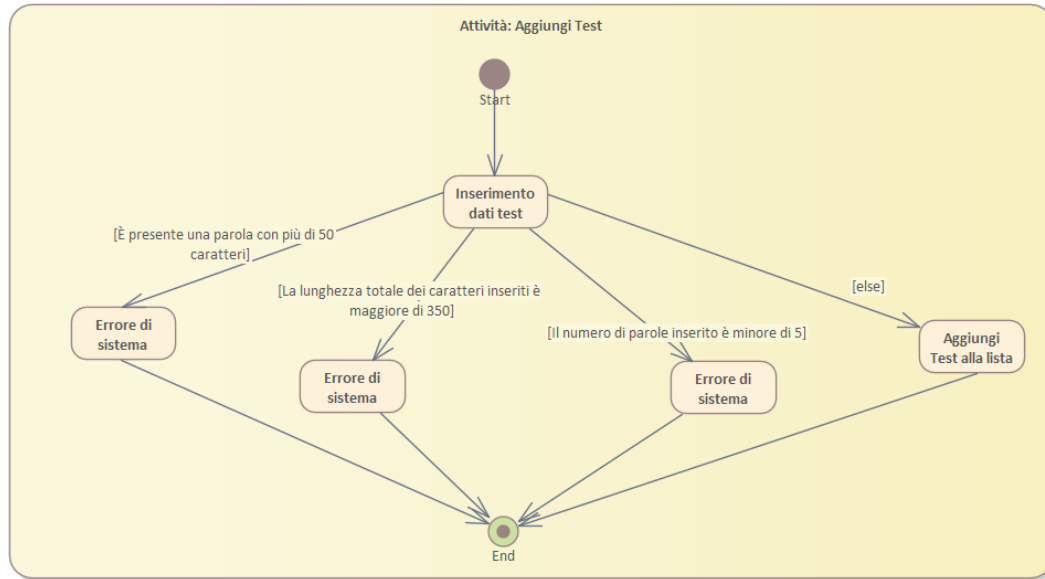


### 3.3.2 Registrazione



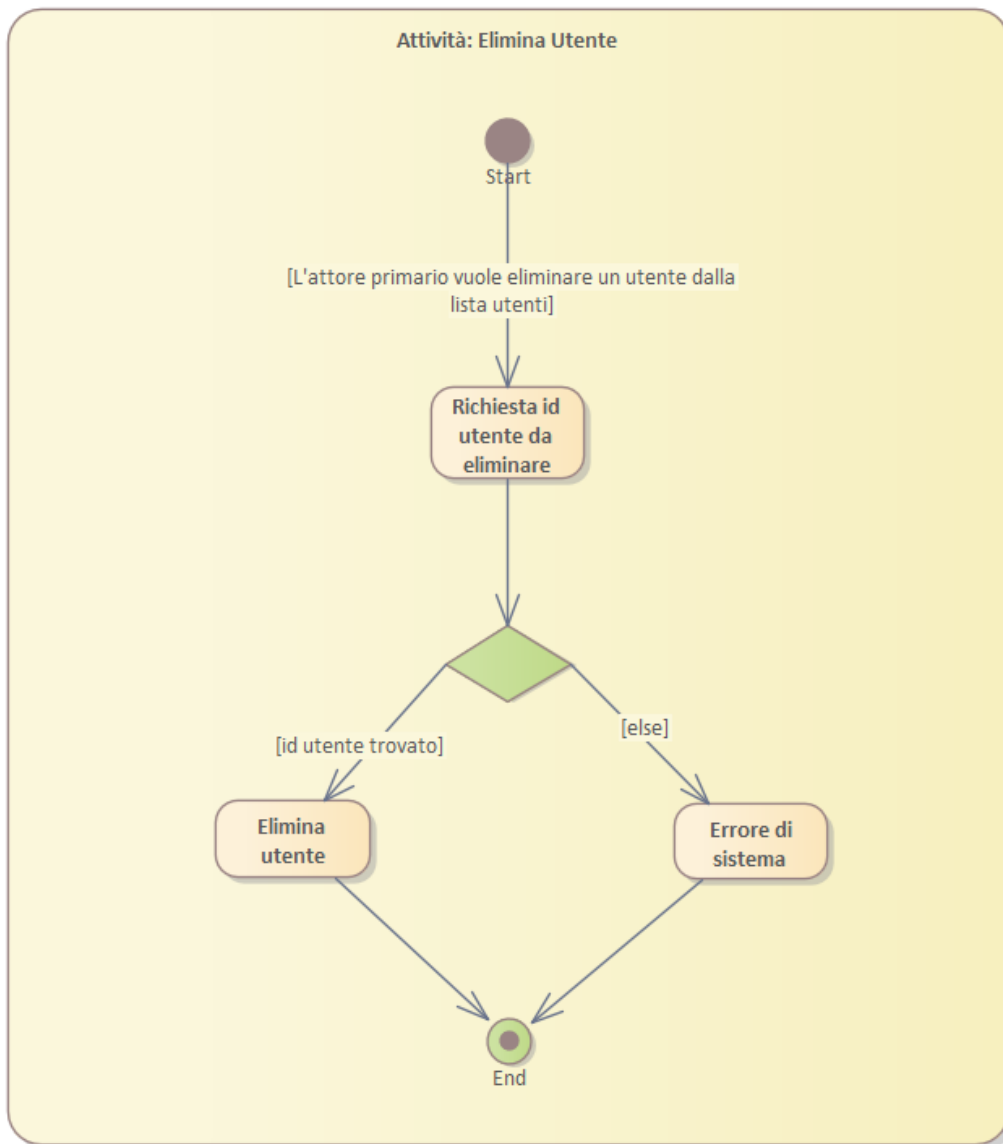
### 3.3.3 Aggiungi Test

act Attività: Aggiungi Test



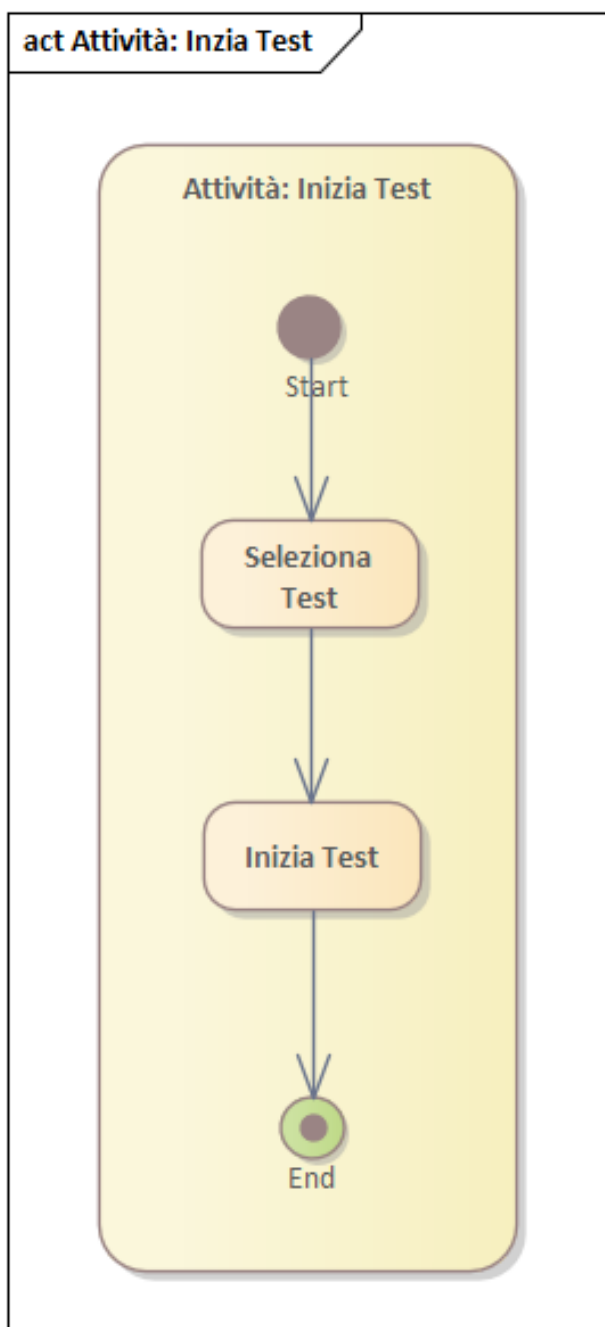
### 3.3.4 Elimina Utente

act Attività: Elimina Utente





