



## ***“SISTEMA SOFTWARE PER LA CONDIVISIONE DI RICETTE ONLINE”***

# INDICE

## **1. SPECIFICHE INFORMALI1**

## **2. ANALISI E SPECIFICA DEI REQUISITI2**

- 2.1 ANALISI NOMI-VERBI2
- 2.2 REVISIONE DEI REQUISITI3
- 2.3 GLOSSARIO DEI TERMINI3
- 2.4 CLASSIFICAZIONE DEI REQUISITI4
  - 2.4.1 Requisiti funzionali4
  - 2.4.2 Requisiti sui dati5
  - 2.4.3 Vincoli / Altri requisiti5
- 2.5 MODELLAZIONE DEI CASI D'USO6
  - 2.5.1 Attori e casi d'uso6
  - 2.5.2 Diagramma dei casi d'uso8
  - 2.5.3 Scenari8
- 2.6 DIAGRAMMA DELLE CLASSI16
  - 2.6.1 Responsabilità**Errore. Il segnalibro non è definito.**
- 2.7 DIAGRAMMI DI SEQUENZA19
  - 2.7.1 Registrazione**Errore. Il segnalibro non è definito.**
  - 2.7.2 Accesso**Errore. Il segnalibro non è definito.**
  - 2.7.3 AggiungiAuto**Errore. Il segnalibro non è definito.**

## **3. PIANO DI TEST FUNZIONALE23**

- 3.1 REGISTRAZIONE**ERRORE. IL SEGNALIBRO NON È DEFINITO.**

## **4. PROGETTAZIONE32**

- 4.1 DIAGRAMMA DELLE CLASSI32
  - 4.1.1 Traduzione classi ed associazioni33
  - 4.1.2 Pattern BCED34
    - 4.1.2.1 Package Boundary34
    - 4.1.2.2 Package Controller35
    - 4.1.2.3 Package Entity35
    - 4.1.2.4 Package Database37
- 4.2 DIAGRAMMI DI SEQUENZA38
  - 4.2.1 Registrazione39
  - 4.2.2 Accesso**Errore. Il segnalibro non è definito.**

## **5. IMPLEMENTAZIONE42**

- 5.1 PACKAGE DATABASE42
- 5.2 PACKAGE ENTITY43
- 5.3 PACKAGE CONTROLLER43
- 5.4 PACKAGE BOUNDARY43
- 5.5 PACKAGE DTO**ERRORE. IL SEGNALIBRO NON È DEFINITO.**
- 5.6 DIAGRAMMA DI DEPLOYMENT44

## **6. TESTING45**

- 6.1 TEST STRUTTURALE45
  - 6.1.1 Complessità ciclomatica45
    - 6.1.1.1 inserisciAutoModifiche – GestioneParcoAuto45
- 6.2 JUNIT – TEST DI UNITÀ48
- 6.3 TEST FUNZIONALE48

# 1. Specifiche informali

## *Riportare la Traccia assegnata*

### **Traccia: Piattaforma di Condivisione di Ricette**

Si desidera sviluppare un sistema software per la gestione di una piattaforma sociale dedicata alla condivisione di ricette culinarie semplici e veloci. La piattaforma è pensata per permettere agli utenti di pubblicare le proprie ricette, organizzarle in raccolte tematiche e interagire con i contenuti pubblicati dagli altri utenti attraverso apprezzamenti e commenti.

### **Descrizione del Sistema:**

Il sistema consente la registrazione degli utenti tramite l'inserimento di e-mail, nome, cognome e password. Ogni utente dispone di un profilo personale modificabile che include i propri dati anagrafici, una breve biografia ed eventualmente un'immagine del profilo. Dopo l'autenticazione, l'utente può pubblicare nuove ricette tramite un'apposita interfaccia grafica. Ogni ricetta è caratterizzata da un titolo, una lista di ingredienti, una descrizione dei passaggi (con un limite massimo di 800 caratteri), un tempo stimato di preparazione espresso in minuti, uno o più tag tematici (ad esempio "vegetariana", "dolci", "pranzo veloce"). L'utente può scegliere se aggiungere la ricetta a una raccolta già esistente oppure crearne una nuova al momento della pubblicazione.

Ogni raccolta è composta da ricette dello stesso autore, è identificata da un titolo e da una descrizione e può contenere ricette pubbliche o private. Le ricette pubbliche sono visibili nel feed principale degli utenti.

Una volta autenticato, ciascun utente può visualizzare nel proprio feed personale un insieme di nuove ricette pubblicate da altri utenti. Il feed mostra al massimo cinque ricette, ordinate in ordine cronologico decrescente, selezionate tra le ultime pubblicazioni effettuate da altri autori.

Gli utenti hanno la possibilità di interagire con le ricette pubblicate da altri tramite un sistema di apprezzamenti (rappresentati da un "like") e con l'aggiunta di commenti testuali. Ogni commento riporta l'autore, il testo e la data di pubblicazione. Quando una ricetta viene visualizzata, il sistema mostra anche il numero totale di like ricevuti e i tre commenti più recenti associati.

La piattaforma offre a ogni utente una sezione dedicata alle statistiche personali. In tale sezione l'utente può visualizzare il numero complessivo di like ricevuti dalle proprie ricette, il numero totale di commenti ottenuti e la ricetta più apprezzata tra quelle pubblicate, ossia quella con il numero più alto di like.

Gli amministratori della piattaforma, previa autenticazione, possono accedere a funzionalità di reportistica. Tali funzionalità consentono di generare report contenenti dati aggregati come il numero totale di ricette pubblicate in un determinato intervallo temporale, l'elenco degli autori più attivi in base al numero di ricette inserite, i tag tematici più utilizzati e le ricette con il maggior numero di interazioni.

Il sistema dovrà essere accessibile sia da dispositivi desktop che mobili, prevedere un sistema di notifiche in tempo reale per segnalare all'utente l'arrivo di nuovi commenti o apprezzamenti, garantire la protezione dei dati personali, la sicurezza delle informazioni archiviate e l'adozione di un meccanismo di autenticazione robusto e affidabile.



## 2. Analisi e specifica dei requisiti

### 2.1 Analisi nomi-verbi

Il sistema consente la registrazione degli **utenti** tramite l'inserimento **di e-mail, nome, cognome e password**. Ogni utente dispone di un **profilo personale** modificabile che include i propri **dati anagrafici, una breve biografia ed eventualmente un'immagine del profilo**. Dopo l'autenticazione, **l'utente può pubblicare nuove ricette tramite un'apposita interfaccia grafica**. Ogni **ricetta** è **caratterizzata da un titolo, una lista di ingredienti, una descrizione dei passaggi (con un limite massimo di 800 caratteri), un tempo stimato di preparazione espresso in minuti, uno o più tag tematici (ad esempio "vegetariana", "dolci", "pranzo veloce")**. **L'utente può scegliere se aggiungere la ricetta a una raccolta già esistente oppure crearne una nuova al momento della pubblicazione**.

Ogni **raccolta** è composta da ricette dello stesso autore, è identificata da un titolo e da una descrizione e può **contenere ricette pubbliche o private**. Le ricette pubbliche sono visibili nel feed principale degli utenti.

Una volta autenticato, **ciascun utente può visualizzare nel proprio feed personale un insieme di nuove ricette pubblicate da altri utenti**. Il **feed** mostra al massimo cinque ricette, ordinate in ordine cronologico decrescente, **selezionate tra le ultime pubblicazioni effettuate da altri autori**.

Gli utenti hanno la possibilità di interagire con le ricette pubblicate da altri tramite un sistema di apprezzamenti (rappresentati da un "like") e con l'aggiunta di commenti testuali. Ogni **commento riporta l'autore, il testo e la data di pubblicazione**. Quando una ricetta viene visualizzata, il sistema mostra anche il numero totale di like ricevuti e i tre commenti più recenti associati.

La piattaforma offre a ogni utente una sezione dedicata alle statistiche personali. In tale sezione l'utente può visualizzare il numero complessivo di like ricevuti dalle proprie ricette, il numero totale di commenti ottenuti e la ricetta più apprezzata tra quelle pubblicate, ossia quella con il numero più alto di like.

Gli **amministratori** della piattaforma, previa autenticazione, possono accedere a funzionalità di reportistica. Tali funzionalità consentono di generare report contenenti dati aggregati come il numero totale di ricette pubblicate in un determinato intervallo temporale, l'elenco degli autori più attivi in base al numero di ricette inserite, i tag tematici più utilizzati e le ricette con il maggior numero di interazioni.

Il sistema dovrà essere accessibile sia da dispositivi desktop che mobili, prevedere un sistema di notifiche in tempo reale per segnalare all'utente l'arrivo di nuovi commenti o apprezzamenti, **garantire la protezione dei dati personali, la sicurezza delle informazioni archiviate e l'adozione di un meccanismo di autenticazione robusto e affidabile**.

#### LEGENDA:

**Classe**

**Attributo**

**Funzionalità**

**Attore**

**Classe-Attore**



## 2.2 Revisione dei requisiti

1. il sistema consente la registrazione degli utenti con e-mail, nome, cognome, password.
2. Il sistema deve permettere all'utente l'autenticazione.
3. Il sistema deve permettere all'Utente autenticato di modificare il proprio profilo personale, includendo biografia e immagine.
4. dopo l'autenticazione l'utente può pubblicare nuove ricette.
5. ogni ricetta è caratterizzata da titolo, lista ingrediente, descrizione di passaggi con max di 800 caratteri, tempo stimato di preparazione in minuti, tag tematici tipo vegetariano, dolci e...
6. Il sistema offre all'utente la possibilità di scegliere se aggiungere la ricetta ad una raccolta già esistente oppure crearne una nuova.
7. ogni raccolta è composta da ricette dello stesso autore identificate da un titolo, descrizione e può contenere ricette pubbliche, **le quali possono essere visibili nel feed principale degli utenti**, o private
8. dopo l'autenticazione il sistema permette a l'utente di visualizzare nel proprio feed un insieme di nuove ricette pubblicate da nuovi utenti nel quale è possibile visualizzare al max 5 ricette e ordinate in ordine cronologico decrescente.
9. gli utenti possono interagire con le ricette tramite un commento o like.
10. ogni commento è caratterizzato da un autore, testo, data
11. quando una ricetta viene visualizzata il sistema mostra il numero totale di like ricevuti e tre commenti più recenti
12. il sistema offre a ogni utente una sezione per le statistiche personali in cui l'utente può visualizzare il numero complessivo di like ricevuti e il numero totale di commenti per le proprie ricette e la più apprezzata quelle con più like
13. gli amministratori previa autenticazione possono accedere a funzionalità di reportistica, per la generazione di report
14. il sistema deve essere accessibile sia da desktop che da dispositivi mobile
15. il sistema deve prevedere un sistema di notifiche in tempo reale in modo da notificare l'utente di nuovi apprezzamenti e commenti e garantire la sicurezza delle info
16. il sistema garantisce protezione dei dati personali e deve garantire un meccanismo di autenticazione robusto

## 2.3 Glossario dei termini

[OPZIONALE] Riportare un glossario dei termini in una tabella termine-descrizione (significato)-eventuali sinonimi, come nel seguente esempio.

Termine	Descrizione	Sinonimi
<b>Utente</b>	Si autentica, pubblica le ricette nel sistema e visualizza statistiche	



<b><u>Ricetta</u></b>	<u>Pubblicate da utente e caratterizzate da titolo, ingredienti, descrizione ,tempo stimato e tag</u>	
<b><u>Raccolta</u></b>	<u>Collezione di ricette sia pubbliche che private, caratterizzate da autore, titolo e una descrizione</u>	<u>collezione</u>
<b><u>Feed</u></b>	<u>Lista dinamica di ricette pubbliche di altri utenti, contiene al max 5 ricette in ordine cronologico decrescente</u>	
<b><u>Commento</u></b>	<u>Apprezzamento testuale di un utente verso una ricetta, che definisce l'autore, il testo e la data di pubblicazione</u>	
<b><u>Like</u></b>	<u>Apprezzamento positivo di un utente verso una ricetta</u>	<u>Mi piace</u>
<b><u>Notifica</u></b>	<u>Segnala l'utente l'arrivo di commenti e/o apprezzamenti</u>	
<b><u>Amministratore</u></b>	<u>Coloro che accedono a funzionalità di reportistica</u>	
<b><u>Report</u></b>	<u>Documento contenente dati aggregati generato da un amministratore</u>	
<b><u>Tag</u></b>	<u>Etichetta tematica associata a una ricetta</u>	<u>Categoria</u>

## 2.4 Classificazione dei requisiti

### 2.4.1 Requisiti funzionali

<u>ID</u>	<u>Requisito</u>	<u>Origine (n. frase dei requisiti revisionati)</u>
<b><u>RF01</u></b>	<u>Il sistema deve permettere la registrazione degli utenti</u>	<u>1</u>
<b><u>RF02</u></b>	<u>Il sistema deve permettere all'utente l'autenticazione</u>	<u>2</u>
<b><u>RF03</u></b>	<u>Il sistema deve consentire la modifica del profilo personale.</u>	<u>3</u>
<b><u>RF04</u></b>	<u>il sistema permette all'utente autenticato di pubblicare nuove ricette tramite un interfaccia e aggiungerle in una raccolta esistente o crearne una nuova</u>	<u>4</u>
<b><u>RF05</u></b>	<u>e il sistema permette all'utente autenticato di poter visualizzare nuove ricette pubblicate da altri utenti nel feed principale</u>	<u>8</u>
<b><u>RF06</u></b>	<u>il sistema permette all'utente di interagire con altri ricette pubblicate da altri utenti attraverso like e commenti</u>	<u>9</u>
<b><u>RF07</u></b>	<u>Il sistema permette, che quando una ricetta viene visualizzata mostra il numero totale di like e i tre commenti più recenti</u>	<u>11</u>
<b><u>RF08</u></b>	<u>il sistema offre agli utenti una sezione per accedere alle statistiche personali</u>	<u>12</u>
<b><u>RF09</u></b>	<u>il sistema prevede un sistema di notifiche per segnalare all'utente l'arrivo di commenti e like</u>	<u>15</u>
<b><u>RF10</u></b>	<u>Il sistema permette agli amministratori la generazione di report</u>	<u>13</u>



## 2.4.2 Requisiti sui dati

<u>ID</u>	<u>Requisito</u>	<u>Origine (n. frase dei requisiti revisionati)</u>
<u>RD01</u>	<u>Di ogni utente si vuole memorizzare e-mail, nome, cognome, password</u>	<u>1</u>
<u>RD02</u>	<u>Di ogni profilo personale si vuole memorizzare i dati anagrafici, biografia e immagine del profilo</u>	<u>3</u>
<u>RD03</u>	<u>di ogni ricetta si vuole memorizzare un titolo, ingredienti, descrizione dei passaggi, un tempo stimato di preparazione e tag tematici</u>	<u>5</u>
<u>RD04</u>	<u>di ogni raccolta si vuole memorizzare l'autore, le ricette, il titolo e descrizione, stato (pubblica o privata)</u>	<u>7</u>
<u>RD05</u>	<u>di ogni commento si vuole memorizzare autore, testo e la data di pubblicazione</u>	<u>10</u>

## 2.4.3 Vincoli / Altri requisiti

<u>ID</u>	<u>Requisito</u>	<u>Origine (n. frase dei requisiti revisionati)</u>
<u>V01</u>	<u>Il sistema deve limitare la lunghezza della descrizione di ogni ricetta a <b>800 caratteri</b> al massimo</u>	<u>5</u>
<u>V02</u>	<u>Il sistema deve richiedere che il <b>tempo stimato</b> di una ricetta sia espresso in <b>minuti</b> (come numero intero)</u>	<u>5</u>
<u>V03</u>	<u>Il feed pubblico delle ricette deve mostrare <b>al più 5 ricette</b>, ordinate in <b>ordine cronologico decrescente</b> di pubblicazione</u>	<u>8</u>
<u>V04</u>	<u>Il sistema deve adottare un meccanismo di autenticazione sicuro.</u>	<u>16</u>
<u>V05</u>	<u>il sistema dovrà essere accessibile sia da dispositivi desktop che dispositivi mobile</u>	<u>14</u>
<u>V06</u>	<u>il sistema garantisce protezione dei dati personali</u>	<u>16</u>



## 2.5 Modellazione dei casi d'uso

### 2.5.1 Attori e casi d'uso

#### Attori Primari:

- Utente
- Amministratore

#### Attori Secondari:

- Servizio di messaggistica

#### Casi d'uso:

- UC1: Registrazione
- UC2: Login
- UC3: Modifica profilo
- UC4: Pubblicazione ricette
- UC5: Creazione Raccolta
- UC6: Visualizzazione feed
- UC7: Interazione con ricetta(Like/Commento)
- UC8: Aggiungi like
- UC9: Aggiungi Commento
- UC10: Visualizzazione statistiche personali\*
- UC11: Generazione report(Ammministratore)
- UC12: Report per periodo(Ammministratore)
- UC13: Report per autore(Ammministratore)
- UC14: Report per tag(Ammministratore)
- UC15: Report per Interazione(Ammministratore)

