

Sistema di Voto e Scrutinio Elettronico

Autore: Andrei Georgiani Talpalaru

Matricola: 942824

1 Descrizione del problema

1.1 Scopo del sistema

Il sistema di voto e scrutinio elettronico è un'applicazione che permette agli elettori di votare secondo diverse modalità di voto e di definire un vincitore secondo diverse modalità di definizione del vincitore.

1.2 Glossario

Utente

Utilizzatore del sistema, che può essere un'**elettore** o un'**impiegato/gestore** del sistema.

Elettore

Utente del sistema di voto e scrutinio elettronico che ha il diritto di voto.

Impiegato/Gestore del sistema

Utente del sistema di voto e scrutinio elettronico che può configurare una sessione di voto, far partire o chiudere la fase di scrutinio e visualizzare l'esito del voto.

Scheda

Strumento con cui l'**elettore** esprime la propria preferenza in occasione di votazioni.

Quorum

Il numero minimo di partecipanti o elettori necessario affinché una votazione sia valida.

Totem

Computer disponibile all'interno del seggio elettorale che viene usato solitamente dall'**elettore** per esprimere il proprio voto di **persona**. Viene attivato da parte di un gestore.

Applicazione

Programma utilizzabile su computer, attraverso cui si può accedere al sistema di voto e scrutinio elettronico.

Seggio Elettorale

Il luogo dove gli **elettori** si recano per votare.

Scheda Bianca

Scheda di un **elettore** che si è astenuto dal votare.

Maggioranza Assoluta

Si ha la maggioranza assoluta quando si hanno i **50% + 1** dei voti espressi.

Identificazione e Verifica del Diritto al Voto

L'identificazione rappresenta il riconoscimento della persona che vuole votare. Nel caso il voto venga espresso in persona, su un totem, il riconoscimento e la verifica del diritto al voto vengono fatte dal gestore presente. Nel caso il voto venga espresso da remoto, su un computer, il riconoscimento e la verifica del diritto al voto vengono fatte attraverso un'operazione di Login.

1.3 Requisiti

1.3.1 Requisiti Funzionali

1.3.1.1 Requisiti Utente

1. L'**utente** deve potersi autenticare nell'**applicazione** come **elettore** o **gestore del sistema**.
2. L'**elettore** deve potere esprimere il suo voto di **persona**, in un **seggio elettorale**, utilizzando un **totem**, o a **distanza**, utilizzando l'**applicazione** su un dispositivo elettronico, ma comunque non prima di aver svolto la fase d'identificazione e di verifica del diritto al voto.
3. Come **gestore di sistema**, voglio poter configurare una sessione di voto, inserire le liste dei candidati/domanda del referendum, far partire/fermare la fase di scrutinio e visualizzare l'esito del voto.
4. Come **gestore di sistema**, quando sto configurando una sessione di voto, voglio poter scegliere tra le seguenti modalità di voto:
 1. Voto Ordinale
 2. Voto Categorico
 3. Voto Categorico con Preferenze
 4. Referendum
5. Come **gestore di sistema**, quando sto configurando una sessione di voto, voglio poter scegliere tra le seguenti modalità di scelta del vincitore:
 1. Maggioranza
 2. Maggioranza Assoluta
 3. Referendum senza Quorum
 4. Referendum con Quorum
6. Come **elettore**, devo poter presentare una **scheda bianca**.
7. Come **elettore**, voglio potermi registrare nel sistema.
8. Come **gestore di sistema**, voglio poter configurare un totem per farlo usare da parte di un'elettore che vota **in persona**.

1.3.1.2 Requisiti di sistema

- 1

1. Il sistema deve considerare le due tipologie d'**utente**.
2. Il sistema deve permettere operazioni diverse all'interno del applicazione rispetto al tipo d'**utente** autenticato.

- 2

1. Nel caso si votasse di persona, il sistema non deve fare la fase d'identificazione e verifica del diritto di voto perché si assume che queste avvengano manualmente.

2. Nel caso si votasse a distanza, il sistema deve implementare un processo di Login per la fase d'identificazione per l'**elettore**.

- 3

1. Il sistema deve implementare un'apposita interfaccia grafica che permette al **gestore di sistema** di scegliere la modalità di voto e di calcolo del vincitore.
2. Il sistema deve implementare un modo con cui il **gestore di sistema** può inserire le liste dei candidati.
3. Il sistema deve implementare un modo con cui il **gestore di sistema** può inserire la domanda del referendum.
4. Il sistema deve permettere al **gestore di sistema** di far partire e di fermare la fase di voto.
5. Il sistema deve visualizzare automaticamente, dopo la chiusura della fase di voto, l'esito del voto.

- 4.1

1. Il sistema deve richiedere all'**elettore** di ordinare i candidati (o gruppi/partiti) presenti nella **scheda** in base alle proprie preferenze.

- 4.2

1. Il sistema deve richiedere all'**elettore** di scegliere una preferenza per il candidato (o gruppo/partito) presente nella scheda.

- 4.3

1. Il sistema deve richiedere all'**elettore** di scegliere una preferenza per un gruppo/partito presente nella **scheda**, e in seguito di indicare una o più preferenze tra i candidati del gruppo/partito selezionato.

- 4.4

1. Il sistema deve presentare all'**elettore** una domanda, con la quale si chiede se si sia favorevoli o contrari a un determinato quesito mostrato sullo schermo, e deve dare la possibilità di rispondere con **Favorevole** o **Non Favorevole**.

- 5

1. Il sistema deve impedire al **gestore di sistema** di selezionare altre opzioni tranne **Referendum senza Quorum** e **Referendum con Quorum** quando **Referendum** è selezionato come modalità di voto.
2. Il sistema deve impedire al **gestore di sistema** di selezionare **Referendum senza Quorum** e **Referendum con Quorum** quando la modalità di voto non è **Referendum**.

- 5.1

1. Dopo la chiusura della fase di voto, il sistema deve considerare vincitore il candidato che ha ottenuto il maggior numero di voti.
2. Dopo la chiusura della fase di voto, nel caso ci sia pareggio, il sistema deve mostrare un messaggio significativo che indica al **gestore di sistema** questa situazione.

- 5.2

1. Dopo la chiusura della fase di voto, il sistema deve considerare vincitore il candidato che ha ottenuto la **maggioranza assoluta** dei voti.
2. Dopo la chiusura della fase di voto, nel caso non ci sia un candidato che abbia **maggioranza assoluta** dei voti, il sistema deve mostrare un messaggio significativo che indica al **gestore di sistema** questa situazione.