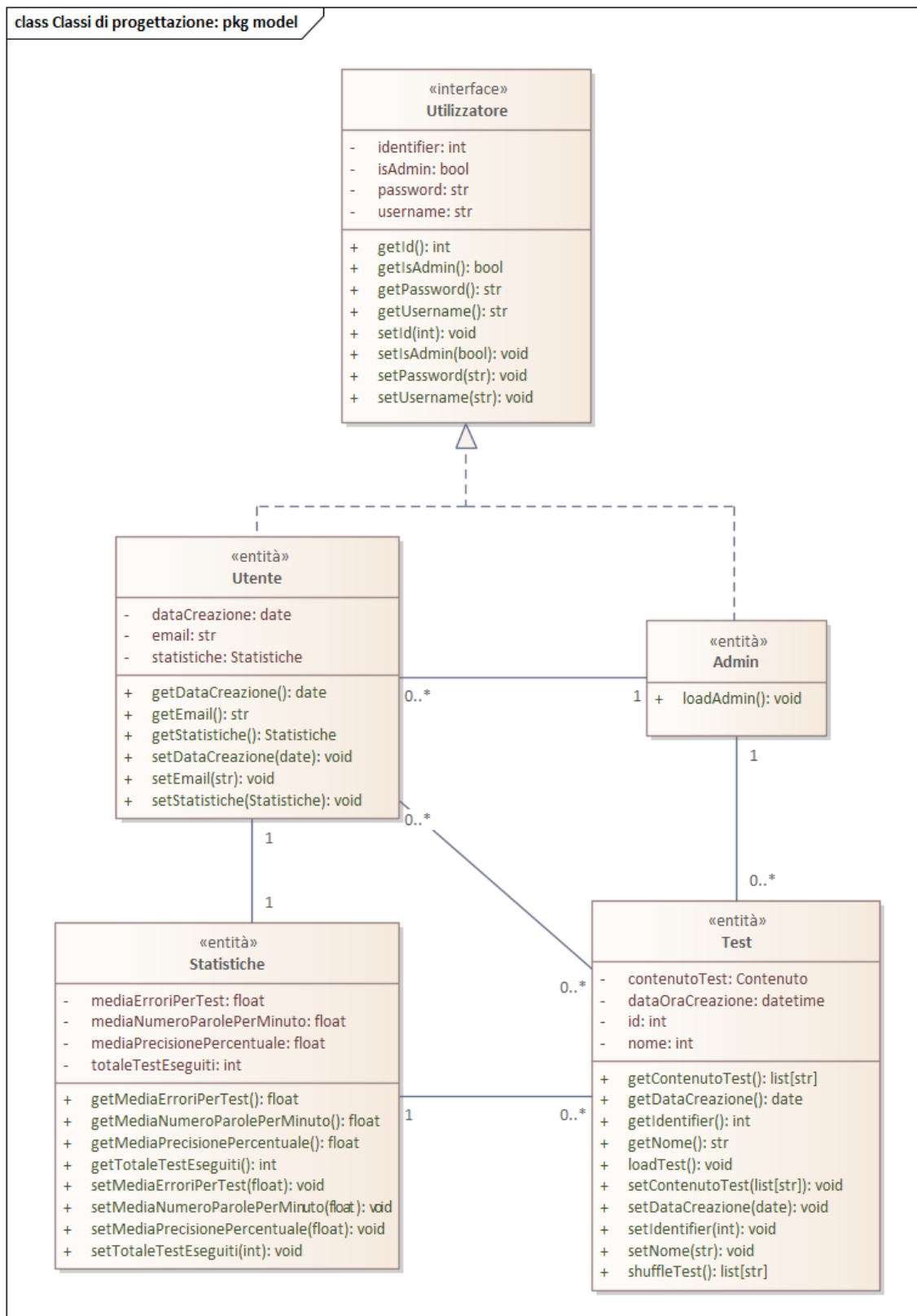
### 3.3.5 Inizia Test



## 4 Diagrammi di Progettazione

Per il flusso di lavoro della progettazione sono state raffinate le classi di analisi includendo una rappresentazione completa degli attributi e dei metodi, sono state create tre macchine a stati.