#### Casi d'uso di inclusione:

- UC16: Notifica interazione(Servizio di messaggistica)

<u>Caso d'uso</u>	<u>Attori Primari</u>	<u>Attori Secondari</u>	<u>Incl. / Ext.</u>	<u>Requisiti corrispondenti</u>
<u>UC1: Registrazione</u>	<u>Utente</u>	<u>Servizio di messaggistica</u>		<u>RF01</u>
<u>UC2: Login</u>	<u>Utente</u>	<u>Servizio di messaggistica,</u>		<u>RF02</u>
<u>UC3: Modifica Profilo</u>	<u>Utente</u>			<u>RF03</u>
<u>UC4: Pubblicazione ricette</u>	<u>Utente</u>		<u>Extends: Creazione Raccolta, Raccolta Esistente</u>	<u>RF04</u>
<u>UC5: Creazione raccolta</u>	<u>Utente</u>		<u>Esteso da UC4</u>	<u>RF04</u>
<u>UC6: Visualizzazione feed</u>	<u>Utente</u>			<u>RF05</u>
<u>UC7: Interazione con ricetta</u>	<u>Utente</u>		<u>Gen: Aggiungi like, Aggiungi commento</u>	<u>RF06</u>



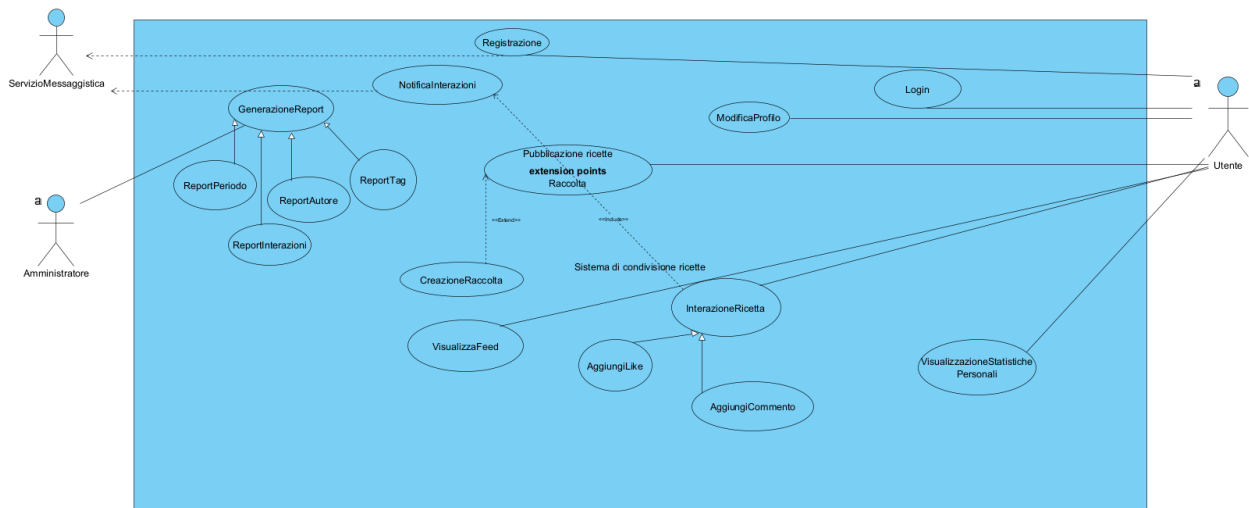


			<u>Include:</u> <u>Notifica interazione</u>	
<u>UC8: Aggiungi like</u>	<u>Utente</u>			<u>RF06</u>
<u>UC9: Aggiungi commento</u>	<u>Utente</u>			<u>RF06</u>
<u>UC10: Visualizzazione statistiche personali</u>	<u>Utente</u>			<u>RF07,RF08</u>
<u>UC11: Generazione Report</u>	<u>Amministratore</u>		<u>Gen:UC12,UC13,UC14, UC15</u>	<u>RF10</u>
<u>UC12: Report per periodo</u>	<u>Amministratore</u>			<u>RF10</u>
<u>UC13: Report per autore</u>	<u>Amministratore</u>			<u>RF10</u>
<u>UC14: Report per tag</u>	<u>Amministratore</u>			<u>RF10</u>
<u>UC15: Report Interazioni</u>	<u>Amministratore</u>			<u>RF10</u>
<u>UC16: Notifica di interazione</u>	<u>Utente</u>	<u>Servizio di messaggistica</u>	<u>Incluso in interazione con ricetta</u>	<u>RF09</u>



## 2.5.2 Diagramma dei casi d'uso

Riportare il diagramma dei casi d'uso.



## 2.5.3 Scenari

Caso d'uso:	Visualizza Statistiche Personali
<b>Attore primario</b>	Utente
<b>Attore secondario</b>	=
<b>Descrizione</b>	L'utente tramite una sezione dedicata alle statistiche personali può visualizzare il numero complessivo di like ricevuti dalle proprie ricette, il numero totale di commenti ottenuti e la ricetta più apprezzata tra quelle pubblicate ossia quella con il numero più alto di like
<b>Pre-Condizioni</b>	L'utente ha effettuato l'accesso al sistema
<b>Sequenza di eventi principale</b>	<ol style="list-style-type: none"> <li>1. l'utente seleziona il menù statistiche personali</li> <li>2. il sistema interroga il database per mostrare <ul style="list-style-type: none"> <li>. il numero totale di like ricevuti per tutte le ricette</li> <li>. il numero totale di commenti</li> <li>. la ricetta col maggior numero di like</li> </ul> </li> </ol>
<b>Post-Condizioni</b>	=
<b>Casi d'uso correlati</b>	=
<b>Sequenza di eventi alternativi</b>	=
<b>Caso d'uso:</b>	Registrazione
<b>Attore primario</b>	Utente
<b>Attore secondario</b>	Servizio di messaggistica, Database



<b><u>Descrizione</u></b>	<u>Il sistema consente la registrazione degli utenti tramite l'inserimento di e-mail, nome, cognome e password</u>
<b><u>Pre-Condizioni</u></b>	<u>L'utente non è registrato al sistema</u>
<b><u>Sequenza di eventi principale</u></b>	<ol style="list-style-type: none"> <li>1. <u>Il caso d'uso inizia quando l'utente seleziona l'opzione registrati</u></li> <li>2. <u>Il sistema mostra un modulo da compilare con: nome, cognome, e-mail, password.</u></li> <li>3. <u>Il sistema inserisce informazioni richieste e conferma</u></li> <li>4. <u>Il sistema verifica i dati inseriti.</u></li> <li>5. <u>IF (Dati validi) 5.1 Il sistema registra l'utente nel database. 5.2 Il servizio di messaggistica invia una notifica di conferma registrazione. 5.3 Il sistema mostra un messaggio di successo e reindirizza alla pagina di login.</u></li> </ol>
<b><u>Post-Condizioni</u></b>	<u>Una volta registrato il sistema salva le info dell'utente ossia nome, cognome, e-mail e password in un apposito database</u>
<b><u>Casi d'uso correlati</u></b>	=
<b><u>Sequenza di eventi alternativi</u></b>	<i><u>Al caso 5 se i dati non sono validi , il sistema mostra un messaggio di errore tornando al caso 2 per modificare i dati inseriti</u></i>
<b><u>Caso d'uso:</u></b>	<u>Login</u>
<b><u>Attore primario</u></b>	<u>Utente</u>
<b><u>Attore secondario</u></b>	=
<b><u>Descrizione</u></b>	<u>L'utente accede al sistema tramite e-mail e password</u>
<b><u>Pre-Condizioni</u></b>	<u>L'utente ha effettuato la registrazione al sistema</u>
<b><u>Sequenza di eventi principale</u></b>	<ol style="list-style-type: none"> <li>1. <u>Il caso d'uso inizia quando l'utente seleziona "Login" dalla pagina iniziale.</u></li> <li>2. <u>Il sistema mostra il form di accesso</u></li> <li>3. <u>L'utente inserisce e-mail e password.</u></li> <li>4. <u>Il sistema verifica le credenziali inserite.</u></li> <li>5. <u>IF (Credenziali corrette) 5.1 Il sistema autentica l'utente. 5.2 L'utente viene reindirizzato alla homepage. 5.3 Il caso d'uso termina.</u></li> </ol>
<b><u>Post-Condizioni</u></b>	=
<b><u>Casi d'uso correlati</u></b>	=



<b><u>Sequenza di eventi alternativi</u></b>	Al caso 3 Se l'utente dimentica la password, il sistema fornisce un link per il recupero delle credenziali tramite e-mail, al caso 5 se le credenziali non sono corrette, il sistema mostra un messaggio di errore ritornando al caso 2 per l'inserimento dei dati
<b><u>Caso d'uso:</u></b>	<b><u>Modifica Profilo</u></b>
<b><u>Attore primario</u></b>	<u>Utente</u>
<b><u>Attore secondario</u></b>	=
<b><u>Descrizione</u></b>	L'utente una volta autenticato ha una sezione dove può modificare il proprio profilo personale che include una breve biografia e una immagine del profilo
<b><u>Pre-Condizioni</u></b>	<u>L'utente ha effettuato l'accesso al sistema</u>
<b><u>Sequenza di eventi principale</u></b>	<ol style="list-style-type: none"> <li>1. <u>Il caso d'uso inizia quando l'utente seleziona la voce "Profilo" dal menu principale.</u></li> <li>2. <u>Il sistema mostra le informazioni attuali del profilo.</u></li> <li>3. <u>L'utente seleziona l'opzione "Modifica profilo".</u></li> <li>4. <u>L'utente inserisce una nuova biografia e/o carica una nuova immagine.</u></li> <li>5. <u>Il sistema valida i dati inseriti.</u></li> </ol>
<b><u>Post-Condizioni</u></b>	<u>Profilo modificato</u>
<b><u>Casi d'uso correlati</u></b>	=
<b><u>Sequenza di eventi alternativi</u></b>	=
<b><u>Caso d'uso:</u></b>	<b><u>Pubblicazione ricette</u></b>
<b><u>Attore primario</u></b>	<u>Utente</u>
<b><u>Attore secondario</u></b>	=
<b><u>Descrizione</u></b>	Il sistema permette all'utente di pubblicare nuove ricette tramite una apposita interfaccia grafica
<b><u>Pre-Condizioni</u></b>	<u>L'utente ha effettuato l'accesso al sistema</u>
<b><u>Sequenza di eventi principale</u></b>	<ol style="list-style-type: none"> <li>1. <u>Il caso d'uso inizia quando l'utente accede al sistema</u></li> <li>2. <u>Il sistema presenta un modulo per inserire titolo, ingredienti, descrizione dei passaggi, tempo stimato di preparazione e uno o più tag tematici.</u></li> <li>3. <u>IF (Descrizione ≤ 800 caratteri) 4.1 Il sistema salva la ricetta nel database. 4.2 La ricetta viene associata alla raccolta selezionata. 4.3 Il sistema conferma la pubblicazione.</u></li> <li>4. <u>L'utente può scegliere se pubblicare la ricetta in una raccolta esistente o crearne una nuova</u> <u>Punto di estensione: &lt;&lt;extend&gt;&gt; Creazione Raccolta</u></li> <li>5. <u>La ricetta è stata aggiunta alla raccolta</u></li> </ol>
<b><u>Post-Condizioni</u></b>	<u>La ricetta è visibile nella raccolta d'interesse</u>
<b><u>Casi d'uso correlati</u></b>	



<b><u>Sequenza di eventi alternativi</u></b>	<i>Al caso 4 se la descrizione è troppo lunga il sistema blocca la pubblicazione, mostrando un messaggio di errore all'utente</i>
<b><u>Caso d'uso:</u></b>	<b><u>Creazione Raccolta</u></b>
<b><u>Attore primario</u></b>	<u>Utente</u>
<b><u>Attore secondario</u></b>	-
<b><u>Descrizione</u></b>	<u>L'utente decide di creare una nuova raccolta che funge da contenitore di ricette</u>
<b><u>Pre-Condizioni</u></b>	<u>L'utente ha effettuato l'accesso al sistema ed ha creato una nuova ricetta da pubblicare in una nuova raccolta</u>
<b><u>Sequenza di eventi principale</u></b>	<ol style="list-style-type: none"> <li>1. <u>Il caso d'uso inizia quando l'utente, durante la pubblicazione di una ricetta, seleziona l'opzione per creare una nuova raccolta.</u></li> <li>2. <u>Il sistema mostra un modulo per l'inserimento del titolo e della descrizione della raccolta.</u></li> <li>3. <u>L'utente inserisce i dati richiesti e conferma.</u></li> <li>4. <u>Il sistema verifica la validità dei dati.</u></li> <li>5. <u>IF (Dati validi) 5.1 Il sistema crea la nuova raccolta nel database. 5.2 La ricetta viene associata alla nuova raccolta. 5.3 Il sistema conferma la creazione e l'associazione.</u></li> </ol>
<b><u>Post-Condizioni</u></b>	<u>La ricetta è visibile nella nuova raccolta</u>
<b><u>Casi d'uso correlati</u></b>	<u>Pubblicazione ricette</u>
<b><u>Sequenza di eventi alternativi</u></b>	<i>Al caso 5 se i dati non sono validi il sistema mostra un messaggio di errore e il sistema reindirizza al caso 2</i>
<b><u>Caso d'uso:</u></b>	<b><u>Visualizzazione feed</u></b>
<b><u>Attore primario</u></b>	<u>Utente</u>
<b><u>Attore secondario</u></b>	-
<b><u>Descrizione</u></b>	<u>Una volta autenticato ciascun utente può visualizzare nel proprio feed personale un insieme di nuove ricette pubblicate da altri utenti e queste ultime devono essere pubbliche</u>
<b><u>Pre-Condizioni</u></b>	<u>L'utente ha effettuato l'accesso al sistema</u>
<b><u>Sequenza di eventi principale</u></b>	<ol style="list-style-type: none"> <li>1. <u>Il caso d'uso inizia quando l'utente accede al proprio feed personale</u></li> <li>2. <u>Il sistema interroga il database per recuperare le ultime ricette pubbliche pubblicate da altri utenti.</u></li> <li>3. <u>Il sistema seleziona al massimo 5 ricette, ordinate in ordine cronologico decrescente.</u></li> <li>4. <u>IF (Ricette disponibili) 4.1 Il sistema visualizza le ricette nel feed dell'utente.</u></li> </ol>
<b><u>Post-Condizioni</u></b>	<u>-il feed personale dell'utente è aggiornato con le ultime ricette</u>



<b>Casi d'uso correlati</b>	=
<b>Sequenza di eventi alternativi</b>	<i>-al caso 4 se non ci sono ricette disponibili il sistema mostra un messaggio di errore</i>
<b>Caso d'uso:</b>	<u>Interazione con Ricetta</u>
<b>Attore primario</b>	Utente
<b>Attore secondario</b>	=
<b>Descrizione</b>	Il sistema permette all'utente di interagire con le ricette tramite like e commenti
<b>Pre-Condizioni</b>	L'utente ha effettuato l'accesso al sistema
<b>Sequenza di eventi principale</b>	<ol style="list-style-type: none"> <li>1. <u>l'utente visualizza il proprio feed</u></li> <li>2. <u>Il sistema mostra il dettaglio della ricetta, inclusi like e commenti.</u></li> <li>3. <u>L'utente sceglie di interagire tramite like e/o commento</u></li> <li>4. <u>IF(interazione completata) 4.1 il sistema registra l'interazione. 4.2 &lt;&lt;include&gt;&gt; Notifica di interazione</u></li> </ol>
<b>Post-Condizioni</b>	L'interazione è aggiunta alla ricetta
<b>Casi d'uso correlati</b>	<u>Aggiungi Like, Aggiungi Commento, Notifica di interazione</u>
<b>Sequenza di eventi alternativi</b>	<i>Al caso 4 se l'interazione non viene completata, non viene salvata nessuna modifica, e non è inviata alcuna notifica</i>
<b>Caso d'uso:</b>	<u>Notifica di interazione</u>
<b>Attore primario</b>	Utente
<b>Attore secondario</b>	Servizio di Messaggistica
<b>Descrizione</b>	Una volta effettuata l'interazione il sistema invia una notifica all'utente interessato
<b>Pre-Condizioni</b>	L'interazione è stata aggiunta alla ricetta di interesse
<b>Sequenza di eventi principale</b>	<ol style="list-style-type: none"> <li>1. <u>Il caso d'uso inizia quando l'utente interagisce con la ricetta</u></li> <li>2. <u>il sistema registra l'interazione nel database</u></li> <li>3. <u>il sistema attiva il servizio di messaggistica</u></li> <li>4. <u>il servizio di messaggistica invia una notifica al creatore della ricetta</u></li> </ol>
<b>Post-Condizioni</b>	L'utente interessato viene avvisato dell'interazione
<b>Casi d'uso correlati</b>	<u>Interazione con ricetta</u>
<b>Sequenza di eventi alternativi</b>	=
<b>Caso d'uso:</b>	<u>Aggiungi Like</u>
<b>Attore primario</b>	Utente
<b>Attore secondario</b>	=
<b>Descrizione</b>	Il sistema permette all'utente di interagire con altre ricette tramite un like
<b>Pre-Condizioni</b>	L'utente ha effettuato l'accesso al sistema
<b>Sequenza di eventi principale</b>	<ol style="list-style-type: none"> <li>1. <u>Il caso d'uso inizia quando l'utente accede al sistema</u></li> <li>2. <u>L'utente visualizza il proprio feed</u></li> <li>3. <u>L'utente decide di interagire con la ricetta tramite un like</u></li> <li>4. <u>il sistema incrementa il contatore del like.</u></li> </ol>
<b>Post-Condizioni</b>	Il numero totale di like della ricetta è aggiornato
<b>Casi d'uso correlati</b>	<u>Interazione con ricetta</u>



