

**Università Politecnica delle Marche**

**Dipartimento di Ingegneria dell'Informazione**

**Corso di Laurea Triennale in Ingegneria Informatica e  
dell'Automazione**

---

**Software per la gestione di un negozio  
dell'usato**

**Ursu Leon**

**Perazzoli Leonardo**

---

## Indice

1 INTRODUZIONE .....	4
1.1 DESCRIZIONE IN LINGUAGGIO NATUREALE .....	4
1.2 GLOSSARIO DEI TERMINI .....	5
2 ANALISI DEI REQUISITI.....	6
2.1 REQUISITI DEL SISTEMA.....	6
2.1.1 REQUISITI FUNZIONALI .....	7
2.1.2 REQUISITI NON FUNZIONALI .....	10
2.2 DIAGRAMMI DEI CASI D'USO .....	12
2.2.1 ATTORI .....	12
2.2.2 GESTIONE AMMINISTRATORE .....	13
2.2.3 GESTIONE CLIENTE-PROPRIETARIO .....	25
2.2.4 GESTIONE SISTEMA.....	28
2.2.5 GESTIONE USER .....	32
2.4 MATRICE MAPPING DEI REQUISITI.....	35
3 DIAGRAMMI DI ANALISI .....	36
3.1 PACKAGE DI ANALISI.....	36
3.2 DIAGRAMMI DELLE CLASSI DI ANALISI .....	37
3.3 DIAGRAMMI DI SEQUENZA .....	38
3.3.1 BACKUP .....	39
3.3.2 CONTROLLASTATOPRODOTTO .....	39
3.3.3 CRUD PRODOTTO.....	40
3.3.4 LOGIN .....	46
3.3.5 NOTIFICA.....	47
3.3.6 RICERCACLIENTE .....	48
3.3.7 RICERCAPRODOTTO .....	49
3.3.8 VENDIPRODOTTO .....	50
3.3.9 VISUALIZZASTATISTICHE .....	51
4 DIAGRAMMI DI PROGETTAZIONE .....	52
4.1 DIAGRAMMI DELLE CLASSI DI PROGETTAZIONE .....	52
4.2 DIAGRAMMI DEI COMPONENTI .....	57
4.3 DIAGRAMMI DELLE MACCHINE A STATI.....	58

4.3.1 MACCHINE A STATI: BACKUP .....	58
4.3.2 MACCHINE A STATI: CATEGORIA .....	59
4.3.3 MACCHINE A STATI: LOGIN .....	59
4.3.4 MACCHINE A STATI: PRODOTTO.....	60
4.3.5 MACCHINE A STATI: RICEVUTA.....	61
5 IMPLEMENTAZIONE .....	62
5.1 DIAGRAMMA DI DEPLOYMENT .....	62
6 MOCKUP.....	63
6.1 VISTA LOGIN .....	63
6.2 VISTA USER .....	64
6.3 VISTA AMMINISTRATORE .....	65
6.4 VISTA CLIENTE PROPRIETARIO .....	71
7 PYUNIT.....	74

# 1 INTRODUZIONE

## 1.1 DESCRIZIONE IN LINGUAGGIO NATUREALE

### DESCRIZIONE NEGOZIO

Il progetto proposto consiste nella realizzazione di un sistema informativo per la gestione di un negozio dell'usato.

L'immobile è composto da una sala di esposizione e un magazzino, nella sala di esposizione gli oggetti vengono esposti sugli scaffali, ogni oggetto ha una determinata categoria, un id oggetto, un prezzo e una data di esposizione.

L'unico lavoratore presente nel negozio è colui che lo gestisce interamente cioè l'amministratore.

### REGISTRAZIONE UTENTE

Il cliente-proprietario arrivato al negozio, se non lo ha già fatto in precedenza, dovrà registrarsi insieme all'amministratore fornendo i propri dati personali, la propria e-mail e la password concordata con l'amministratore e gli verrà fornito un codice cliente. Il sistema invierà un'e-mail di registrazione effettuata con successo, dove verranno indicati i rispettivi dati di registrazione per l'accesso locale al sistema.

### ACCETTAZIONE OGGETTO

Solo se già registrato, il cliente-proprietario consegnerà l'oggetto destinato alla vendita, al cassiere(amministratore), il quale provvederà ad inserirlo nel sistema compilando i dati dell'oggetto: nome categoria, id cliente-proprietario, data di registrazione e il prezzo iniziale che verrà concordato da ambo le parti (il cassiere e il cliente-proprietario).

L'oggetto rimarrà in vendita per un massimo di 5 mesi:

- Al secondo mese dalla data di registrazione il prezzo iniziale del prodotto subirà un decremento del 30%.
- Al terzo mese dalla data di registrazione il prezzo iniziale del prodotto subirà un decremento del 40%.
- Al quarto mese dalla data di registrazione il prezzo iniziale del prodotto subirà un decremento del 50%.
- Al quinto mese dalla data di registrazione il prodotto verrà eliminato dalla lista di oggetti disponibili.

Nel caso in cui l'oggetto giunga alla data di scadenza invenduto, l'oggetto dovrà essere ritirato dal cliente-proprietario altrimenti si procederà allo smaltimento.

### VENDITA OGGETTO

Al momento della vendita, l'amministratore effettuerà la ricerca nel sistema del prodotto che ha intenzione di vendere, lo seleziona ed effettua la sua vendita con il prezzo che corrisponde al prezzo corrente, questo accade perché al prezzo corrente corrisponde già un eventuale sconto applicato. La data e l'ora di vendita vengono acquisite automaticamente. L'oggetto venduto viene

rimosso dalla lista degli oggetti disponibili e viene archiviato, Inoltre, verrà inoltrata un'e-mail per notificare la vendita al corrispettivo cliente-proprietario.

## ALTRE FUNZIONALITA'

Nella struttura è presente un terminale accessibile ai visitatori del negozio dove potranno consultare il catalogo degli oggetti.

Il cliente-acquirente, non sarà vincolato dalla registrazione, potrà visualizzare l'intera lista dei prodotti disponibili ed avrà a disposizione strumenti per effettuare filtraggio in base alle proprie preferenze (categoria, prezzo, data).

L'amministratore avrà a disposizione degli strumenti per visualizzare le statistiche sui prodotti venduti e sul numero di clienti, in questo modo potrà analizzare l'andamento della propria attività commerciale.

L'amministratore avrà la possibilità di effettuare un backup manualmente.

Alla fine della giornata lavorativa il sistema dovrà essere chiuso per effettuare la creazione delle statistiche ed un backup dei dati, che consisterà nel copiare tutti i dati che costituiscono il database, sul disco, al fine di riparare a eventuali perdite di dati o anomalie.

## 1.2 GLOSSARIO DEI TERMINI

In seguito, si trova il glossario del progetto in questione. Tutti i termini presenti nella colonna denominata "PAROLA" sono stati individuati ed estratti dalla "1.1 Descrizione in Linguaggio naturale". Tali termini sono stati definiti per ridurre l'ambiguità generale nella comprensione con i corrispettivi, se presenti, sinonimi e omonimi.

PAROLA	DEFINIZIONE	SINONIMI	OMONIMI
Cliente-Proprietario	Colui che possiede un oggetto in vendita nel negozio.	ClienteP	nessuno
Amministratore	Colui che ha privilegi massimi all'interno del sistema informativo, gestisce l'intero negozio e il software gestionale.	Admin	nessuno
User	Colui che utilizza il sistema software per la sola ricerca di oggetti disponibili senza avere privilegi particolari.	Utente	nessuno
Codice cliente	Codice univoco associato a ciascun Cliente-Proprietario	Id Cliente	nessuno
Dati di registrazione	E-mail e password di accesso al sistema.	Credenziali	nessuno
Prezzo iniziale	Prezzo concordato per l'oggetto al momento della messa in vendita.	Prezzo originale	nessuno
Prezzo di vendita	Prezzo al quale è stato venduto l'oggetto, si ricava dal prezzo iniziale applicando qualche sconto.	Prezzo corrente	nessuno
Categoria	Ogni oggetto appartiene ad una categoria	nessuno	nessuno

	che li contraddistingue per specifiche fisiche e di utilizzo.		
Account	Dati di informazioni associati a persone fisiche che funge da identificativo.	nessuno	nessuno
Prodotto	Oggetto registrato nel sistema, che è nello stato: in vendita, venduto, eliminato, scaduto.	oggetto	nessuno

## 2 ANALISI DEI REQUISITI

Dall'analisi della documentazione prodotta durante le interviste con gli stakeholder, sono stati individuati dei requisiti che fungeranno come vincoli da rispettare durante la progettazione del sistema. Essi sono suddivisi in requisiti funzionali e requisiti non funzionali.

- requisiti funzionali: sono delle caratteristiche che il sistema deve garantire all'utilizzatore.
- requisiti non funzionali: sono dei vincoli imposti dal sistema.

Dall'analisi dei requisiti del sistema e alla descrizione nel linguaggio naturale è stato possibile individuare i casi d'uso del sistema.

### 2.1 REQUISITI DEL SISTEMA

In questa sezione verranno trattati i requisiti del sistema. Come da figura numero 1, i requisiti sono stati suddivisi in package diversi.

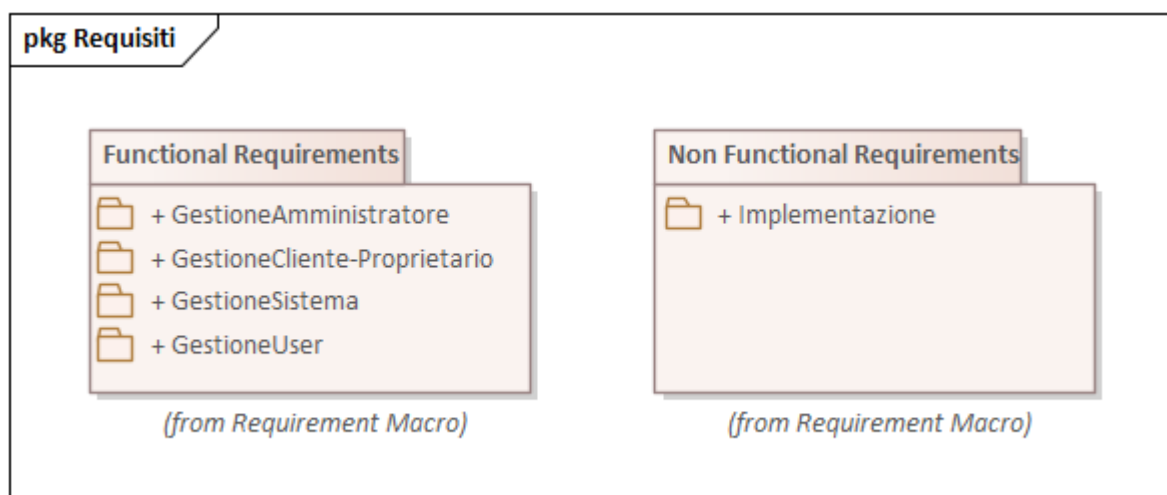


Figura 1 - Package dei requisiti

### 2.1.1 REQUISITI FUNZIONALI

Nella figura numero 2 troviamo la rappresentazione grafica di come abbiamo suddiviso i requisiti funzionali. La suddivisione in package diversi, permette di individuare aree diverse del nostro sistema

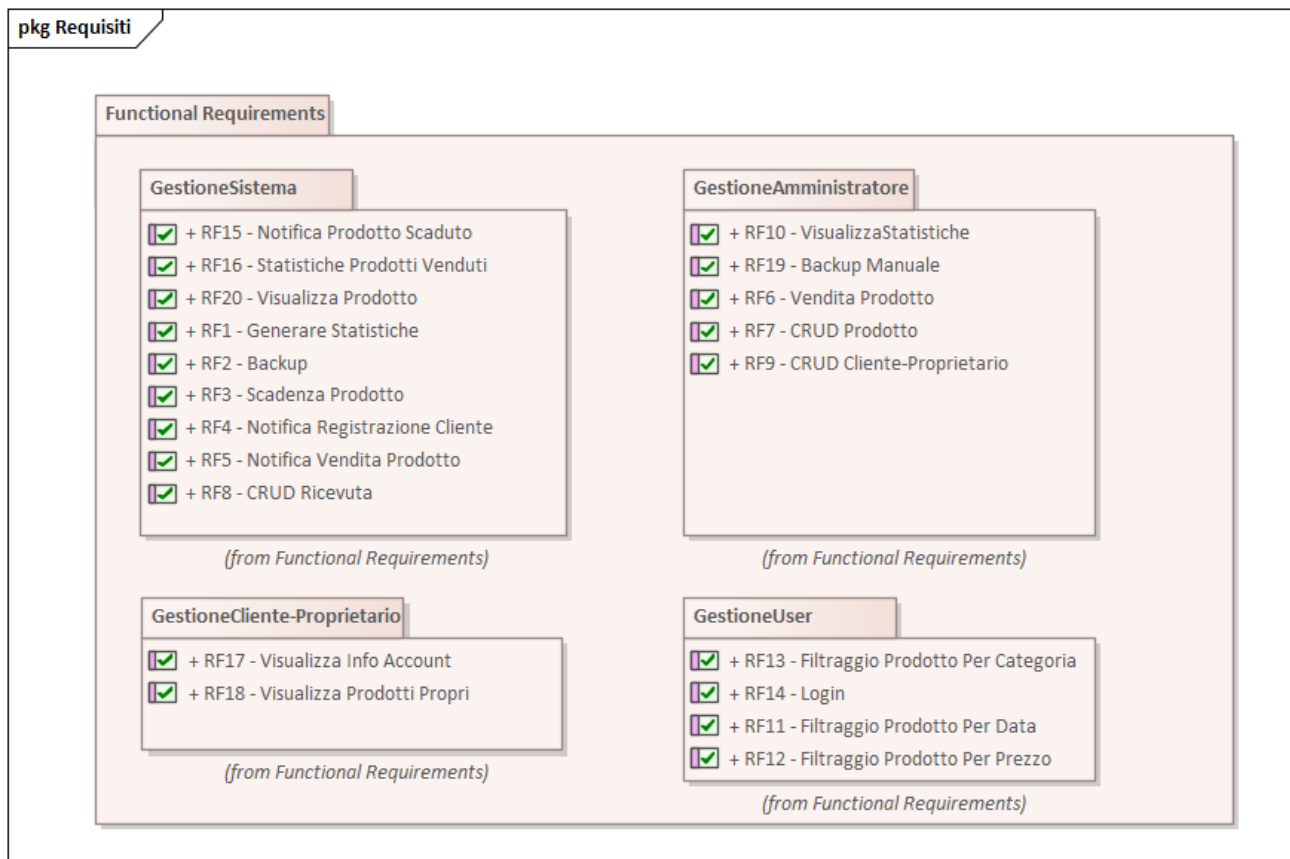


Figura 2 - Requisiti funzionali

#### RF1 – Generare Statistiche

Type: **Requirement**

Status: proposed

Package: *GestioneSistema*

*Il sistema dovrà generare delle statistiche sull'andamento economico*

#### RF2 - Backup

Type: **Requirement**

Status: proposed

Package: *GestioneSistema*

Il sistema dovrà effettuare un backup dei dati

**RF3 – Scadenza Prodotto**

Type: **Requirement**

Status: proposed

Package: *GestioneSistema*

Il sistema dovrà gestire la scadenza dei prodotti non venduti

**RF4 - Notifica Registrazione Cliente**

Type: **Requirement**

Status: proposed

Package: *GestioneSistema*

Il sistema dovrà notificare all'e-mail del cliente la propria registrazione

**RF5 - Notifica Vendita Prodotto**

Type: **Requirement**

Status: proposed

Package: *GestioneSistema*

Il sistema dovrà notificare all'e-mail del cliente la vendita del proprio oggetto

**RF6 - Vendita Prodotto**

Type: **Requirement**

Status: proposed

Package: *GestioneAmministratore*

Il sistema dovrà permettere all'Amministratore di vendere un prodotto

**RF7 - CRUD Prodotto**

Type: **Requirement**

Status: proposed

Package: *GestioneAmministratore*

Il sistema dovrà essere in grado di effettuare operazioni CRUD sul prodotto

**RF8 - CRUD Ricevuta**

Type: **Requirement**

Status: proposed

Package: *GestioneSistema*

Il sistema dovrà essere in grado di effettuare operazioni CRUD sulla ricevuta

**RF9 - CRUD Cliente-Proprietario**



Type: **Requirement**

Status: proposed

Package: *GestioneAmministratore*

Il sistema dovrà effettuare operazioni CRUD sul Cliente-Proprietario

#### **RF10 - Visualizza Statistiche**

Type: **Requirement**

Status: proposed

Package: *GestioneAmministratore*

Il sistema dovrà poter visualizzare statistiche sul prodotto

#### **RF11 - Filtraggio Prodotto Per Data**

Type: **Requirement**

Status: proposed

Package: *GestioneUser*

Il sistema dovrà effettuare il filtraggio dei prodotti con la data di inserimento

#### **RF12 - Filtraggio Prodotto Per Prezzo**

Type: **Requirement**

Status: proposed

Package: *GestioneUser*

Il sistema dovrà effettuare il filtraggio dei prodotti con il loro prezzo corrente

#### **RF13 - Filtraggio Prodotto Per Categoria**

Type: **Requirement**

Status: proposed

Package: *GestioneUser*

Il sistema dovrà effettuare il filtraggio dei prodotti con la loro categoria

#### **RF14 - Login**

Type: **Requirement**

Status: proposed

Package: *GestioneUser*

Il sistema dovrà permettere di effettuare il login

#### **RF15 - Notifica Prodotto Scaduto**

Type: **Requirement**

Status: proposed

Package: *GestioneSistema*

Il sistema dovrà notificare all'e-mail cliente che il suo prodotto è scaduto

**RF16 - Statistiche Prodotti Venduti**

Type: **Requirement**

Status: proposed

Package: *GestioneSistema*

Il sistema dovrà effettuare statistiche su prodotti venduti

**RF17 – Visualizza Info Account**

Type: **Requirement**

Status: proposed

Package: *GestioneCliente-proprietario*

Il sistema dovrà permettere di visualizzare le info del proprio account

**RF18 – Visualizza Prodotti Propri**

Type: **Requirement**

Status: proposed

Package: *GestioneCliente-proprietario*

Il sistema dovrà permettere di visualizzare i propri prodotti presenti nel sistema

**RF19 – Backup Manuale**

Type: **Requirement**

Status: proposed

Package: *GestioneAmministratore*

Il sistema dovrà permettere all'Amministratore di effettuare un backup manuale

**RF20 - Visualizza Prodotto**

Type: **Requirement**

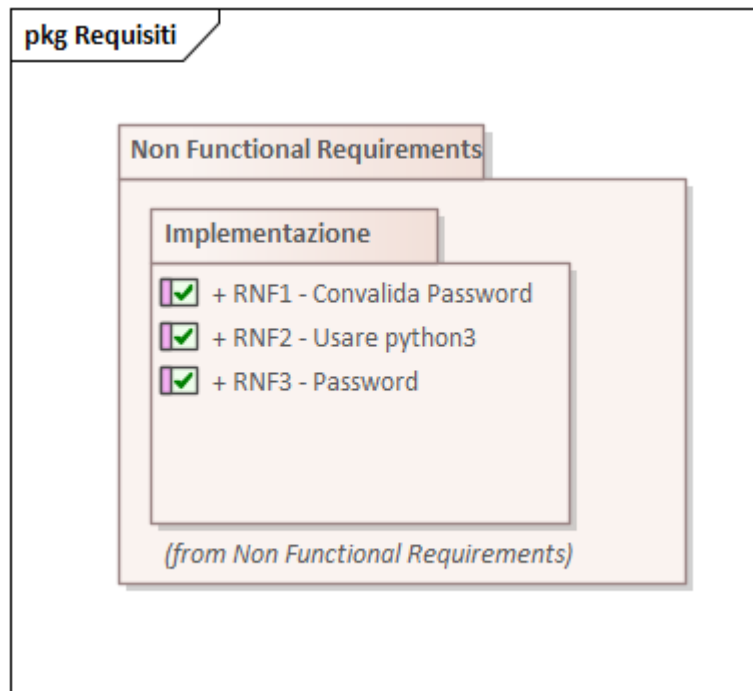
Status: proposed

Package: *GestioneSistema*

Il sistema dovrà permettere di visualizzare e info di un prodotto

**2.1.2 REQUISITI NON FUNZIONALI**

Nella figura numero 3 troviamo la rappresentazione grafica di come abbiamo suddiviso i requisiti non funzionali. La suddivisione in package diversi permette di individuare aree diverse del nostro sistema.



*Figura 3 - Requisiti non funzionali*

**RNF1 - Convalida Password**

Type: **Requirement**

Status: proposed

Package: Requisiti non funzionali

Il sistema dovrà effettuare la convalida della password

**RNF2 - Usare Python 3**

Type: **Requirement**

Status: proposed

Package: Requisiti non funzionali

Il sistema dovrà essere implementato in Python 3

**RNF3 - Password**

Type: **Requirement**

Status: proposed

Package: Requisiti non funzionali

Il sistema dovrà poter gestire una password

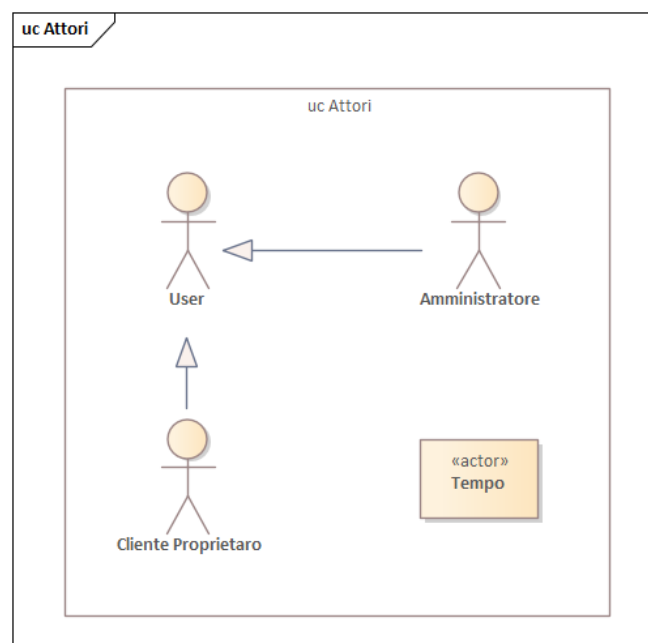
## 2.2 DIAGRAMMI DEI CASI D'USO

Con l'analisi dei requisiti è stato possibile ricavare i casi d'uso del sistema, scenari che si possono verificare durante l'esecuzione del sistema.

I seguenti diagrammi sono stati suddivisi in base alle varie aree di interazione da parte degli attori con il sistema in uso, in modo da rendere indipendenti, dove è possibile, le interazioni dei vari attori con il sistema.

### 2.2.1 ATTORI

Nella figura 4 vi è il diagramma degli attori individuati durante l'analisi, gli attori: Amministratore e Cliente-Proprietario sono delle generalizzazioni dell'attore User, ciò significa che potranno trovarsi nelle situazioni dell'attore User.