## 4.1 Diagramma delle Classi di Progettazione



ControllerUtente
<ul style="list-style-type: none"> <li>+ controllaUsernameModificabile(str, list[Utente], Utente): void</li> <li>+ eliminaUtente(int): bool</li> <li>+ modificaProfilo(Utente, str, str, str): void</li> <li>+ modificaUtente(utente, str, str): bool</li> </ul>

ControllerPickle
<ul style="list-style-type: none"> <li>+ caricaListaTest(): void</li> <li>+ caricaListaUtilizzatori(): void</li> <li>+ salvaListaTest(): void</li> <li>+ salvaListaUtilizzatori(): void</li> </ul>

ControllerStatistiche
<ul style="list-style-type: none"> <li>+ aggiornaMediaErroriPerTest(Utente, float): void</li> <li>+ aggiornaMediaNumeroParolePerMinuto(Utente, float): void</li> <li>+ aggiornaMediaPrecisionePercentuale(Utente, float): void</li> <li>+ calcolaNumeroParolePerMinutoTest(int, int): float</li> <li>+ calcolaPrecisionePercentualeTest(int, int): float</li> <li>+ calcolaTotaleTestEseguiti(Utente): void</li> </ul>

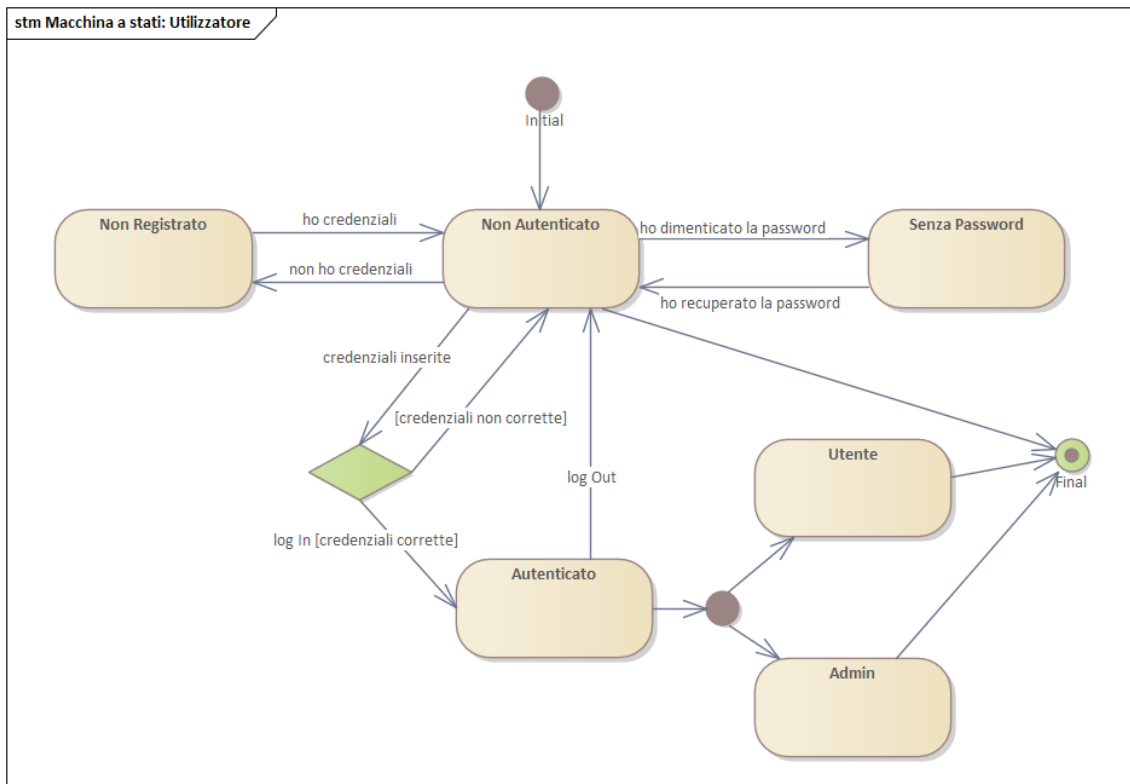
ControllerAutenticazione
<ul style="list-style-type: none"> <li>+ controllaIdentificatoreUnivoco(int): int</li> <li>+ controllaUsernameUnivoco(str): str</li> <li>+ login(str, str): tuple</li> <li>+ recuperaPassword(int, str): bool</li> <li>+ registrazione(): void</li> </ul>

ControllerTest
<ul style="list-style-type: none"> <li>+ aggiungiTest(str, str): void</li> <li>+ assegnaContenutoTest(str): list[str]</li> <li>+ controllaIdentificatoreUnivoco(int): int</li> <li>+ controllaNomeTestModificabile(str, list[Test], Test): void</li> <li>+ controllaNomeUnivoco(str): str</li> <li>+ eliminaTest(int): bool</li> <li>+ modificaTest(Test, str, str): bool</li> </ul>

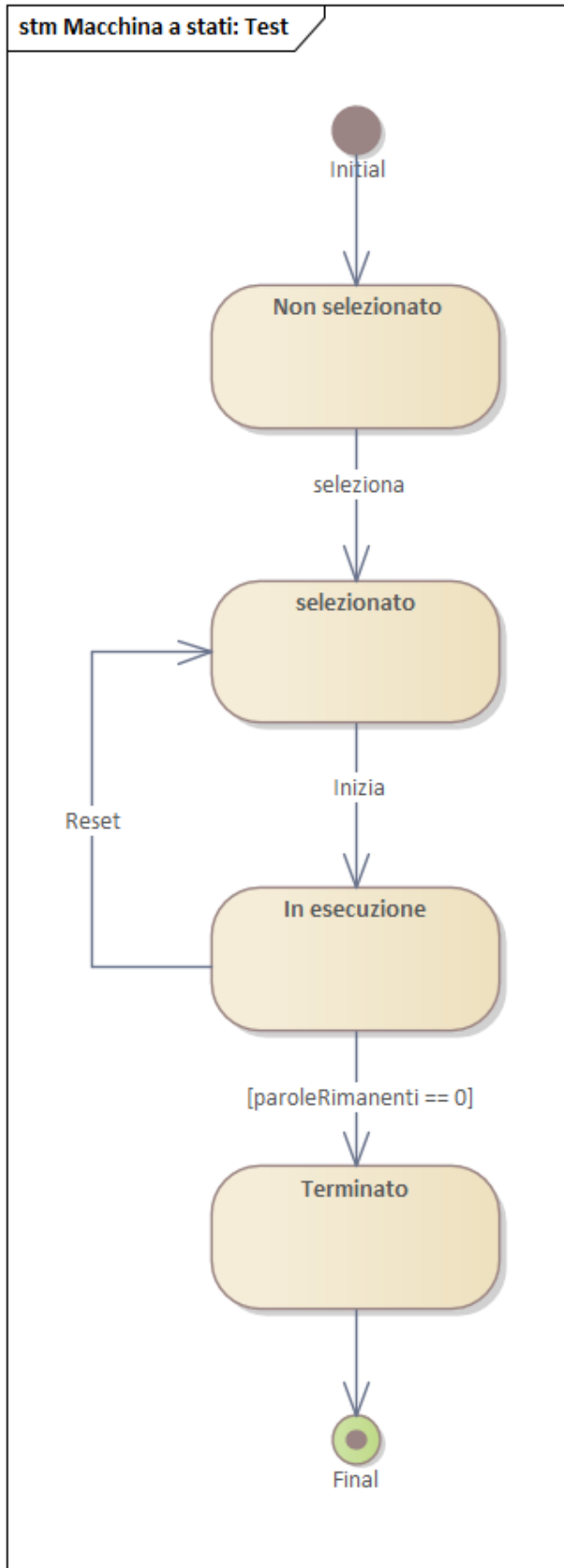
ControllerBackup
<ul style="list-style-type: none"> <li>+ backupDatiTest(str): void</li> <li>+ backupDatiUtenti(str): void</li> <li>+ effettuaBackup(): void</li> </ul>