- 5.3

1. Dopo la chiusura della fase di voto, il sistema deve procedere al conteggio dei voti indipendentemente se abbia partecipato o meno alla consultazione la maggioranza degli aventi diritto al voto.
2. Dopo la chiusura della fase di voto, nel caso ci sia pareggio, il sistema deve mostrare un messaggio significativo che indica al **gestore di sistema** questa situazione.

- 5.4

1. Dopo la chiusura della fase di voto, il sistema deve procedere al conteggio dei voti espressi solo nel caso in cui abbia partecipato alla consultazione la maggioranza degli aventi diritto al voto.
2. Dopo la chiusura della fase di voto, nel caso in cui non abbia partecipato alla consultazione la maggioranza degli aventi diritto al voto, il sistema

deve mostrare un messaggio significativo che indica al **gestore di sistema** questa situazione.

3. Dopo la chiusura della fase di voto, nel caso ci sia pareggio, il sistema deve mostrare un messaggio significativo che indica al **gestore di sistema** questa situazione.

- 6

1. Il sistema deve implementare un modo per permettere all'**elettore** di continuare anche se non ha fatto una selezione all'interno della **scheda**.

- 7

1. Il sistema deve implementare un modo per registrarsi attraverso l'**applicazione**.

- 8

1. Il sistema deve implementare per il **gestore di sistema** un modo per far comunicare applicazione e totem.

1.3.2 Requisiti Non Funzionali

1.3.2.1 Requisiti di sicurezza

1. Il voto espresso deve rimanere segreto e non deve essere riconducibile all'**elettore**.
2. Si deve prevedere un sistema di auditing per la verifica del corretto funzionamento dell'applicazione, basato su un sistema di log.

1.3.2.2 Requisiti di sviluppo

1. Il sistema dev'essere implementato in Java.
2. L'interfaccia grafica del sistema dovrebbe essere implementata usando JavaFX.
3. Il progetto deve contenere i seguenti Design Pattern:
 1. Singleton
 2. MVC
 3. DAO
 4. Observer
 5. Abstract Factory

1.3.2.3 Requisiti di affidabilità

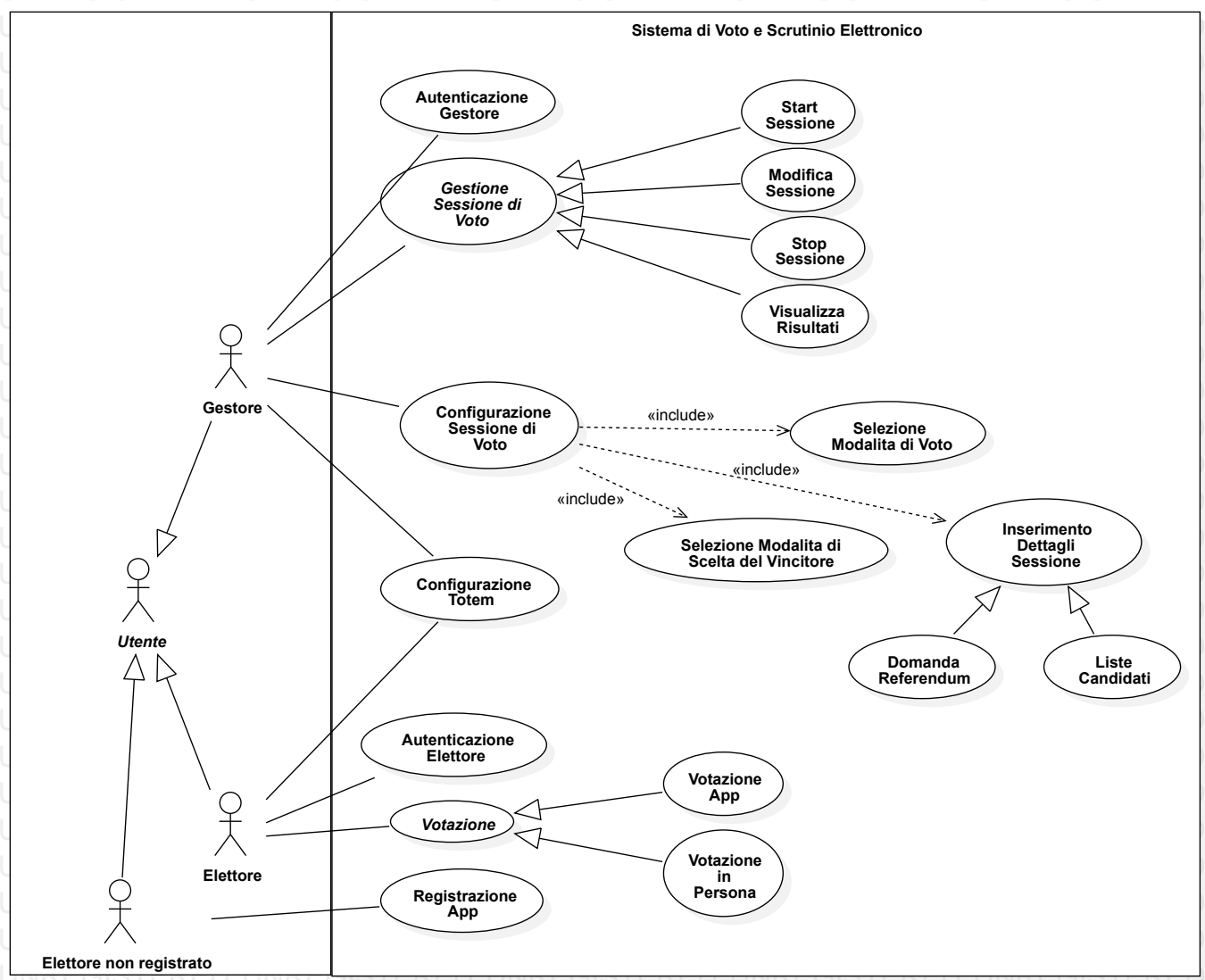
1. Dopo aver fatto la selezione per il voto, il sistema dovrebbe richiedere la conferma del voto prima di continuare.