*Figura 4 - Diagramma degli attori*

## 2.2.2 GESTIONE AMMINISTRATORE

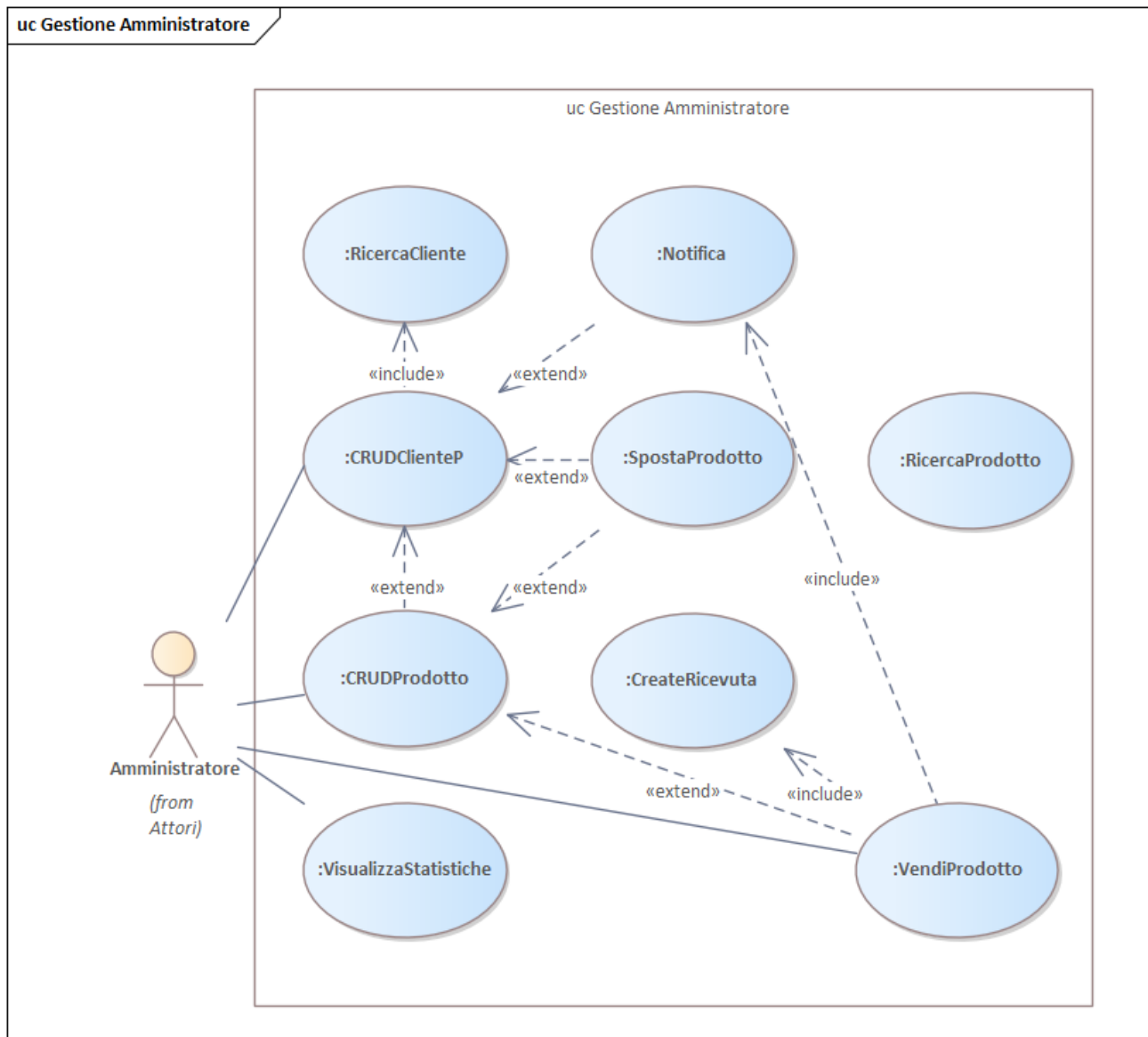


Figura 5 – Diagramma dei casi d'uso: Gestione Amministratore

Caso d'uso: CreateRicevuta
ID: 1
Descrizione: Questo caso d'uso permette di effettuare operazioni di creazione di una ricevuta
Attori primari: Amministratore
Attori secondari: nessuno
Precondizioni: l'Amministratore ha venduto dei prodotti
Sequenza eventi principali: 1. il caso d'uso inizia quando sono stati venduti dei prodotti 2. il sistema crea la ricevuta 3. il sistema aggiunge i prodotti nella ricevuta 4. il sistema stampa la ricevuta
Postcondizioni: il sistema si rende nuovamente disponibile per altre operazioni
Sequenza degli eventi alternativa: nessuna

Caso d'uso: CRUDClienteProprietario
ID: 2
Descrizione: Questo caso d'uso permette di effettuare operazioni CRUD sul ClienteProprietario
Attori primari: Amministratore
Attori secondari: ClienteProprietario
Precondizioni: l'amministratore del sistema desidera effettuare operazioni CRUD per un ClienteProprietario
<p>Sequenza eventi principali:</p> <ol style="list-style-type: none"> <li>1. il caso d'uso inizia quando l'attore primario desidera effettuare un operazione CRUD di un ClienteProprietario</li> <li>2. <b>if</b> l'Amministratore vuole inserire nel database un ClienteProprietario <ol style="list-style-type: none"> <li>2.1 <b>include</b> (:RicercaCliente)</li> <li>2.2 <b>if</b> il cliente risulta esistente nel sistema <ol style="list-style-type: none"> <li>2.2.1 il sistema restituisce un errore</li> </ol> </li> <li>2.3 <b>else</b> <ol style="list-style-type: none"> <li>2.3.1 L'Amministratore inserisce i dati del ClientePr nel database</li> <li>2.3.2 il sistema memorizza i dati del cliente nel database</li> <li>2.3.3 il sistema manda una notifica</li> </ol> </li> </ol> </li> <li>3. <b>else if</b> l'amministratore vuole visualizzare un ClienteProprietario <ol style="list-style-type: none"> <li>3.1 <b>include</b> (:RicercaCliente)</li> <li>3.2 <b>if</b> il cliente risulta inesistente nel database <ol style="list-style-type: none"> <li>3.2.1 il sistema restituisce un errore</li> </ol> </li> <li>3.3 <b>else</b> il sistema mostra il cliente</li> </ol> </li> <li>4. <b>else if</b> l'amministratore vuole aggiornare i dati relativi ad un ClienteP <ol style="list-style-type: none"> <li>4.1. <b>include</b> (:RicercaCliente);</li> <li>4.2. <b>if</b> il cliente risulta esistente nel database <ol style="list-style-type: none"> <li>4.2.1 l'amministratore specifica i nuovi dati</li> <li>4.2.2 Il sistema aggiorna i dati del cliente nel database</li> </ol> </li> <li>4.3. <b>else</b> <ol style="list-style-type: none"> <li>4.3.1 il sistema restituisce un errore</li> </ol> </li> </ol> </li> <li>5. <b>else if</b> l'amministratore vuole cancellare un ClienteP dal database <ol style="list-style-type: none"> <li>5.1 <b>include</b> (:RicercaCliente)</li> <li>5.2 <b>if</b> il cliente risulta esistente nel database <ol style="list-style-type: none"> <li>5.2.1 <b>include</b> (:RicercaProdotto)</li> <li>5.2.2 <b>while</b> (esistono prodotti del ClienteP) <ol style="list-style-type: none"> <li>5.2.2.1 il sistema elimina il prodotto</li> </ol> </li> <li>5.2.3 il sistema elimina il ClienteP dal database</li> </ol> </li> <li>5.3. <b>else</b></li> </ol> </li> </ol>

5.3.1 il sistema restituisce un errore
Postcondizioni: il sistema si rende nuovamente disponibile per altre operazioni
Sequenza degli eventi alternativa: nessuna



Caso d'uso: CRUDProdotto
ID: 3
Descrizione: Questo caso d'uso permette di effettuare operazioni CRUD sul prodotto
Attori primari: Amministratore
Attori secondari: Nessuno
Precondizioni: l'Amministratore vuole effettuare un'operazione CRUD su un prodotto
<p>Sequenza eventi principali:</p> <p>0. Il caso d'uso inizia quando l'amministratore seleziona l'operazione</p> <p>1. <b>if</b> l'amministratore vuole creare un prodotto</p> <p>    1.1 il sistema chiede di inserire i dati del prodotto da creare</p> <p>    1.2 L'Amministratore inserisce i dati del prodotto nel sistema</p> <p>    1.3 il sistema memorizza i dati del prodotto nel database</p> <p>    1.4 <b>if</b> prodotto inserito correttamente</p> <p>        1.4.1 il sistema comunica il successo dell'operazione</p> <p>    1.5 <b>else</b></p> <p>        1.5.1 il sistema restituisce un errore</p> <p>2. <b>else if</b> l'Amministratore primario vuole visualizzare il prodotto</p> <p>    2.1 <b>include</b> (:RicercaProdotto)</p> <p>    2.2 <b>if</b> prodotto trovato nel database</p> <p>        2.2.1 il sistema restituisce l'oggetto trovato</p> <p>    2.3 <b>else</b></p> <p>        2.3.1 il sistema restituisce un errore</p> <p>3. <b>else if</b> l'Amministratore primario vuole aggiornare i dati relativi ad un prodotto</p> <p>    3.1. <b>include</b> (:RicercaProdotto);</p> <p>    3.2. <b>if</b> il prodotto risulta esistente nel database</p> <p>        3.2.1 l'attore primario specifica i nuovi dati</p> <p>        3.2.2 Il sistema aggiorna i dati del prodotto nel database</p> <p>    3.3. <b>else</b></p> <p>        3.3.1. il sistema restituisce un errore</p> <p>4. <b>else if</b> l'Amministratore vuole cancellare il prodotto dal database</p> <p>    4.1 <b>include</b> (:RicercaProdotto)</p> <p>    4.2 <b>if</b> il prodotto risulta esistente nel database</p> <p>        4.2.1 il sistema elimina il prodotto dal database</p> <p>    4.3. <b>else</b></p> <p>        4.3.1. il sistema restituisce un errore</p>
Postcondizioni: il sistema si rende nuovamente disponibile per altre operazioni

Sequenza degli eventi alternativa: nessuna

Caso d'uso: Notifica
ID: 4
Descrizione: Invia una notifica al cliente registrato
Attori primari: Amministratore
Attori secondari: Nessuno
<p>Precondizioni Segmento 1: L'utente è stato creato</p> <p>Precondizioni Segmento 2: Il prodotto è stato rimosso</p> <p>Precondizioni Segmento 3: Il prodotto è stato venduto</p>
<p>Sequenza eventi Segmento 1:</p> <ol style="list-style-type: none"> <li>1. il caso d'uso inizia quando l'amministratore crea un nuovo Cliente Proprietario</li> <li>2. il sistema invia una e-mail al Cliente Proprietario</li> </ol> <p>Sequenza eventi Segmento 2:</p> <ol style="list-style-type: none"> <li>1. il caso d'uso inizia quando il Sistema rimuove il prodotto di un Cliente Proprietario</li> <li>2. <b>if</b> il prodotto è stato venduto <ol style="list-style-type: none"> <li>2.1 il sistema invia una e-mail al cliente proprietario con l'esito della vendita</li> </ol> </li> <li>3. <b>else if</b> il tempo di vendita del prodotto è scaduto <ol style="list-style-type: none"> <li>3.1 il sistema invia una e-mail al cliente proprietario indicando la scadenza del prodotto registrato</li> </ol> </li> </ol> <p>Sequenza eventi Segmento 3:</p> <ol style="list-style-type: none"> <li>1. Il caso d'uso inizia quando il prodotto viene venduto</li> <li>2. il sistema invia un'e-mail al ClienteProprietario</li> </ol>
Postcondizioni: il sistema si rende nuovamente disponibile per altre operazioni
Sequenza degli eventi alternativa: Nessuna

Caso d'uso: RicercaCliente
ID: 5
Descrizione: Questo caso d'uso permette di effettuare la ricerca di un cliente nel sistema
Attori primari: Amministratore
Attori secondari: Cliente
Precondizioni: l'attore primario desidera effettuare una ricerca del cliente nel database
<p>Sequenza eventi principali:</p> <ol style="list-style-type: none"> <li>1. il caso d'uso inizia quando l'attore primario desidera effettuare la ricerca di un Cliente nel database del sistema</li> <li>2. l'attore primario specifica i dati di ricerca</li> <li>3. il sistema effettua la ricerca con i dati forniti dall'attore primario</li> <li>4. <b>if</b> il Cliente è stato trovato <ol style="list-style-type: none"> <li>4.1 il sistema restituisce i dati del Cliente</li> </ol> </li> <li>5. <b>else</b> <ol style="list-style-type: none"> <li>5.1 il sistema restituisce un errore</li> </ol> </li> </ol>
<p>Postcondizioni:</p> <ol style="list-style-type: none"> <li>1. il sistema ha restituito dei dati</li> </ol>
Sequenza degli eventi alternativa: nessuna

Caso d'uso: RicercaProdotto
ID: 6
Descrizione: Questo caso d'uso permette di effettuare la ricerca di un prodotto nel sistema
Attori primari: Amministratore, ClienteP, User
<u>Attori secondari: sistema</u>
Precondizioni: l'attore primario desidera effettuare la ricerca di un prodotto
<p>Sequenza eventi principali:</p> <ol style="list-style-type: none"> <li>1. il caso d'uso inizia quando l'attore primario desidera effettuare la ricerca di un prodotto nel database del sistema</li> <li>2. il sistema richiede all'attore primario di inserire i dati di ricerca</li> <li>3. l'attore primario specifica i dati di ricerca</li> <li>4. il sistema effettua la ricerca con i dati forniti dall'attore primario</li> <li>5. <b>if</b> il prodotto è stato trovato <ol style="list-style-type: none"> <li>5.1 il sistema restituisce i dati del prodotto</li> </ol> </li> <li>6. <b>else</b> <ol style="list-style-type: none"> <li>6.1 il sistema restituisce un errore</li> </ol> </li> </ol>
Postcondizioni: nessuna
Sequenza degli eventi alternativa: nessuna

Caso d'uso: SpostaProdotto
ID: 7
Descrizione: Questo caso d'uso permette di effettuare lo spostamento di un prodotto nel sistema
Attori primari: Amministratore
Attori secondari: nessuno
Precondizioni: è necessario lo spostamento di un prodotto
<p>Sequenza eventi principali:</p> <ol style="list-style-type: none"> <li>1. Il caso d'uso inizia quando l'attore primario richiede di spostare un prodotto</li> <li>2. Il sistema richiede all'attore primario un luogo di A e un luogo B</li> <li>3. l'attore primario inserisce i dati richiesti</li> <li>4. Il sistema richiede il prodotto o un suo riferimento da spostare dal luogo A ad un luogo B</li> <li>5. l'attore primario inserisce i dati richiesti</li> <li>6. Il sistema sposta il prodotto dal luogo A ad un luogo B</li> <li>7. <b>if</b> l'operazione è stata effettuata <ol style="list-style-type: none"> <li>7.1. il sistema comunica il successo dell'operazione</li> </ol> </li> <li>8. <b>else</b> <ol style="list-style-type: none"> <li>8.1 Il sistema restituisce un errore</li> </ol> </li> </ol>
Postcondizioni: nessuna
Sequenza degli eventi alternativa: nessuna

Caso d'uso: VendiProdotto
ID: 8
Descrizione: Questo caso d'uso permette di effettuare la vendita di un prodotto
Attori primari: Amministratore
Attori secondari: Nessuno
Precondizioni: l'amministratore desidera effettuare la vendita di un prodotto
<p>Sequenza eventi principali:</p> <ol style="list-style-type: none"> <li>1. il caso d'uso inizia quando l'Amministratore vuole vendere un prodotto</li> <li>2. l'attore primario inserisce l'oggetto da vendere</li> <li>3. il sistema inizia la ricerca del prodotto <ol style="list-style-type: none"> <li>3.1 <b>include</b> (:RicercaProdotto)</li> </ol> </li> <li>4. il sistema inizia a preparare la ricevuta di vendita <ol style="list-style-type: none"> <li>4.1 <b>include</b> (:CRicevuta)</li> </ol> </li> <li>5. il sistema elimina il prodotto <ol style="list-style-type: none"> <li>5.1 <b>include</b> (:DProdotto)</li> </ol> </li> <li>6. il sistema manda una notifica di vendita prodotto <ol style="list-style-type: none"> <li>6.1 <b>include</b> (:Notifica)</li> </ol> </li> </ol>
Postcondizioni: il prodotto è stato venduto
Sequenza degli eventi alternativa: nessuna

Caso d'uso: VisualizzaStatistiche
ID: 9
Descrizione: Questo caso d'uso permette di effettuare la visualizzazione delle statistiche disponibili
Attori primari: Amministratore
Attori secondari: Nessuno
Precondizioni: l'amministratore desidera visualizzare le statistiche
Sequenza eventi principali: 1. Il caso d'uso inizia quando l'attore primario seleziona "visualizzare le statistiche" 2. Il sistema preleva dal database le statistiche 3. <b>if</b> statistiche disponibili 3.1 il sistema mostra le statistiche disponibili 4. <b>else</b> 4.1 il sistema restituisce un errore
Postcondizioni: nessuna
Sequenza degli eventi alternativa: nessuna



### 2.2.3 GESTIONE CLIENTE-PROPRIETARIO

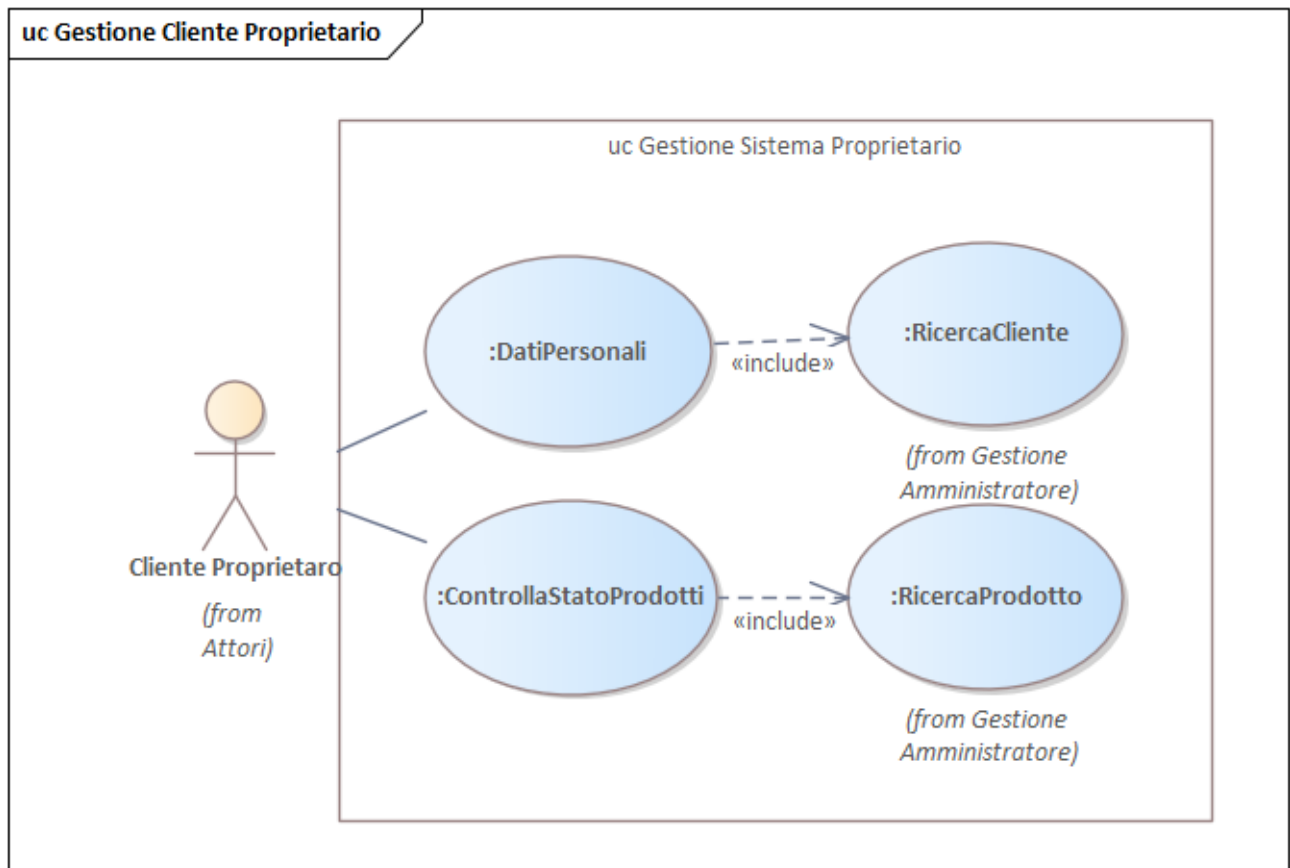


Figura 6 - Diagramma dei casi d'uso: Gestione Cliente-Proprietario

Caso d'uso: ControllaStatoProdotti
ID: 10
Descrizione: Questo caso d'uso permette di effettuare in controllo dello stato dei prodotti di un ClienteProprietario
Attori primari: ClienteProprietario
Attori secondari: nessuno
Precondizioni: L'attore primario deve aver effettuato il login
<p>Sequenza eventi principali:</p> <ol style="list-style-type: none"> <li>1. Il caso d'uso inizia quando l'attore primario desidera visualizzare lo stato dei prodotti</li> <li>2. Il sistema ricerca i prodotti associati all'attore primario</li> <li>3. <b>if</b> esistono oggetti associati <ol style="list-style-type: none"> <li>3.1 il sistema preleva dal database gli oggetti associati all'attore Primario</li> <li>3.2 Il sistema mostra all'attore primario lo stato dei prodotti associati</li> </ol> </li> <li>4. <b>else</b> il sistema restituisce un errore</li> </ol>
Postcondizioni: Nessuna

Caso d'uso: DatiPersonali
ID: 11
Descrizione: Questo caso d'uso permette ad un Cliente proprietario di effettuare un controllo dei propri dati presenti nel sistema
Attori primari: ClienteProprietario
Attori secondari: Nessuno
Precondizioni: L'attore primario deve aver effettuato il login
<p>Sequenza eventi principali:</p> <ol style="list-style-type: none"> <li>1. Il caso d'uso inizia quando l'attore primario desidera visualizzare i propri dati</li> <li>2. Il sistema ricerca l'account associato all'attore primario</li> <li>3. <b>if</b> trovato account associato <ol style="list-style-type: none"> <li>3.1. il sistema preleva dal database i dati associati all'attore primario</li> <li>3.2 Il sistema mostra all'attore primario i propri dati</li> </ol> </li> <li>4. <b>else</b> <ol style="list-style-type: none"> <li>4.1 Il sistema restituisce un errore</li> </ol> </li> </ol>
Postcondizioni: Nessuna

## 2.2.4 GESTIONE SISTEMA

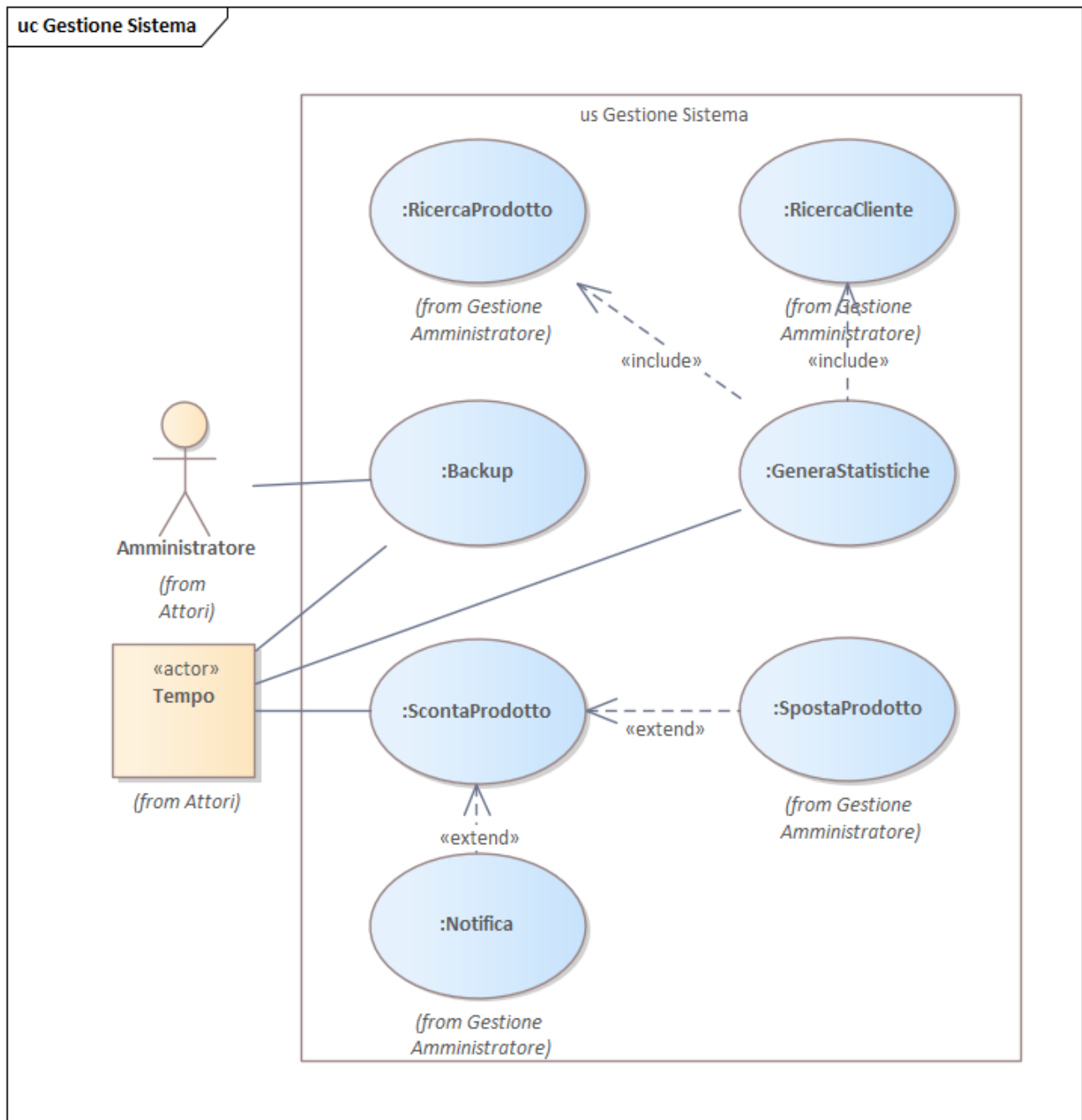


Figura 7 - Diagramma dei casi d'uso: Gestione Sistema

Caso d'uso: Backup
ID: 12
Descrizione: il sistema effettua il backup dei dati
Attori primari: Tempo
Attori secondari: Nessuno
Precondizioni: Nessuno il programma è stato chiuso
Sequenza eventi principali: 1. Il caso d'uso inizia quando il programma è stato chiuso 2. Il sistema preleva i dati dal database 3. Il sistema copia su disco i dati prelevati
Postcondizioni: I dati sono stati copiati.
Sequenza degli eventi alternativa: Nessuna