## 4.2 Diagrammi delle Macchine a Stati

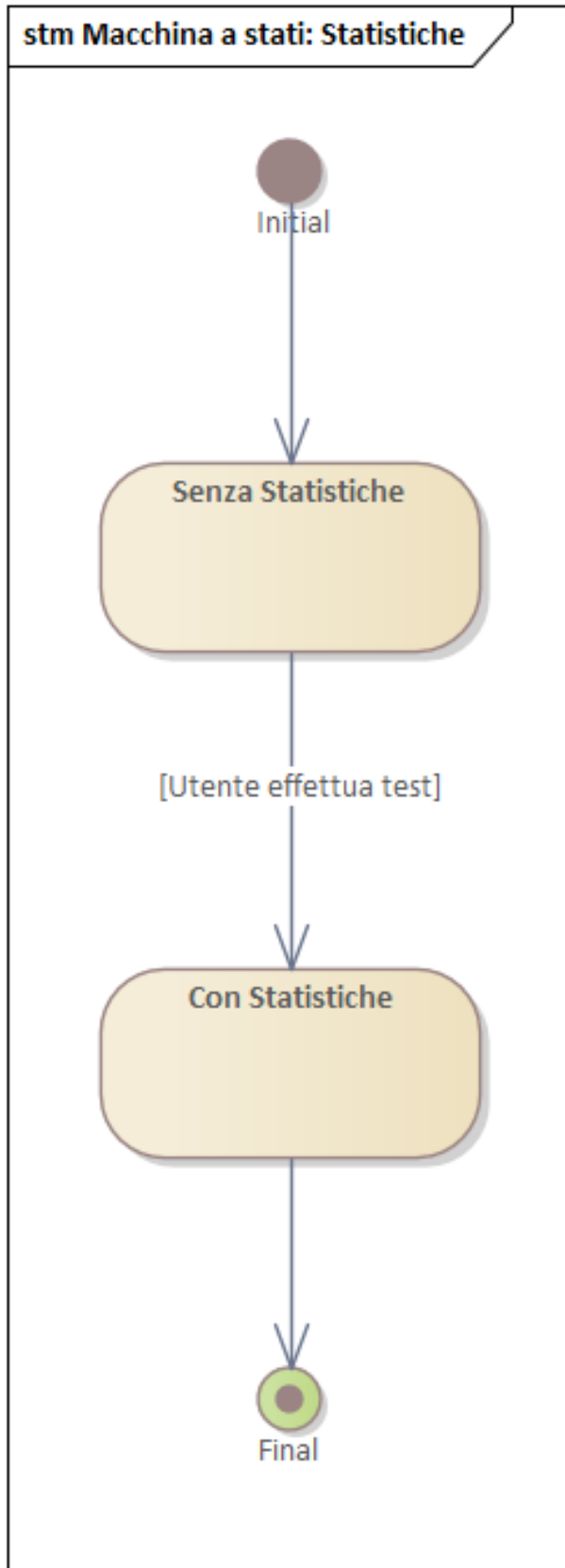
### 4.2.1 Macchina a stati: Utilizzatore