1.3.2.4 Requisiti normativi

1. Ogni **elettore** può votare una volta sola in una sessione di voto.

2 Progettazione del sistema

2.1 Diagramma dei casi d'uso



Di seguito una breve descrizione testuale per ogni scenario:

Registrazione App: L'**elettore** non registrato può registrarsi nel sistema da applicazione per avere un'account con cui votare da **app**.

Requisito funzionale associato: Come **elettore**, voglio potermi registrare nel sistema.

Autenticazione: L'**utente** può autenticarsi nel sistema tramite applicazione usando le sue credenziali. L'**utente** può accedere al sistema tramite applicazione solo se già registrato con le credenziali inserite.

Requisito funzionale associato: L'**utente** si può autenticare nell'applicazione come **elettore** o **gestore del sistema**.

Votazione: L'**elettore** può votare sia da remoto, da un dispositivo elettronico, sia in persona, usando un totem.

Requisito funzionale associato: L'**elettore** deve potere esprimere il suo voto di **persona**, in un **seggio elettorale**, utilizzando un **totem**, o a **distanza**, utilizzando l'**applicazione** su un dispositivo elettronico, ma comunque non prima di aver svolto la fase d'identificazione e di verifica del diritto al voto.

Configurazione Sessione di Voto: Il **gestore di sistema** può configurare una sessione di voto, scegliendo delle preferenze.

Requisito funzionale associato: Come **gestore di sistema**, voglio poter configurare una sessione di voto, inserire le liste dei candidati, far partire la fase di scrutinio e visualizzare l'esito del voto.

Gestione Sessione di Voto: Il **gestore di sistema**, dopo essersi autenticato nel sistema come gestore, può gestire le sessioni esistenti.

Requisito funzionale associato: Come **gestore di sistema**, voglio poter configurare una sessione di voto, inserire le liste dei candidati, far partire la fase di scrutinio e visualizzare l'esito del voto.

Configurazione Totem: Il **gestore di sistema** può configurare una sessione di voto per un'**elettore** che vota in presenza su un totem.

Requisito funzionale associato: Come **gestore di sistema**, voglio poter configurare un totem per farlo usare da parte di un'elettore che vota **in persona**.

2.2 Descrizione degli scenari

Nome	Registrazione App
Scopo	Registrazione di un'elettore nel sistema attraverso l'applicazione.
Attore/i	Elettore non registrato.
Pre-condizioni	Elettore non registrato.
Trigger	Elettore che vuole fare la registrazione.
Descrizione sequenza eventi	<ol style="list-style-type: none">1. L'elettore preme il pulsante Registra ed entra sulla pagina di registrazione2. L'elettore inserisce le sue informazioni personali come: nome, cognome, eta, codice fiscale.3. L'elettore inserisce una password ripetuta 2 volte.4. L'elettore preme il pulsante Registra e gli compare un>alert che gli chiede se sia sicuro di voler creare l'account.5. L'elettore preme il pulsante OK.6. L'account viene creato e si ritorna alla pagina di autenticazione.
Alternativa/e	<ol style="list-style-type: none">4a. L'elettore viene avvisato che alcuni dati non sono stati inseriti o non sono validi.5a. L'elettore preme il pulsante Cancel e ritorna a modificare i dati inseriti.
Post-condizioni	L'utente e in possesso di un'account.

Nome	Autenticazione
Scopo	Entrare nel sistema in modo da poter partecipare a una sessione di voto.

Attore/i	Elettore, Gestore di Sistema
Pre-condizioni	Elettore registrato, Gestore esistente
Trigger	Elettore/Gestore vuole entrare sull'applicazione
Descrizione sequenza eventi	1. Utente inserisce le sue credenziali nella sua sezione specifica di Login. 2. Utente preme il pulsante Autentifica.
Alternativa/e	1a. Utente inserisce un codice fiscale non associato ad un account. 1b. Utente inserisce una password errata.
Post-condizioni	Utente autenticato nell'applicazione.

Nome	Votazione su App
Scopo	Esprimere il proprio voto sull'applicazione
Attore/i	Elettore
Pre-condizioni	Autenticazione Elettore su App
Trigger	L'elettore vuole esprimere il suo voto in una sessione di voto.
Descrizione sequenza eventi	1. L'elettore sceglie dalla pagina principale una sessione di voto disponibile e premendo Entra, entra nella sessione di voto. 2. L'elettore fa la sua scelta dalla scheda. 3. L'elettore preme il pulsante Avanti e gli compare un>alert che gli chiede se vuole confermare il voto. 4. Se l'elettore e sicuro, preme OK e gli compare un>alert che conferma la registrazione del voto. 5. L'elettore viene mandato alla pagina di login.
Alternativa/e	2a. L'elettore lascia la scheda bianca. 4a. Dopo aver visto che il voto e sbagliato, l'elettore preme Cancel.
Post-condizioni	L'elettore ha espresso il suo voto nella sessione in cui ha partecipato.

Nome	Votazione in Persona
------	-----------------------------

Scopo	Esprimere il proprio voto sul totem in persona.
Attore/i	Elettore
Pre-condizioni	
Trigger	L'elettore vuole esprimere il suo voto in una sessione di voto.
Descrizione sequenza eventi	<ol style="list-style-type: none"> 1. L'elettore indica al gestore presente la sessione in cui vuole votare e il gestore configura un totem con quella sessione di voto preparata per l'elettore. 2. L'elettore fa la sua scelta nella sessione di voto. 3. L'elettore preme il pulsante Avanti e gli compare un>alert che gli chiede se vuole confermare il voto. 4. Se l'elettore e sicuro, preme OK e gli compare un>alert che conferma la registrazione del voto.
Alternativa/e	<p>2a. L'elettore lascia la scheda bianca.</p> <p>4a. Dopo aver visto che il voto e sbagliato, l'elettore preme Cancel.</p>
Post-condizioni	L'elettore ha espresso il suo voto nella sessione in cui ha partecipato.

Nome	Configurazione Sessione di Voto
Scopo	Creare una sessione di voto con delle proprieta scelte.
Attore/i	Gestore di Sistema
Pre-condizioni	Gestore autenticato nell'applicazione.
Trigger	Gestore decide di creare una sessione di voto.
Descrizione sequenza eventi	<ol style="list-style-type: none"> 1. Il gestore preme il pulsante "+" nella lista di sessioni. 2. Il gestore sceglie la modalita di voto. 3. Il gestore sceglie la modalita di scelta del vincitore. 4. Il gestore inserisce le liste dei candidati/la domanda del referendum. 5. Il gestore preme il pulsante Avanti e gli compare un>alert che gli chiede se vuole salvare la sessione. 6. Il gestore preme il pulsante OK.
Alternativa/e	6a. Il gestore preme il pulsante Cancel.