Caso d'uso: GeneraStatistiche
ID: 13
Descrizione: Il sistema genera statistiche
Attori primari: Tempo
Attori secondari: Nessuno
Precondizioni: Nessuna
Sequenza eventi principali: 1. Il caso d'uso inizia quando il sistema viene chiuso 2. Il sistema preleva dati riguardanti i prodotti 3. Il sistema preleva dati riguardanti le ricevute 4. Il sistema preleva dati riguardanti i Clienti Proprietari 5. Il sistema genera statistiche 6. Il sistema salva i dati su disco
Postcondizioni: Le statistiche generate sono salvate su disco
Sequenza degli eventi alternativa: Nessuna

Caso d'uso: ScontaProdotto
ID: 14
Descrizione: Il sistema sconta il prodotto
Attori primari: Tempo
Attori secondari: Nessuno
<p>Precondizioni Segmento 1: Sono passati 60 giorni dalla registrazione del prodotto</p> <p>Precondizioni Segmento 2: Sono passati 90 giorni dalla registrazione del prodotto</p> <p>Precondizioni Segmento 3: Sono passati 120 giorni dalla registrazione del prodotto</p>
<p>Sequenza eventi Segmento 1:</p> <ol style="list-style-type: none"> <li>1. Il caso d'uso inizia quando sono passati 60 giorni dalla registrazione del prodotto.</li> <li>2. Il prodotto viene scontato del 30% rispetto al prezzo iniziale</li> </ol> <p>Sequenza eventi Segmento 2:</p> <ol style="list-style-type: none"> <li>1. Il caso d'uso inizia quando sono passati 90 giorni dalla registrazione del prodotto.</li> <li>2. Il prodotto viene scontato del 40% rispetto al prezzo iniziale</li> </ol> <p>Sequenza eventi Segmento 3:</p> <ol style="list-style-type: none"> <li>1. Il caso d'uso inizia quando sono passati 120 giorni dalla registrazione del prodotto</li> <li>2. Il prodotto viene scontato del 50% rispetto al prezzo iniziale</li> </ol>
Postcondizioni: Il prodotto viene scontato
Sequenza degli eventi alternativa: Nessuna

## 2.2.5 GESTIONE USER

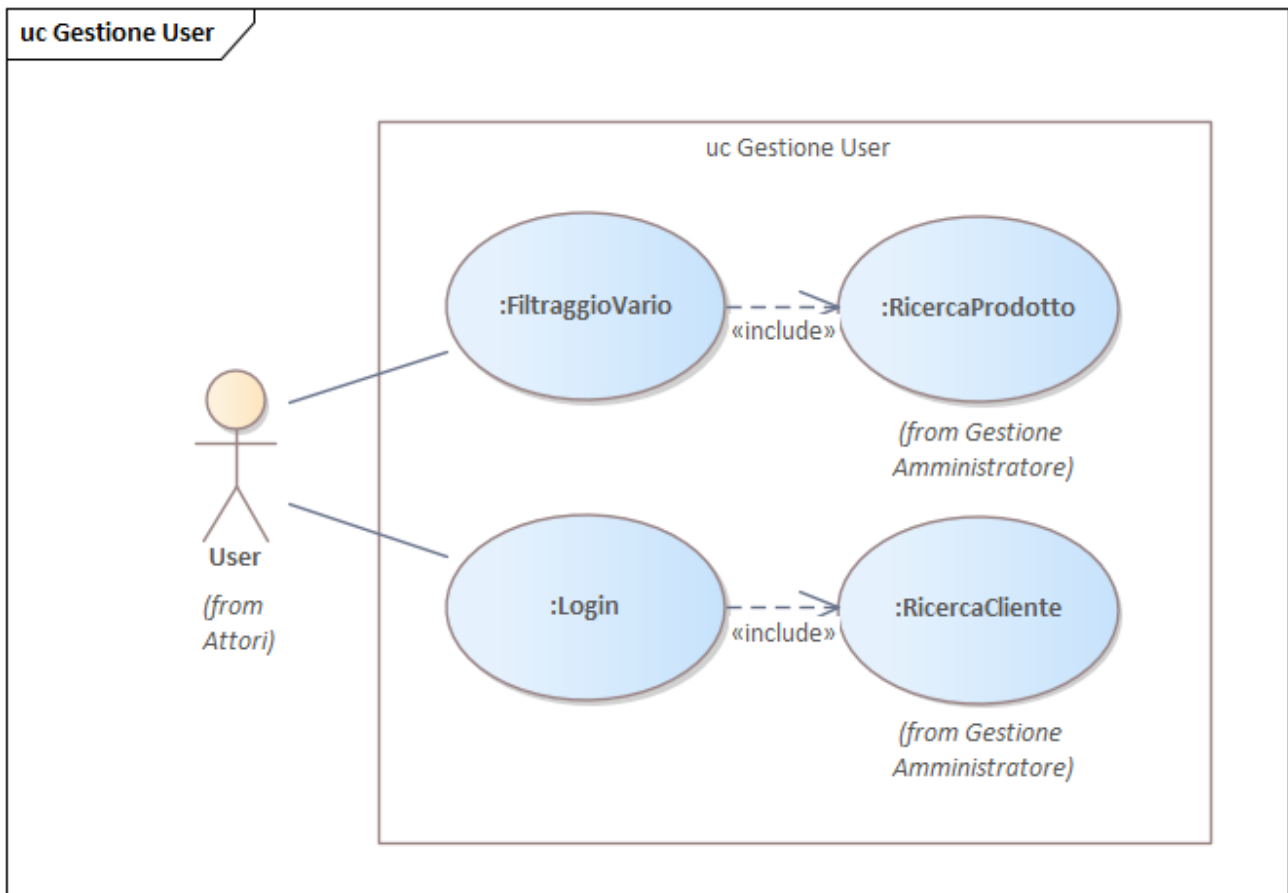


Figura 8 - Diagramma dei casi d'uso: Gestione User



Caso d'uso: FiltraggioVario
ID: 15
Descrizione: Filtraggio dei prodotti in varie modalità
Attori primari: User
Attori secondari: Nessuno
Precondizioni: l'utente vuole effettuare il filtraggio dei prodotti
<p>Sequenza eventi principali:</p> <ol style="list-style-type: none"> <li>1. il caso d'uso inizia quando l'utente vuole filtrare i prodotti disponibili</li> <li>2. l'utente inserisce la modalità di filtraggio e preme invio</li> <li>3. il sistema effettua il filtraggio con il filtro applicato</li> <li>4. <b>if</b> ci sono oggetti nella categoria cercata <ol style="list-style-type: none"> <li>4.1. mostra all'utente la lista degli oggetti trovati</li> </ol> </li> <li>5. <b>else</b> mostra all'utente un messaggio indicando che non sono stati trovati oggetti nella categoria</li> </ol>
Postcondizioni: Nessuna
Sequenza degli eventi alternativa: nessuna

Caso d'uso: Login
ID: 16
Descrizione: L'User accede al sistema
Attori primari: User
Attori secondari: Nessuno
Precondizioni: Nessuna
<p>Sequenza eventi principali:</p> <ol style="list-style-type: none"> <li>1. Il caso d'uso inizia quando l'User vuole accedere al sistema</li> <li>2. L'User inserisce il nome utente e la password</li> <li>3. Il Sistema controlla i dati inseriti</li> <li>4. Login</li> </ol>
Postcondizioni: L'User è loggato
<p>Sequenza degli eventi alternativa:</p> <p>La sequenza degli eventi alternativa inizia dal punto 3</p> <ol style="list-style-type: none"> <li>4. Il nome utente e la password non esistono o sono errati</li> <li>5. Il sistema restituisce un messaggio di errore</li> </ol>

## 2.4 MATRICE MAPPING DEI REQUISITI

È utile tenere traccia della realizzazione tra i casi d'uso individuati e i requisiti in modo da comprendere se la progettazione sta andando in linea con l'idea di cosa il sistema debba essere in grado di svolgere.

Target \ Source	RF1 - Generare Statistiche	RF10 - Visualizza Statistiche	RF11 - Filtraggio Prodotto Per Data	RF12 - Filtraggio Prodotto Per Prezzo	RF13 - Filtraggio Prodotto Per Categoria	RF14 - Login	RF15 - Notifica Prodotto Scaduto	RF16 - Statistiche Prodotti Venduti	RF17 - Visualizza Info Account	RF18 - Visualizza Prodotti Propri	RF19 - Backup Manuale	RF2 - Backup	RF20 - Visualizza Prodotto	RF3 - Scadenza Prodotto	RF4 - Notifica Registrazione Cliente	RF5 - Notifica Vendita Prodotto	RF6 - Vendita Prodotto	RF7 - CRUD Prodotto	RF8 - CRUD Ricevuta	RF9 - CRUD Cliente-Proprietario	RNF1 - Convalida Password	RNF2 - Usare python3	RNF3 - Password
:CRUDClienteP	↑								↑											↑			
:CRUDProdotto	↑		↑	↑	↑					↑			↑	↑		↑		↑					
:CreateRicevuta																			↑				
:VendiProdotto																	↑	↑					
:SpostaProdotto																	↑	↑					
:RicercaCliente	↑																			↑			
:VisualizzaStatistiche		↑						↑															
:ControllaStatoProdotti			↑						↑														
:Notifica							↑								↑	↑							
:Backup											↑	↑											
:ScontaProdotto														↑				↑					
:GeneraStatistiche	↑							↑															
:FiltraggioVario			↑				↑																
:Login						↑																	
:DatiPersonali									↑	↑													
:RicercaProdotto	↑		↑	↑	↑					↑			↑	↑			↑	↑					

Figura 9 - Matrice mapping dei requisiti

### 3 DIAGRAMMI DI ANALISI

Il diagramma di analisi permette una prima rappresentazione delle classi di oggetti individuate durante l'analisi, la loro individuazione è stata frutto delle tecniche di associazione tra ogni aspetto del negozio fisico con una rappresentazione a classi di oggetti e l'interazione tra classi con altre classi. La suddivisione in macroaree *Attività & Servizio* è stata realizzata valutando l'interazione tra le classi di oggetti e vedendo che gli oggetti appartenenti alla stessa macroarea sono maggiormente dipendenti tra essi. Sono state attribuite due definizioni diverse alle macroaree:

*Attività*: rappresenta il negozio nel suo complesso di utilizzo da parte di utenti.

*Servizio*: rappresenta a seconda degli aspetti fisici del negozio in questione i servizi che offre.

L'accesso ai membri interni alle classi di oggetti è stato reso pubblico per assecondare le implicazioni derivanti dal vincolo del RNF2.

#### 3.1 PACKAGE DI ANALISI

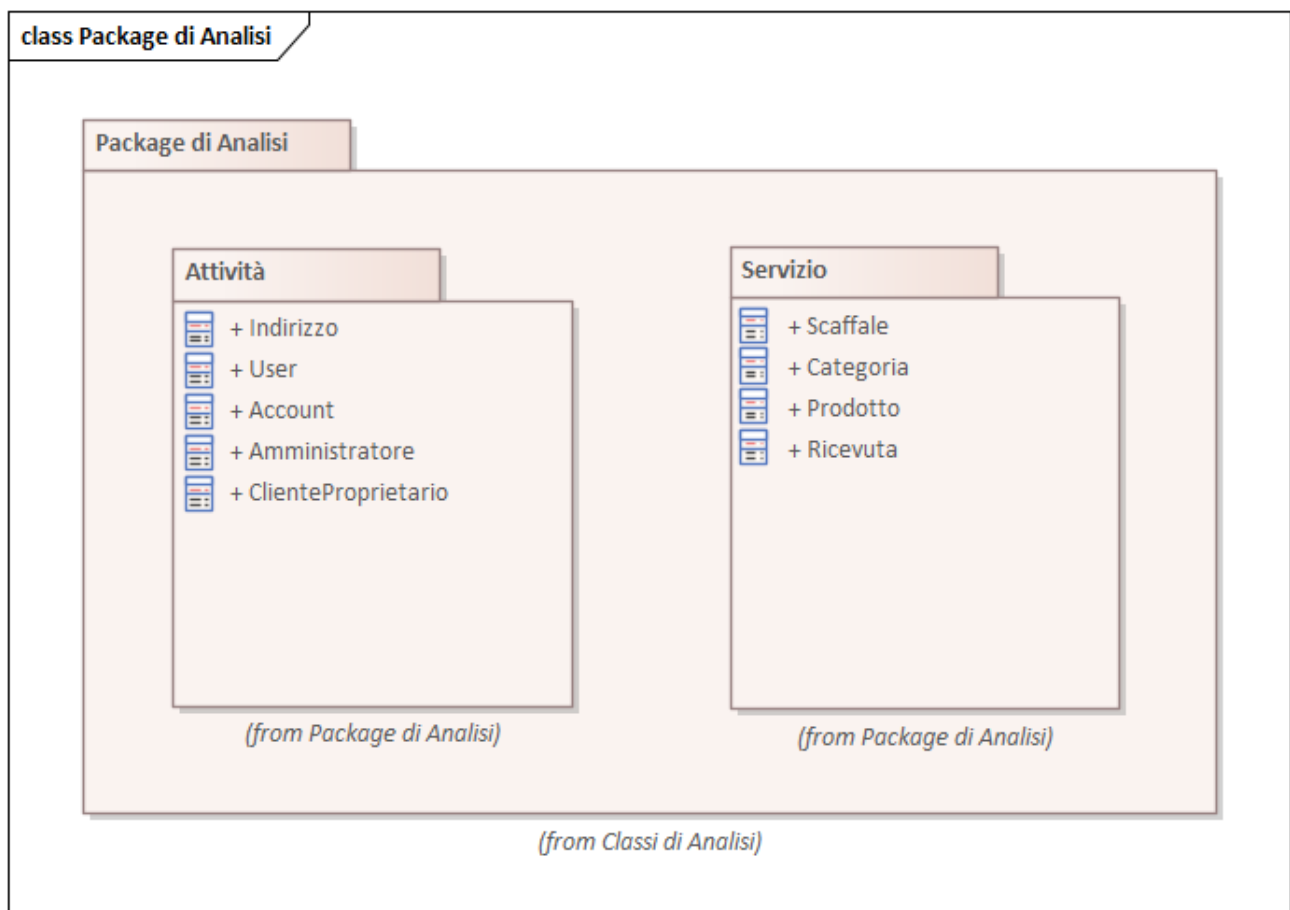


Figura 10 - Package di analisi

### 3.2 DIAGRAMMI DELLE CLASSI DI ANALISI

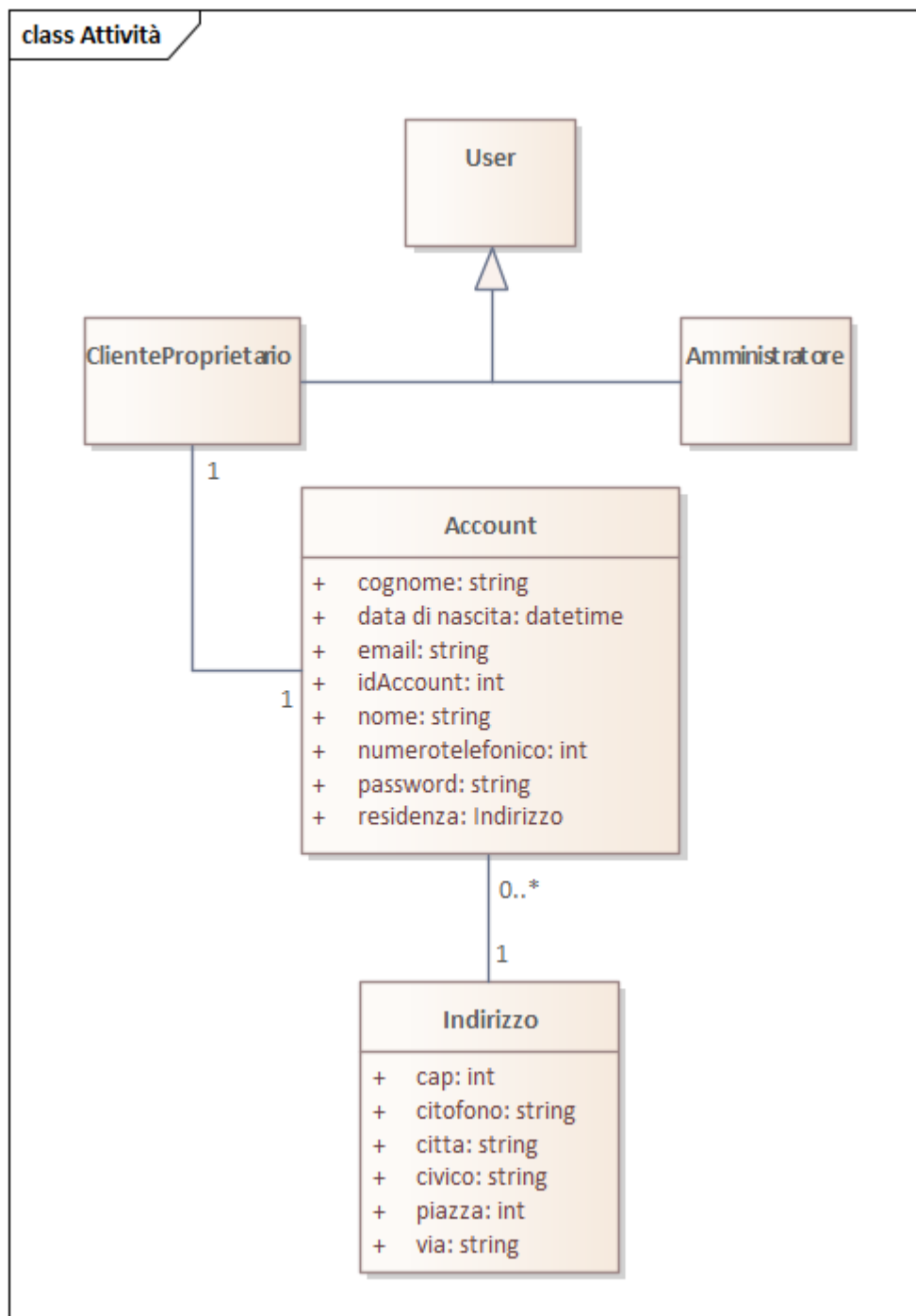
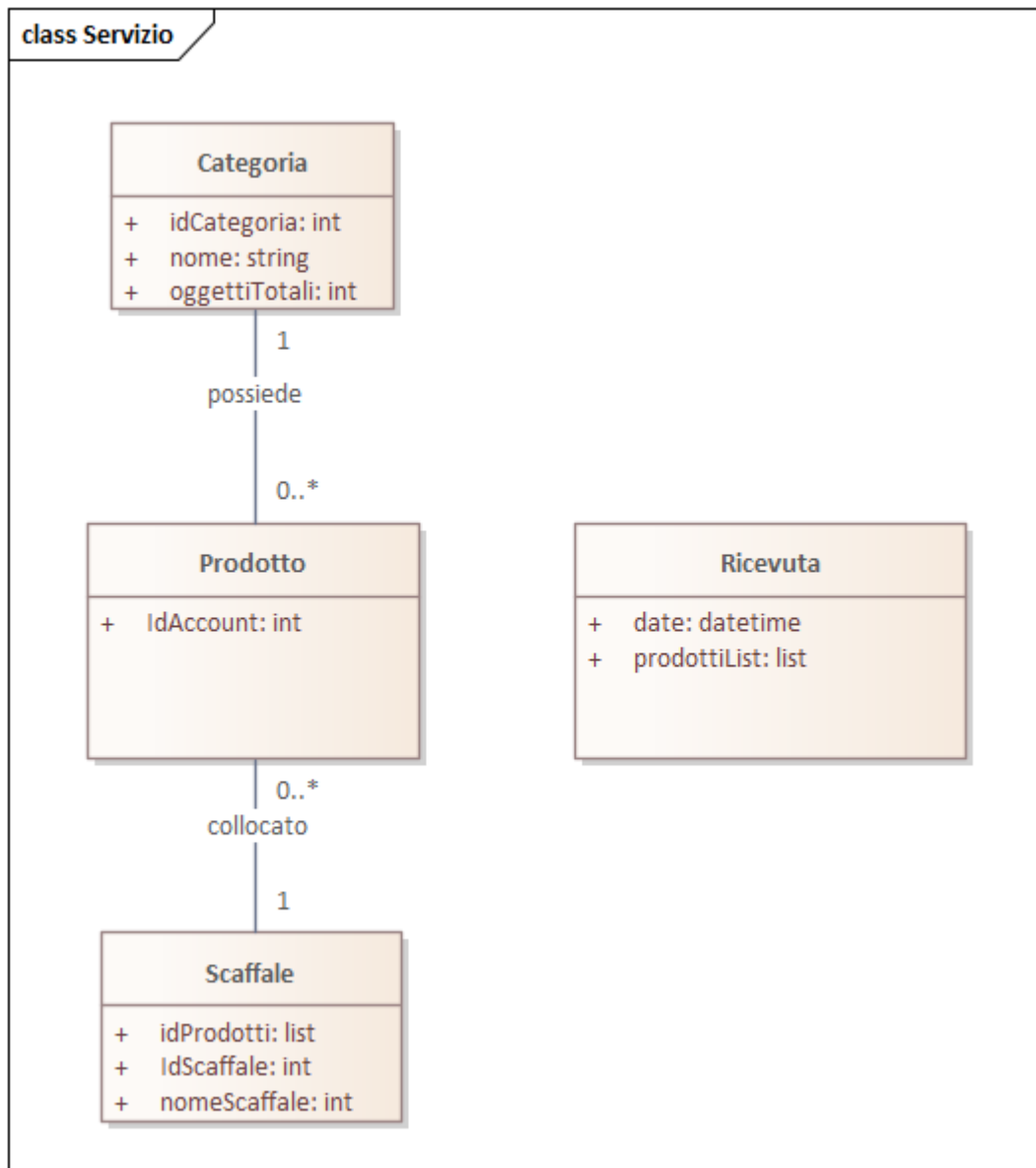


Figura 11- Diagramma di analisi delle classi di Attività



*Figura 12 - Diagramma di analisi delle classi di Servizio*

### 3.3 DIAGRAMMI DI SEQUENZA

I diagrammi di sequenza modellano le interazioni tra oggetti e attori, rappresentano una specifica sequenza di interazione per svolgere una funzione, l'ordine in cui le interazioni avvengono dopo l'attivazione da parte di un attore che interagisce con il sistema è stabilito dai diagrammi, questo permetterà un'implementazione più agevole.

### 3.3.1 BACKUP

Tale diagramma modella l'interazione da parte dell'attore Tempo con la richiesta di effettuare un backup dei dati. L'interazione con il Database avviene nella breve linea di vita in cui vi è la copia dei dati in esso contenuti.

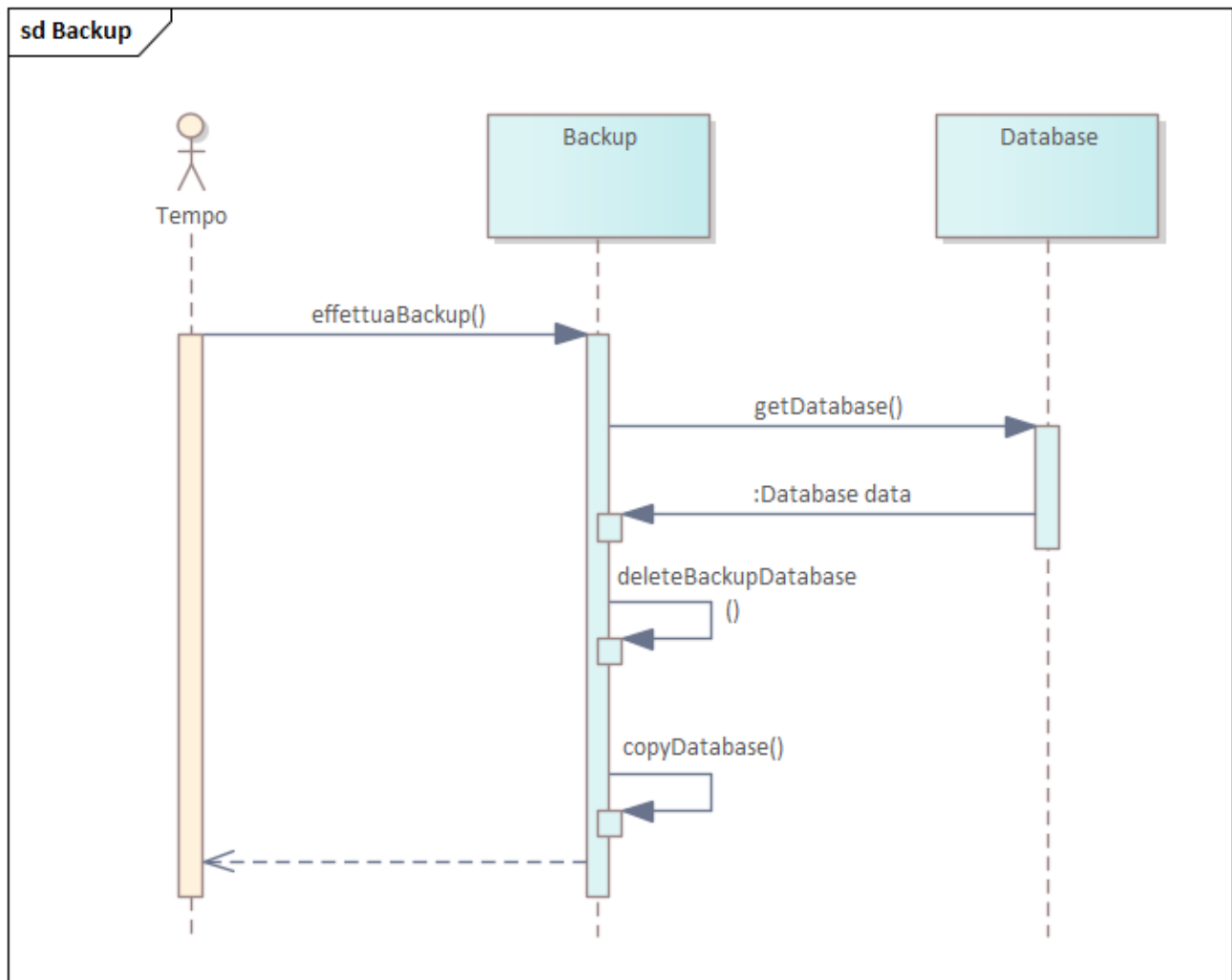


Figura 13 - Diagramma di sequenza: Backup

### 3.3.2 CONTROLLASTATOPRODOTTO

Tale diagramma modella l'interazione dell'attore Cliente Proprietario per controllare lo stato degli oggetti registrati nel sistema.