<b><u>Sequenza di eventi alternativi</u></b>	-
<b><u>Caso d'uso:</u></b>	<i>Aggiungi commento</i>
<b><u>Attore primario</u></b>	<u>Utente</u>
<b><u>Attore secondario</u></b>	-
<b><u>Descrizione</u></b>	<u>Il sistema permette all'utente di interagire con altre ricette tramite un like</u>
<b><u>Pre-Condizioni</u></b>	<u>L'utente ha effettuato l'accesso al sistema</u>
<b><u>Sequenza di eventi principale</u></b>	<u>1. Il caso d'uso inizia quando l'utente accede al sistema</u> <u>2. L'utente visualizza il proprio feed</u> <u>3. L'utente decide di interagire con la ricetta tramite un commento</u> <u>4. il sistema mostra l'area dedicata ai commenti</u> <u>5. l'utente inserisce testo e conferma</u> <u>6. IF ( testo non vuoto)</u> <u>6.1 Il sistema salva il commento associandolo alla ricetta.</u> <u>6.2 Il sistema aggiorna l'elenco dei commenti visibili.</u>
<b><u>Post-Condizioni</u></b>	<u>Il commento associato alla ricetta è salvato tra i più recenti</u>
<b><u>Casi d'uso correlati</u></b>	<u>Interazione con ricetta</u>
<b><u>Sequenza di eventi alternativi</u></b>	<u>Al caso 6 se il testo è vuoto il sistema mostra un messaggio di errore chiedendo l'utente di reinscrivere il commento</u>
<b><u>Caso d'uso:</u></b>	<i>Generazione report</i>
<b><u>Attore primario</u></b>	<u>Amministratore</u>
<b><u>Attore secondario</u></b>	-
<b><u>Descrizione</u></b>	<u>Gli amministratori della piattaforma, possono accedere a funzionalità di reportistica.</u>
<b><u>Pre-Condizioni</u></b>	<u>-Amministratore ha accesso al sistema</u>
<b><u>Sequenza di eventi principale</u></b>	<u>1. Il caso d'uso inizia quando l'amministratore seleziona la sezione "GeneraReport"</u> <u>2. Il sistema mostra le opzioni disponibili per la generazione del report.</u> <u>3. L'amministratore seleziona una tipologia di report da generare.</u> <u>4. Il sistema attiva il relativo caso d'uso specifico (&lt;&gt; Report per periodo, autore, tag o interazioni).</u>
<b><u>Post-Condizioni</u></b>	
<b><u>Casi d'uso correlati</u></b>	<u>Report per periodo, Report per autori, Report per tag</u>
<b><u>Sequenza di eventi alternativi</u></b>	-
<b><u>Caso d'uso:</u></b>	<i>Report per periodo</i>
<b><u>Attore primario</u></b>	<u>Amministratore</u>
<b><u>Attore secondario</u></b>	-
<b><u>Descrizione</u></b>	<u>Gli amministratori della piattaforma, possono generare report contenenti dati aggregati come il numero totale di ricette pubblicate in un dato intervallo di tempo</u>



<b><u>Pre-Condizioni</u></b>	<u>-Amministratore ha accesso al sistema</u>
<b><u>Sequenza di eventi principale</u></b>	<ol style="list-style-type: none"> <li>1. <u>Il caso d'uso inizia quando l'amministratore seleziona l'opzione "Report per periodo".</u></li> <li>2. <u>Il sistema richiede l'inserimento delle date di inizio e fine periodo.</u></li> <li>3. <u>L'amministratore inserisce l'intervallo temporale desiderato.</u></li> <li>4. <u>IF(periodo valido)</u></li> </ol> <p><u>4.1 Il sistema interroga il database per calcolare il numero totale di ricette pubblicate nel periodo indicato.</u></p> <p><u>4.2 IF(Dati trovati) 4.2.1 Il sistema genera e visualizza il report con il conteggio delle ricette.</u></p>
<b><u>Post-Condizioni</u></b>	<u>L'amministratore visualizza il report per intervallo temporale.</u>
<b><u>Casi d'uso correlati</u></b>	<u>Generazione di report</u>
<b><u>Sequenza di eventi alternativi</u></b>	<u>-al caso 4 se l'utente inserisce un periodo non valido il sistema mostra un messaggio di errore reindirizzando al caso 3</u> <u>al caso 4.2 se non è trovata alcuna ricetta il sistema mostra un messaggio informativo</u>
<b><u>Caso d'uso:</u></b>	<u>Report per autore</u>
<b><u>Attore primario</u></b>	<u>Amministratore</u>
<b><u>Attore secondario</u></b>	<u>=</u>
<b><u>Descrizione</u></b>	<u>Gli amministratori della piattaforma, possono generare report contenenti dati aggregati come l'elenco degli autori più attivi in base al numero di ricette pubblicate</u>
<b><u>Pre-Condizioni</u></b>	<u>-l'amministratore ha accesso al sistema</u>
<b><u>Sequenza di eventi principale</u></b>	





<ol style="list-style-type: none"> <li>1. <u>Il caso d'uso inizia quando l'amministratore seleziona l'opzione "Report per autore".</u></li> <li>2. <u>Il sistema interroga il database per ottenere la lista degli utenti ordinati per numero di ricette pubblicate.</u></li> <li>3. <u>IF (Autori trovati)</u></li> </ol> <p><u>3.1 Il sistema genera il report e visualizza i dati.</u></p>	
<b><u>Post-Condizioni</u></b>	<u>L'amministratore visualizza un report con gli autori più attivi.</u>
<b><u>Casi d'uso correlati</u></b>	<u>Generazione report</u>
<b><u>Sequenza di eventi alternativi</u></b>	<i>-al caso 3 se non ci sono autori, il sistema informa che non sono presenti autori con ricette registrate</i>
<b><u>Caso d'uso:</u></b>	<u>Report per tag</u>
<b><u>Attore primario</u></b>	<u>Amministratore</u>
<b><u>Attore secondario</u></b>	-
<b><u>Descrizione</u></b>	<u>Gli amministratori della piattaforma, possono generare report contenenti dati aggregati come i tag tematici più utilizzati</u>
<b><u>Pre-Condizioni</u></b>	<u>- l'amministratore ha accesso al sistema</u>
<b><u>Sequenza di eventi principale</u></b>	<p>-</p> <ol style="list-style-type: none"> <li>1. <u>Il caso d'uso inizia quando l'amministratore seleziona l'opzione "Report per tag"</u></li> <li>2. <u>Il sistema interroga il database per recuperare e contare i tag associati alle ricette pubblicate.</u></li> <li>3. <u>IF (Tag disponibili) 3.1 Il sistema elabora i dati e genera il report. 3.2 Il sistema permette la visualizzazione del report all'amministratore.</u></li> </ol>
<b><u>Post-Condizioni</u></b>	<u>-L'amministratore visualizza un report dei tag più usati.</u>
<b><u>Casi d'uso correlati</u></b>	<u>Generazione report</u>
<b><u>Sequenza di eventi alternativi</u></b>	<i>-al caso 3 se nessun tag è stato trovato viene inviato un messaggio informativo all'amministratore</i>
<b><u>Caso d'uso:</u></b>	<u>Report per Interazioni</u>
<b><u>Attore primario</u></b>	<u>Amministratore</u>
<b><u>Attore secondario</u></b>	-
<b><u>Descrizione</u></b>	<u>Gli amministratori della piattaforma, possono generare report contenenti dati aggregati per le ricette con il maggior numero di interazioni</u>
<b><u>Pre-Condizioni</u></b>	<u>-l'amministratore ha accesso al sistema</u>



<b><u>Sequenza di eventi principale</u></b>	<ol style="list-style-type: none"> <li>1. <u>Il caso d'uso inizia quando l'amministratore seleziona l'opzione "Report per interazioni".</u></li> <li>2. <u>Il sistema interroga il database per identificare le ricette con il maggior numero complessivo di like e commenti.</u></li> <li>3. <u>Il sistema elabora i dati e calcola le ricette piu interattive</u></li> <li>4. <u>il sistema mostra e genera il report</u></li> </ol>
<b><u>Post-Condizioni</u></b>	<u>-L'amministratore visualizza le ricette più interattive della piattaforma.</u>
<b><u>Casi d'uso correlati</u></b>	<u>Generazione report</u>
<b><u>Sequenza di eventi alternativi</u></b>	-

## 2.6 Diagramma delle classi

Class Diagramm corrispondente al System Domain Model ossia quello individuato nella fase di analisi



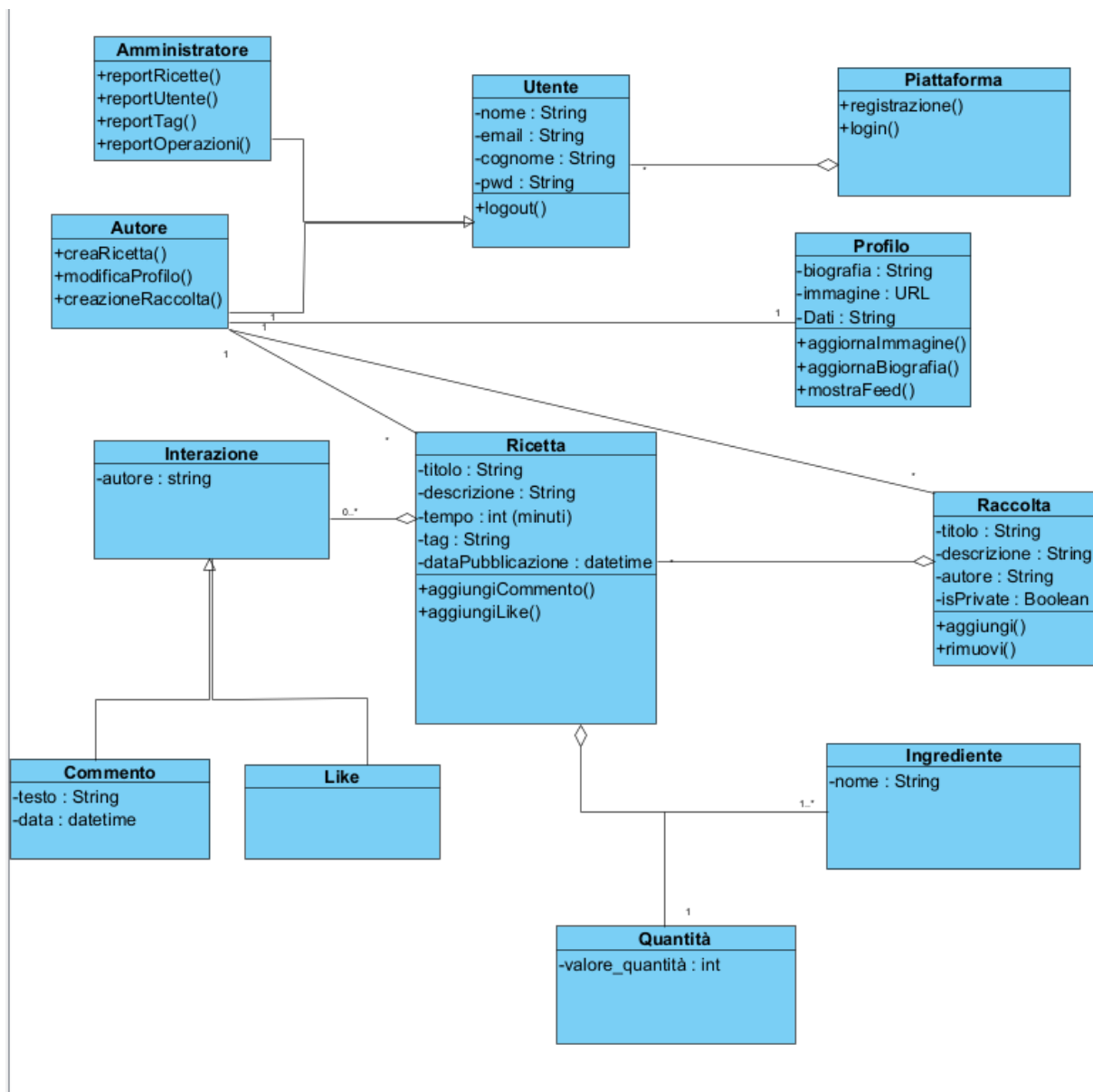


Tabella che riassume alcune responsabilità:

RESPONSABILITÀ	CLASSE
<u>Registrazione</u>	<u>Sistema</u>
<u>Login</u>	<u>Sistema</u>
<u>Logout</u>	<u>Autore</u>
<u>ModificaProfilo</u>	<u>Autore</u>
<u>CreaRicetta</u>	<u>Autore</u>
<u>AggiungiLike</u>	<u>Ricetta</u>
<u>AggiungiCommento</u>	<u>Ricetta</u>
<u>CreazioneRaccolta</u>	<u>Autore</u>



<u>ReportRicette</u>	<u>Amministratore</u>
<u>ReportUtente</u>	<u>Amministratore</u>
<u>ReportTag</u>	<u>Amministratore</u>
<u>ReportInterazioni</u>	<u>Amministratore</u>

**Giustificazione di come è stata assegnata la responsabilità tenendo presente anche i pattern GRASP**

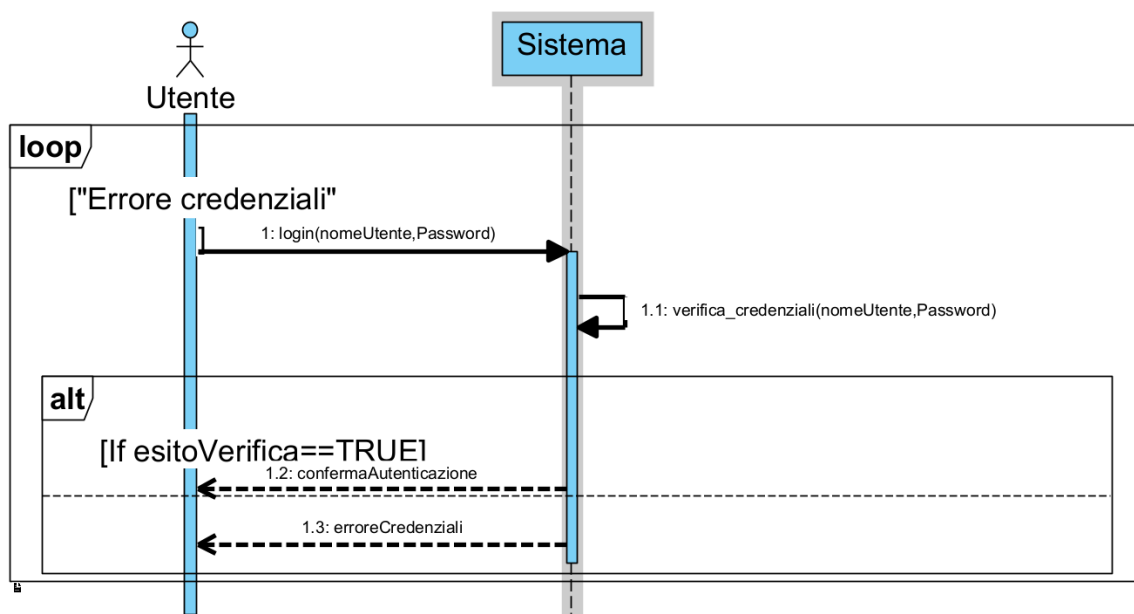
- Registrazione e Accesso(login e logout) sono responsabilità di **Sistema**, in quanto <<information expert>> di Utente.
- PubblicaRicetta è responsabilità di **Autore**, perché è <<Creator>> della classe Ricetta.
- ModificaProfilo è responsabilità di **Autore**, perché è colui che può modificare di fatto il suo profilo personale.
- AggiungiLike e AggiungiCommento è responsabilità di **Ricetta** perché è colei che incrementa il numero di like/commenti.
- CreazioneRaccolta è responsabilità di **Autore** perché è <<creator>> della classe Raccolta.
- GeneraReport(ReportRicette,ReportUtente,ReportTag,ReportInterazione) è responsabilità di **Amministratore** perché è colui, previa autenticazione, può accedere a funzionalità di reportistica.