Post-condizioni	Una nuova sessione di voto viene creata.
-----------------	--

Nome	Gestione Sessione di Voto
Scopo	Avviare, fermare, modificare o visualizzare l'esito di una sessione di voto.
Attore/i	Gestore di Sistema
Pre-condizioni	Gestore autenticato nell'applicazione.
Trigger	Gestore decide di avviare/fermare una sessione di voto o di visualizzare l'esito di una sessione di voto finita.
Descrizione sequenza eventi	1. Il gestore preme il pulsante Avvia su una sessione di voto.
Alternativa/e	1a. Il gestore preme il pulsante Stop su una sessione di voto. 1b. Il gestore preme il pulsante Risultati su una sessione di voto.
Post-condizioni	La sessione di voto viene avviata, fermata, modificata o l'esito viene visualizzato.

Nome	Configura Totem
Scopo	Configurare un Totem in modo che un'elettore puo votare in persona.
Attore/i	Gestore di Sistema, Elettore
Pre-condizioni	Gestore autenticato nell'applicazione, Elettore presente nel seggio elettorale.
Trigger	Elettore si presenta nel seggio elettorale con l'intenzione di votare.
Descrizione sequenza eventi	1. Il gestore preme il pulsante Totem su una sessione di voto scelta dall'elettore e passa alla schermata di Configurazione Totem. 2. Il gestore sceglie un Totem disponibile dalla lista di Totem ed inserisce il Codice Fiscale che viene indicato dall'elettore e preme Avanti. 3. L'elettore puo votare ora sul totem che e stato attivato.
Alternativa/e	2a. Non ci sono Totem disponibili e il gestore deve aspettare che

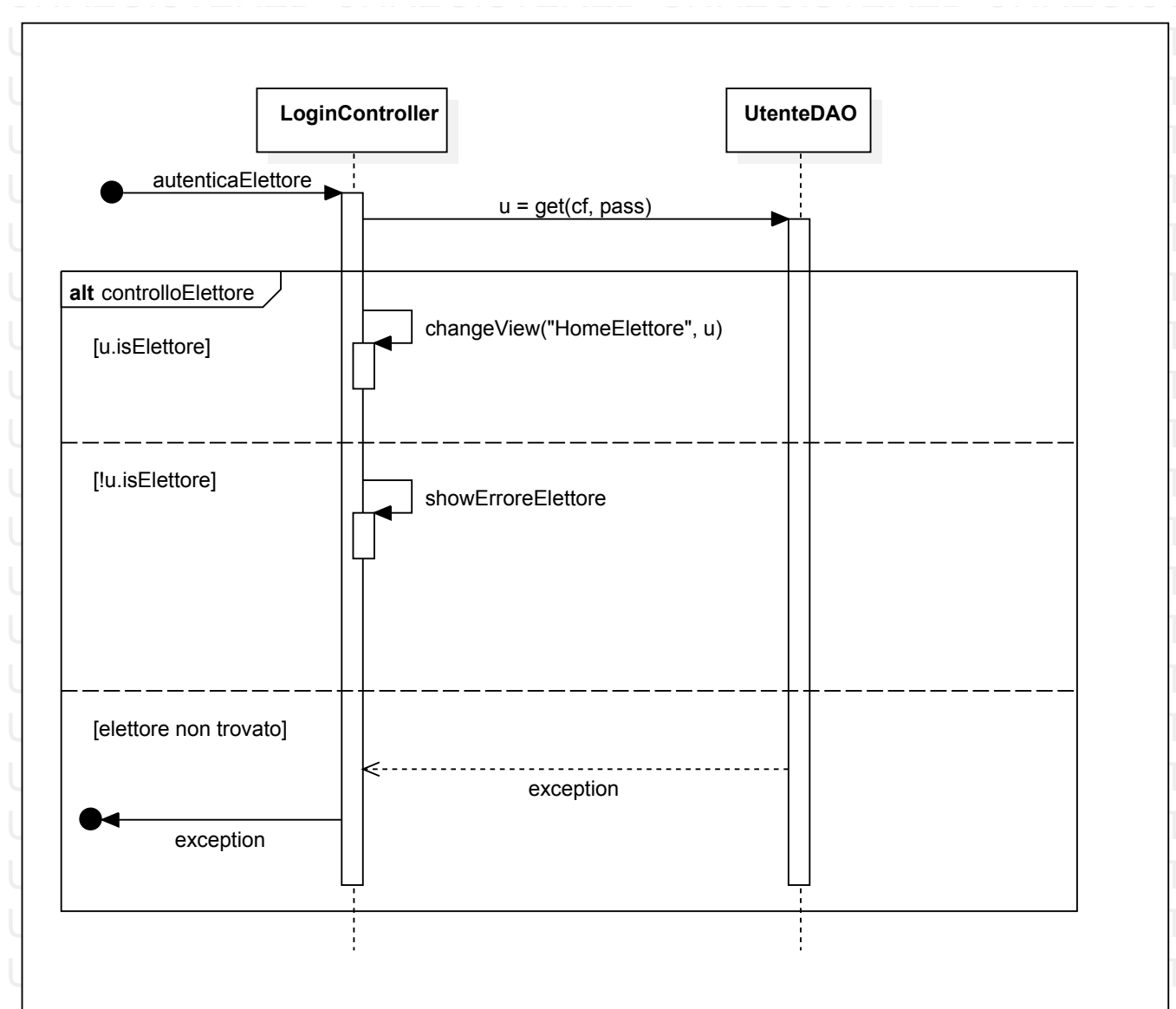
	uno si liberi, verificando se ci sono nuovi liberi tramite il tasto Refresh.
Post-condizioni	Il totem indicato puo essere usato ora, da parte dell'elettore, per fare il suo voto nella sessione scelta.