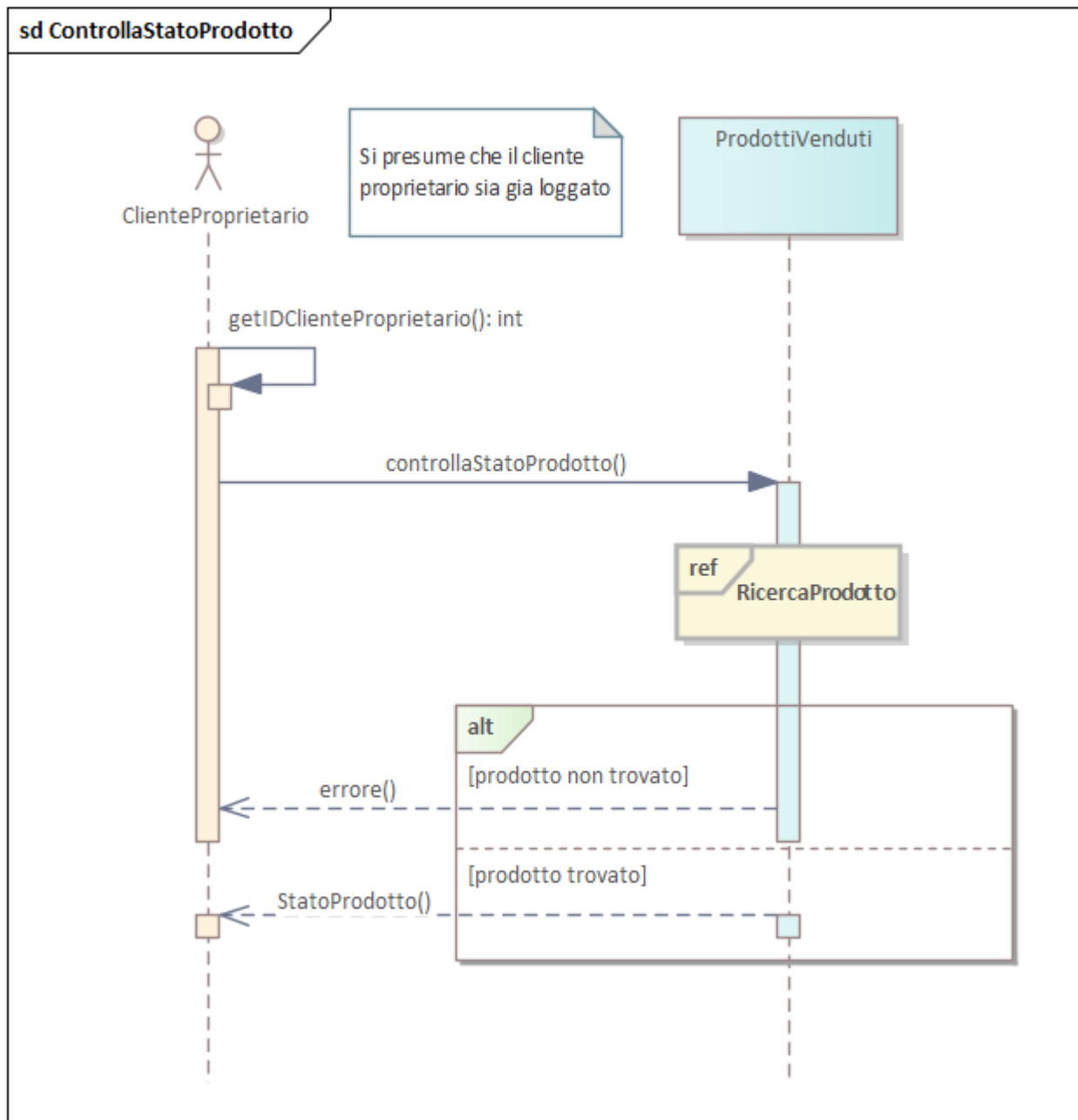


Figura 14 - Diagramma di sequenza: ControllaStatoProdotto

### 3.3.3 CRUD PRODOTTO

Tali diagrammi modellano le interazioni CRUD dell'attore Amministratore con un prodotto presente nel sistema



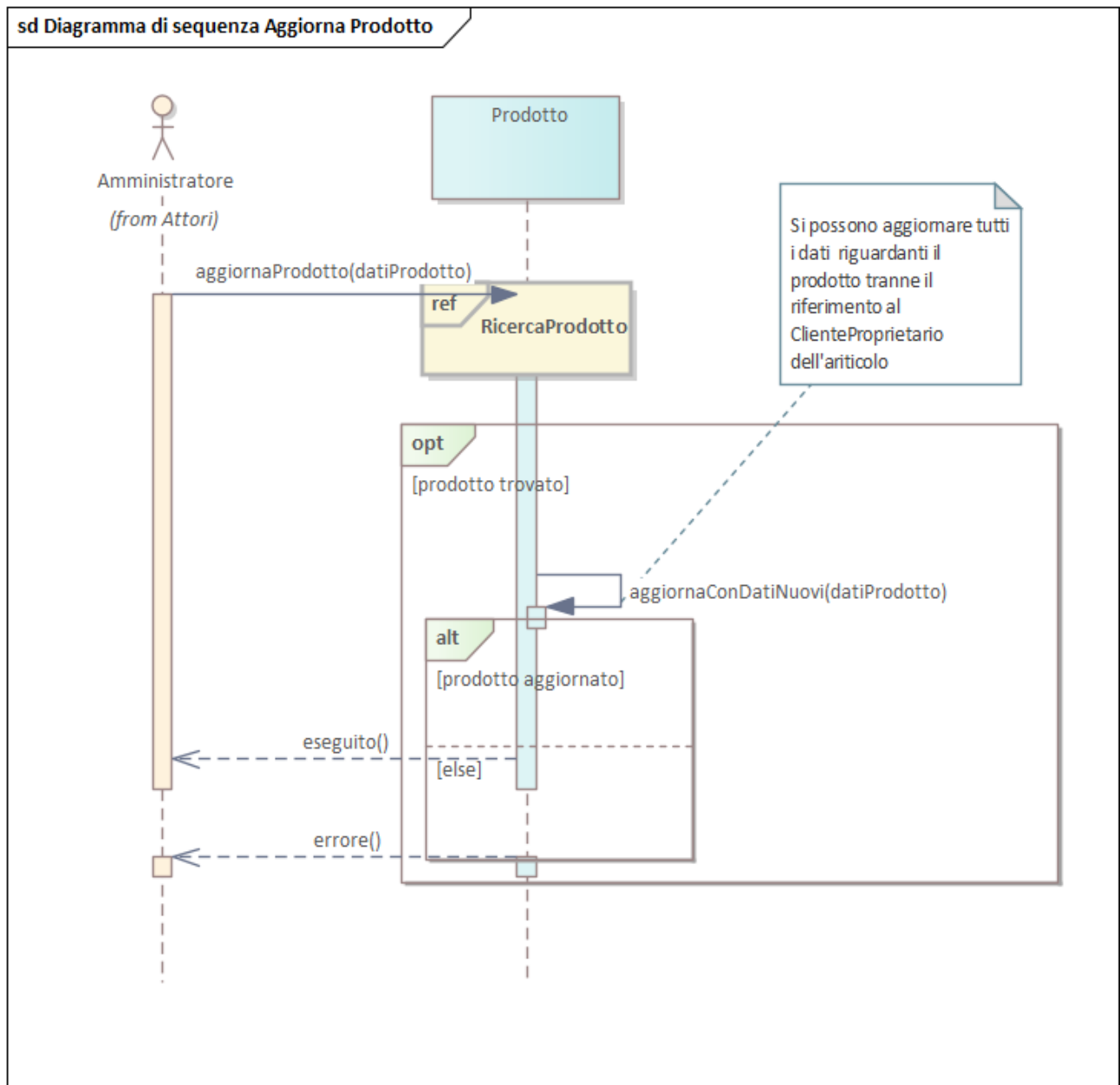


Figura 15 - Diagramma di sequenza: Aggiorna Prodotto

sd Diagramma di sequenza Crea Prodotto

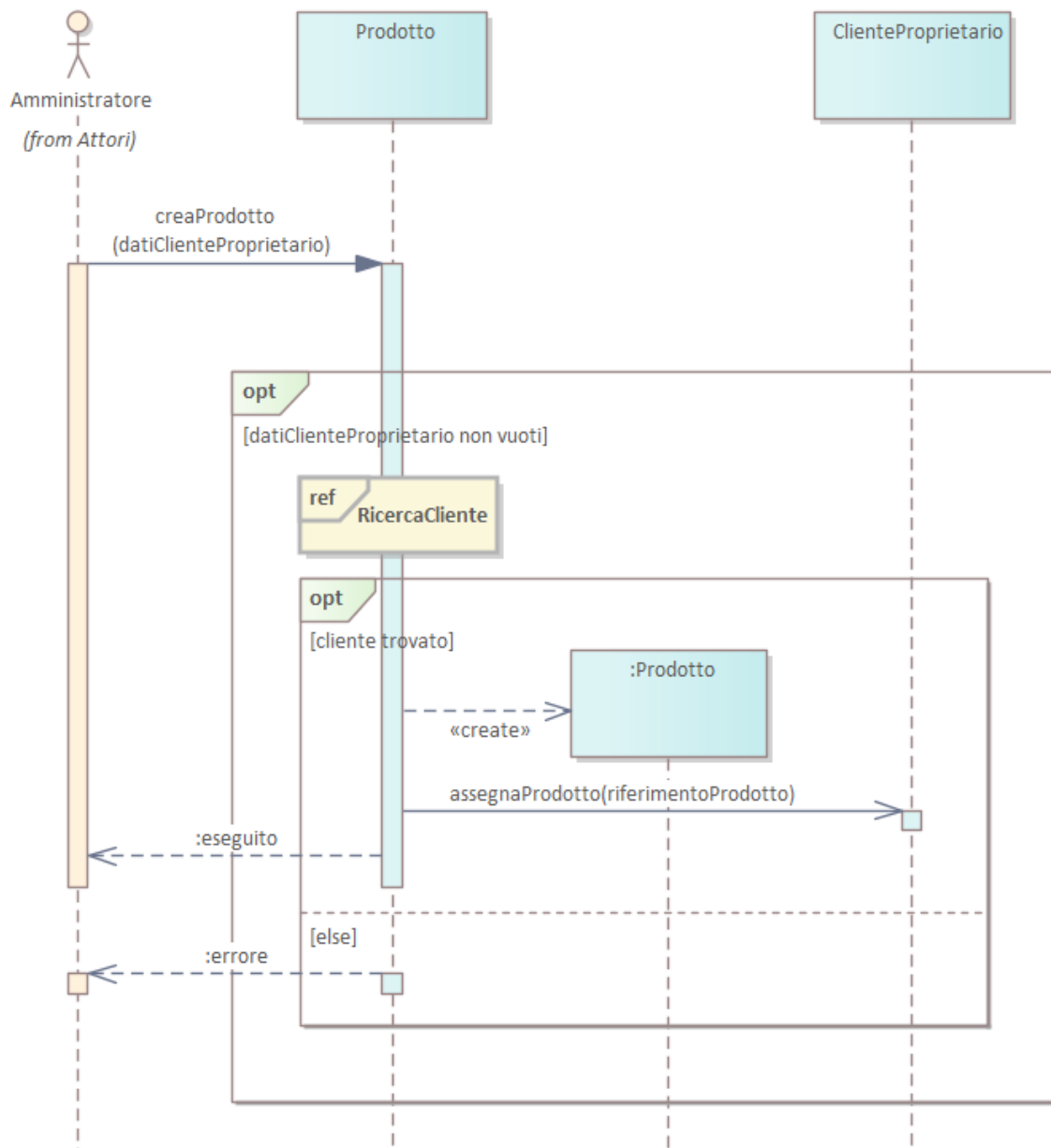


Figura 16 - Diagramma di sequenza: Crea Prodotto

sd Diagramma di sequenza Cancella Prodotto

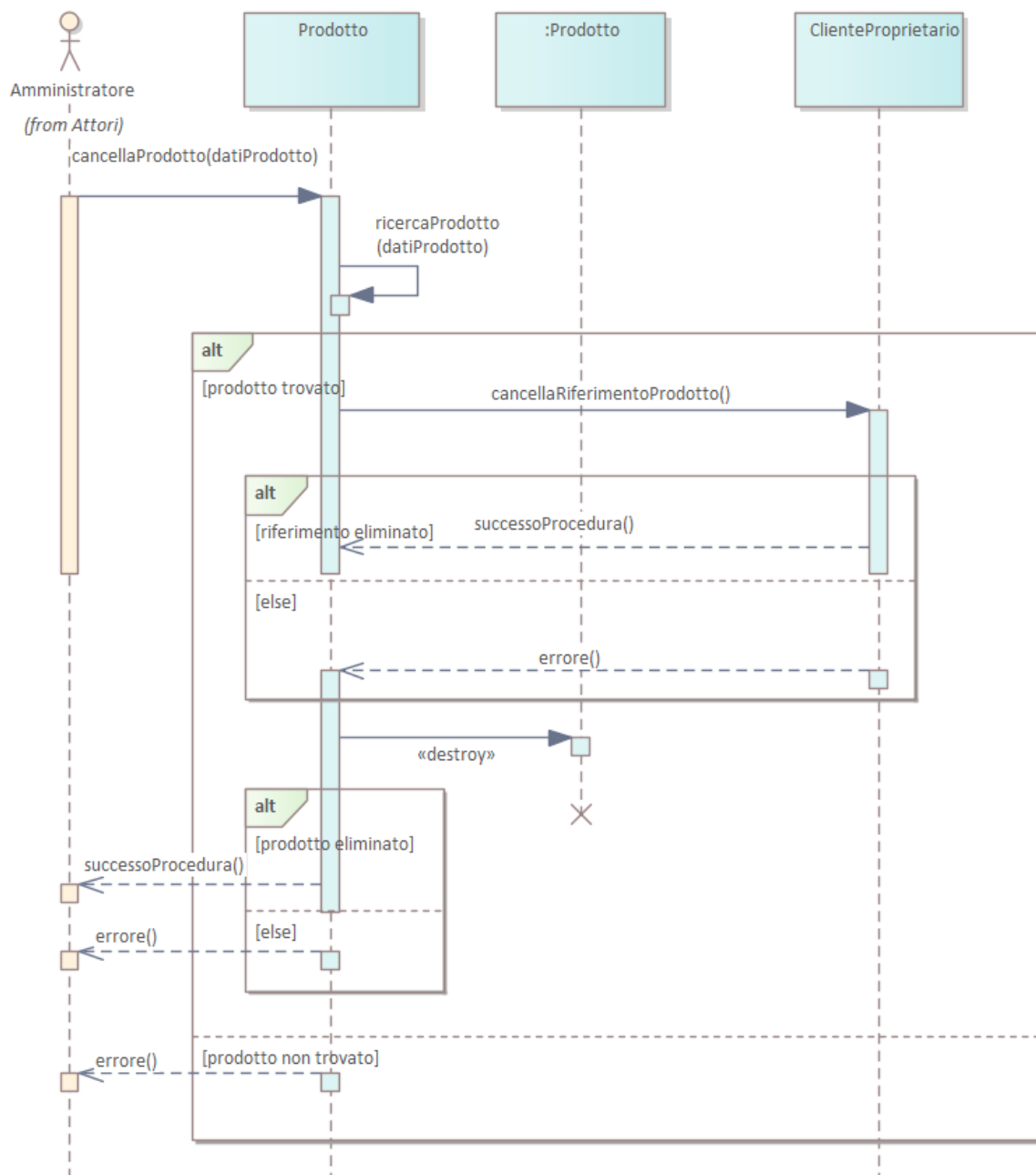


Figura 17 - Diagramma di sequenza: Cancella Prodotto

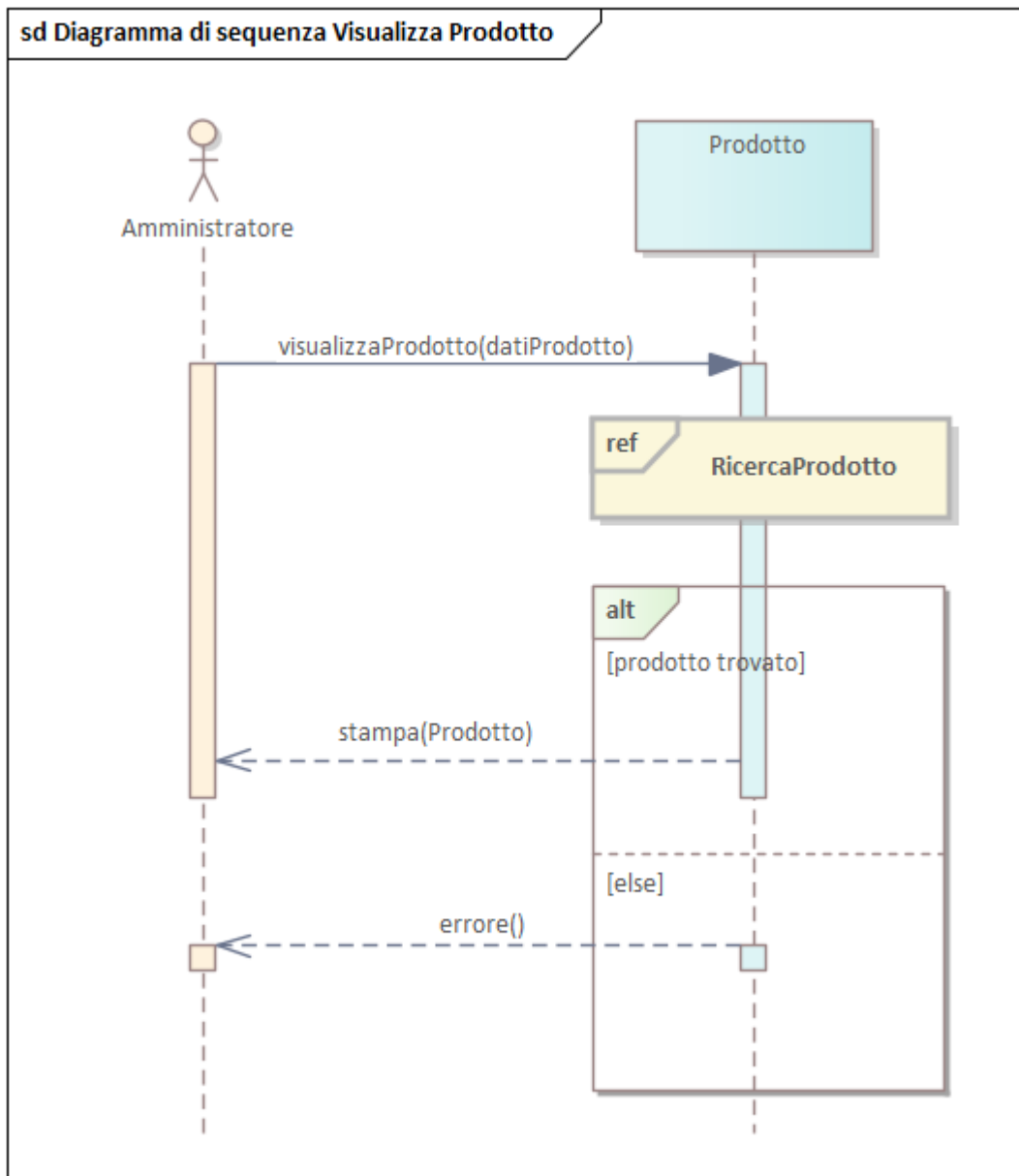


Figura 18 - Diagramma di sequenza: Visualizza Prodotto

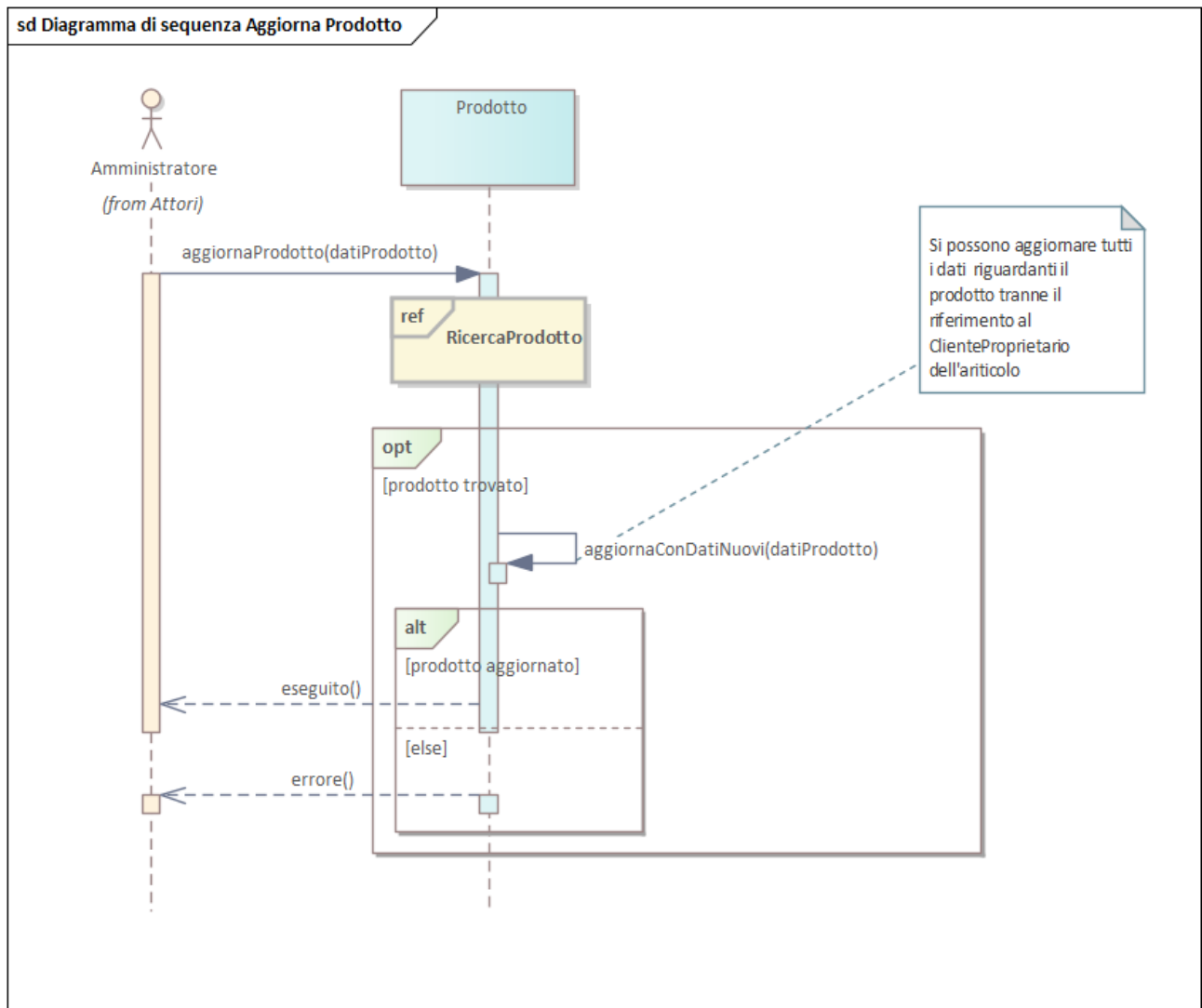


Figura 19 - Diagramma di sequenza: Visualizza Prodotto

### 3.3.4 LOGIN

Tale diagramma modella l'interazione dell'attore User in possesso di un account per l'accesso ai servizi del sistema.