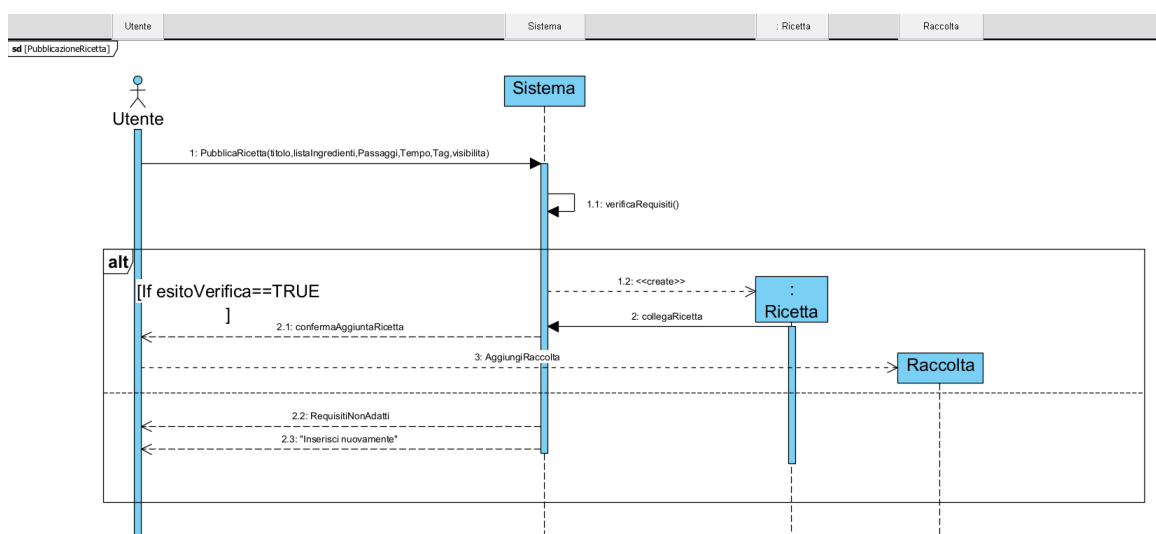
## 2.7 Diagrammi di sequenza

### 2.7.1 Login

La creazione del suddetto sequence diagram, sviluppato a partire dalla descrizione dello scenario del caso d'uso *Login*, ha fatto sorgere la necessità di definire un metodo, specifico per la classe **Sistema**, **verificaCredenziali(nomeUtente, password)**, privato, per consentire al sistema di verificare che le credenziali inserite dall'utente siano valide.



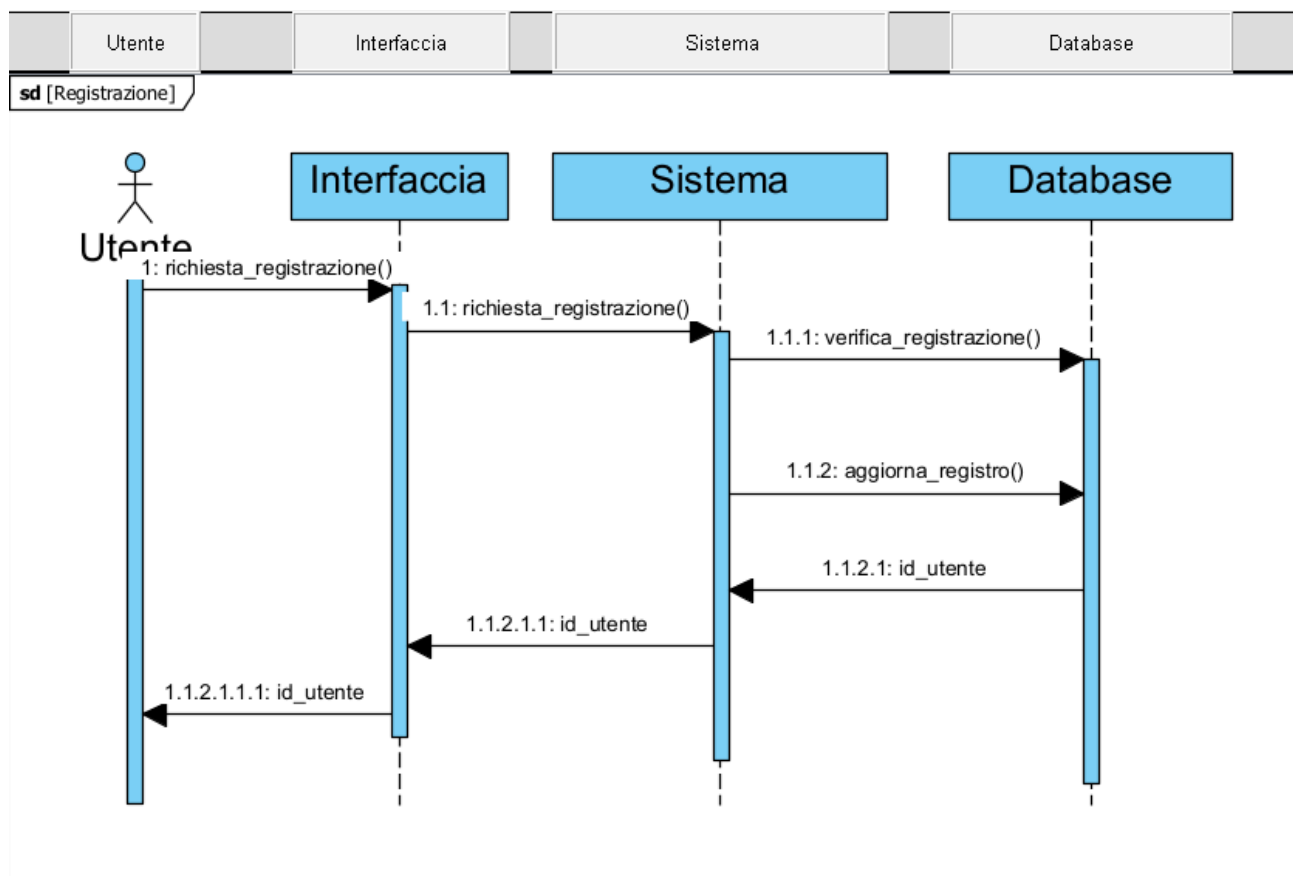
### 2.7.2 Pubblica Ricetta



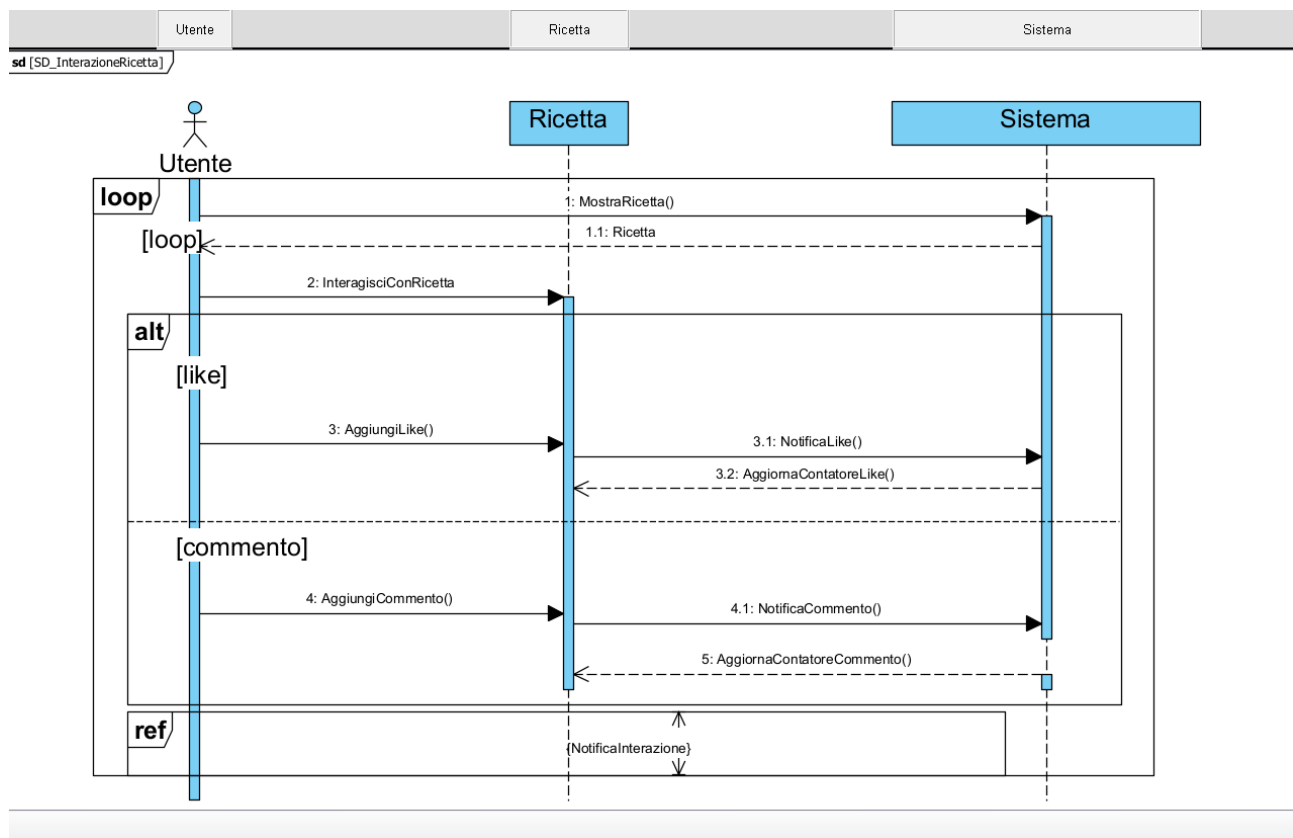
Per le stesse ragioni, è stato necessario inserire all'interno della classe **Sistema** il metodo privato **verificaRequisiti()**, col fine ultimo di individuare se la ricetta che l'utente intende pubblicare rispetta i requisiti del sistema.



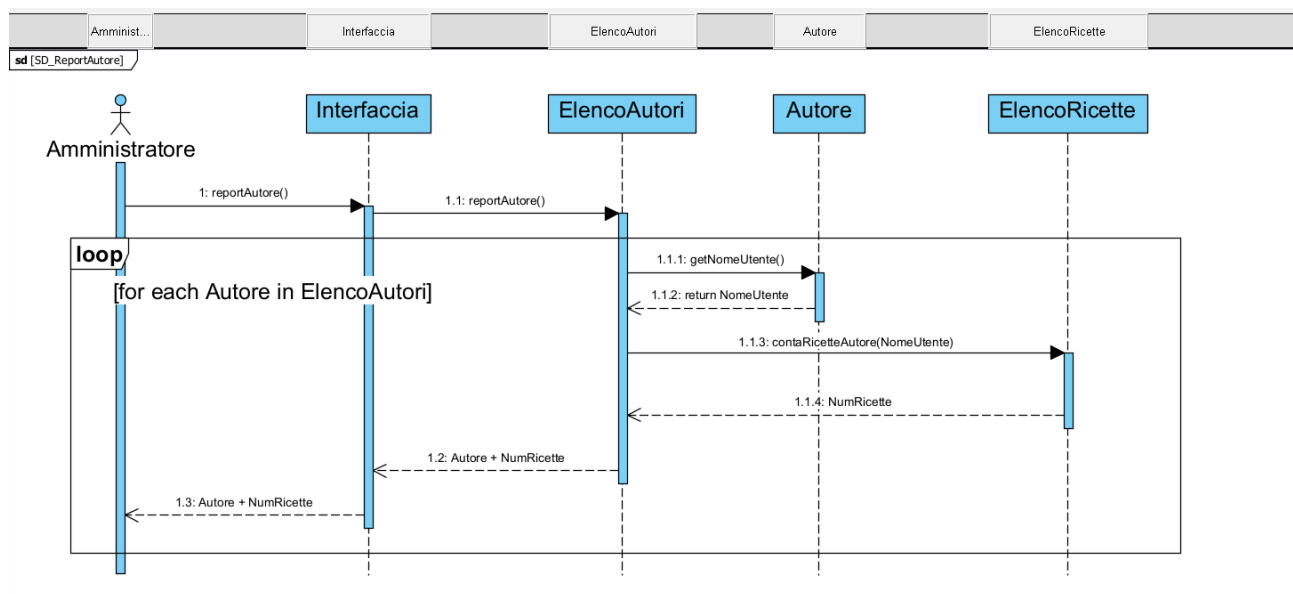
## 2.7.3 Registrazione



## 2.7.4 Interazione Ricetta

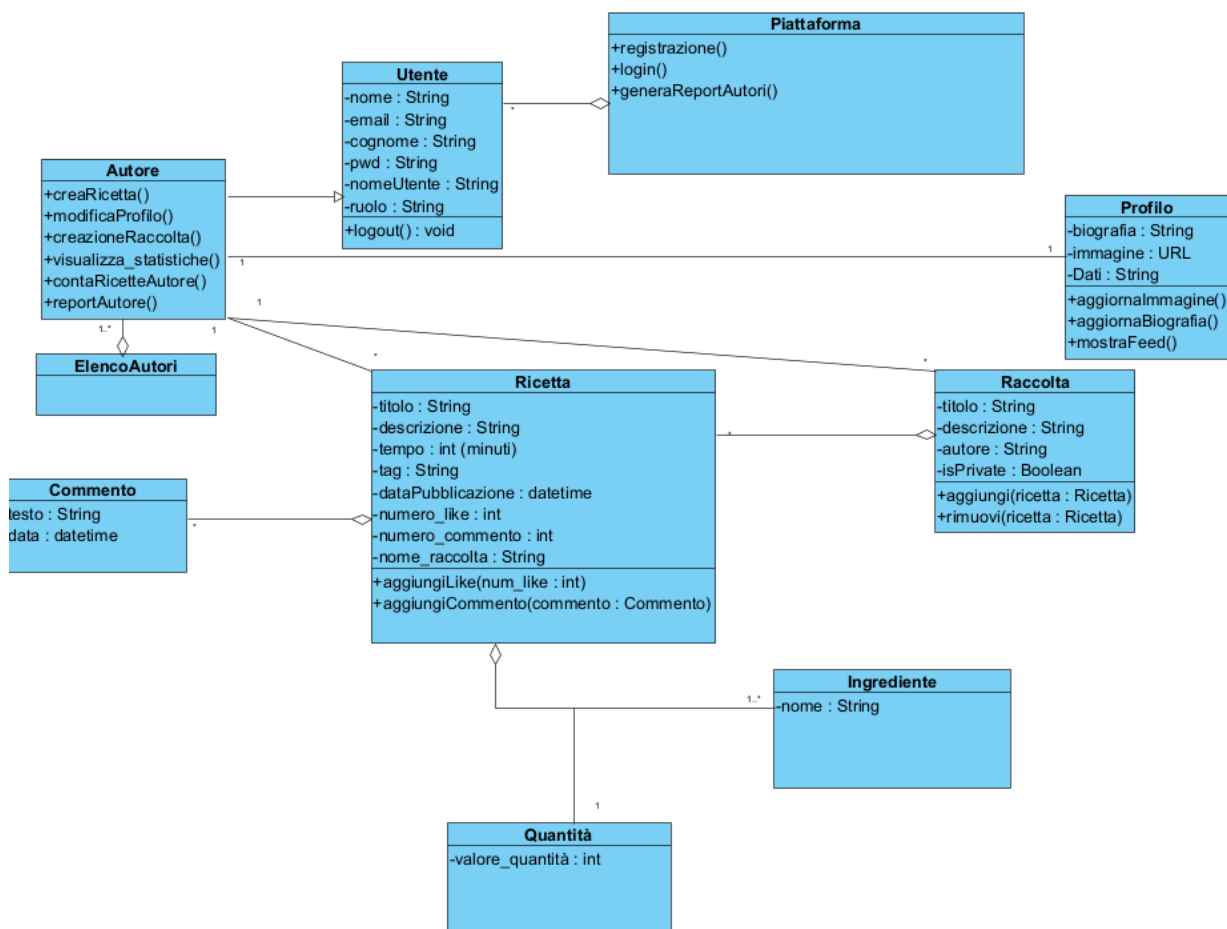


## 2.7.5 Report Autore



## 2.8 Diagramma delle classi raffinato

Le aggiunte e le modifiche fatte nel corso della costruzione dei Sequence Diagrams hanno determinato lo sviluppo di un **Diagramma delle Classi** raffinato che riporta maggiori dettagli sugli attributi e le principali operazioni delle classi:



Breve descrizione del class diagramm raffinato:

il seguente class diagramm è caratterizzata dalla presenza di una classe **piattaforma** che funge da sistema nella quale sono presenti i metodi di registrazione, login e generaReportAutore.