2.3 Diagramma delle classi

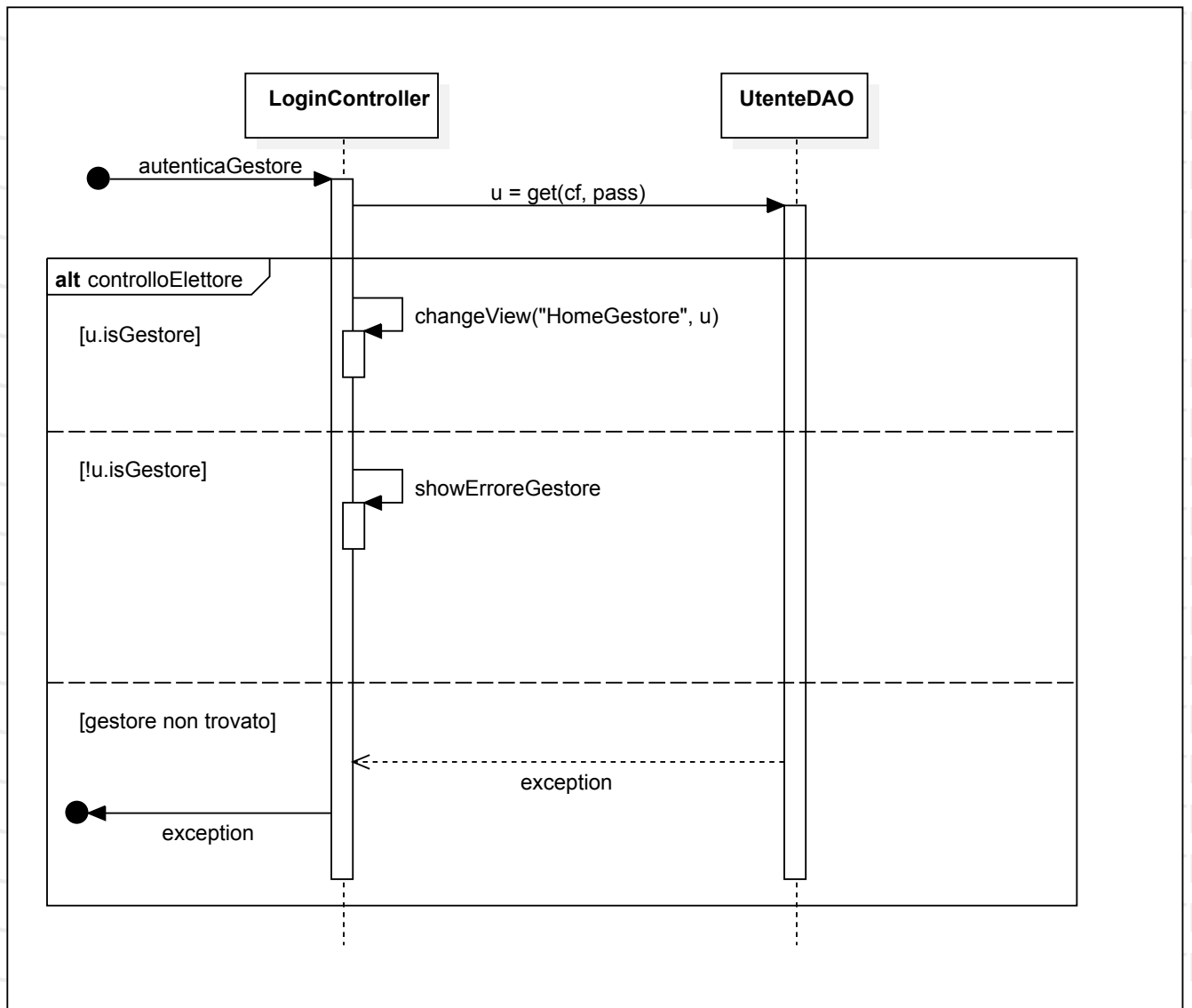
E stato deciso di mostrare solo il modello di programma del diagramma delle classi. (3.1).