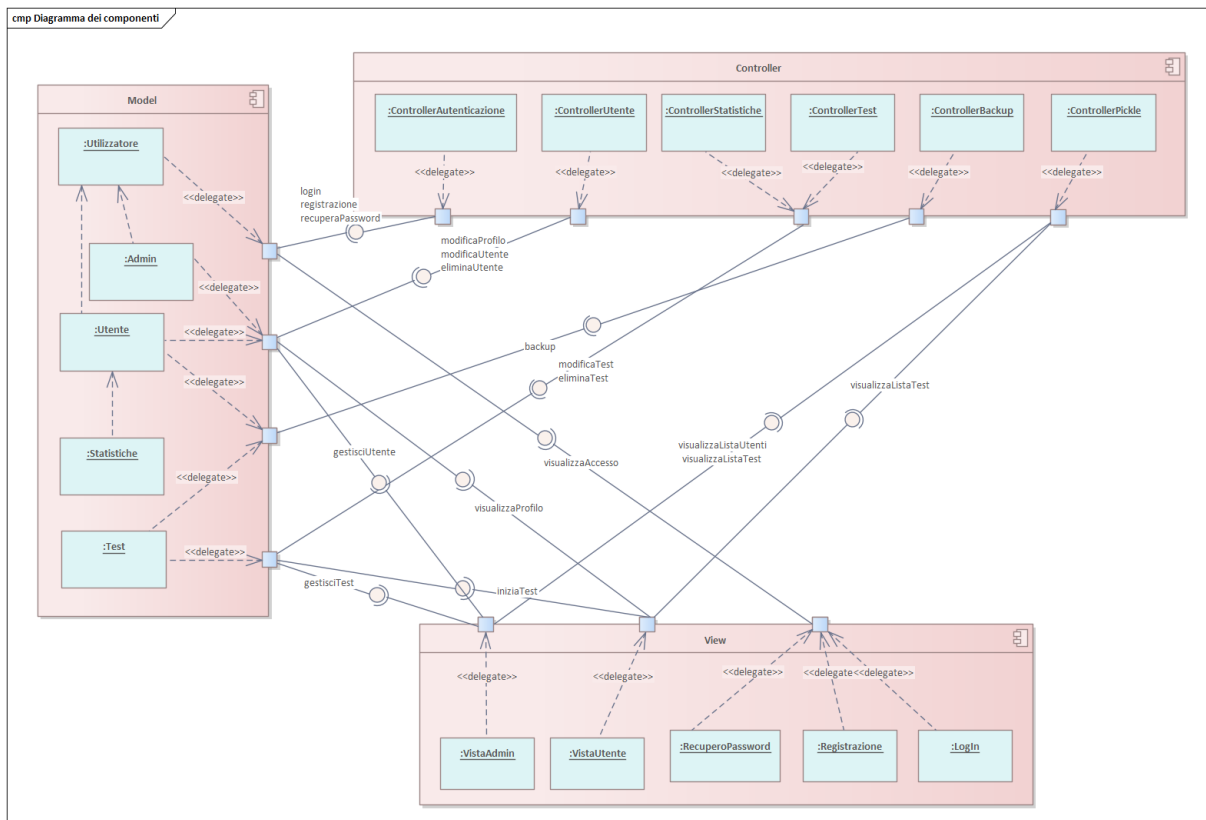
#### 4.2.2 Macchina a stati: Test



#### 4.2.3 Macchina a stati: Statistiche



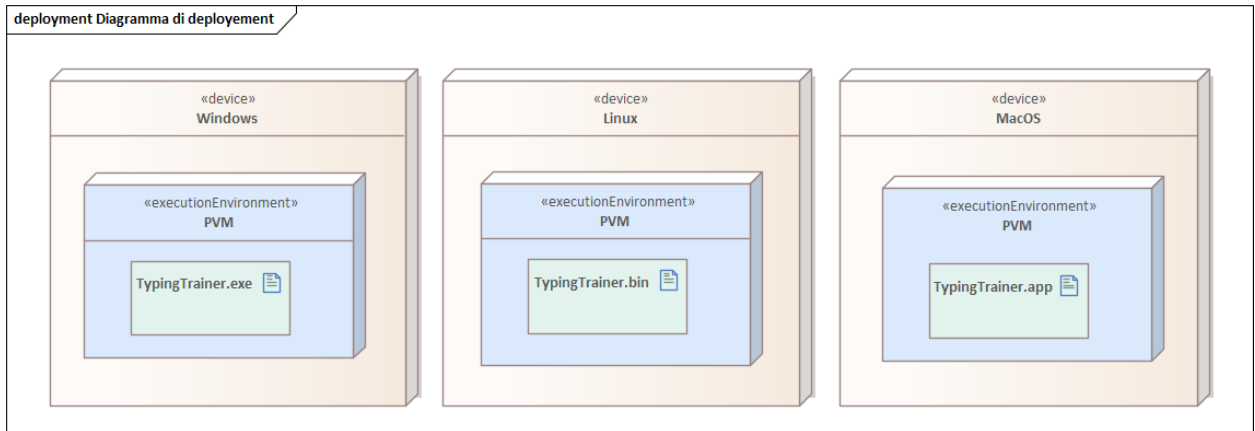
### 4.3 Diagramma dei Componenti





## 5 Implementazione

### 5.1 Diagramma di Deployment



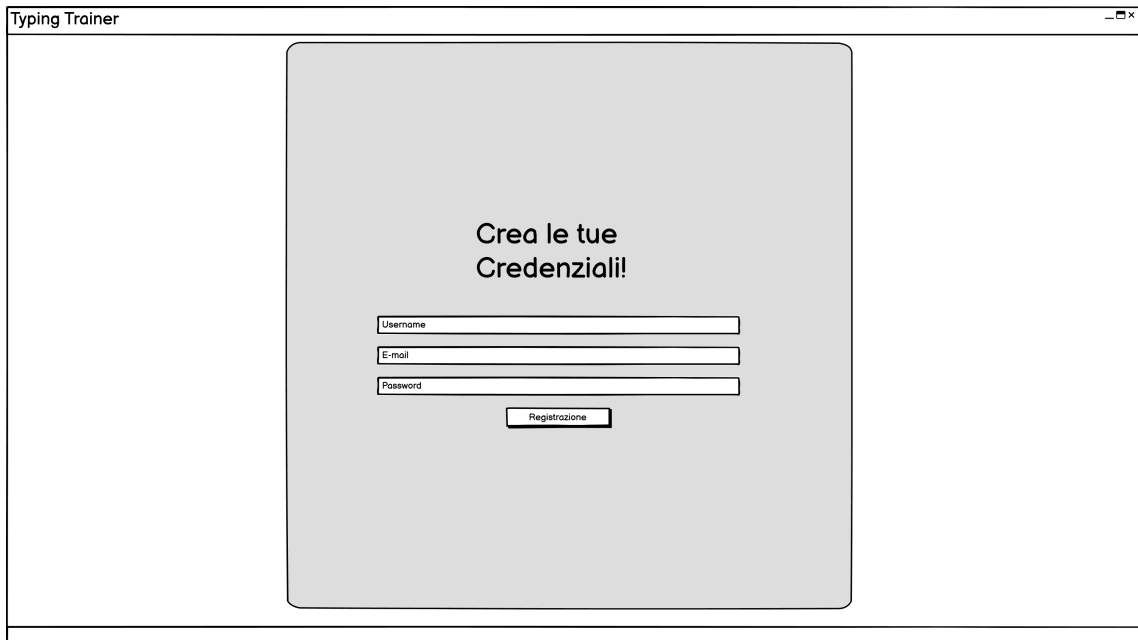
## 5.2 Mockup

Per i mockup è stato usato il software balsamiq wireframes. I mockup servono unicamente da linee guida per visualizzare in modo semplice la struttura dell'interfaccia grafica, per questo motivo potrebbero differire dall'interfaccia finale.

### 5.2.1 LogIn

The mockup shows a web browser window titled "Typing Trainer". Inside the window, there is a light gray rectangular box centered on the page. Within this box, the text "Benvenuto in Typing Trainer!" is displayed in a bold, sans-serif font. Below the text are two input fields: the first is labeled "Username" and the second is labeled "Password". At the bottom of the gray box, there are three buttons: "Log In", "Registrazione", and "Recupero Password". The browser window has a standard title bar with a close button (X) in the top right corner.

## 5.2.2 Registrazione



The image shows a registration form window titled "Typing Trainer". The window has a standard title bar with a close button (X) in the top right corner. The main content area is a light gray rectangle centered on a white background. Inside this rectangle, the text "Crea le tue Credenziali!" is displayed in a bold, black, sans-serif font. Below the text are three input fields, each with a label to its left: "Username", "E-mail", and "Password". The "Username" and "E-mail" labels are in a smaller font, while the "Password" label is in a slightly larger font. Each input field is a simple white rectangle with a thin black border. Below the input fields is a single button labeled "Registrazione" in a small, black, sans-serif font. The button is a white rectangle with a thin black border.

Typing Trainer

Crea le tue  
Credenziali!

Username

E-mail

Password

Registrazione

### 5.2.3 Recupera Password

Typing Trainer

Recupera la  
tua Password!

Username

Nuova Password

Recupera Password

### 5.2.4 Landing Page Utente

Typing Trainer

Benvenuto in  
Typing Trainer!