Possiede i metodi di registrazione e login in quanto essa è <<information expert>> di utente e possiede il metodo di generaReportAutore in quanto, in un certo senso, ha visibilità di tutti gli utenti registrati nella piattaforma.



Tra piattaforma e la classe utente c'è un'associazione di contenimento lasco in quanto la classe piattaforma funge da contenitore della classe utente ma non regola il ciclo di vita di quest'ultima.

La classe utente invece oltre ai vari attribuiti è presente il metodo di logout in quanto è responsabilità di utente nell'uscire dalla piattaforma.

Sempre per tale classe è stata effettuata una relazione di gen-spec tra utente e autore in quanto oltre all'autore ossia colui che pubblica le varie ricette e presente una amministratore che è in grado di effettuare i vari tipi di report.

Non abbiamo aggiunto una classe amministratore ma per evidenziare la differenza in utente è stato inserito un attributo di tipo stringa chiamato ruolo.

Nella classe autore sono presenti i vari metodi di creazione ricetta in quanto è colui che è in grado di creare una ricetta e per questo è stata aggiunta una relazione di tipo 1-N tra la classe Autore e la classe Ricetta.

Analogamente ha un metodo di creazione raccolta in quanto l'autore una volta creata una ricetta è obbligato a inserirla in una raccolta esistente.

Per far sì che l'autore abbia visibilità delle raccolte è stata effettuata un'associazione di tipo 1-N tra autore e raccolta.

Autore ha il metodo di modifica profilo in quanto all'atto della registrazione la piattaforma a quest'ultimo gli fornisce un profilo personale che può modificare e per avere visibilità del suo profilo è stata creata un'associazione di tipo 1-1 in quanto l'autore possiede un suo profilo personale.

A supporto della classe autore c'è una classe che viene chiamata Elenco autore in che servirà per effettuare i report per gli autore stessi.

Tra autore ed elenco è presente un'associazione di contenimento lasco.

Per la classe ricetta oltre ai vari attribuiti sono presenti i metodi di aggiungiLike e aggiungiCommento.

Per il commento è stata creata una classe con gli attribuiti autore e testo e tale funge da oggetto contenuto in ricetta.

Non è stato fatto lo stesso per il like in quanto questo ultimo funge da contatore e per tale lo abbiamo inserito in ricetta con il tipo intero.

Inoltre la ricetta funge da oggetto contenuto di raccolta in quanto ricordiamo che la ricetta deve essere inserita in una raccolta.

L'associazione tra raccolta e ricetta è di tipo contenimento lasco in quanto la raccolta non regola il ciclo di vita dell'oggetto ricetta.

Poiché la ricetta per essere creata ha bisogno di ingredienti è stata creata una classe Ingrediente relazionata con ricetta tramite un contenimento lasco che ha come attributo il nome che è di tipo stringa.

Tra ricetta e ingrediente è presente una classe associativa di nome quantità.

Si è creata la classe associativa in quanto la quantità di un ingrediente può variare.

### **3. Piano di test funzionale**

Si intende progettare i casi di test funzionale con la tecnica del **Category Partition Testing**.

#### **3.1 Pubblicazione Ricette**



Pubblicazione Ricette					
TITOLO	INGREDIENTE	DESCRIZIONE	TEMPO	TAG	ESISTENZA RACCOLTA
<ul style="list-style-type: none"> <li>Stringa di caratteri di lunghezza &lt;=100</li> <li>Stringa di caratteri di lunghezza &lt;1[ERROR]</li> <li>Stringa di caratteri di lunghezza &gt;100[ERROR]</li> <li>Stringa che contiene simboli che non sono caratteri [ERROR]</li> <li>Stringa di caratteri contenente caratteri speciali[ERROR]</li> </ul>	<ul style="list-style-type: none"> <li>Stringa di caratteri di lunghezza &lt;= 40</li> <li>Stringa di caratteri di lunghezza &lt;1[ERROR]</li> <li>Stringa di caratteri di lunghezza &gt; 40 [ERROR]</li> <li>Stringa che contiene simboli che non sono caratteri [ERROR]</li> <li>Stringa di caratteri contenente caratteri speciali[ERROR]</li> </ul>	<ul style="list-style-type: none"> <li>Stringa di caratteri di lunghezza&lt;=800</li> <li>Stringa di caratteri di lunghezza&gt;800 [ERROR]</li> <li>Stringa di caratteri di lunghezza&lt;1 [ERROR]</li> <li>Stringa che contiene simboli che non sono caratteri [ERROR]</li> </ul>	<ul style="list-style-type: none"> <li>Numero intero &gt;=10</li> <li>Numero intero &lt;=300</li> <li>Numero intero &gt;300[ERROR]</li> <li>Numero intero &lt;10[ERROR]</li> <li>Numero che non è un intero[ERROR]</li> <li>Numero che contiene simboli che non sono numeri[ERROR]</li> </ul>	<ul style="list-style-type: none"> <li>Stringa di caratteri di lunghezza &lt;=40</li> <li>Stringa di caratteri di lunghezza &lt;1[ERROR]</li> <li>Stringa di caratteri di lunghezza &gt;40[ERROR]</li> <li>Stringa che contiene simboli che non sono caratteri [ERROR]</li> <li>Stringa di caratteri contenente caratteri speciali al di fuori di #[ERROR]</li> </ul>	<ul style="list-style-type: none"> <li>Esistenza Raccolta =TRUE</li> <li>Esistenza Raccolta= FALSE[ERROR]</li> </ul>

Il numero di test da effettuarsi senza particolari vincoli è:  $5 * 5 * 4 * 6 * 5 = 3000$ .

Con i vincoli [ERROR], invece, il numero di test da eseguire per testare singolarmente i vincoli è 20 (4 per Titolo, 4 per Ingrediente, 3 per Descrizione, 4 per Tempo, 4 per Tag e 1 per Esistenza Raccolta).

Il numero di test risultante è:  $(1*1*1*2*4) + 20 = 28$ .

<b>TEST SUITE</b>						
<u>Test Case ID</u>	<u>Descrizione</u>	<u>Classi di equivalenza coperte</u>	<u>Pre-condizioni</u>	<u>Input</u>	<u>Output attesi</u>	<u>Post-condizioni attese</u>
1	Tutti input validi	Titolo valido Ingrediente valido Descrizione valida Tempo valido Tag valido Esistenza raccolta valido		{Titolo: "Carbonara", Ingrediente: "Uova", Descrizione: "Breve descrizione", Tempo: "10", Tag: "#pranzo veloce", Esistenza Raccolta=True }	Ricetta pubblicata	La ricetta è mostrata nel profilo utente
2	Titolo Stringa > 100 caratteri	Titolo stringa > 40 caratteri [ERROR]. Ingrediente. Descrizione. Tempo, Tag. Esistenza Raccolta validi		{Nome: "carbonaraaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaaaaaa" Ingrediente: "Uova". Descrizione: "Breve descrizione". Tempo: "10", Tag: "#pranzo veloce". Esistenza Raccolta=True }	Titolo troppo lungo!	
3	Titolo stringa < 1 caratteri	Titolo stringa < 1 caratteri [ERROR]. Ingrediente. Descrizione. Tempo, Tag. Esistenza Raccolta validi		{Nome: "", Ingrediente: "Uova", Descrizione: "Breve descrizione", Tempo: "10", Tag: "#pranzo veloce". Esistenza Raccolta=True }	Non è presente un Titolo	
4	Ingrediente stringa > 40 caratteri	Titolo valido. Ingrediente stringa > 40.		{Nome: "parmigiana", Ingrediente: "melanzaneeeeeeeeeeeeeeeeee"	Ingrediente troppo lungo!	

		<u>Descrizione.</u> <u>Tempo, Tag.</u> <u>Esistenza</u> <u>Raccolta validi</u>		<u>eeeeeeeeeeeeeeeeeeeeeeeeeeeeeeee</u> <u>eeeeeeeeeeeeeeeeeeee</u> <u>Descrizione: "descrizione".</u> <u>Tempo:</u> <u>"15",Tag:"#vegetariana".</u> <u>Esistenza Raccolta=True }</u>		
5	<u>Tempo in</u> <u>decimale</u>	<u>Titolo valido.</u> <u>Ingrediente</u> <u>valido.</u> <u>Descrizione.tag.</u> <u>Esistenza</u> <u>Raccolta validi</u> <u>Tempo in</u> <u>decimale[ERROR</u> <u>]</u>		<u>{Nome: "caprese",Ingrediente:</u> <u>"cioccolato".</u> <u>Descrizione:"desc".</u> <u>Tag:"#dolce",tempo:10.45,</u> <u>Esistenza Raccolta=True}</u>	<u>Tempo deve</u> <u>essere</u> <u>espresso in</u> <u>minuti</u>	
6	<u>Descrizione</u> <u>stringa &lt;1</u>	<u>Titolo valido.</u> <u>Ingrediente</u> <u>valido.</u> <u>Descrizione</u> <u>stringa &lt;1.</u> <u>[ERROR]</u> <u>Tempo, Tag.</u> <u>Esistenza</u> <u>Raccolta validi</u>		<u>{Nome: "caprese",Ingrediente:</u> <u>"cioccolato", Descrizione:"".</u> <u>Tag:"dolce",tempo:15,</u> <u>Esistenza Raccolta=True}</u>	<u>Non è</u> <u>presente una</u> <u>descrizione</u>	
7	<u>Tag Stringa</u> <u>&lt;1</u>	<u>Titolo valido.</u> <u>Ingrediente</u> <u>valido.</u> <u>Descrizione</u> <u>valida, Tempo.</u> <u>Esistenza</u> <u>Raccolta validi.</u> <u>Tag stringa di</u> <u>caratteri&lt;1</u> <u>[ERROR]</u>		<u>{Nome: "millefoglie".</u> <u>Ingrediente: "crema e</u> <u>fragolina".</u> <u>Descrizione:"descrizione</u> <u>breve "Tempo: 50, tag:"".</u> <u>Esistenza Raccolta=True }</u>	<u>Il Tag non è</u> <u>stato inserito</u>	
8	<u>Tag stringa</u> <u>contenente</u> <u>carattere</u> <u>speciale al di</u> <u>fuori di #</u>	<u>Titolo.</u> <u>Ingrediente.</u> <u>descrizione.</u> <u>Tempo validi.</u> <u>Tag non valido</u> <u>[ERROR].</u> <u>Esistenza</u> <u>Raccolta valido</u>		<u>{Nome: "millefoglie".</u> <u>Ingrediente: "crema e</u> <u>fragolina".</u> <u>Descrizione:"descrizione</u> <u>breve "Tempo: 50,</u> <u>tag:"@dolce".</u> <u>Esistenza Raccolta=True }</u>	<u>Il tag inserito</u> <u>non è</u> <u>corretto</u>	
9	<u>Raccolta non</u> <u>esistenza</u> <u>(Esistenza R</u> <u>accolta=False</u> <u>)</u>	<u>Titolo.</u> <u>Ingrediente.</u> <u>descrizione.</u> <u>Tempo, Tag</u> <u>valido.</u> <u>Esistenza Raccol</u> <u>ta non valido</u>		<u>{Nome: "caprese",Ingrediente:</u> <u>"cioccolato".</u> <u>Descrizione:"desc".</u> <u>Tag:"dolce",tempo:15,</u> <u>Esistenza Raccolta=False}</u>	<u>La raccolta</u> <u>non è</u> <u>esistente.</u> <u>creane una</u> <u>nuova</u>	



### 3.2 Commento

COMMENTO			
AUTORE	TESTO	GIORNO	MESE
<ul style="list-style-type: none"> <li>Stringa di caratteri di lunghezza &lt;= 40</li> <li>Stringa di caratteri di lunghezza &gt; 40 [ERROR]</li> <li>Stringa di caratteri di lunghezza &lt;1[ERROR]</li> </ul>	<ul style="list-style-type: none"> <li>Stringa di caratteri di lunghezza &gt;= 2</li> <li>Stringa di caratteri di lunghezza &lt;= 500</li> <li>Stringa di caratteri di lunghezza &gt;500[ERROR]</li> <li>Stringa di caratteri di lunghezza &lt;1[ERROR]</li> <li>Stringa che contiene simboli che non sono caratteri [ERROR]</li> </ul>	<ul style="list-style-type: none"> <li>Numero intero &gt;=1</li> <li>Numero intero&lt;=31</li> <li>Numero intero &lt;1[ERRORE]</li> <li>Numero intero &gt;31[ERROR]</li> <li>Giorno non è un numero intero[ERROR]</li> </ul>	<ul style="list-style-type: none"> <li>Stringa di caratteri=Gennaio</li> <li>Stringa di caratteri=Febbraio</li> <li>Stringa di caratteri=Marzo</li> <li>Stringa di caratteri=Aprile</li> <li>Stringa di caratteri=Maggio</li> <li>Stringa di caratteri=Giugno</li> <li>Stringa di caratteri=Luglio</li> <li>Stringa di caratteri=Agosto</li> <li>Stringa di caratteri=Settembre</li> <li>Stringa di caratteri=Ottobre</li> <li>Stringa di caratteri=Novembre</li> <li>Stringa di caratteri=Dicembre</li> <li>Stringa di caratteri diversa di quelle elencate[ERROR]</li> <li>Stringa di caratteri di lunghezza&lt;1[ERROR]</li> </ul>

Il numero di test da effettuarsi senza particolari vincoli è:  $3 * 5 * 5 * 14 = 1050$ .

Con i vincoli [ERROR], invece, il numero di test da eseguire per testare singolarmente i vincoli è 10 (2 per Autore, 3 per Testo, 3 per Giorno, 2 per Mese).

Il numero di test risultante è:  $(1 \cdot 2 \cdot 2 \cdot 12) + 10 = 58$ .

TEST SUITE						
Test Case ID	Descrizione	Classi di equivalenza coperte	Pre-condizioni	Input	Output attesi	Post-condizioni attese
1	Tutti input validi	Autore, Testo, Giorno, Mese validi		{Autore: "Mario", Test:"ok",Giorno: 3,Mese:Luglio }	Commento registrato	Il sistema incrementa il contatore dei commenti della ricetta
2	Autore stringa > 40 caratteri	Autore stringa > 40 caratteri [ERROR], Testo, Giorno, Mese validi		{Nome: "Mariooooooooooooooooooooo oooooooooooooooooooooooooooo oooooooooooooooooooooooooooo oooooooooooooo Testo:"ok",Giorno: 3,Mese:Luglio}	Autore troppo lungo	
3	Autore stringa < 1 caratteri	Autore stringa > 40 caratteri [ERROR],		{Nome: "" Test:"ok",Giorno: 3,Mese:Luglio	L'autore non è presente	



		<u>Testo.</u> <u>Giorno, Mese</u> <u>validi</u>				
<u>4</u>	<u>Testo stringa</u> <u>&lt;2 caratteri</u>	<u>Autore valido.</u> <u>Testo &lt; 2</u> <u>caratteri</u> [ERROR], <u>Giorno.</u> <u>Mese validi</u>		<u>{Nome: "Mario".</u> <u>Testo:"i",Giorno:3,Mese:"Lugli</u> <u>o" }</u>	<u>Testo di</u> <u>lunghezza</u> <u>troppo corta</u>	
<u>5</u>	<u>Testo</u> <u>Stringa&gt;500</u> <u>caratteri</u>	<u>Autore valido.</u> <u>Testo &gt;500</u> <u>caratteri</u> [ERROR], <u>Giorno.</u> <u>Mese validi</u>		<u>{Nome: "Mario".</u> <u>Testo:"commento.....",Giorn</u> <u>o:3,Mese:"Luglio" }</u>	<u>Testo troppo</u> <u>lungo!</u>	
<u>6</u>	<u>Testo con</u> <u>stringa</u> <u>contenente</u> <u>caratteri</u> <u>speciali</u>	<u>Autore valido.</u> <u>Testo caratteri</u> <u>speciali [ERROR].</u> <u>Giorno, Mese.</u> <u>validi</u>		<u>{Nome: "Mario".</u> <u>Testo:"cia@o"</u> <u>Giorno:3,Mese:"Luglio" }</u>	<u>Testo non</u> <u>consentito a</u> <u>causa dei</u> <u>caratteri</u> <u>speciali</u>	
<u>7</u>	<u>Giorno &gt;31</u>	<u>Autore valido.</u> <u>Testo valido.</u> <u>Giorno&gt;31[ERRO</u> <u>R]. Mese, validi</u>		<u>{Nome: "Mario", Testo:"ciao"</u> <u>Giorno:32,Mese:"Luglio" }</u>	<u>Giorno &gt;31</u>	
<u>8</u>	<u>Giorno&lt;1</u>	<u>Autore valido.</u> <u>Testo valido.</u> <u>Giorno&lt;1[ERRO</u> <u>R]. Mese validi</u>		<u>{Nome: "Mario", Testo:"ciao"</u> <u>Giorno:1,Mese:"Luglio" }</u>	<u>Giorno&lt;1</u>	
<u>9</u>	<u>Giorno che</u> <u>non è un</u> <u>intero</u>	<u>Autore valido.</u> <u>Testo valido.</u> <u>Giorno che non è</u> <u>intero[ERROR].</u> <u>Mese validi</u>		<u>{Nome: "Mario", Testo:"ciao"</u> <u>Giorno:2.3,Mese:"Luglio" }</u>	<u>Giorno deve</u> <u>essere un</u> <u>numero</u> <u>intero</u>	
<u>10</u>	<u>Mese che non</u> <u>è tra quelli</u> <u>consentiti</u>	<u>Autore valido.</u> <u>Testo valido.</u> <u>Giorno valido.</u> <u>Mese diverso da</u> <u>quelli</u> <u>consentiti[ERRO</u> <u>R]</u>		<u>{Nome: "Mario", Testo:"ciao"</u> <u>Giorno:2,Mese:"Ciao" }</u>	<u>Mese deve</u> <u>essere uguale</u> <u>a quelli</u> <u>consentiti</u>	
<u>11</u>	<u>Mese Stringa</u> <u>&lt;1 caratteri</u>	<u>Autore valido.</u> <u>Testo valido.</u> <u>Giorno valido.</u> <u>Mese</u> <u>Stringa&lt;1[ERRO</u> <u>R].</u>		<u>{Nome: "Mario", Testo:"ciao"</u> <u>Giorno:1,Mese:"" }</u>	<u>Il mese non è</u> <u>stato</u> <u>specificato</u>	