2.4 Diagrammi di sequenza

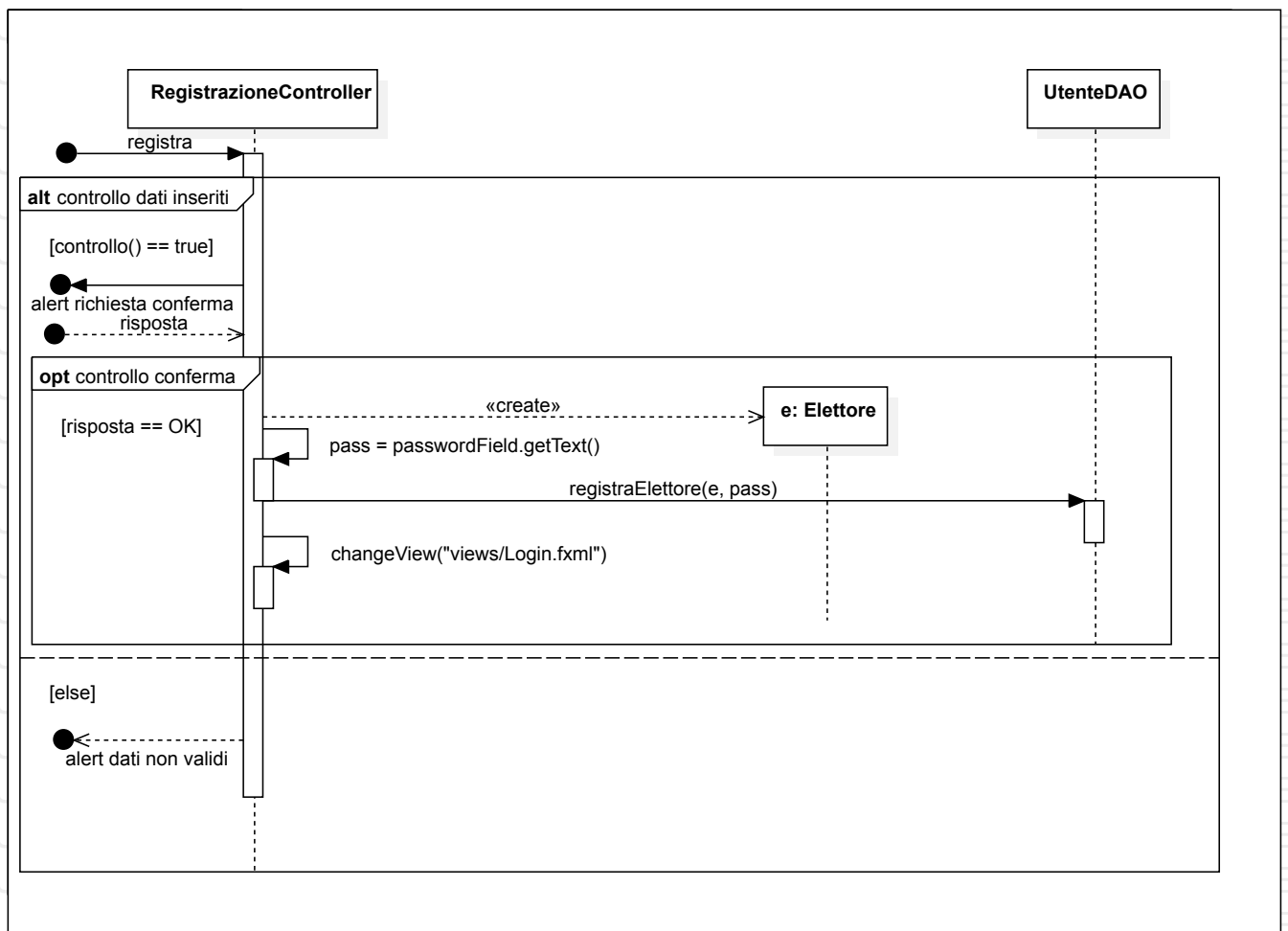
2.4.1 Autenticazione Elettore



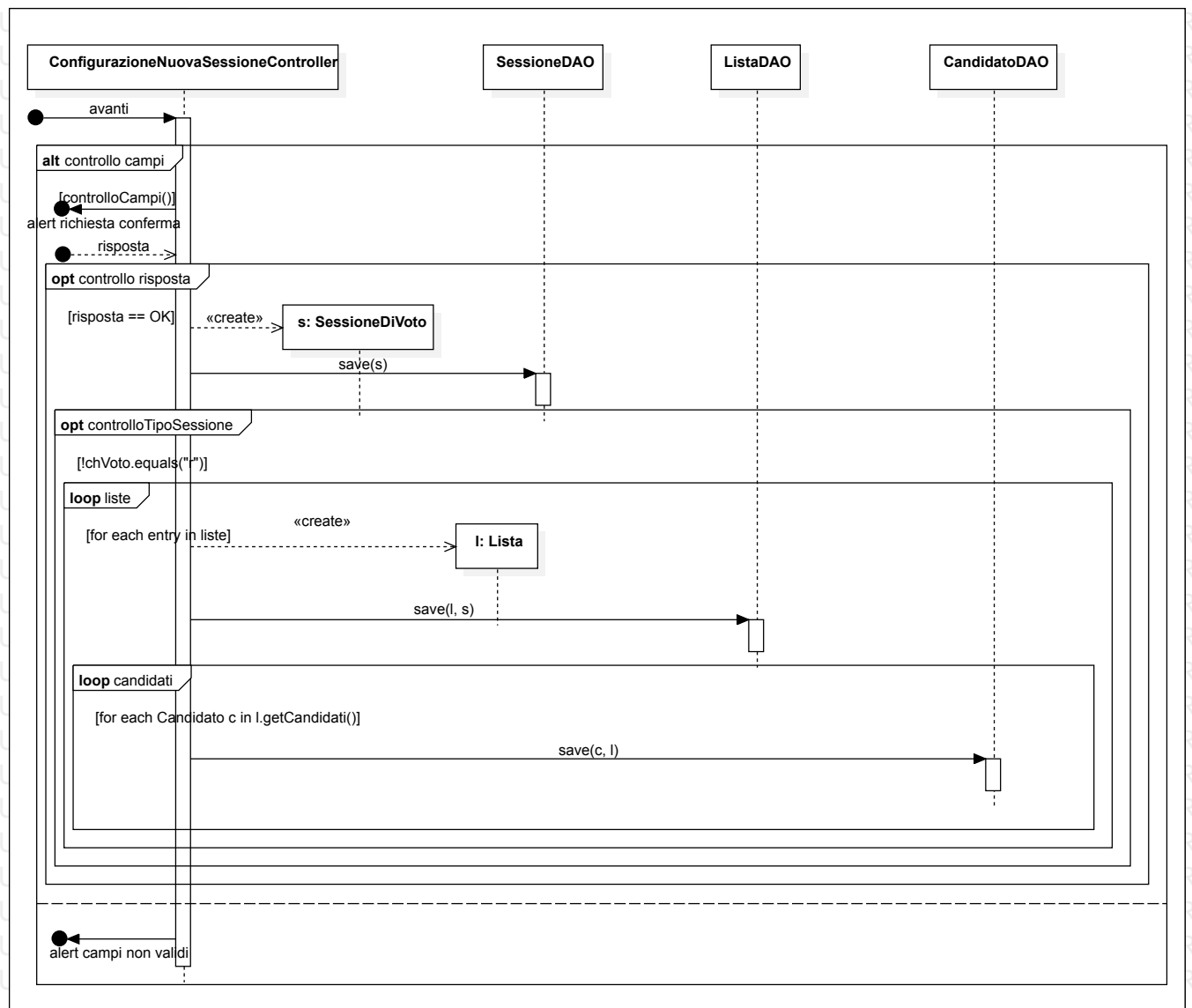
2.4.2 Autenticazione Gestore



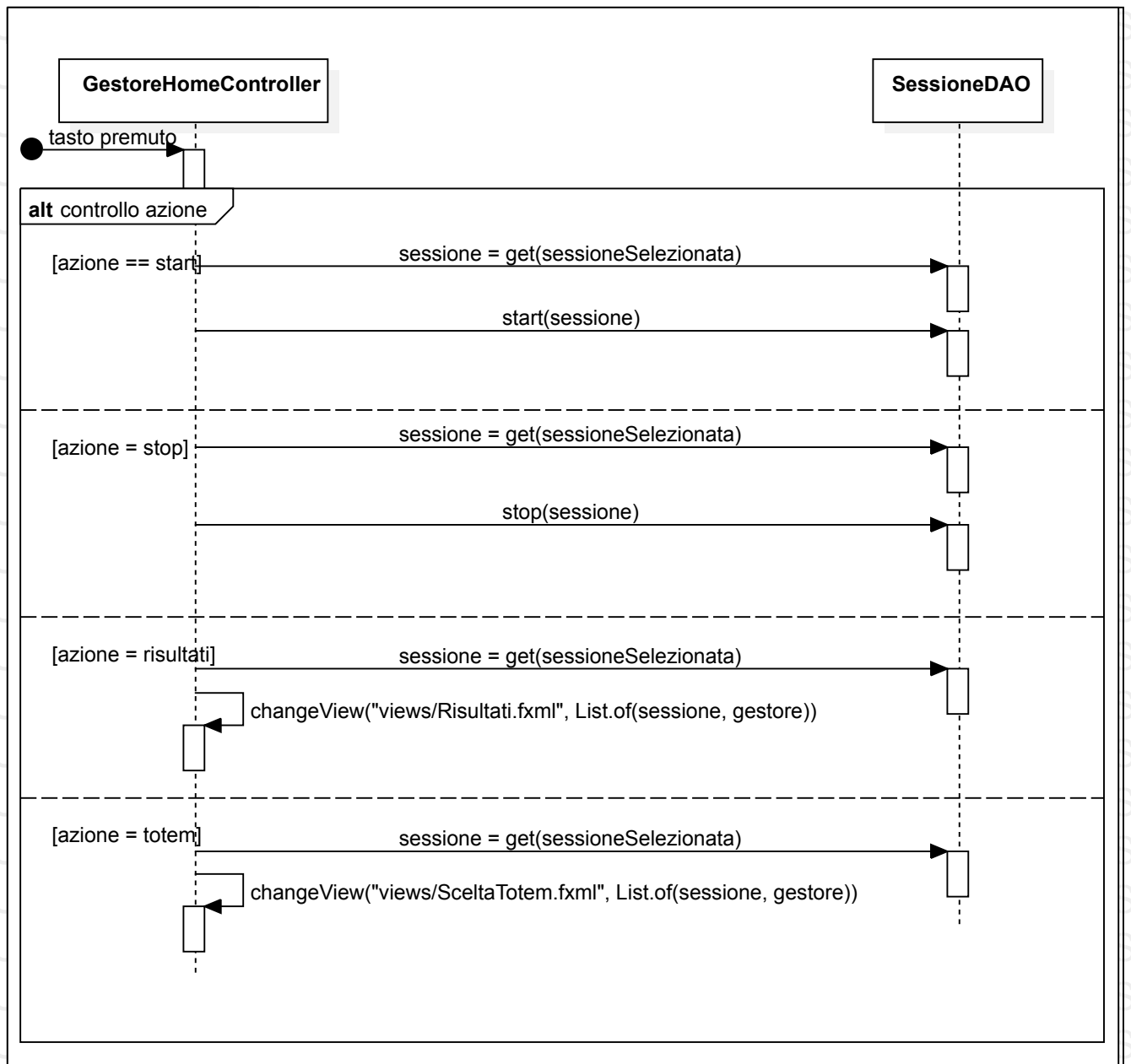