Username

🔍

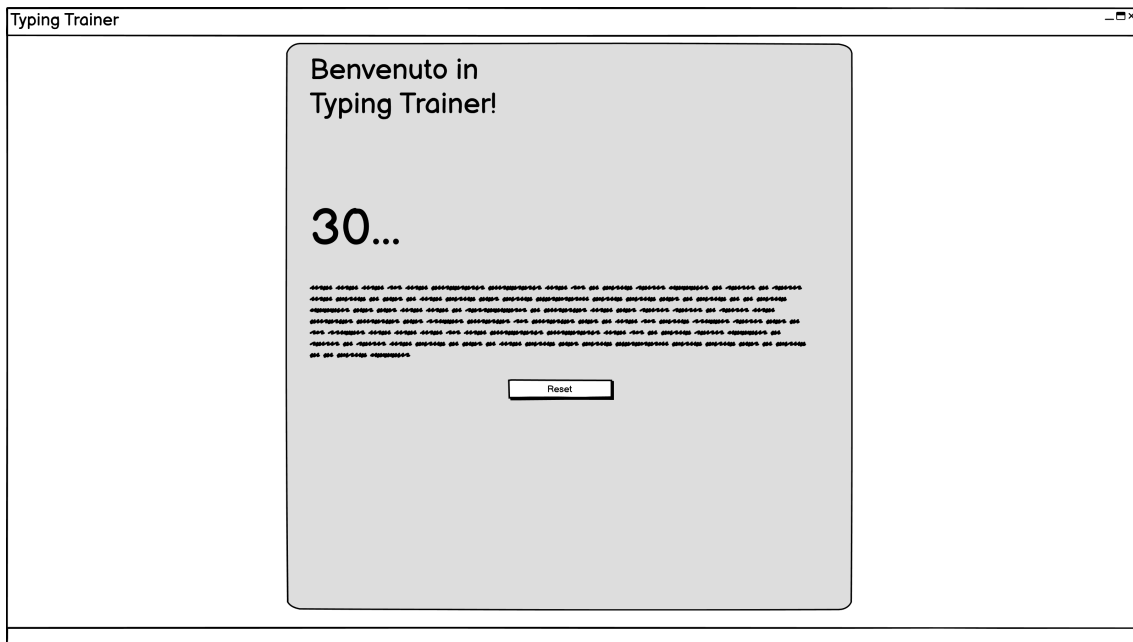
Inserisci nome oppure ID

Cerca

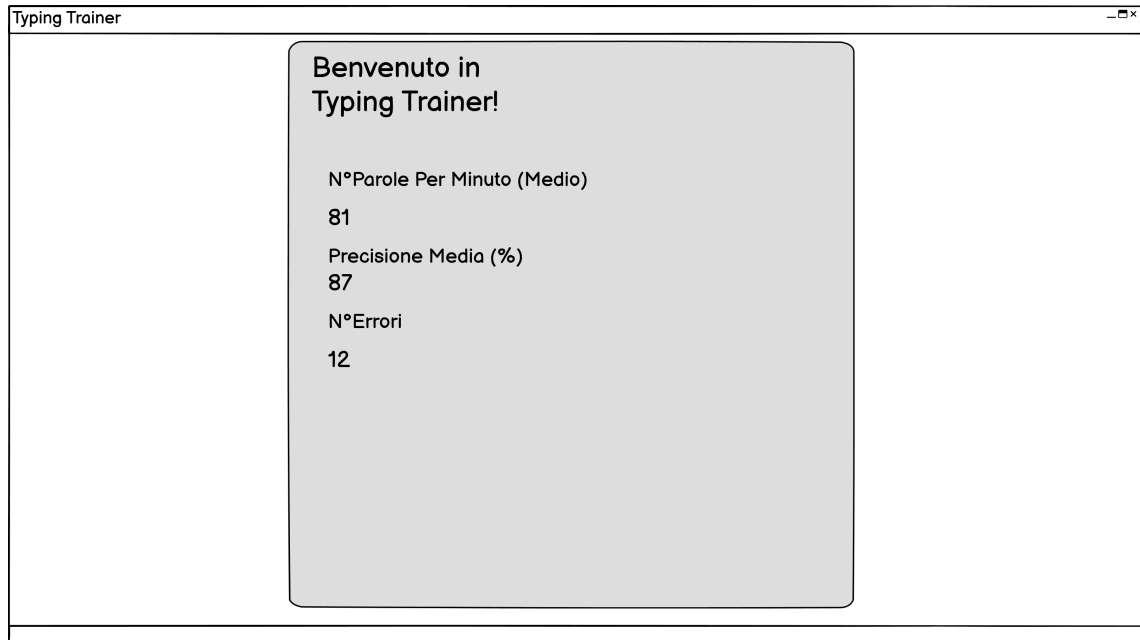
ID	Nome	Data creazione	Selezione
1	Test 1	YYYY/MM/DD	<input type="checkbox"/>
2	Test 2	YYYY/MM/DD	<input type="checkbox"/>
3	Test 3	YYYY/MM/DD	<input type="checkbox"/>
4	Test 4	YYYY/MM/DD	<input checked="" type="checkbox"/>
5	Test 5	YYYY/MM/DD	<input type="checkbox"/>
6	Test 6	YYYY/MM/DD	<input type="checkbox"/>
7	Test 7	YYYY/MM/DD	<input type="checkbox"/>
8	Test 8	YYYY/MM/DD	<input type="checkbox"/>
9	Test 9	YYYY/MM/DD	<input type="checkbox"/>
10	Test 10	YYYY/MM/DD	<input type="checkbox"/>
11	Test 11	YYYY/MM/DD	<input type="checkbox"/>
12	Test 12	YYYY/MM/DD	<input type="checkbox"/>
13	Test 13	YYYY/MM/DD	<input type="checkbox"/>
14	Test 14	YYYY/MM/DD	<input type="checkbox"/>
15	Test 15	YYYY/MM/DD	<input type="checkbox"/>
16	Test 16	YYYY/MM/DD	<input type="checkbox"/>
17	Test 17	YYYY/MM/DD	<input type="checkbox"/>
18	Test 18	YYYY/MM/DD	<input type="checkbox"/>