GENERA REPORT AUTORE				
AUTORE	RICETTA	GIORNO	MESE	ANNO
<ul style="list-style-type: none"> <li>Stringa di caratteri di lunghezza &lt;= 40</li> <li>Stringa di caratteri di lunghezza &gt; 40 [ERROR]</li> <li>Stringa di caratteri di lunghezza &lt;1[ERROR]</li> </ul>	<ul style="list-style-type: none"> <li>Vettore di ricette &gt;=1</li> <li>Vettore di ricetta &lt;1[ERROR]</li> </ul>	<ul style="list-style-type: none"> <li>Numero intero &gt;=1</li> <li>Numero intero &lt;=31</li> <li>Numero intero &lt;1[ERRORE]</li> <li>Numero intero &gt;31[ERROR]</li> <li>Giorno non è un numero intero[ERROR]</li> </ul>	<ul style="list-style-type: none"> <li>Stringa di caratteri=Gennaio</li> <li>Stringa di caratteri=Febbraio</li> <li>Stringa di caratteri=Marzo</li> <li>Stringa di caratteri=Aprile</li> <li>Stringa di caratteri=Maggio</li> <li>Stringa di caratteri=Giugno</li> <li>Stringa di caratteri=Luglio</li> <li>Stringa di caratteri=Agosto</li> <li>Stringa di caratteri=Settembre</li> <li>Stringa di caratteri=Ottobre</li> <li>Stringa di caratteri=Novembre</li> <li>Stringa di caratteri=Dicembre</li> <li>Stringa di caratteri diversa di quelle elencate[ERROR]</li> <li>Stringa di caratteri di lunghezza&lt;1[ERROR]</li> </ul>	<ul style="list-style-type: none"> <li>Numero intero &gt;=2020</li> <li>Numero intero &lt;=2025</li> <li>Numero intero &lt;2020[ERROR]</li> <li>Numero intero &gt;2025[ERROR]</li> <li>Numero che non è un intero[ERROR]</li> </ul>

Il numero di test da effettuarsi senza particolari vincoli è:  $3 * 2 * 5 * 14 * 5 = 2100$ .

Con i vincoli [ERROR], invece, il numero di test da eseguire per testare singolarmente i vincoli è 11 (2 per Autore, 1 per Ricetta, 3 per Giorno, 2 per Mese, 3 per Anno).

Il numero di test risultante è:  $(1*1*2*12*2) + 11 = 59$ .

TEST SUITE						
Test Case ID	Descrizione	Classi di equivalenza coperte	Pre-condizioni	Input	Output attesi	Post-condizioni attese
1	Tutti input validi	Autore, Ricetta Giorno, Mese, Anno validi		{Autore: "Mario", Ricetta:3, Giorno: 3, Mese: Luglio, Anno:2024 }	Report Effettuato	Il sistema registrata il report effettuato dall'amministratore
2	Autore stringa > 40 caratteri	Autore stringa > 40 caratteri [ERROR], Ricetta, Giorno, Mese, Anno validi		{Nome: "Mariooooooooooooooooooooo oooooooooooooooooooooooooooo oooooooooooooooooooooooooooo ooooooooooooo Ricetta:1, Giorno: 3, Mese: Luglio, Anno:2024 }	Autore troppo lungo	
3	Autore stringa < 1 caratteri	Autore stringa < 1 caratteri [ERROR], Ricetta,		{Nome: "" Ricetta:4, Giorno: 3, Mese: Luglio, Anno:2024"}	L'autore non è presente	



		<u>Giorno, Mese, Anno validi</u>				
4	<u>Ricetta&lt;1</u>	<u>Autore, Giorno, Mese, Anno validi</u> <u>Ricetta&lt;1</u>		<u>{Nome: "Michele"</u> <u>Ricetta:0, Giorno:</u> <u>3, Mese: Luglio, Anno: 2024"}</u>	<u>L'autore non ha pubblicato ricette</u>	
5	<u>Giorno &gt;31</u>	<u>Autore valido,</u> <u>Ricetta valido,</u> <u>Giorno&gt;31[ERROR],</u> <u>Mese, Anno validi</u>		<u>{Nome: "Mario", Ricetta:3</u> <u>Giorno:32, Mese:"Luglio", Anno:</u> <u>2023}</u>	<u>Giorno &gt;31</u>	
6	<u>Giorno&lt;1</u>	<u>Autore valido,</u> <u>Ricetta valido,</u> <u>Giorno&lt;1[ERROR],</u> <u>Mese, Anno validi</u>		<u>{Nome: "Mario", Ricetta:3</u> <u>Giorno:1, Mese:"Luglio", Anno:</u> <u>2023}</u>	<u>Giorno&lt;1</u>	
7	<u>Giorno che non è un intero</u>	<u>Autore valido,</u> <u>Ricetta valido,</u> <u>Giorno che non è intero[ERROR],</u> <u>Mese, Anno validi</u>		<u>{Nome: "Mario", Ricetta:3</u> <u>Giorno:2.3, Mese:"Luglio", Anno:</u> <u>2023}</u>	<u>Giorno deve essere un numero intero</u>	
8	<u>Mese che non è tra quelli consentiti</u>	<u>Autore valido,</u> <u>Ricetta valido,</u> <u>Giorno valido,</u> <u>Mese diverso da quelli consentiti[ERROR],</u> <u>Anno validi</u>		<u>{Nome: "Mario", Ricetta:3</u> <u>Giorno:2, Mese:"Ciao", Anno:20</u> <u>23}</u>	<u>Mese deve essere uguale a quelli consentiti</u>	
9	<u>Mese Stringa &lt;1 caratteri</u>	<u>Autore valido,</u> <u>Ricetta valido,</u> <u>Giorno valido,</u> <u>Mese Stringa&lt;1[ERROR],</u> <u>Anno validi</u>		<u>{Nome: "Mario", Ricetta:3</u> <u>Giorno:1, Mese:"", Anno:2022}</u>	<u>Il mese non è stato specificato</u>	
13	<u>Anno &lt;2020</u>	<u>Autore valido,</u> <u>Ricetta valido,</u> <u>Giorno valido,</u> <u>Mese valido,</u> <u>Anno&lt;2020 [ERROR]</u>		<u>{Nome: "Mario", Ricetta:3</u> <u>Giorno:7, Mese:"Agosto", Anno:</u> <u>2019}</u>	<u>L'anno deve essere compreso tra 2020 e 2025</u>	
14	<u>Anno&gt;2025</u>	<u>Autore valido,</u> <u>Ricetta valido,</u> <u>Giorno valido,</u> <u>Mese valido,</u> <u>Anno&lt;2025 [ERROR]</u>		<u>{Nome: "Mario", Ricetta:3</u> <u>Giorno:5, Mese:"Dicembre", Anno:</u> <u>2026}</u>	<u>L'anno deve essere compreso tra 2020 e 2025</u>	
15	<u>Anno non è un intero</u>	<u>Autore valido,</u> <u>Ricetta valido,</u> <u>Giorno valido,</u> <u>Mese valido,</u> <u>Anno non è un intero[ERROR]</u>		<u>{Nome: "Mario", Ricetta:3</u> <u>Giorno:5, Mese:"Dicembre", Anno:</u> <u>2023.5}</u>	<u>L'anno deve essere un intero</u>	



### 3.3 LOGIN

EMAIL	PASSWORD
<ul style="list-style-type: none"><li>• <u>Stringa alfanumerica</u> <u><math>\geq 5</math> e <math>\leq 64</math></u></li><li>• <u>Stringa alfanumerica</u> <u>contenente il</u> <u>simbolo '@'</u></li><li>• <u>Stringa alfanumerica</u> <u>contenente il simbolo</u> <u>'.'</u></li><li>• <u>Stringa alfanumerica</u> <u><math>&lt; 5</math> [ERROR]</u></li><li>• <u>Stringa alfanumerica</u> <u><math>&gt; 64</math> [ERROR]</u></li><li>• <u>Stringa alfanumerica</u> <u>contenente caratteri</u> <u>speciali al di fuori</u> <u>della '@' (come %,</u> <u>\$, &amp;, etc.)</u> <u>[ERROR]</u></li><li>• <u>Stringa alfanumerica</u> <u>non contenente il</u> <u>simbolo '@'</u> <u>[ERROR]</u></li><li>• <u>Stringa alfanumerica</u> <u>non contenente il</u> <u>simbolo '.'</u> <u>[ERROR]</u></li></ul>	<ul style="list-style-type: none"><li>• <u>Stringa alfanumerica</u> <u><math>\geq 6</math> e <math>\leq 16</math></u></li><li>• <u>Stringa alfanumerica</u> <u>contenente almeno un</u> <u>numero</u></li><li>• <u>Stringa alfanumerica</u> <u>contenente almeno un</u> <u>simbolo speciale</u></li><li>• <u>Stringa alfanumerica</u> <u><math>&lt; 6</math> [ERROR]</u></li><li>• <u>Stringa alfanumerica</u> <u><math>&gt; 16</math> [ERROR]</u></li><li>• <u>Stringa alfanumerica</u> <u>che non contiene</u> <u>nessun numero</u> <u>[ERROR]</u></li><li>• <u>Stringa alfanumerica</u> <u>che non contiene</u> <u>nessun carattere</u> <u>speciale [ERROR]</u></li></ul>

Il numero di test da effettuarsi senza particolari vincoli è:  $8 * 7 = 56$ .

Con i vincoli [ERROR], invece, il numero di test da eseguire per testare singolarmente i vincoli è 10 (6 per Email, 4 per Password).

Il numero di test risultante è  $(3*3)+10=19$





TEST SUITE						
Test Case ID	Descrizione	Classi di equivalenza coperte	Pre-condizioni	Input	Output attesi	Post-condizioni attese
1	Tutti input validi	Email, Password		{Email:micheleamoroso@gmail.com,Password:"Michele1!"}	Utente Loggato	l'utente si è loggato al sistema
2	Email Stringa alfanumerica <5	Email Stringa<5[ERROR] Password valida		{Email:m@gm.c,Password:"Michele1!"}	Email non valida	
3	Email Stringa non contenente la @	Email no @[ERROR],Password valida		{Email:micheleamorosogmail.com,Password:"Michele1!"}	Email non valida	
4	Email Stringa non contenente il .	Email no .[ERROR], Password valida		{Email:luigiamoroso@gmailcom,Password:"Luigi1!"}	Email non valida	
5	Password Stringa alfanumerica <6	Email valida, Password<6[ERROR]		{Email:lucaanastasio3@libero.it,Password:"luc1!"}	Password non valida	
6	Password non contenente numeri	Email valida, password no numeri[ERROR]		{Email:lucaanastasio3@libero.it,Password:"lucaAna!"}	Password non valida	
7	Password che non contiene caratteri speciali	Email valida, password no caratteri speciali[ERROR]		{Email:lucaanastasio3@libero.it,Password:"ciaociao1!"}	Password non valida	



## DIAGRAMMA DELLE CLASSI DI PROGETTAZIONE SECONDO IL MODELLO BCED



### 4.1.1 Traduzione classi ed associazioni

#### PACKAGE BOUNDARY:

nel package boundary si è inserito:

- la classe BoundaryUtenteNonRegistrato che fornisce il metodo di registrazione per l'utente
- la classe BoundaryAutore che fornisce i metodi di login(), creaRicetta(), modificaProfilo(), creazioneRaccolta(), aggiungiCommento(), mostraFeed() e mostraMieRicette() e logout().
- La classe BoundaryStart fornisce i metodi registrazione() e login()

La classe BoundaryStart istanzia la GUI di partenza per far sì che l'utente possa scegliere di effettuare il login o la registrazione.

Tale classe richiama sia la classe BoundaryAutoreNonRegistrato che possiede il metodo di registrazione e la classe BoundaryAutore che possiede il metodo di login.

Quando avviene ciò automaticamente, a seconda della scelta fatta dall'autore, viene invocato il metodo di gestisciRegistrazione() di controlAutoreNonRegistrato e gestisciLogin() della classe ControlAutore del package Control.

#### PACKAGE CONTROL:

Nel package Control si è inserito:

- La classe ControlUtenteNonRegistrato che fornisce il metodo Registrazione()
- La classe ControllAutore che fornisce i metodi login(), creaRicetta(), modificaProfilo(), creazioneRaccolta(), aggiungiCommento(), mostraFeed() e mostraMieRicette() e logout()

Nel package control sono implementate le responsabilità individuate dall'analisi dei casi d'uso come ad esempio login e logout sia per autore, la creazione delle ricette da parte dell'autore e...

Tale package fa da intermediario tra il Boundary e il package Entity dove quest'ultimo fornisce i dati acceduti a loro volta tramite il package Database.

#### PACKAGE ENTITY

Il package Entity raccoglie le classi già individuate in fase di analisi del dominio.

Nelle classi del package Entity sono presenti metodi che servono per salvare "l'entità" nel Database.

Le classi inserite nel package Entity sono quelle che serviranno a realizzare il nostro sistema software.

La persistenza delle informazioni non è necessaria in tutte le classi.



## PACKAGE DAO

Tale package serve realizzare la comunicazione tra l'applicazione e il DBMS.

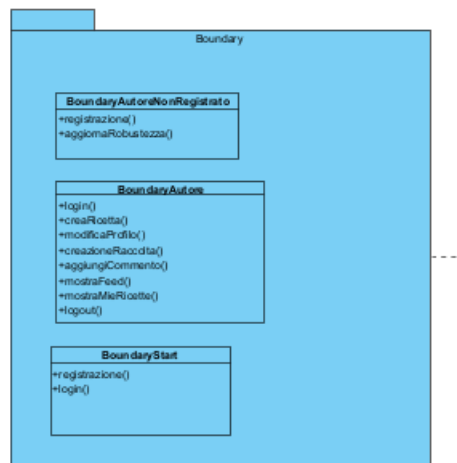
In tale package abbiamo inserito 6 classi ossia AutoreDAO, CommentoDAO, RicettaDAO, ProfiloDAO, RaccoltaDAO, IngredienteDAO.

I metodi presenti nelle classi sono create, read, update, delete e SalvaSuDB che sono metodi pubblici e un metodo privato per ognuno di loro che è quello di ExecuteQuery().