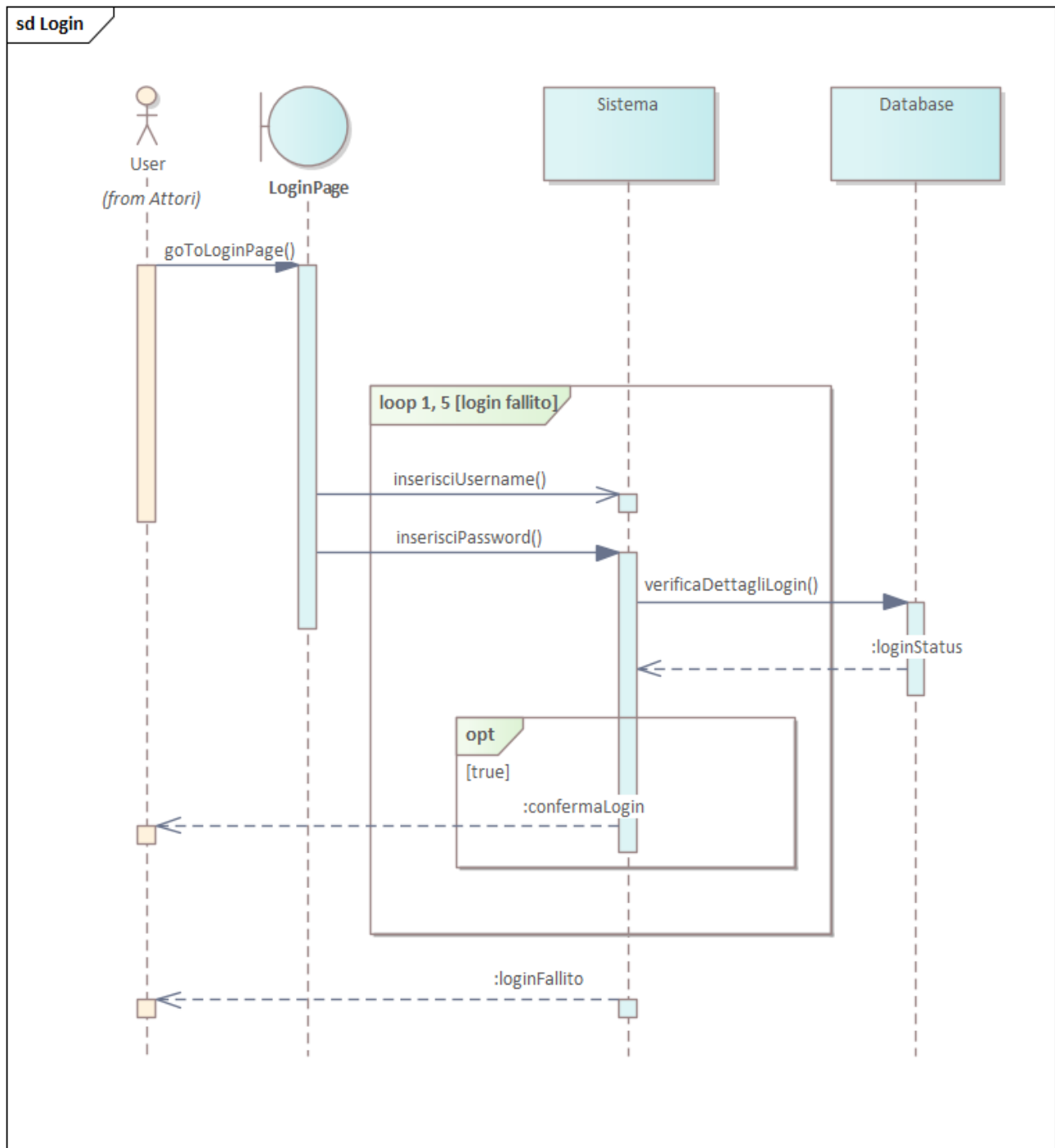


Figura 20 - Diagramma di sequenza: Login

### 3.3.5 NOTIFICA

Tale diagramma modella l'interazione dell'attore Tempo con l'invio di una e-mail di notifica.

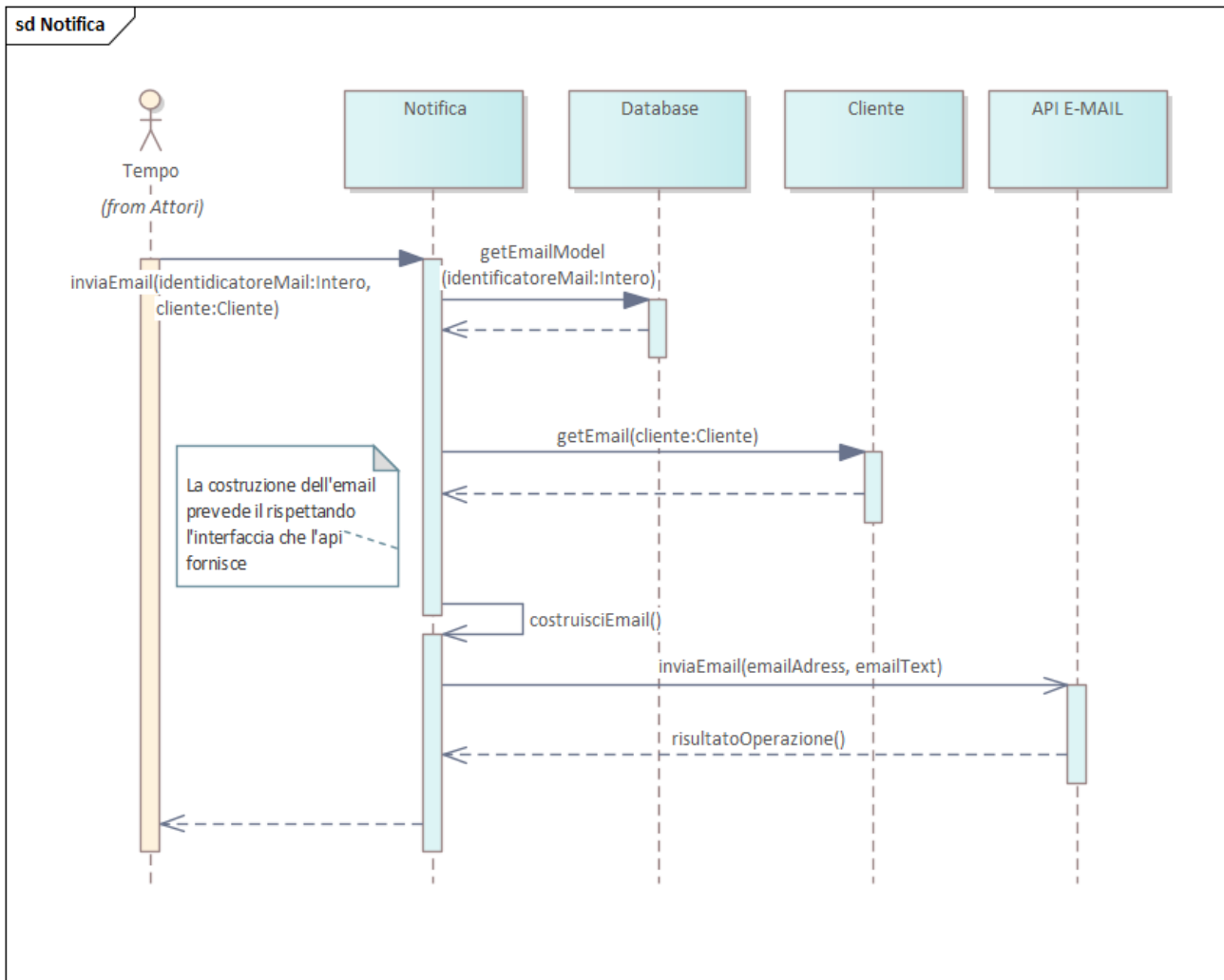


Figura 21 - Diagramma di sequenza: Notifica

### 3.3.6 RICERCACLIENTE

Tale diagramma modella l'interazione dell'attore Amministratore con il sistema per ricercare un Cliente-Proprietario registrato.

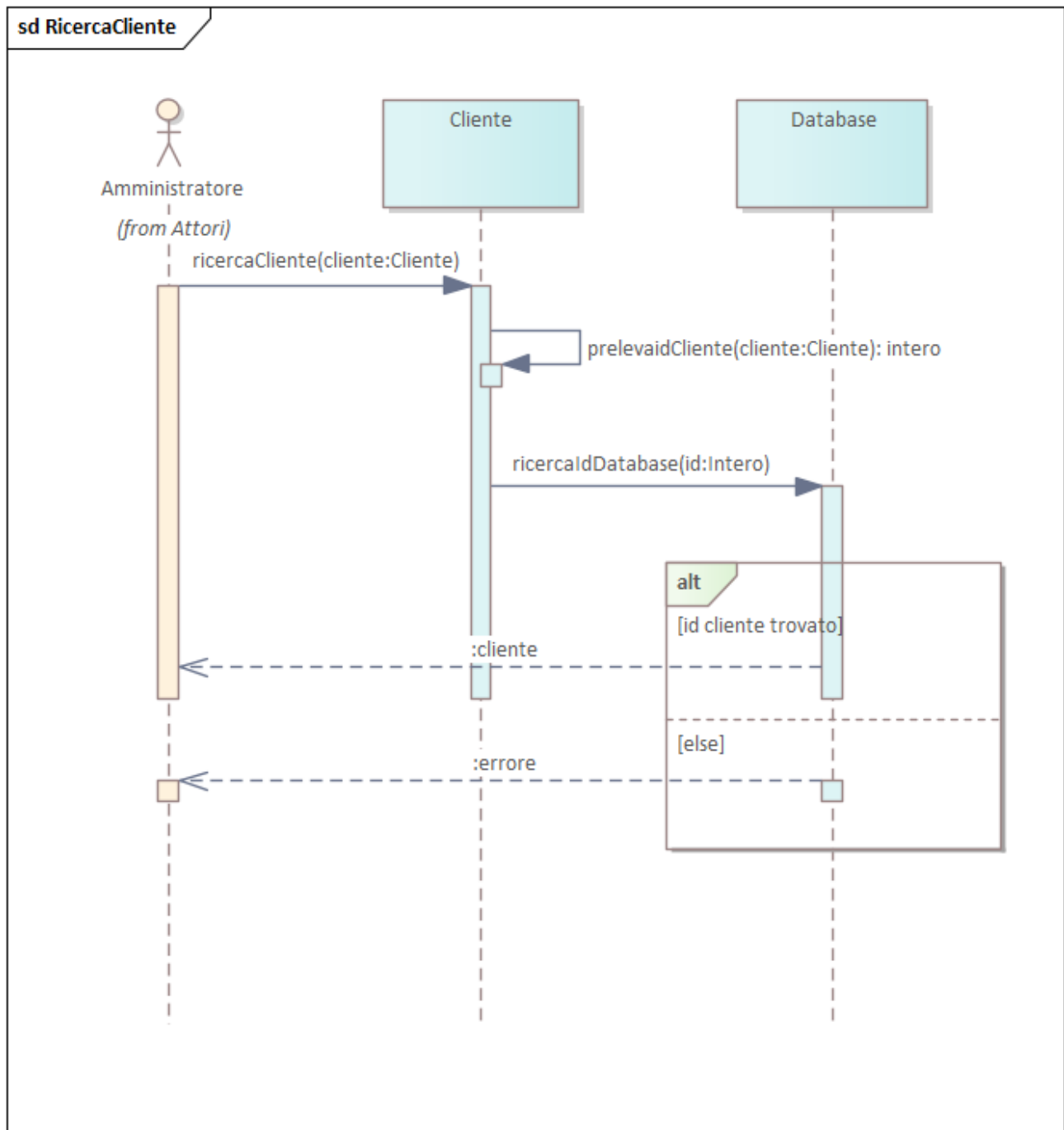


Figura 22 - Diagramma di sequenza: Ricerca Cliente



### 3.3.7 RICERCAPRODOTTO

Tale diagramma modella l'interazione dell'attore Amministratore con il sistema per la ricerca di un prodotto.

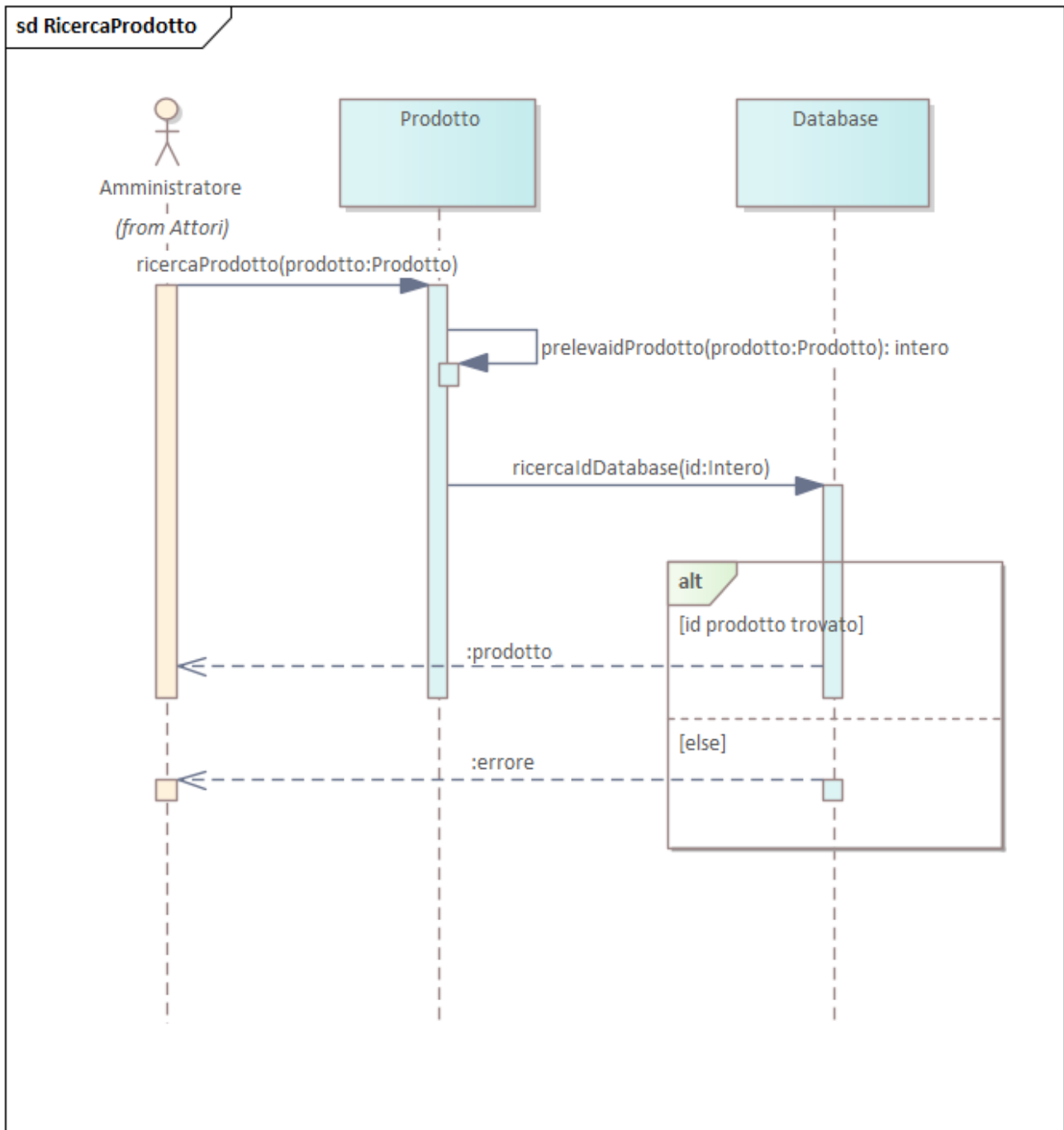


Figura 23 - Diagramma di sequenza: RicercaProdotto

### 3.3.8 VENDIPRODOTTO

Tale diagramma modella l'interazione dell'attore Amministratore con il sistema per la vendita di un prodotto.

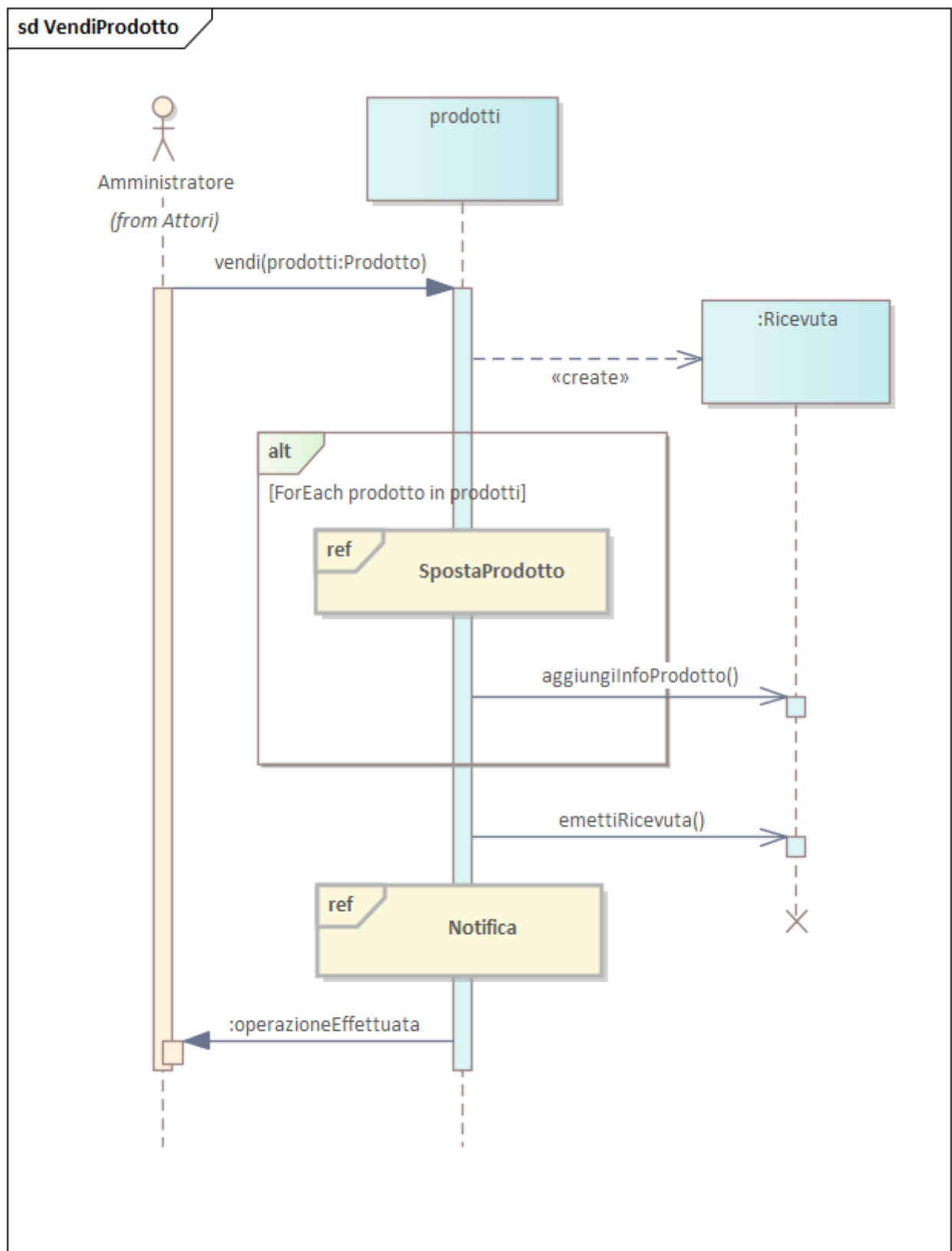


Figura 24 - Diagramma di sequenza: VendiProdotto

### 3.3.9 VISUALIZZASTATISTICHE

Tale diagramma modella l'interazione dell'attore Amministratore con il sistema per la visualizzazione di statistiche disponibili.

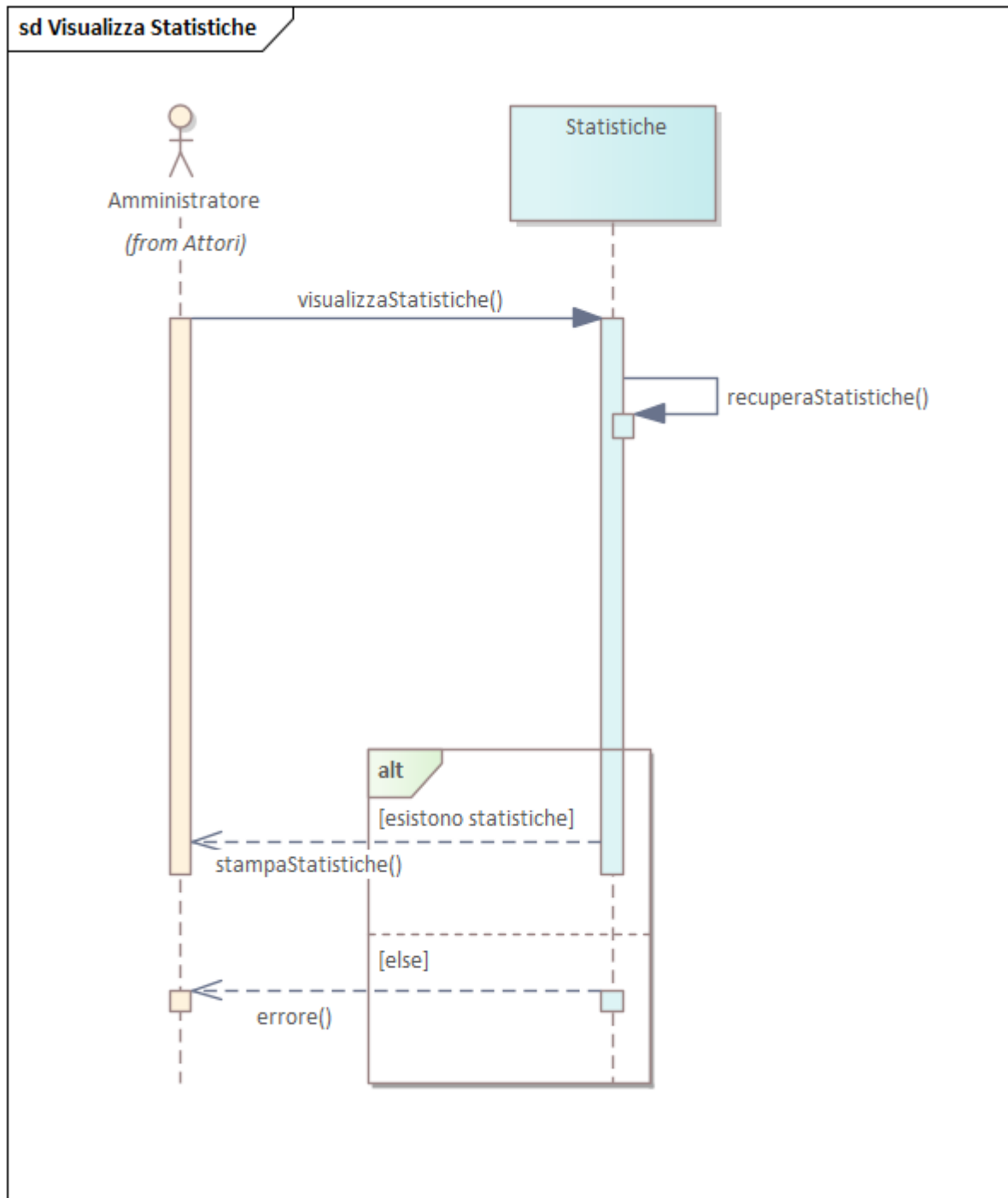


Figura 25 - Diagramma di sequenza: Visualizza Statistiche

## 4 DIAGRAMMI DI PROGETTAZIONE

Sono una raffinazione dei diagrammi delle classi di analisi infatti le seguono temporalmente. Procedendo con la progettazione, ci siamo resi conto che le classi individuate durante l'analisi, non erano sufficienti per implementare il sistema nel codice, in quanto mancavano degli aspetti accessori che se implementati nelle classi individuate, sarebbero stati fuori luogo.

Per il motivo sopra citato è stato aggiunto un altro package con classi di supporto al sistema chiamato *SistemService*. Esso è stato ideato per una corretta suddivisione dei compiti e delle funzioni del sistema.