Inizia Test

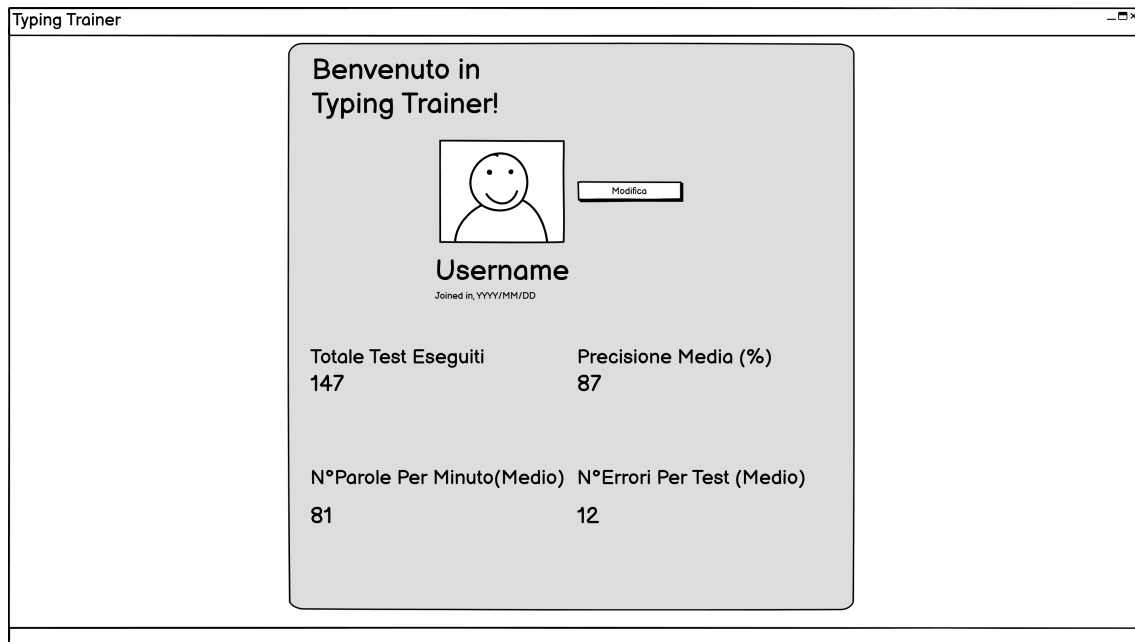
### 5.2.5 Inizia Test



### 5.2.6 Visualizza Statistiche



### 5.2.7 Visualizza Profilo





### 5.2.8 Modifica Profilo

Typing Trainer

Benvenuto in  
Typing Trainer!

Modifica le tue  
Credenziali!

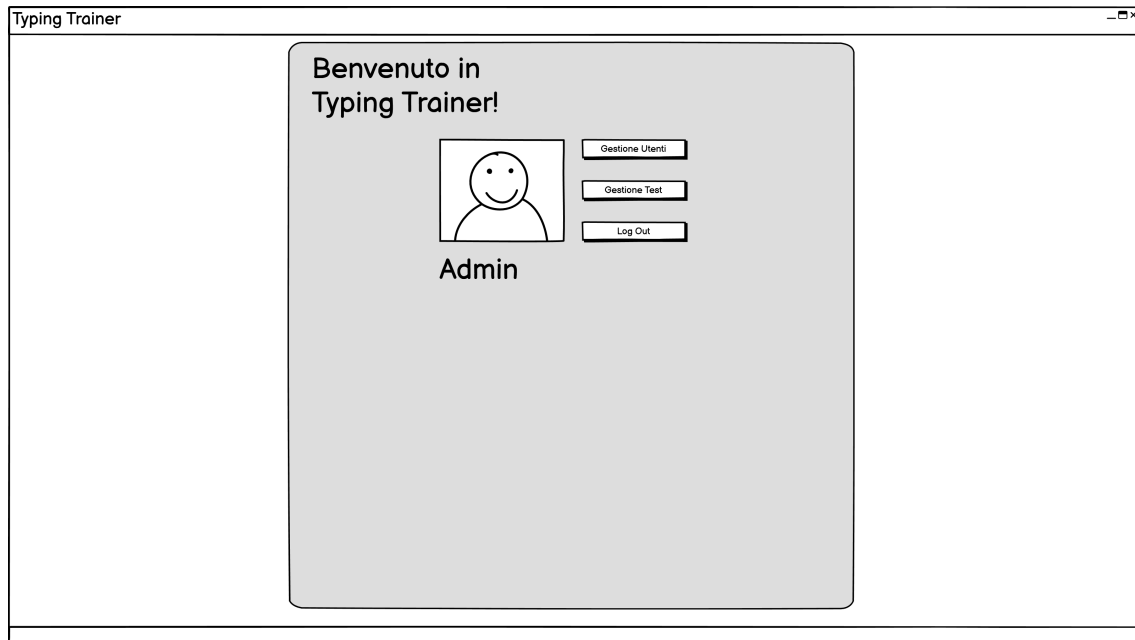
Username

E-mail

Password

Modifica

### 5.2.9 Landing Page Admin



### 5.2.10 Gestione Test

Typing Trainer

Benvenuto in  
Typing Trainer!

Inserisci nome oppure ID

Cerca

+

Aggiungi

ID	Nome	Data creazione	Modifica	Elimina
1	Test 1	YYYY/MM/DD	<a href="#">modifica</a>	<a href="#">elimina</a>
2	Test 2	YYYY/MM/DD	<a href="#">modifica</a>	<a href="#">elimina</a>
3	Test 3	YYYY/MM/DD	<a href="#">modifica</a>	<a href="#">elimina</a>
4	Test 4	YYYY/MM/DD	<a href="#">modifica</a>	<a href="#">elimina</a>
5	Test 5	YYYY/MM/DD	<a href="#">modifica</a>	<a href="#">elimina</a>
6	Test 6	YYYY/MM/DD	<a href="#">modifica</a>	<a href="#">elimina</a>
7	Test 7	YYYY/MM/DD	<a href="#">modifica</a>	<a href="#">elimina</a>
8	Test 8	YYYY/MM/DD	<a href="#">modifica</a>	<a href="#">elimina</a>
9	Test 9	YYYY/MM/DD	<a href="#">modifica</a>	<a href="#">elimina</a>
10	Test 10	YYYY/MM/DD	<a href="#">modifica</a>	<a href="#">elimina</a>
11	Test 11	YYYY/MM/DD	<a href="#">modifica</a>	<a href="#">elimina</a>
12	Test 12	YYYY/MM/DD	<a href="#">modifica</a>	<a href="#">elimina</a>
13	Test 13	YYYY/MM/DD	<a href="#">modifica</a>	<a href="#">elimina</a>
14	Test 14	YYYY/MM/DD	<a href="#">modifica</a>	<a href="#">elimina</a>
15	Test 15	YYYY/MM/DD	<a href="#">modifica</a>	<a href="#">elimina</a>
16	Test 16	YYYY/MM/DD	<a href="#">modifica</a>	<a href="#">elimina</a>
17	Test 17	YYYY/MM/DD	<a href="#">modifica</a>	<a href="#">elimina</a>
18	Test 18	YYYY/MM/DD	<a href="#">modifica</a>	<a href="#">elimina</a>

### 5.2.11 Aggiungi Test

Typing Trainer

Benvenuto in  
Typing Trainer!

Aggiungi il  
nuovo test

Nome

Contenuto

Aggiungi

### 5.2.12 Modifica Test

Typing Trainer

Benvenuto in  
Typing Trainer!

Modifica il test

Nome

Contenuto

Modifica

### 5.2.13 Gestione Utenti

Typing Trainer

Benvenuto in  
Typing Trainer!