### 4.1.2 Pattern BCED

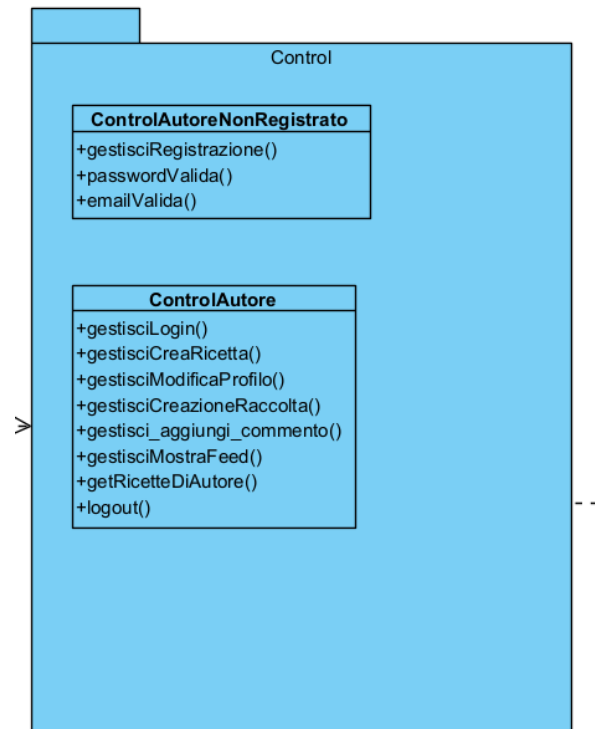
#### 4.1.2.1 Package Boundary

Il package Boundary contiene tutti gli oggetti responsabili dell'interfaccia utente e della logica di presentazione; a questo livello tutte le classi corrispondono a delle interfacce e i relativi attributi non sono altro che gli elementi che le compongono, visualizzati a video.



#### 4.1.2.2 Package Controller

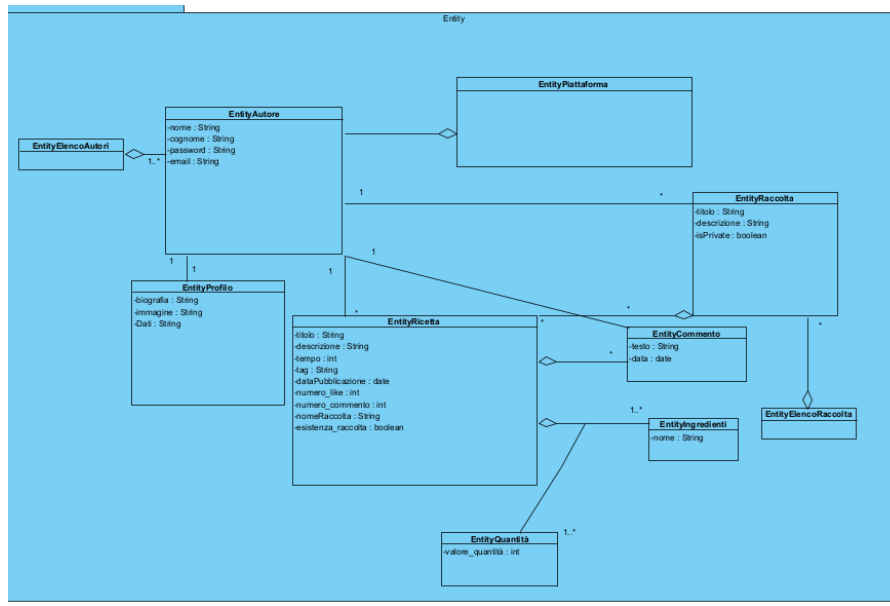
Questo package contiene gli oggetti che percepiscono gli eventi generati dalle interazioni con l'interfaccia utente e ne demandano la gestione all'unico componente del sistema software responsabile della gestione della Business Logic, il package Entity.



#### 4.1.2.3 Package Entity

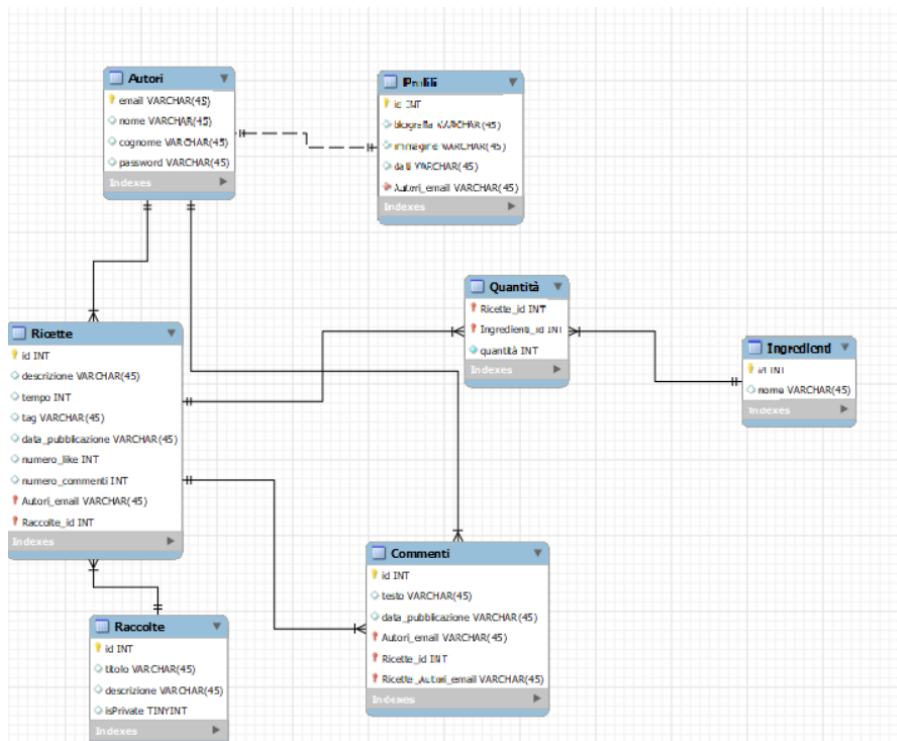
Il Package Entity contiene tutti gli oggetti che rappresentano la semantica delle entità del dominio applicativo e corrispondono alle strutture dati presenti all'interno del database di persistenza.





#### 4.1.2.4 Package Database

Schema E/R:

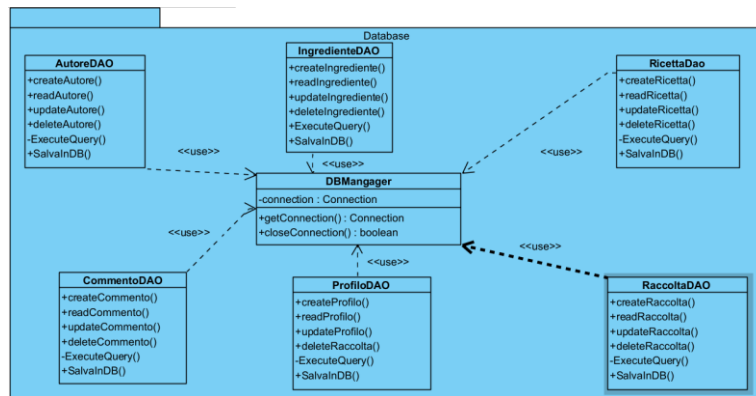


Da questo punto di vista, il Package Database contiene tutte le classi responsabili dell'estrazione dei dati dal DB, esponendo una vera e propria interfaccia che di fatto rende indipendenti le classi della Business Logic (Entity) dalla tecnologia di persistenza utilizzata.

In particolare, tra le strategie di risoluzione del problema dell'**impedance mismatch**, che nasce dalla mancata corrispondenza tra il modello Object Oriented e quello relazionale, si è deciso di adottare quella delle classi **DAO** (Data Access Objects), che consiste nell'utilizzo di appositi oggetti per l'accesso ai dati.



Ognuna di queste classi conterrà i metodi CRUD per l'interrogazione e la manipolazione della corrispondente classe di dominio (*query*), implementati in funzione di un'ulteriore classe, **DBManager**, che costituisce di fatto l'unico punto di accesso vero e proprio al DB, sfruttando i metodi che questa mette a disposizione.