### 4.1 DIAGRAMMI DELLE CLASSI DI PROGETTAZIONE

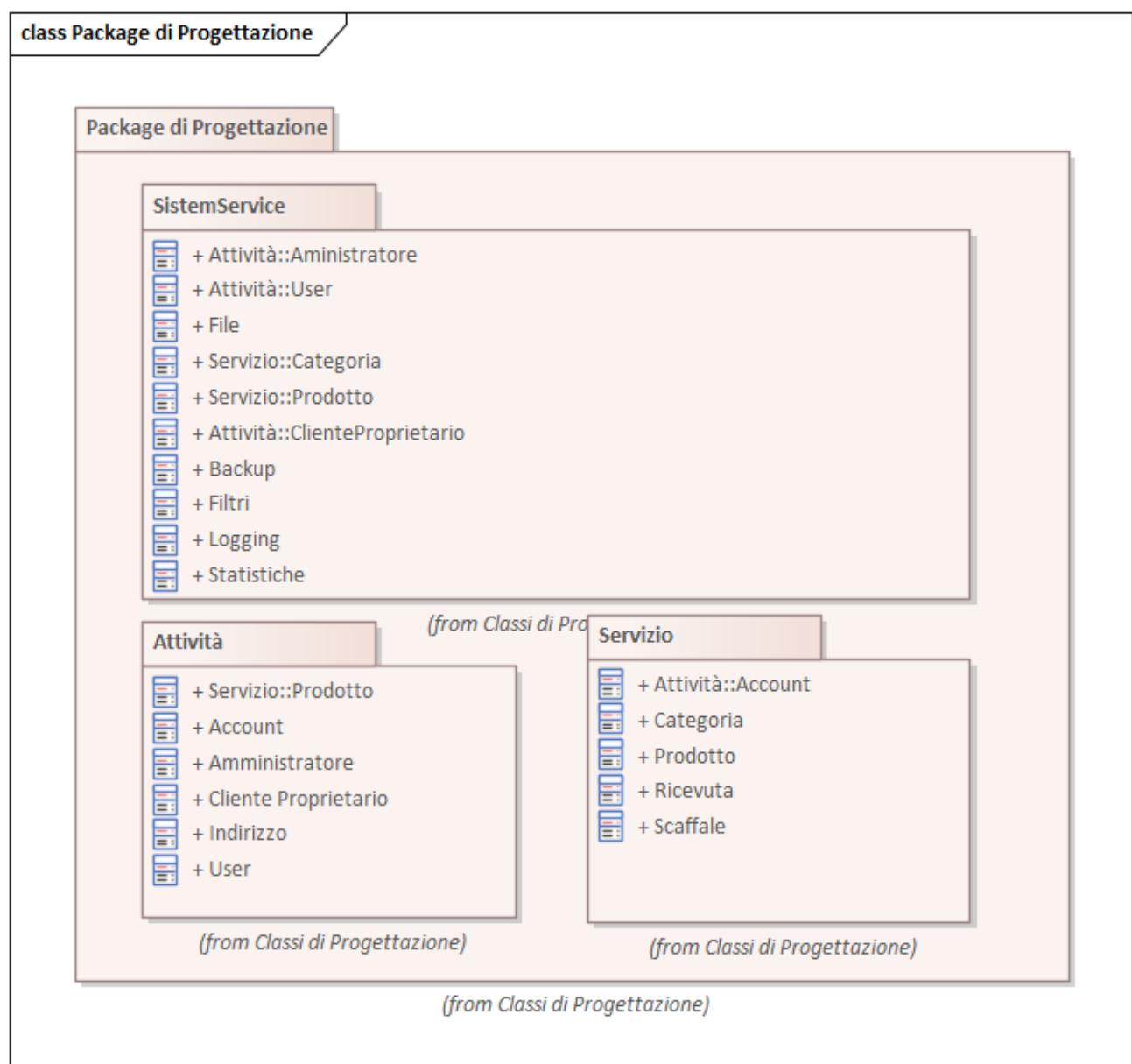


Figura 26 - Package di Progettazione

Nella Figura 27 vediamo il diagramma di progettazione delle classi di *Attività*, questo diagramma rappresenta appieno le potenzialità della classe, dotata di tutti i metodi individuati per il corretto funzionamento delle figure che possono interagire con il sistema



Figura 27 - Diagramma di progettazione delle classi di Attività

Nella Figura 28 vediamo il diagramma di progettazione delle classi di *Servizio*, questo diagramma rappresenta appieno le potenzialità della classe, dotata di tutti i metodi individuati per il corretto funzionamento della struttura e dei servizi che offre.

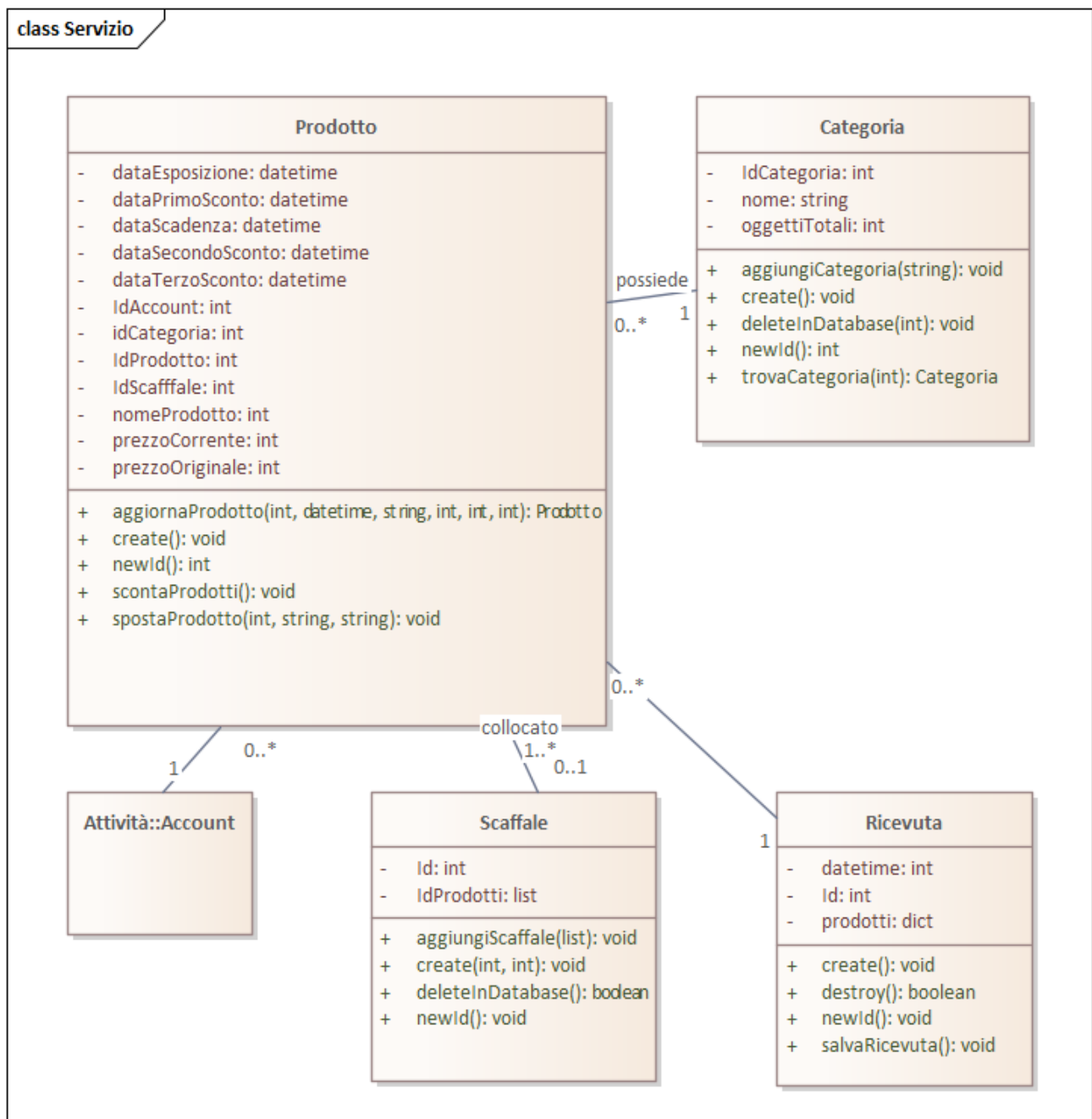


Figura 28 - Diagramma di progettazione delle classi di Servizio

Nella Figura numero 29 troviamo il diagramma di progettazione delle classi di SistemService,

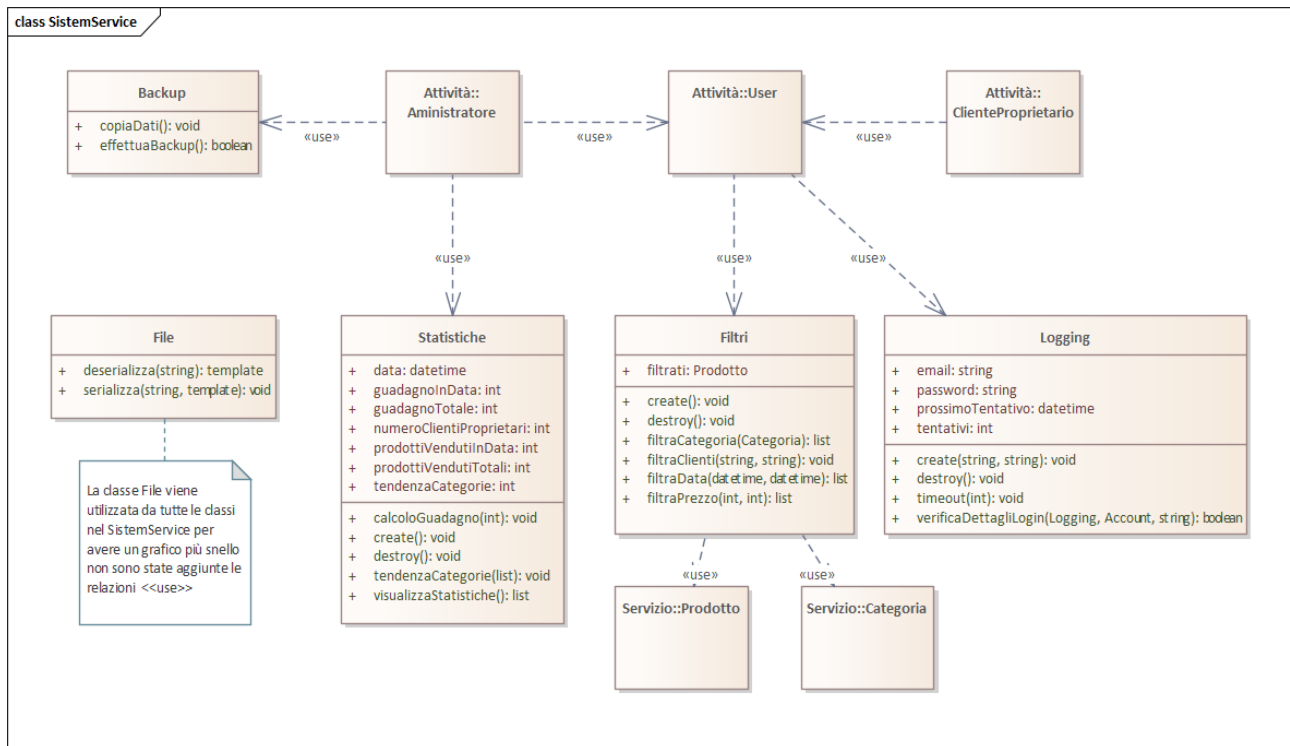


Figura 29 - Diagramma di progettazione delle classi di SistemService

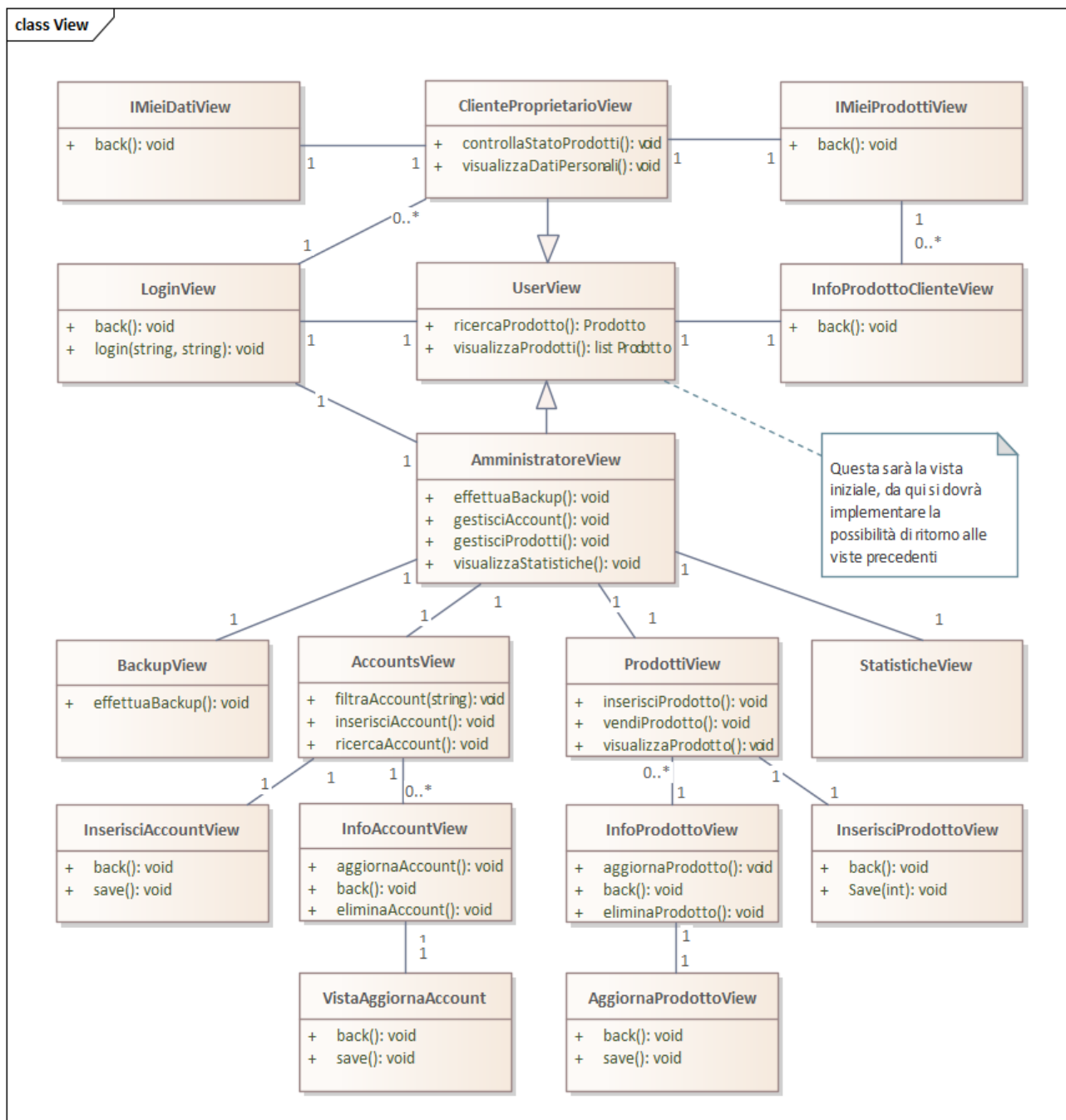


Figura 30 - Diagramma di progettazione delle classi di view



## 4.2 DIAGRAMMI DEI COMPONENTI

Una progettazione per componenti è alla base della moderna ingegneria del software, tale approccio permette di individuare già nella fase di sviluppo dei confini tra varie parti del sistema, grazie a questo il software potrà essere rimodellato e adattato a varie situazioni di funzionamento. Di seguito si riportano alcuni diagrammi di come questo concetto è stato pensato per il nostro sistema con un determinato grado di dettagli interni.