Admin

Inserisci nome oppure ID

Cerca

ID	Nome	Data creazione	Modifica	Elimina
1	Username1	YYYY/MM/DD	<a href="#">modifica</a>	<a href="#">elimina</a>
2	Username2	YYYY/MM/DD	<a href="#">modifica</a>	<a href="#">elimina</a>
3	Username3	YYYY/MM/DD	<a href="#">modifica</a>	<a href="#">elimina</a>
4	Username4	YYYY/MM/DD	<a href="#">modifica</a>	<a href="#">elimina</a>
5	Username5	YYYY/MM/DD	<a href="#">modifica</a>	<a href="#">elimina</a>
6	Username6	YYYY/MM/DD	<a href="#">modifica</a>	<a href="#">elimina</a>
7	Username7	YYYY/MM/DD	<a href="#">modifica</a>	<a href="#">elimina</a>
8	Username8	YYYY/MM/DD	<a href="#">modifica</a>	<a href="#">elimina</a>
9	Username9	YYYY/MM/DD	<a href="#">modifica</a>	<a href="#">elimina</a>
10	Username10	YYYY/MM/DD	<a href="#">modifica</a>	<a href="#">elimina</a>
11	Username11	YYYY/MM/DD	<a href="#">modifica</a>	<a href="#">elimina</a>
12	Username12	YYYY/MM/DD	<a href="#">modifica</a>	<a href="#">elimina</a>
13	Username13	YYYY/MM/DD	<a href="#">modifica</a>	<a href="#">elimina</a>
14	Username14	YYYY/MM/DD	<a href="#">modifica</a>	<a href="#">elimina</a>
15	Username15	YYYY/MM/DD	<a href="#">modifica</a>	<a href="#">elimina</a>
16	Username16	YYYY/MM/DD	<a href="#">modifica</a>	<a href="#">elimina</a>
17	Username17	YYYY/MM/DD	<a href="#">modifica</a>	<a href="#">elimina</a>
18	Username18	YYYY/MM/DD	<a href="#">modifica</a>	<a href="#">elimina</a>

### 5.2.14 Modifica Utente

Typing Trainer

Benvenuto in  
Typing Trainer!

Modifica le  
Credenziali!

Username

E-mail

Modifica

## 5.3 Descrizione Unit Tests

Per verificare l'integrità del codice sono stati creati alcuni UnitTest tramite l'utilizzo di PyUnit.

### 5.3.1 Test Accesso

```
1 import unittest
2 from Controller.ControllerPickle import ControllerPickle
3 from Model import Utente
4
5 class MyTestCase(unittest.TestCase):
6     def setUp(self):
7         self.utente_normale = Utente.Utente()
8         self.utente_normale.setUsername('username1')
9         self.utente_normale.setPassword('pass1')
10        self.utente_normale.setIsAdmin(False)
11
12        self.utente_admin = Utente.Utente()
13        self.utente_admin.setUsername('administrator')
14        self.utente_admin.setPassword('adminpass0')
15        self.utente_admin.setIsAdmin(True)
16
17        self.utente_sbagliato = Utente.Utente()
18        self.utente_sbagliato.setUsername('username2')
19        self.utente_sbagliato.setPassword('pass2')
20        self.utente_sbagliato.setIsAdmin(False)
21
22        self.miaListaUtilizzatori = [
23            self.utente_normale,
24            self.utente_admin,
25            self.utente_sbagliato
26        ]
27
28    def login(self, username, password):
29        for i in self.miaListaUtilizzatori:
30            if username == i.username and password == i.password and i.isAdmin ==
31                False:
32                return "Utente", i
33            elif username == i.username and password == i.password and i.isAdmin
34                == True:
35                return "Admin", i
36            elif username == i.username:
37                return "PasswordErrata", "errore"
38        return "UsernameNonTrovato", "errore"
39
40    def test_login_utente(self):
41        result = self.login('username1', 'pass1')
42        self.assertEqual(result, ("Utente", self.utente_normale))
43
44    def test_login_admin(self):
45        result = self.login('administrator', 'adminpass0')
46        self.assertEqual(result, ("Admin", self.utente_admin))
47
48    def test_login_password_errata(self):
49        result = self.login('username2', 'wrongpass1')
50        self.assertEqual(result, ("PasswordErrata", "errore"))
51
52    def test_login_username_non_trovato(self):
53        result = self.login('nonexistent', 'nopass1')
54        self.assertEqual(result, ("UsernameNonTrovato", "errore"))
55
56    def testCaricaListaUtilizzatori(self):
57        controllerPickle = ControllerPickle()
58        controllerPickle.caricaListaUtilizzatori()
59        listaUtilizzatori = controllerPickle.listaUtilizzatori
60        self.assertIsNotNone(listaUtilizzatori)
61
62    if __name__ == '__main__':
63        unittest.main()
```



### 5.3.2 Test Administrator

```
import unittest

from Model import Test

class TestAdministrator(unittest.TestCase):
    def setUp(self):
        self.test1 = Test.Test()
        self.test1.setIdentifier(1)

        self.test2 = Test.Test()
        self.test2.setIdentifier(2)

        self.mialistaTest = [
            self.test1,
            self.test2
        ]
    def eliminaTest(self, identifier):
        try:
            for i in self.mialistaTest:
                if identifier == i.getIdentifier():
                    self.mialistaTest.remove(i)
            return True
        except Exception:
            return False

    def testEliminaTest(self):
        result = self.eliminaTest(1)
        self.assertEqual(result, True)

if __name__ == '__main__':
    unittest.main()
```

### 5.3.3 Test Utente

```
import unittest
from Model import Utente

class TestUtente(unittest.TestCase):
    def setUp(self):
        self.utente_normale = Utente.Utente()
        self.utente_normale.setIdentifier(0)
        self.utente_normale.setUsername('username1')
        self.utente_normale.setPassword('pass1')
        self.utente_normale.setEmail('utentenormale@gmail.com')
        self.utente_normale.setIsAdmin(False)

        self.mialistaUtilizzatori = [
            self.utente_normale
        ]

    def modificaProfilo(self, utenteDaModificare, nuovoUsername, nuovaEmail):
        try:
            for utente in self.mialistaUtilizzatori:
                if utenteDaModificare.getIdentifier() == utente.getIdentifier():
                    utente.setUsername(nuovoUsername)
                    utente.setEmail(nuovaEmail)
            return True
        except Exception:
            return False

    def testModificaProfilo(self):
        result = self.modificaProfilo(self.utente_normale, 'usernameMod', 'emailmodificata@gmail.com')
        self.assertEqual(result, True)

if __name__ == '__main__':
    unittest.main()
```