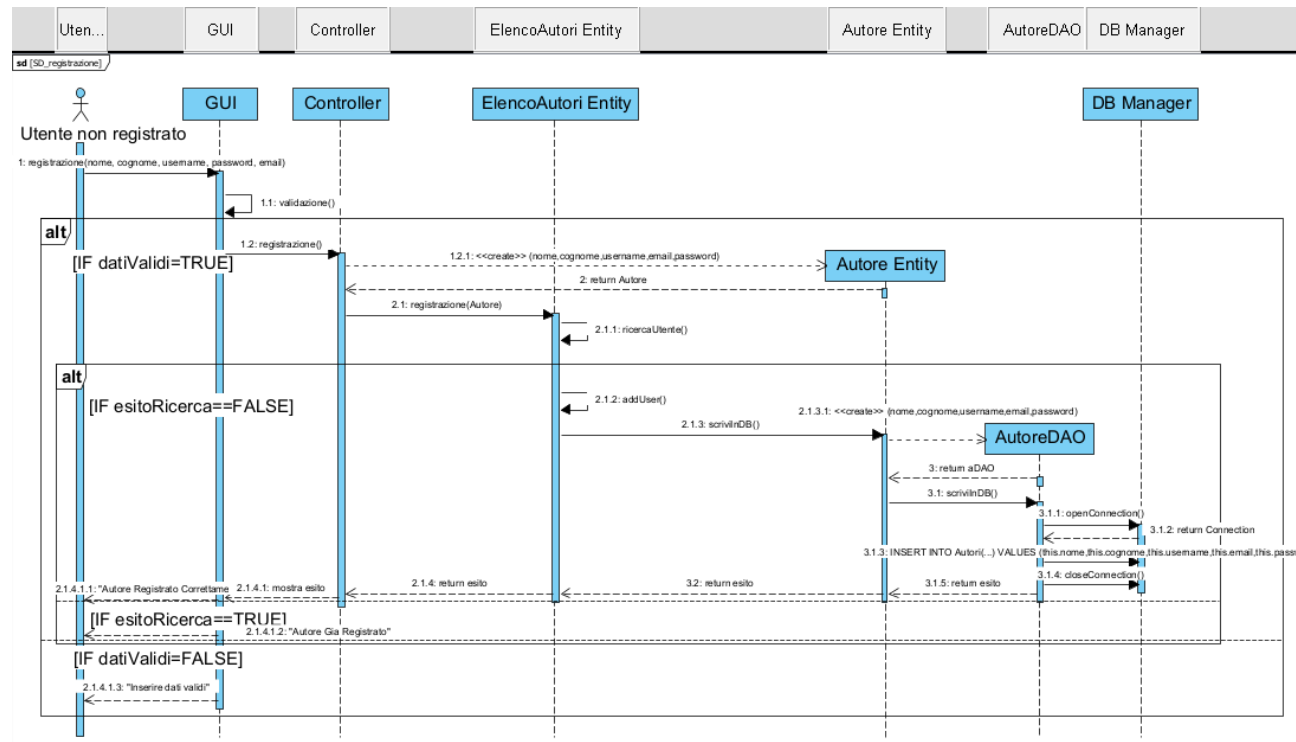


## 4.2 Diagrammi di sequenza

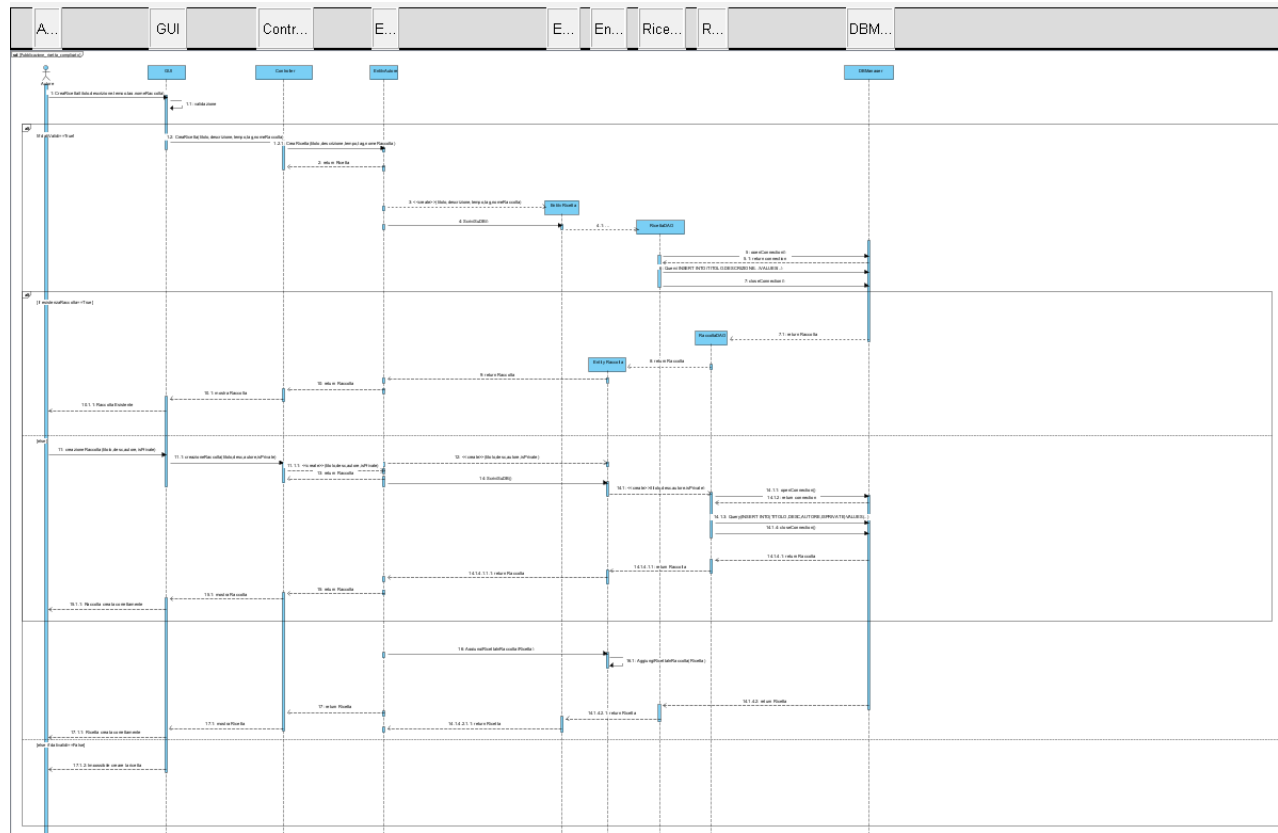




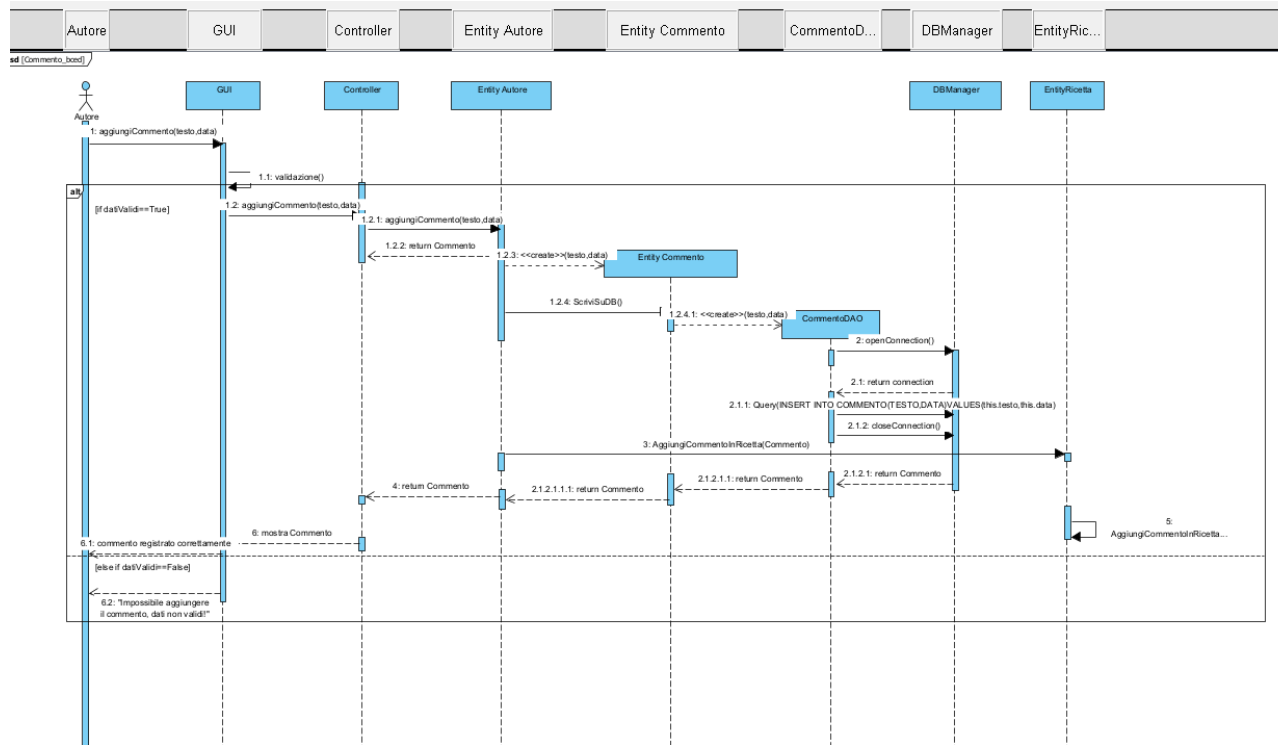
## 4.2.1 Registrazione



## 4.2.2 Pubblicazione Ricetta

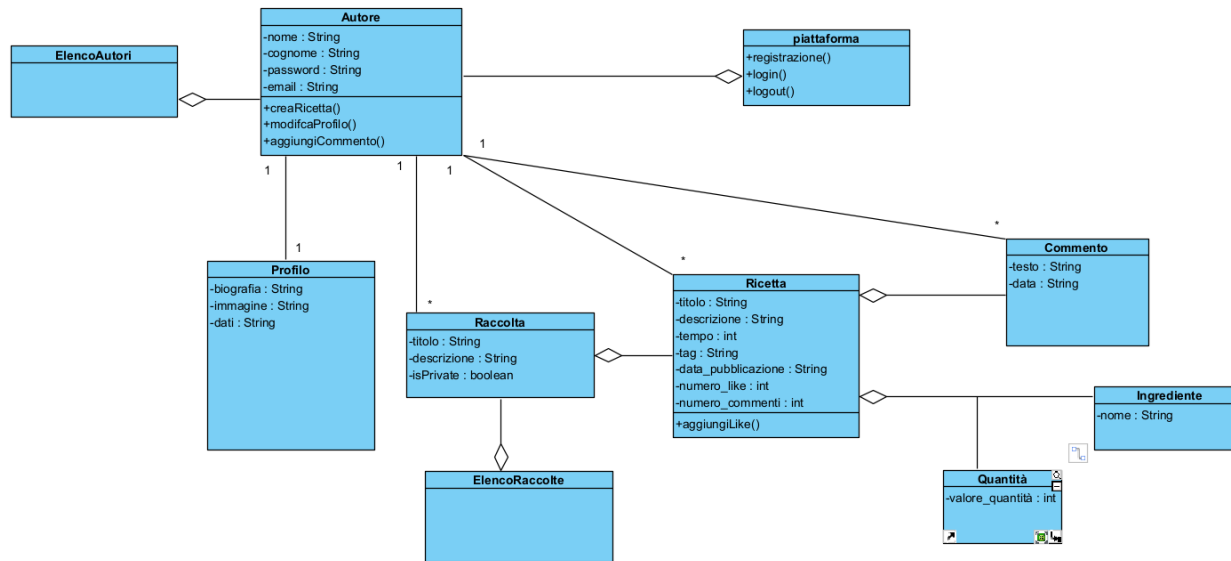


## 4.2.3 Commento



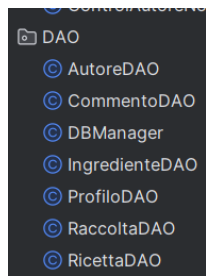
I sequence progettati sono stati fondamentali per la corretta implementazione dell'applicazione software ed ha fatto nascere la necessità di definire ulteriori classi, metodi e funzioni, che hanno arricchito passo dopo passo il **Diagramma delle Classi di Progettazione**, fino ad ottenere la seguente versione finale:





## 5. Implementazione

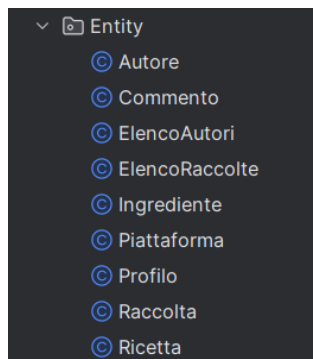
### 5.1 Package Database



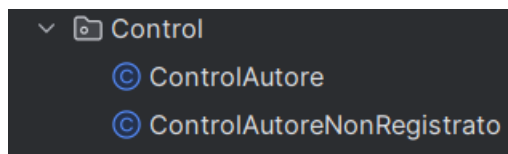
Per la connessione con il database abbiamo usato il driver: `mysql-connector-j-8.0.33.jar`



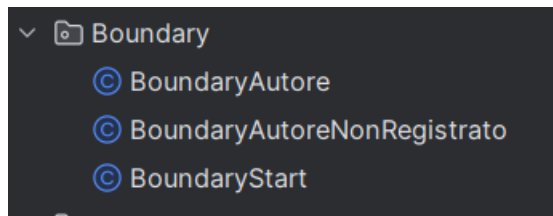
## 5.2 Package Entity



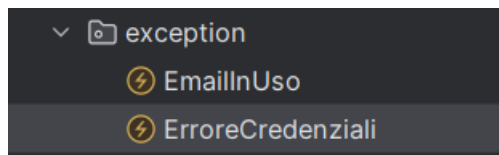
## 5.3 Package Controller



## 5.4 Package Boundary

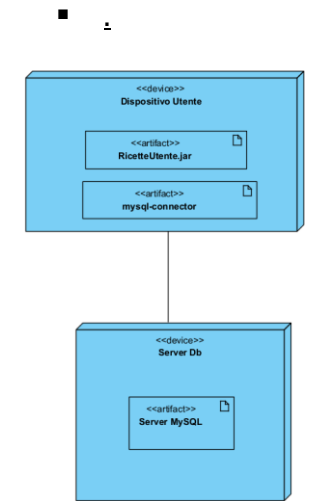


## 5.5 Package Exception



## 5.6 Diagramma di Deployment

I diagrammi di deployment sono utilizzati per mostrare l'architettura fisica del sistema software realizzato; sono particolarmente utili per valutare, durante lo sviluppo, come un'applicazione si distribuisce tra le varie macchine.



La distribuzione del software prevede una versione per l'uso da parte degli Autori che possono essere registrati al sistema oppure non registrati dove quelli registrati possono creare le ricette, creare le raccolte, visualizzare il feed e ...

Quindi tale versione corrisponde all'esecuzione di BoundaryStart che possiede sia BoundaryAutoreNonRegistrato e sia BoundaryAutore.

## 6. Testing(Test Strutturale)

### 6.1.1 Complessità ciclomatica

Si intende costruire il Control Flow Graph per due dei metodi delle classi implementate e mostrare il calcolo del numero ciclomatico e i percorsi linearmente indipendenti.

#### 6.1.1.1 creaRicetta - emailValida

```
public void creaRicetta(Ricetta nuovaRicetta) {  
    try {  
        String titolo = nuovaRicetta.getTitolo(); // (1)  
        if (titolo == null || titolo.trim().isEmpty() || titolo.length() > 100) { // (2)  
            throw new IllegalArgumentException("titolo non valido"); // (22)  
        }  
        String descrizione = nuovaRicetta.getDescrizione(); // (3)  
        if (descrizione == null || descrizione.isEmpty() || descrizione.length() > 800) { // (4)  
            throw new IllegalArgumentException("La descrizione non può essere vuota e non può superare 800 caratteri."); // (22)  
        }  
        int tempo = nuovaRicetta.getTempo(); // (5)  
        if (tempo <= 0) { // (6)  
            throw new IllegalArgumentException("Il tempo deve essere un numero positivo."); // (22)  
        }  
        String tag = nuovaRicetta.getTag(); // (7)  
        if (tag == null || tag.trim().isEmpty()) { // (8)  
            throw new IllegalArgumentException("Il Tag non può essere vuoto."); // (22)  
        }  
    }  
  
    String titoloRaccolta = nuovaRicetta.getRaccolta().getTitolo(); // (9)  
    String descrizioneRaccolta = nuovaRicetta.getRaccolta().getDescrizione(); // (10)  
    String emailAutore = nuovaRicetta.getAutore().getEmail(); // (11)  
    RaccoltaDAO daoRac = new RaccoltaDAO(); // (12)  
    int idRaccolta = daoRac.trovaIdPerTitoloEAutore(titoloRaccolta, emailAutore); // (13)  
    if (idRaccolta == 0) { // (14)  
        Raccolta nuova = new Raccolta(); // (15)  
        nuova.setTitolo(titoloRaccolta); // (15)
```



```

    nuova.setDescrizione(descrizioneRaccolta); // (15)
    nuova.setPrivate(false); // (15)
    nuova.setAutore(nuovaRicetta.getAutore()); // (15)
    Raccolta inserita = creaRaccolta(nuova); // (16)
    idRaccolta = inserita.getId(); // (17)
}
int idRicetta = (int) (System.currentTimeMillis() % Integer.MAX_VALUE); // (18)
RicettaDAO daoRic = new RicettaDAO(); // (19)
daoRic.setId(idRicetta); // (19)
daoRic.scriviSuDB(); // (20)
} catch (Exception e) { // (Exception)
throw new RuntimeException("Errore DB durante creaRicetta: " + e.getMessage(), e) // (22); }}

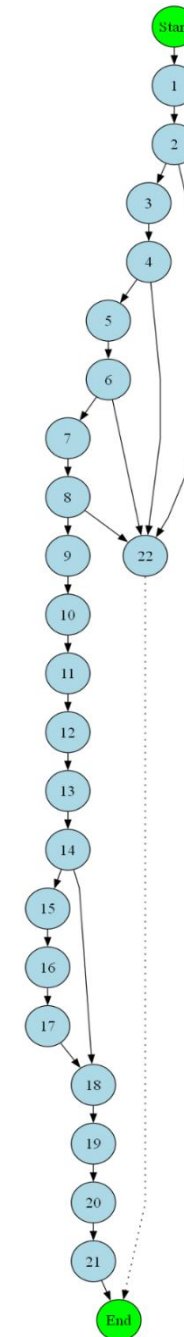
```

### NUMERO CICLOMATICO:

- numero di regioni chiuse del grafo + 1 = 6
- numero di nodi predicato (5) + 1 = 6
- # archi - # nodi + 2 = (26 - 22) + 2 = 6

### CAMMINI:

- 1) 1-2-22
- 2) 1-2-3-4-22
- 3) 1-2-3-4-5-6-22
- 4) 1-2-3-4-7-6-7-8-22
- 5) 1-2-3-4-7-8-9-10-11-12-13-14-18-19-20-21
- 6) 1-2-3-4-7-8-9-10-11-12-13-14-15-16-17-18-19-20-21





```

private boolean emailValida(String email) {           // Start
    if (email == null)                                // (1)
        return false;                                // (2)

    String regexEmail = "^\\w.-]+@\\w.-]+\\.\\w{2,}$"; // (3)
    if (!email.matches(regexEmail))                   // (4)
        return false;                                // (5)

    String[] dominiAmmessi = {"gmail.com", "libero.it", "outlook.com", "studenti.unina.it", "unina.it"}; // (6)
    String dominio = email.substring(email.lastIndexOf("@") + 1).toLowerCase(); // (7)

    for (String d : dominiAmmessi) {                   // (8)
        if (dominio.equals(d))                         // (9)
            return true;                               // (10)
    }

    return false;                                     // (11)
}                                                       // End

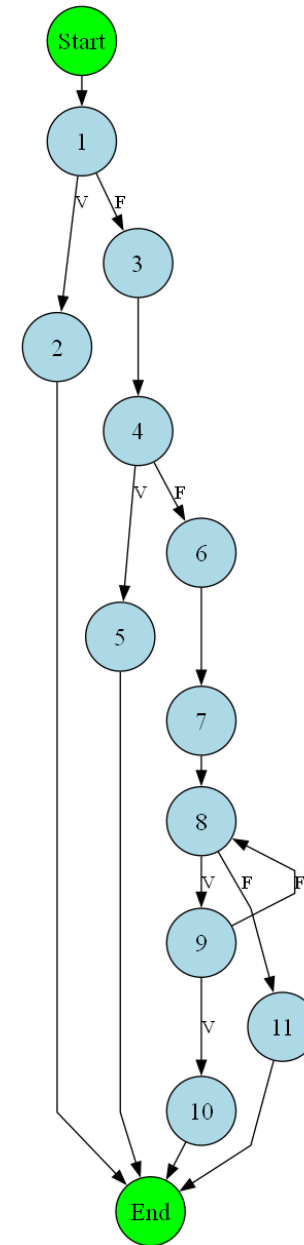
```

### NUMERO CICLOMATICO:

- numero di regioni chiuse del grafo + 1 = 5
- numero di nodi predicato (4) + 1 = 5
- # archi - # nodi + 2 = (16 - 13) + 2 = 5

### CAMMINI:

- 1) 1-2
- 2) 1-3-4-5
- 3) 1-3-4-6-7-8-9-10
- 4) 1-2-3-4-7-6-8-9-8-9-10
- 5) 1-3-4-6-7-8-11

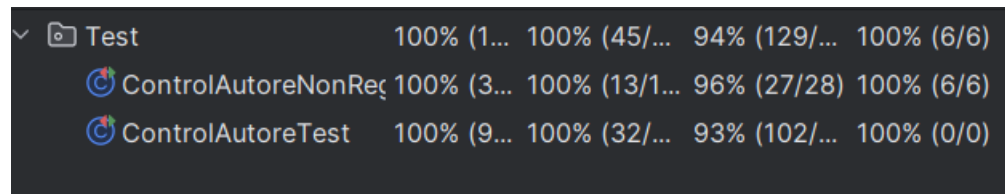


## 6.2 JUnit – Test di Unità

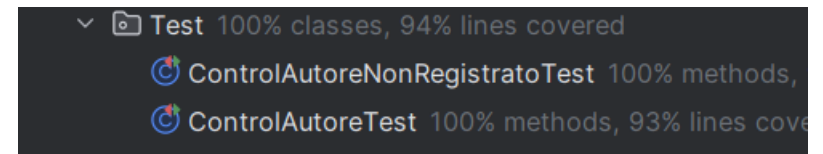
Per il testing in JUnit, abbiamo scelto le classe con più logica e controlli, le classi del package Control.

Inoltre per ragioni di complessità e leggibilità, si è reso necessario creare un Database “fittizio” per evitare di invalidare il database reale.

Riportando la seguente coverage:



Test	100% (1...)	100% (45/...)	94% (129/...)	100% (6/6)
ControlAutoreNonReg	100% (3...)	100% (13/1...)	96% (27/28)	100% (6/6)
ControlAutoreTest	100% (9...)	100% (32/...)	93% (102/...)	100% (0/0)



Test	100% classes, 94% lines covered
ControlAutoreNonRegistratoTest	100% methods,
ControlAutoreTest	100% methods, 93% lines covered

## 6.3 Test funzionale

Segue una descrizione in forma tabellare dei risultati dell'esecuzione dei test funzionali precedentemente pianificati.

creaRicetta									
TEST SUITE									
Test Case ID	Descrizione	Classi di equivalenza coperte	Pre-condizioni	Input	Output attesi	Post-condizioni attese	Output ottenuti	Post-condizioni ottenute	Esito (FAIL, PASS)
1	Tutti input validi	Titolo valido Descrizione valida Tempo valido Tag validi Raccolta valida	L'utente è presente nel database come autore	{Titolo: "Pizza", Descrizione: "Margherita", Tempo: "20", Tag: "#italian", Raccolta: "Piatti Italiani"}	Ricetta registrata	La ricetta è visibile dal profilo dell'autore	Pubblicazione avvenuta	La ricetta è stata pubblicata ed è ora possibile interagire con essa	PASS





						<u>almeno un tag valido</u>		<u>almeno un tag valido</u>	
--	--	--	--	--	--	-----------------------------	--	-----------------------------	--

### emailValida

TEST SUITE									
Test Case ID	Descrizione	Classi di equivalenza coperte	Pre-condizioni	Input	Output attesi	Post-condizioni attese	Output ottenuti	Post-condizioni ottenute	Esito (FAIL, PASS)
1	Email Valida	Email corretta	L'utente si vuole registrare alla piattaforma, ed ha inserito una password valida	{Email: "autore1@unina.it"}	Email valida, continua la registrazione	L'utente prosegue la registrazione	Email valida!	L'utente viene registrato e inserito nel db	PASS
2	Email vuota	Email non valida	L'utente si vuole registrare alla piattaforma, ed ha inserito una password valida	{Email: ""}	Email vuota!	L'utente viene reindirizzato o ad inserire una mail valida	Errore, la mail non può essere vuota!	L'utente viene reindirizzato o ad inserire una mail valida	PASS
3	Email non ammessa	Email non valida	L'utente si vuole registrare alla piattaforma, ed ha inserito una password valida	{Email: "autore1@tempmail.com"}	Dominio non ammesso!	L'utente viene reindirizzato o ad inserire una mail valida	Email non valida!	L'utente viene reindirizzato o ad inserire una mail valida	PASS
4	Formato non valido	Email non valida	L'utente si vuole	{Email: "autore1.com"}	Formato non valido!	L'utente viene	Email non valida!	L'utente viene	PASS



			<u>registrare</u> <u>alla</u> <u>piattaforma</u> <u>, ed ha</u> <u>inserito</u> <u>una</u> <u>password</u> <u>valida</u>			<u>reindirizzat</u> <u>o ad</u> <u>inserire</u> <u>una mail</u> <u>valida</u>		<u>reindirizzat</u> <u>o ad</u> <u>inserire</u> <u>una mail</u> <u>valida</u>	
--	--	--	---	--	--	---	--	---	--