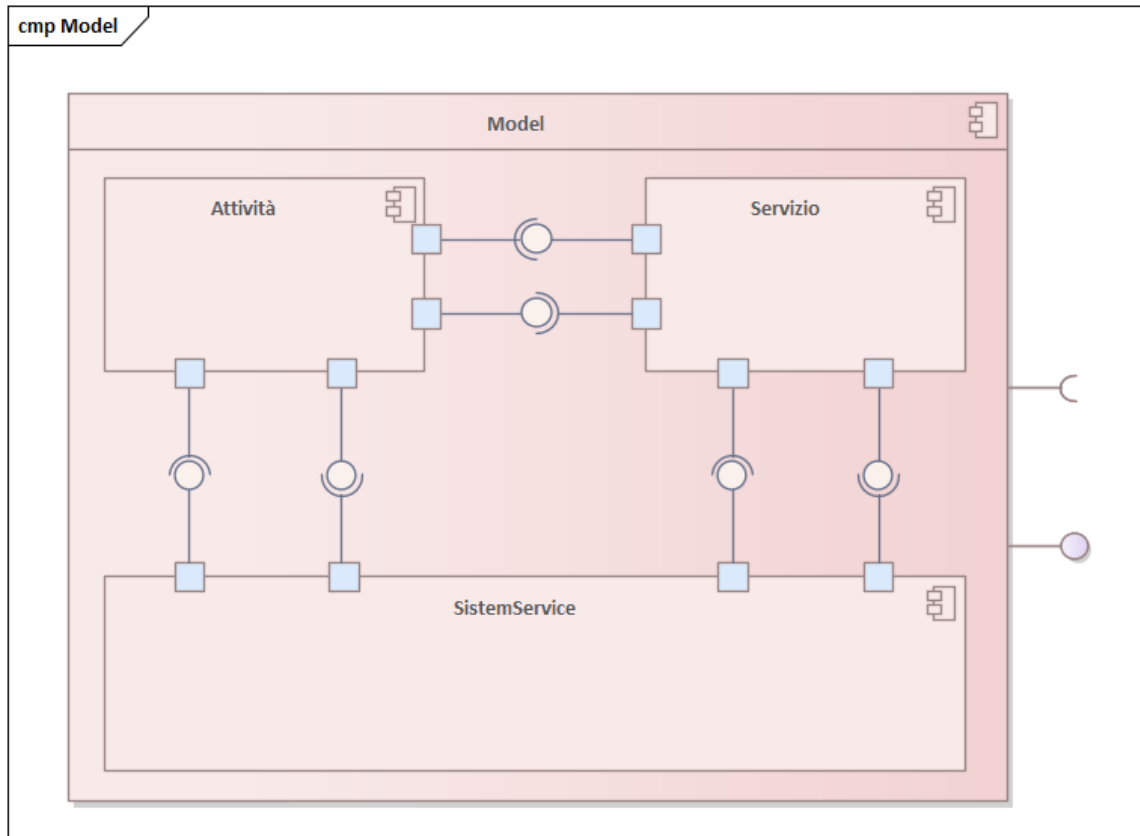


Figura 31 - Diagramma dei componenti: Model

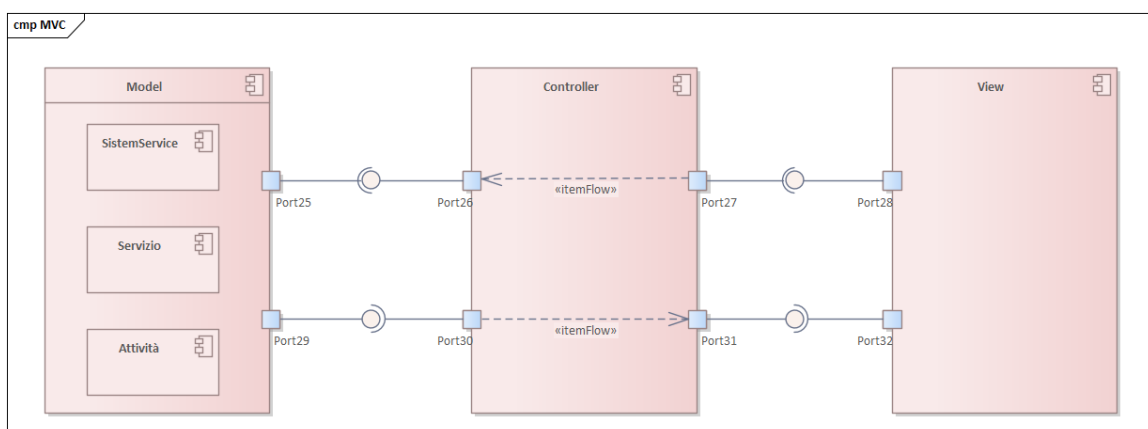


Figura 32 - Diagramma dei componenti: MVC

## 4.3 DIAGRAMMI DELLE MACCHINE A STATI

### 4.3.1 MACCHINE A STATI: BACKUP

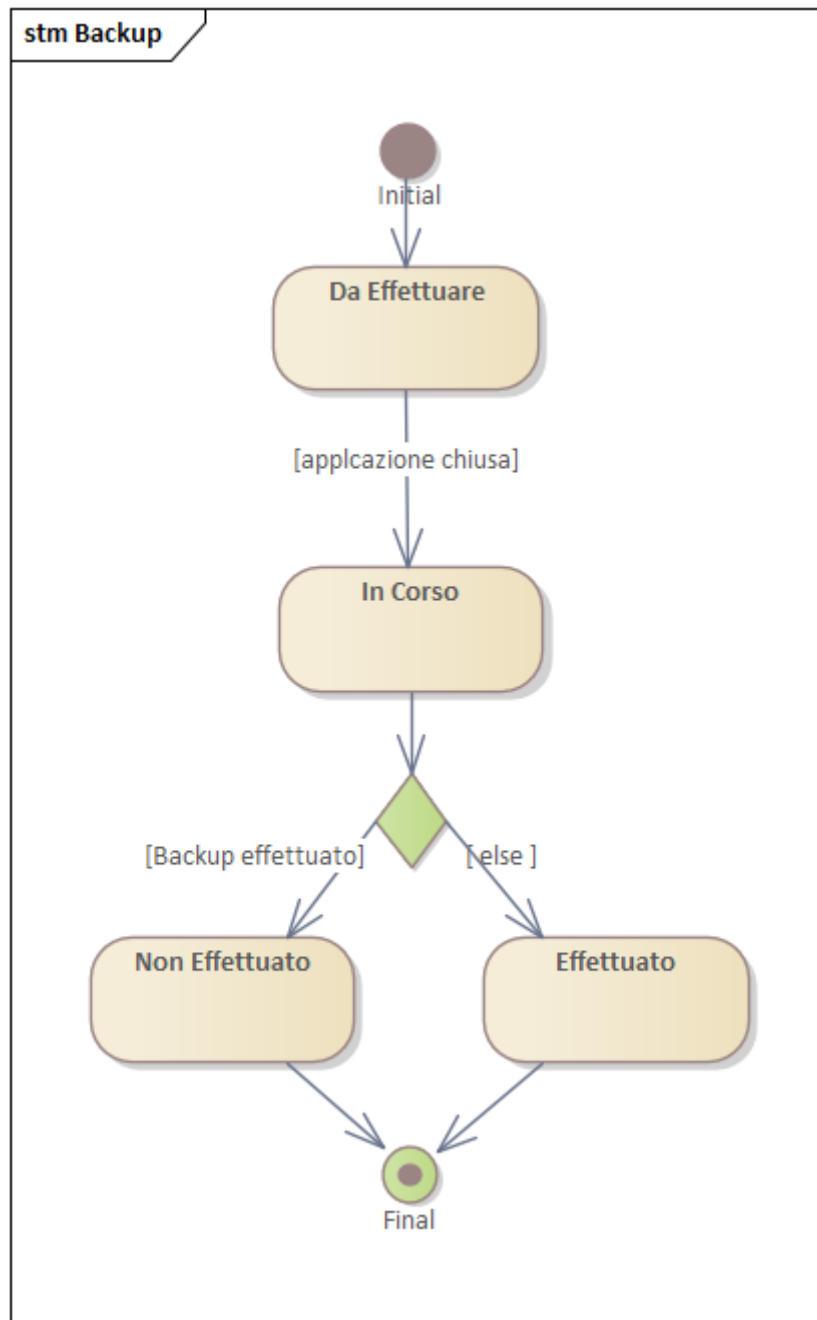


Figura 33 - Diagramma della macchina a stati: Backup

#### 4.3.2 MACCHINE A STATI: CATEGORIA

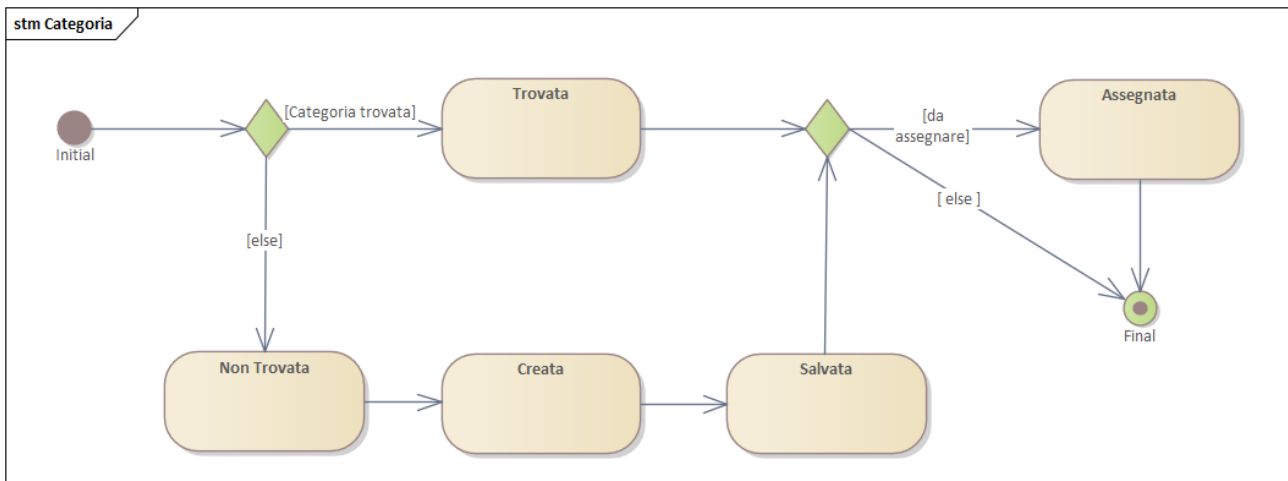


Figura 34 - Diagramma della macchina a stati: Categoria

#### 4.3.3 MACCHINE A STATI: LOGIN

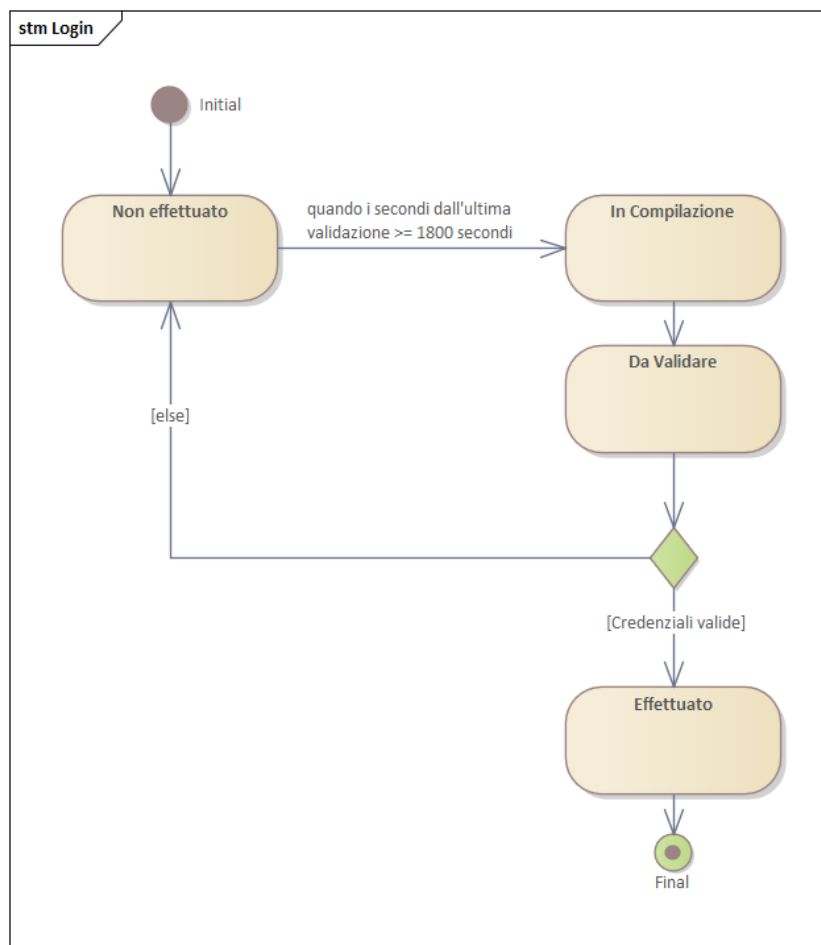


Figura 35 - Diagramma della macchina a stati: Login

#### 4.3.4 MACCHINE A STATI: PRODOTTO

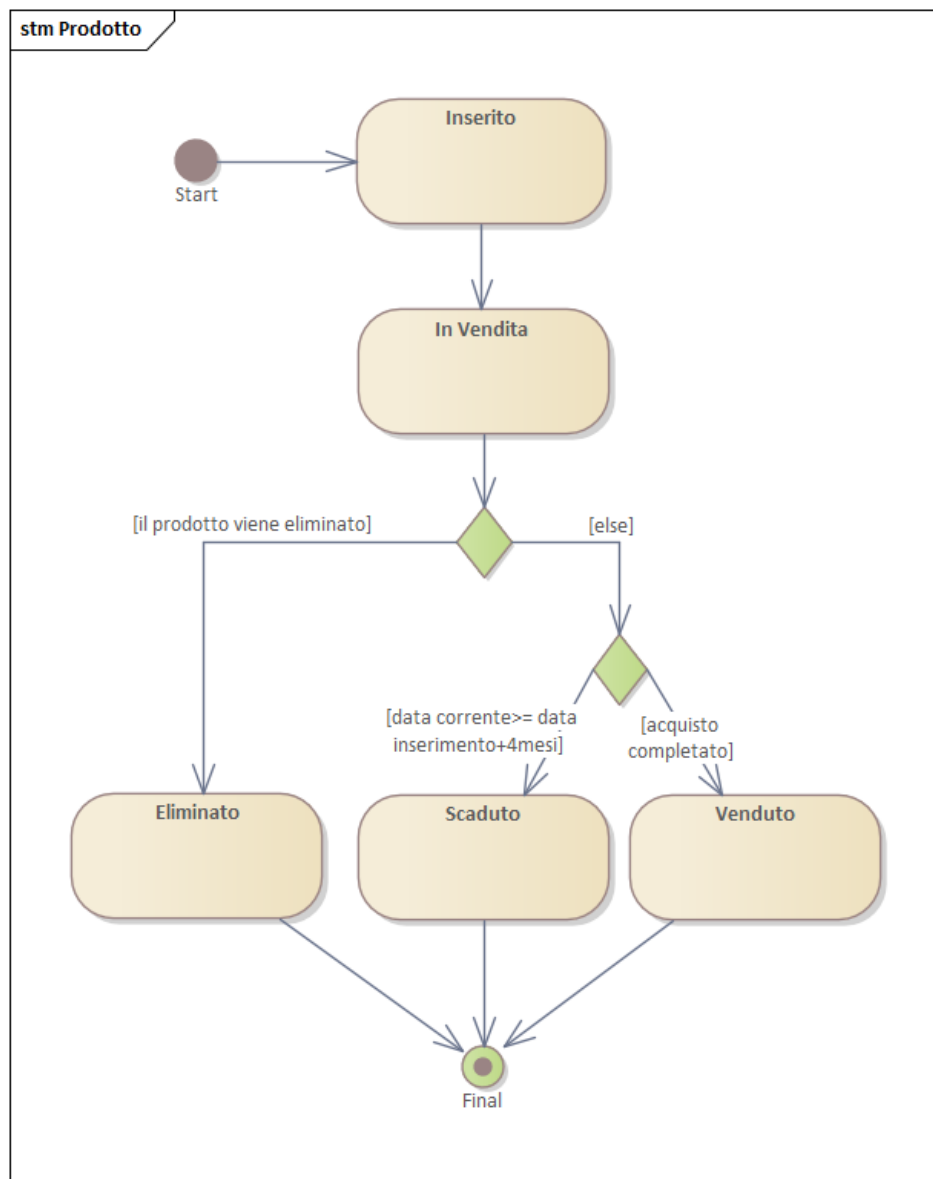


Figura 36 - Diagramma della macchina a stati: Prodotto

#### 4.3.5 MACCHINE A STATI: RICEVUTA

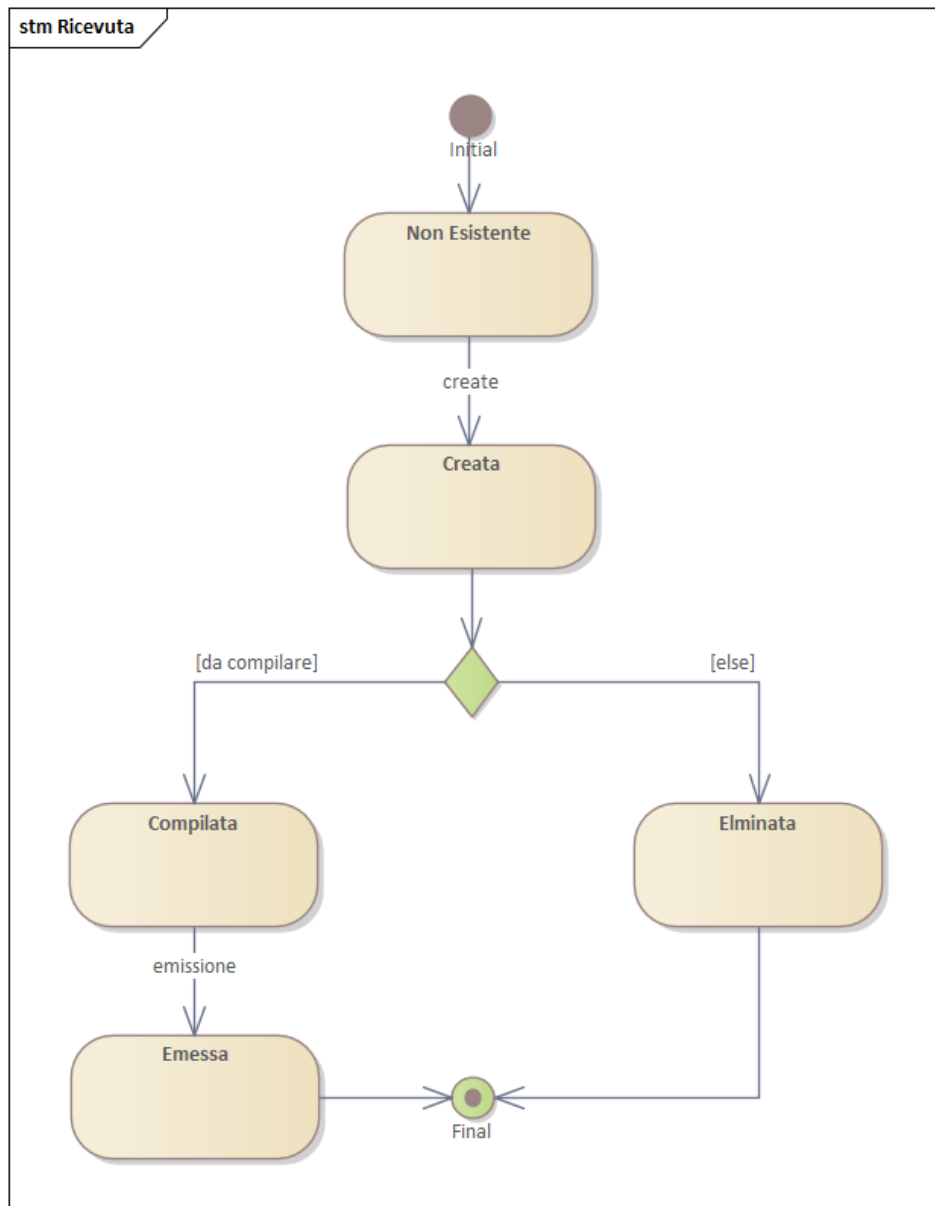


Figura 37 - Diagramma della macchina a stati: Ricevuta

## 5 IMPLEMENTAZIONE

L'implementazione del sistema avverrà in Python 3.10.4

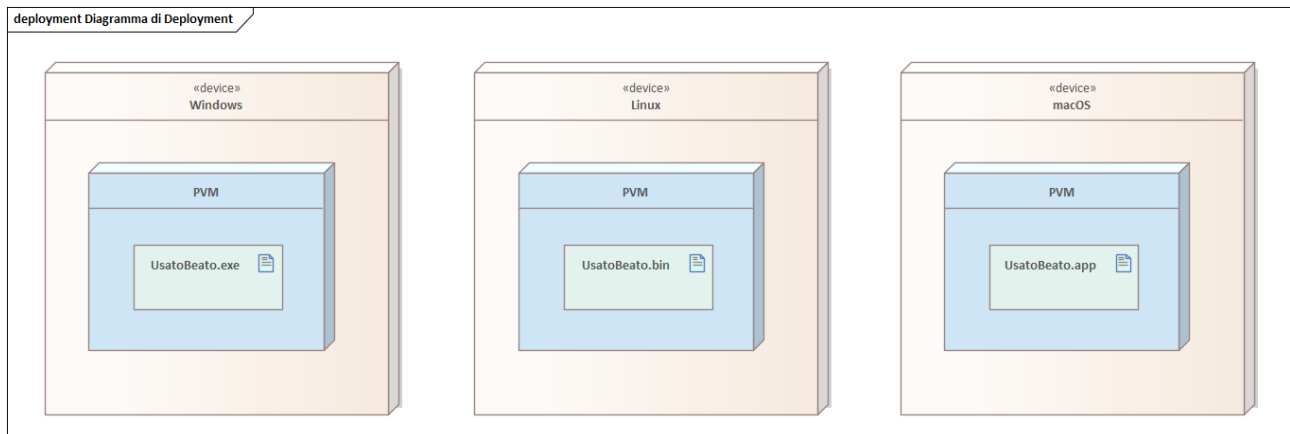
Pattern utilizzati durante la realizzazione:

- MVC (Model View Controller)
- Builder

Il progetto è stato implementato con librerie compatibili con vari sistemi operativi, il codice eseguibile verrà distribuito sui seguenti sistemi operativi:

- Windows
- Linux
- macOS

### 5.1 DIAGRAMMA DI DEPLOYMENT



*Figura 38 - Diagramma di deployment*

## 6 MOCKUP

I seguenti Mockup sono stati realizzati con il modulo ufficiale per Python: PySide6

Tale modulo permette l'accesso completo a **Qt 6.0+ framework** e inoltre integra anche il **QtDesigner**, un software che ci ha permesso la progettazione facilitata dei Mockup.

Il fine è stato rendere dinamica la progettazione delle interfacce grafiche e quindi evitare una struttura monolitica, per questo motivo è stato utilizzato per la progettazione dei Mockup il suddetto software. Vi è la possibilità di compilare la progettazione grafica nel designer in un "NomeFile.ui" e di utilizzarlo come componente nella gestione delle interfacce grafiche. Il fatto di aver reso l'interfaccia grafica un componente permette di effettuare manutenzioni future in maniera estremamente efficiente.

La seguente versione dei Mockup è aggiornata alla prima versione di UsatoBeato rilasciata.

### 6.1 VISTA LOGIN

Tale vista è accessibile dalla vista del User, è necessaria per poter effettuare l'accesso al proprio account

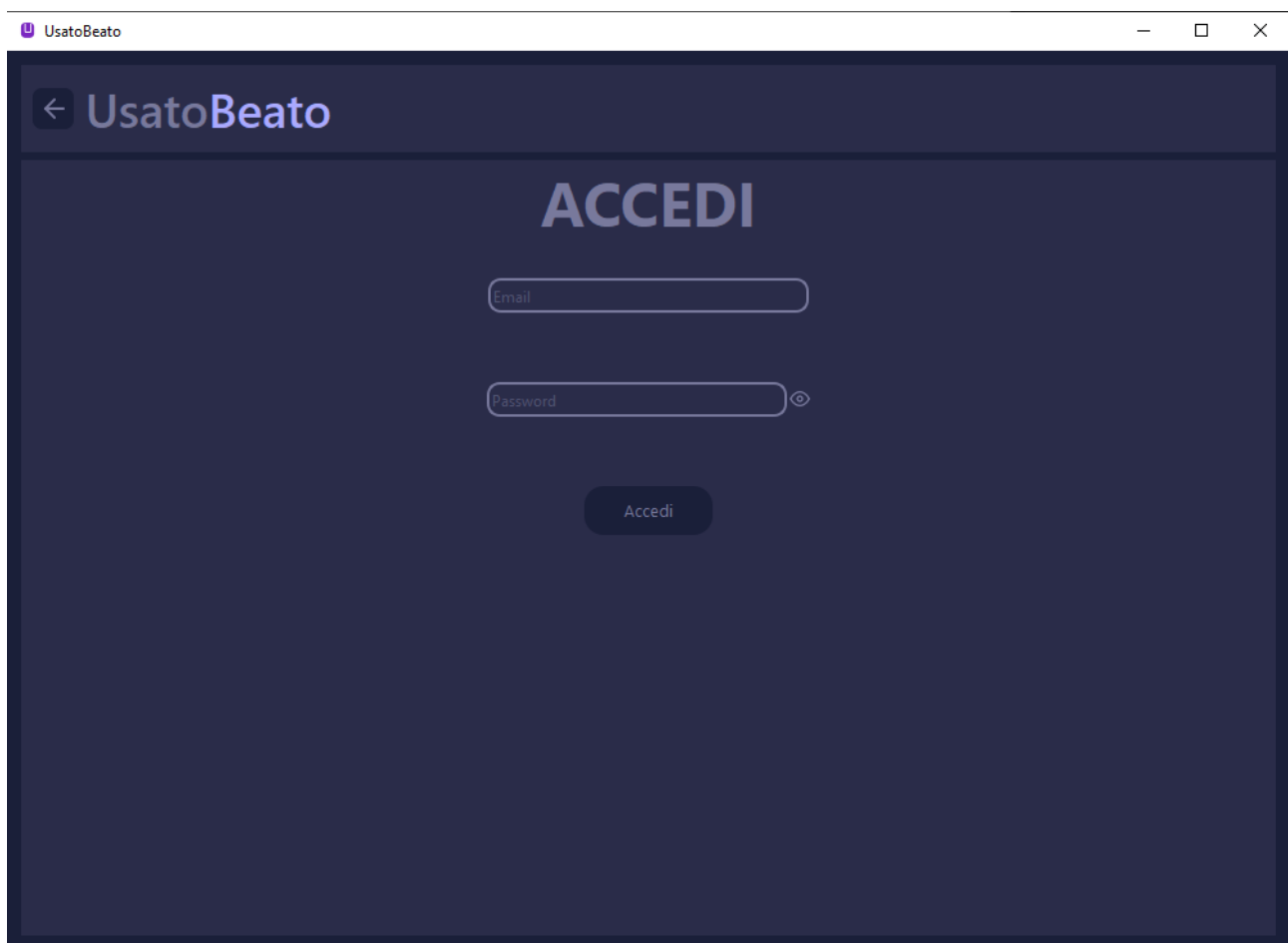


Figura 39 – vista login

## 6.2 VISTA USER

Tale vista è caricata all'avvio dell'eseguibile, vi sono tre barre di attivazione per la selezione di filtri da applicare ai prodotti disponibili nel negozio.

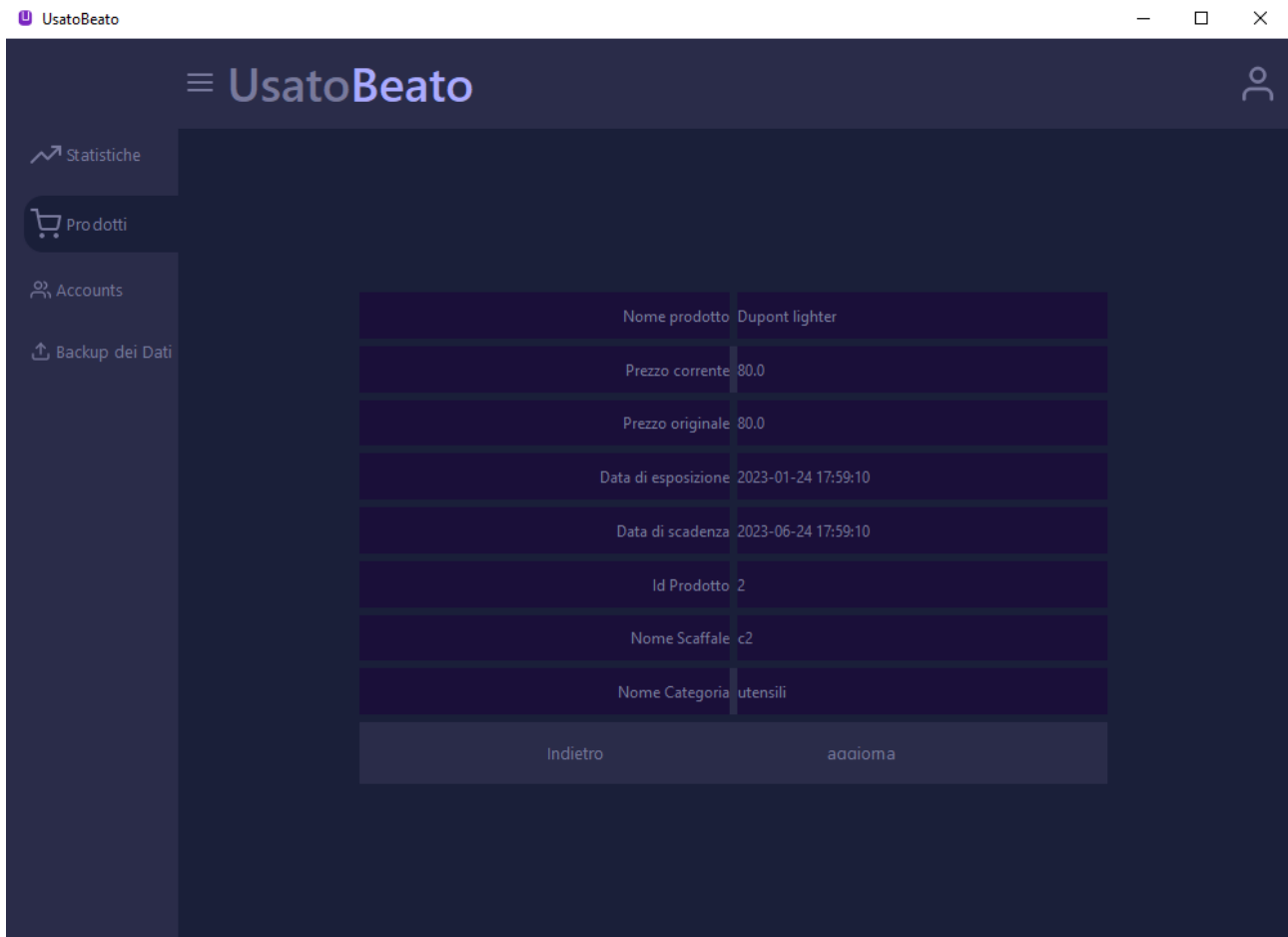


*Figura 40 - vista user*



### 6.3 VISTA AMMINISTRATORE

Nella vista NUMERO che è accessibile dall'utente Amministratore, è possibile visualizzare le informazioni di un prodotto di interesse.



The screenshot displays the 'UsatoBeato' administrator interface. On the left, a sidebar contains navigation links: 'Statistiche', 'Prodotti', 'Accounts', and 'Backup dei Dati'. The main area shows a table with product details for 'Dupont lighter'. The table has two columns: 'Nome prodotto' and 'Valore'. The data rows are as follows:

Nome prodotto	Valore
Dupont lighter	
Prezzo corrente	80.0
Prezzo originale	80.0
Data di esposizione	2023-01-24 17:59:10
Data di scadenza	2023-06-24 17:59:10
Id Prodotto	2
Nome Scaffale	c2
Nome Categoria	utensili

At the bottom of the table, there are two buttons: 'Indietro' and 'aggiorna'.

Figura 41 - vista info prodotto

Nella vista 42 che è accessibile dall'utente Amministratore, è possibile visualizzare le informazioni di un account registrato nel sistema come Cliente-Proprietario.

The screenshot displays the 'UsatoBeato' administrative dashboard. On the left, a sidebar contains navigation links: 'Statistiche', 'Prodotti', 'Accounts', and 'Backup dei Dati'. The main content area shows the details of a specific account. The account information is presented in a table-like format with labels and values. At the bottom of this section, there are two buttons: 'Indietro' (Back) and 'aggiorna' (Update).

Nome	Elliot
Cognome	Alderson
Data di nascita	1986-09-17
Email	elliotalderson86@ecorp.us
Password	password
Id account	1
Numero Telefonico	
Città	New York City
Cap	12342
Citofono	Alderson
Via	East Broadway
Piazza	
Civico	217

[Indietro](#) [aggiorna](#)

*Figura 42 - vista info account*

Nella vista 43 che è accessibile dall'utente Amministratore, è possibile visualizzare le statistiche disponibili per valutare l'andamento del negozio

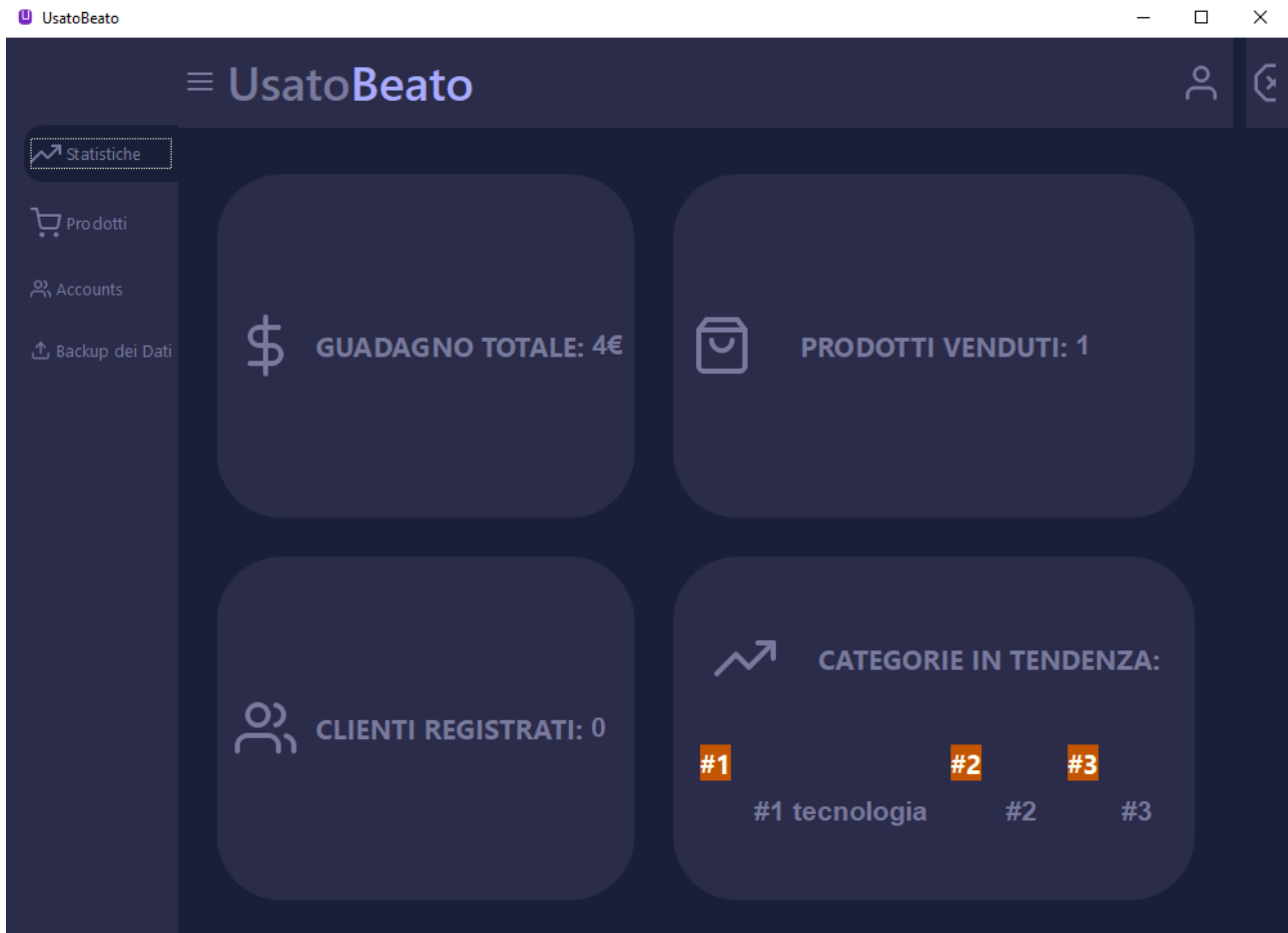


Figura 43 - vista statistiche amministratore

Nella Figura 44 che è accessibile dall'utente Amministratore, è possibile visualizzare la lista dei prodotti disponibili nel negozio.

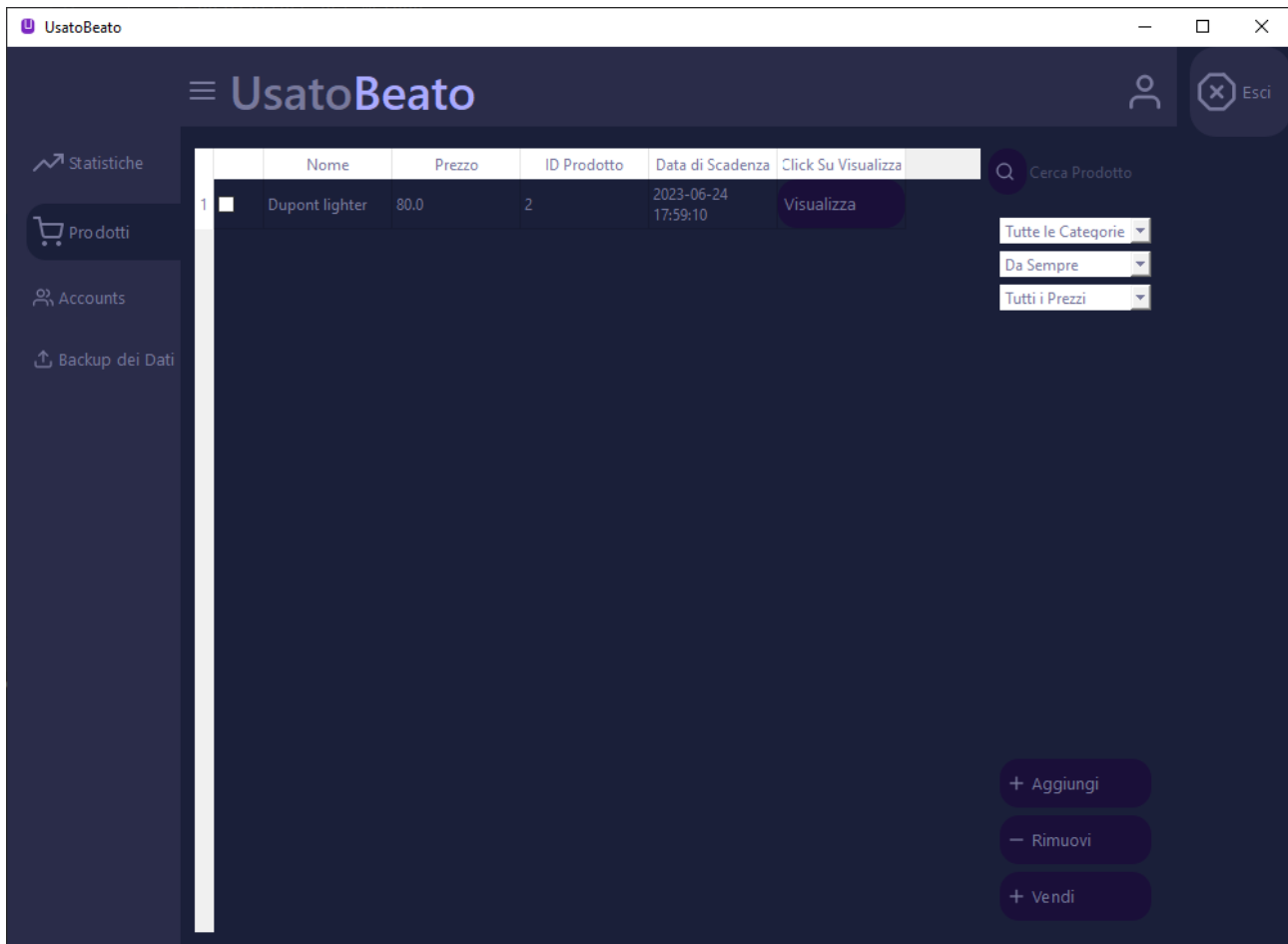
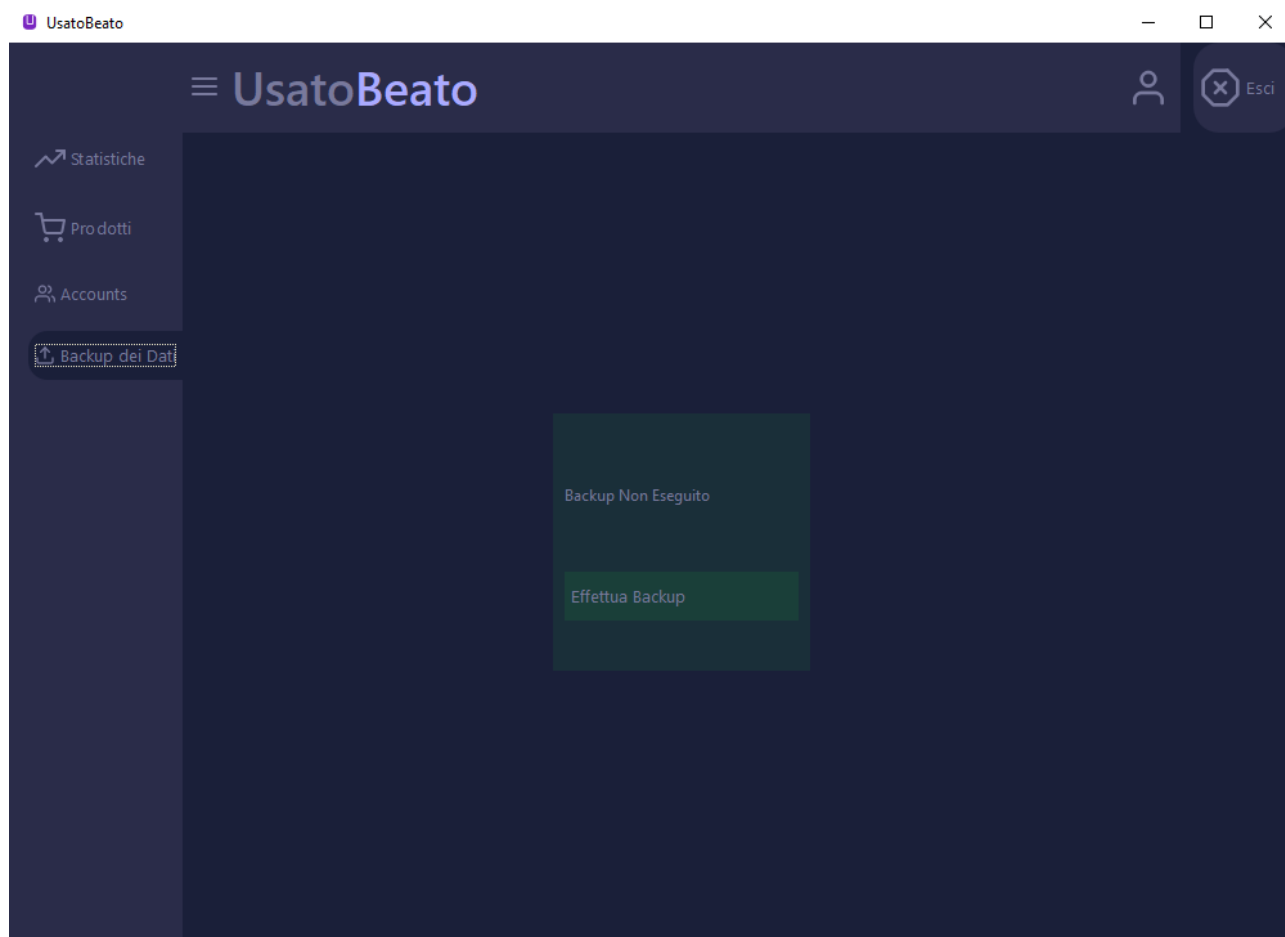


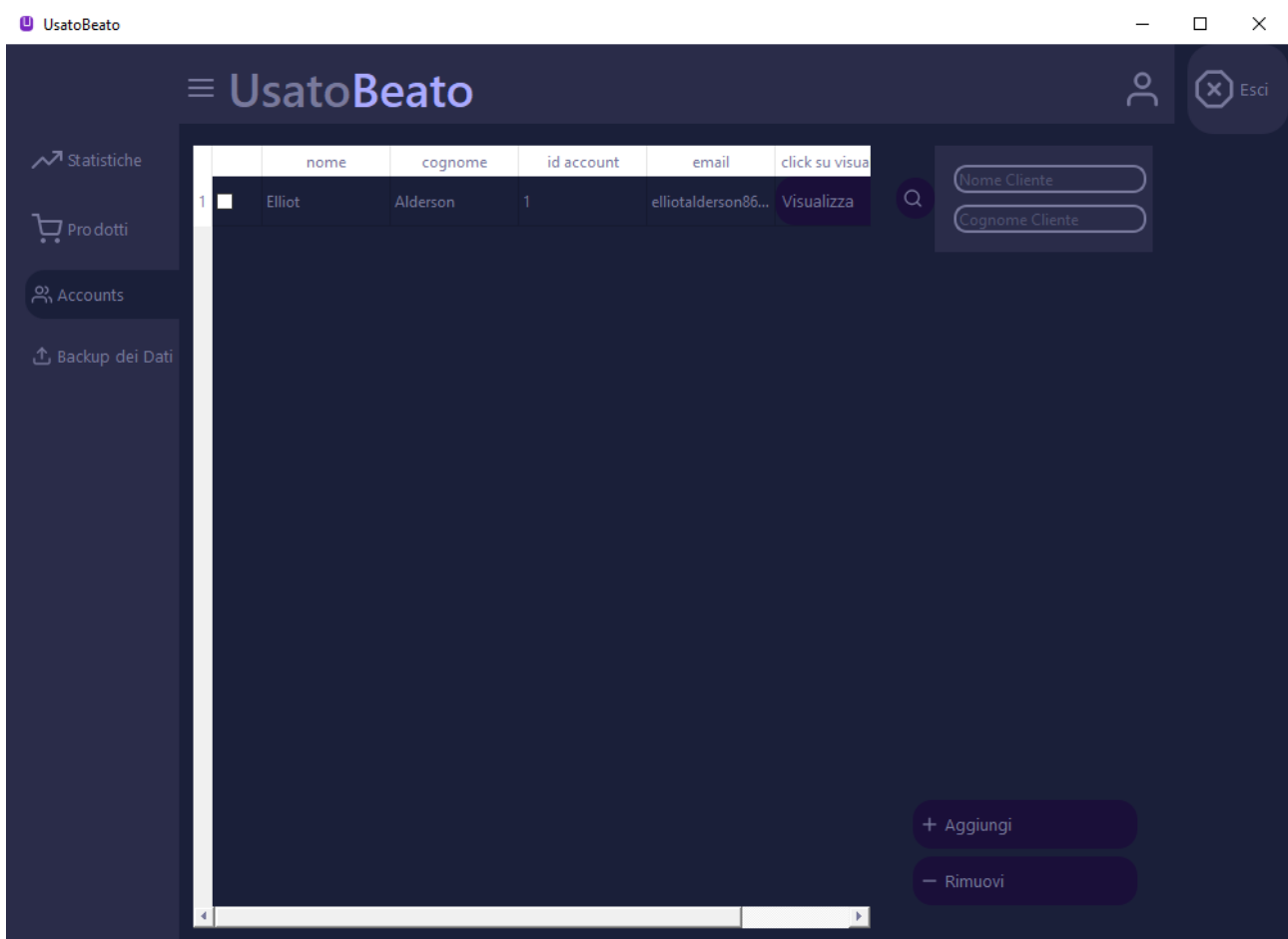
Figura 44 - vista prodotti amministratore

Nella Figura 45 che è accessibile dall'utente Amministratore, è possibile effettuare l'operazione di backup.



*Figura 45 - vista backup*

Nella Figura 46 che è accessibile dall'utente Amministratore, è possibile visualizzare la lista degli utenti registrati nel sistema



*Figura 46 - vista account amministratore*

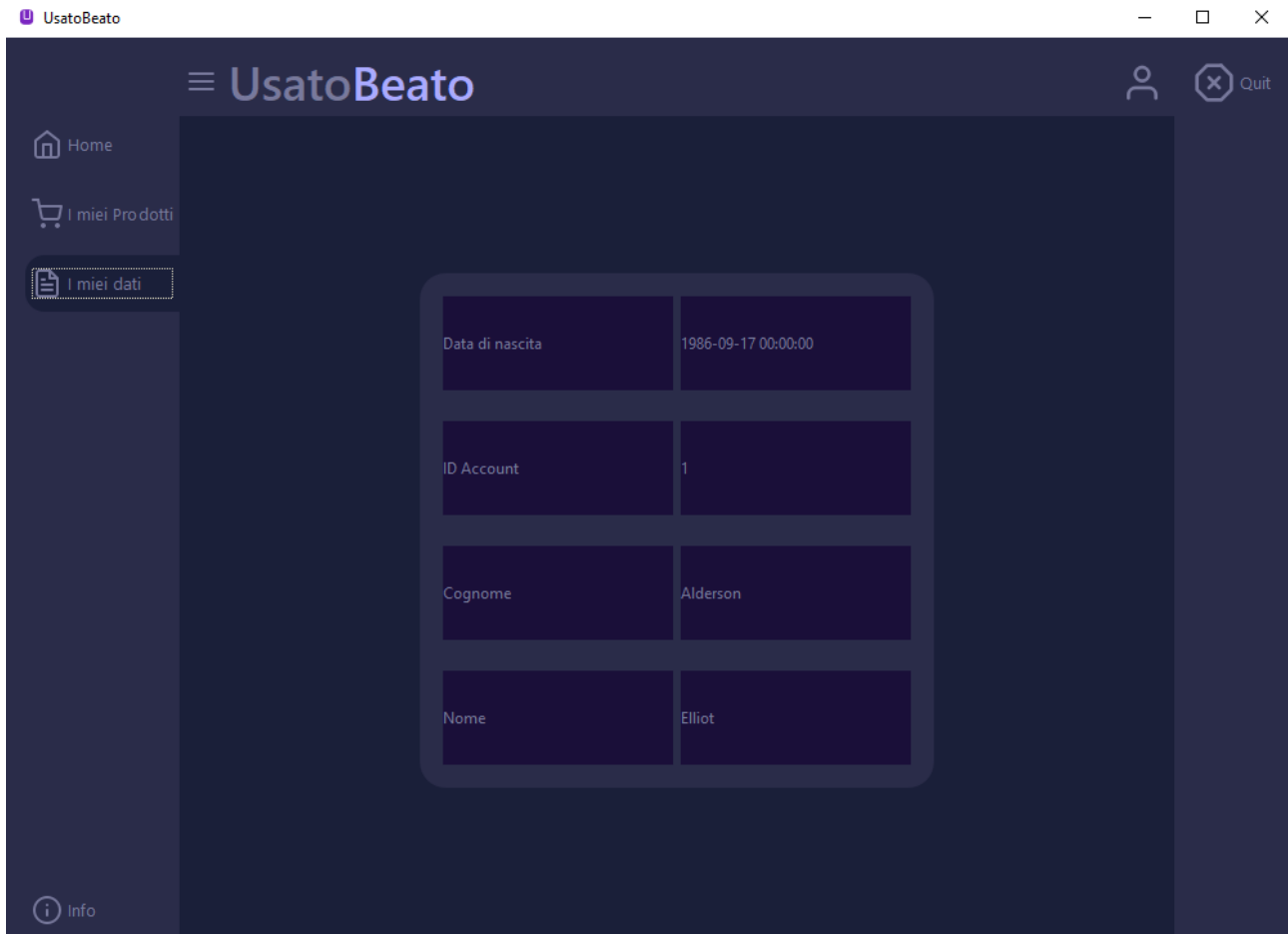
## 6.4 VISTA CLIENTE PROPRIETARIO

Nella Figura 47 accessibile dall'utente Cliente-Proprietario, è possibile visualizzare lo stato degli oggetti associati all'account del cliente.



Figura 47 - vista prodotti del cliente proprietario

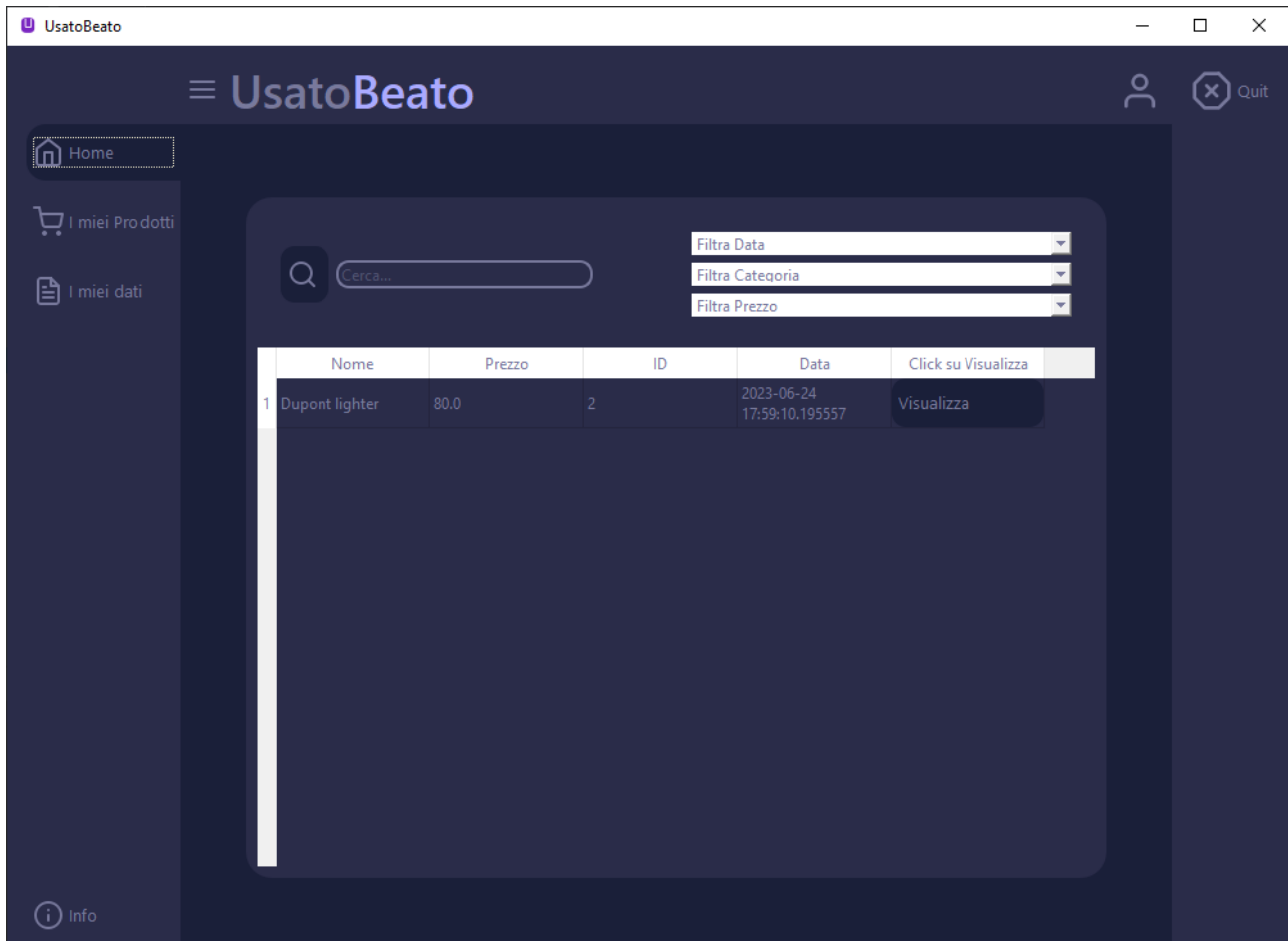
Nella Figura 48 accessibile dall'utente Cliente-Proprietario, è possibile visualizzare i propri dati.



*Figura 48 - vista dati del cliente proprietario*



Nella vista 49 accessibile dall'utente Cliente-Proprietario, è possibile vedere la lista degli oggetti disponibili all'interno del negozio.



*Figura 49 - vista home cliente proprietario*

## 7 PYUNIT

Il testing del sistema è stato portato avanti durante la progettazione con un picco nella fase di fine progettazione del **Model** del sistema, i test eseguiti sono stati fondamentali per il corretto funzionamento del codice, eseguiti grazie alla libreria TestCase del modulo unittest.

Di seguito un estratto di alcuni test eseguiti:

```
def test_vendiProdotti(self):
    listProdotti = list()
    contatore = 5
    for i in range(contatore):
        prodotto = Amministratore().inserisciProdotto(datetime.today(), None, f"{i}", i, "s", "tecnologia")
        listProdotti.append(prodotto)
    Amministratore().vendiProdotti(listProdotti)
    listProdottiVenduti = File().deserializza(PathDatabase().vendutiTxt)
    segnalino = 0
    for venduto in listProdottiVenduti:
        for prodotto in listProdotti:
            if prodotto.idProdotto == venduto.idProdotto:
                segnalino += 1
    self.assertEqual(segnalino, contatore)
```

Figura 50 - test vediProdotti

```
def test_inserisciAccount(self):
    # setup -----
    account1 = Amministratore().inserisciAccount("nuovo", "cognome", '24/05/2000', "nuovoMailodad@gmafil.com", "ciao",
                                                '3812312271',
                                                '63066', 'ciao', 'sbt', '9', 'nessuna', ' ciao1')
    # test-----
    account2 = Account().trovaOggettoTramiteEmail(account1.email)
    self.assertEqual(account1.idAccount, account2.idAccount)
```

Figura 51 - test inserisciAccount

```
def test_aggiornaProdotto(self):
    prodotto = Amministratore().inserisciProdotto(datetime.today(), None, "nomeProdotto", "43", "c3", "Elettronica")
    nuovoNome = "nuovoNomeProdotto"
    Amministratore().aggiornaProdotto("Utensili", "12/10/2021", nuovoNome, "60", "d4", prodotto.idProdotto)
    afterProdotto = Prodotto().trovaOggettoTramiteId(prodotto.idProdotto)
    self.assertEqual(nuovoNome, afterProdotto.nomeProdotto)
```

Figura 52 - test aggiornaProdotto

```

def test_aggiornaAccount(self):
    account1 = Amministratore().inserisciAccount("nome", "cognome", "10/10/2000", "vecchiaEmail@gmail.com",
                                                "password", "12351324", "61200", "nome", "Macerata", "30", None,
                                                "Romaaa")
    account2 = Amministratore().aggiornaAccount(None, "PoLo", "12/12/1987", "nuovaEmial@usatobeato.it",
                                                account1.idAccount, "121313245", None)
    self.assertEqual(account1.idAccount, account2.idAccount)
    self.assertNotEqual(account1.nome, account2.nome)
    self.assertNotEqual(account1.cognome, account2.cognome)
    self.assertNotEqual(account1.dataDiNascita, account2.dataDiNascita)
    self.assertNotEqual(account1.email, account2.email)
    self.assertNotEqual(account1.numeroTelefonico, account2.numeroTelefonico)

```

*Figura 53 - test aggiornaAccount*