2.4.3 Registrazione Elettore



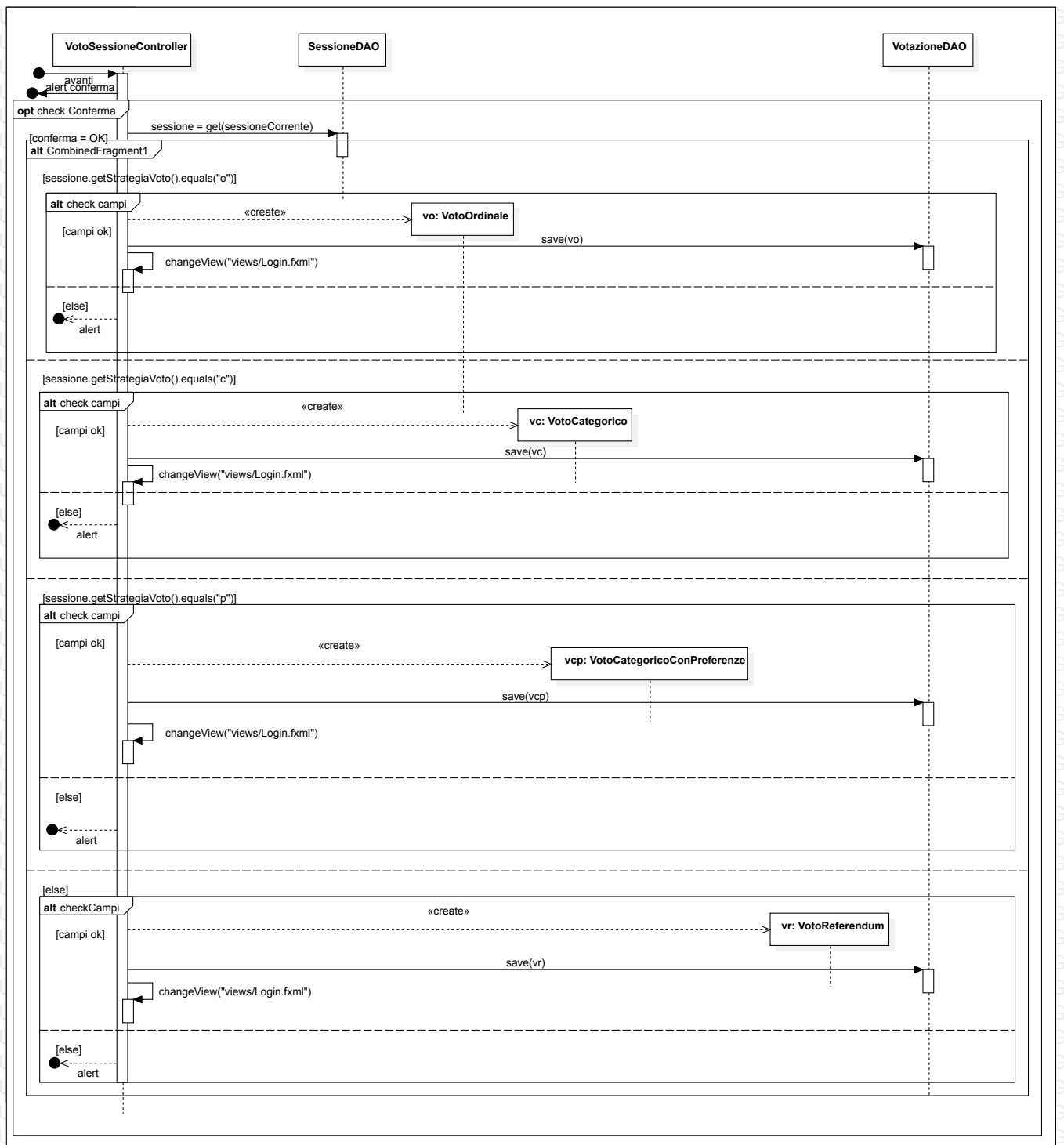
2.4.4 Configurazione Sessione di Voto



2.4.5 Gestione Sessione di Voto

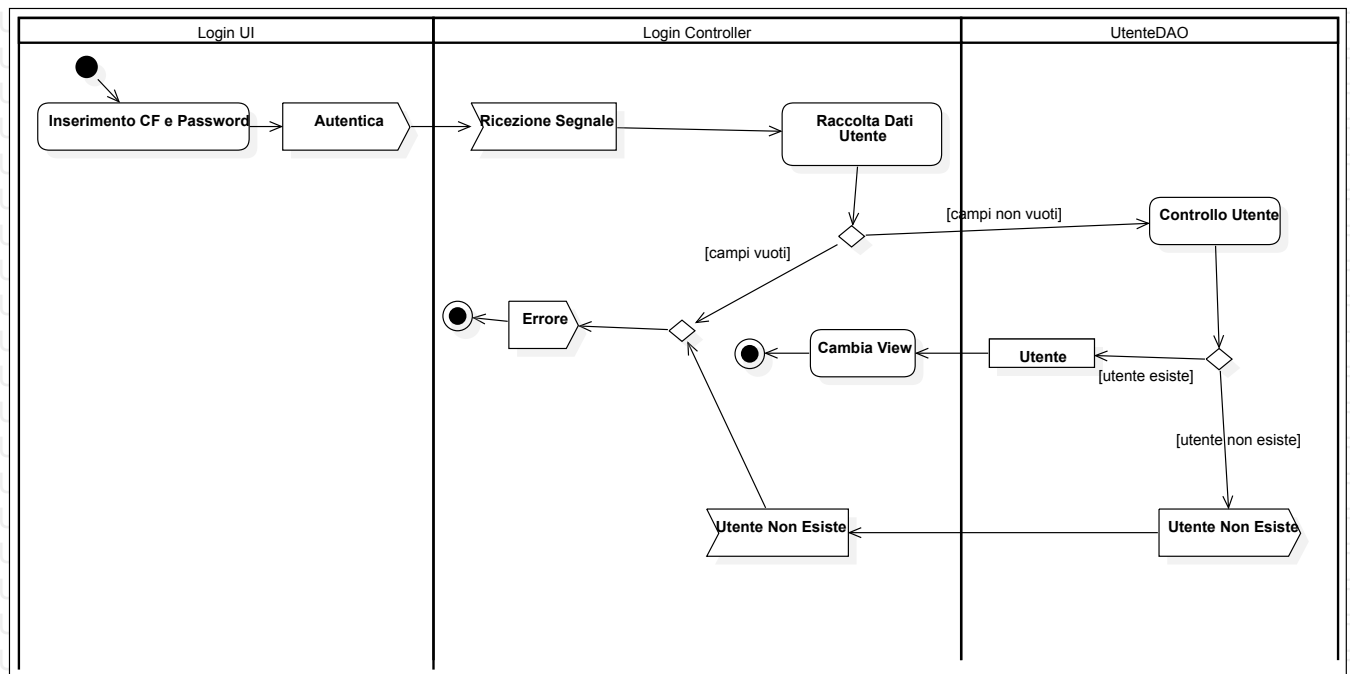


2.4.6 Votazione



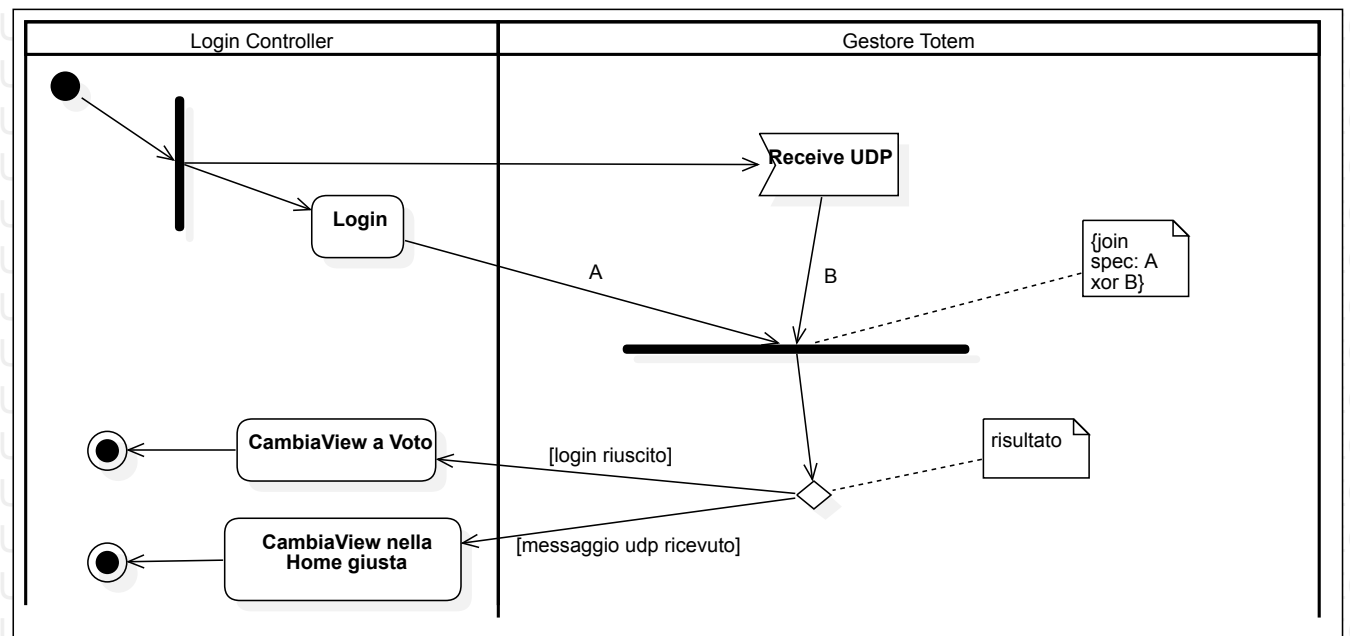
2.5 Diagramma delle attività

2.5.1 Login



2.5.2 Totem

Quando siamo sulla pagina di Login, l'applicazione ascolta con UDP su un'altro thread, in modo da non bloccare azioni che potrebbero essere fatte sulla GUI.

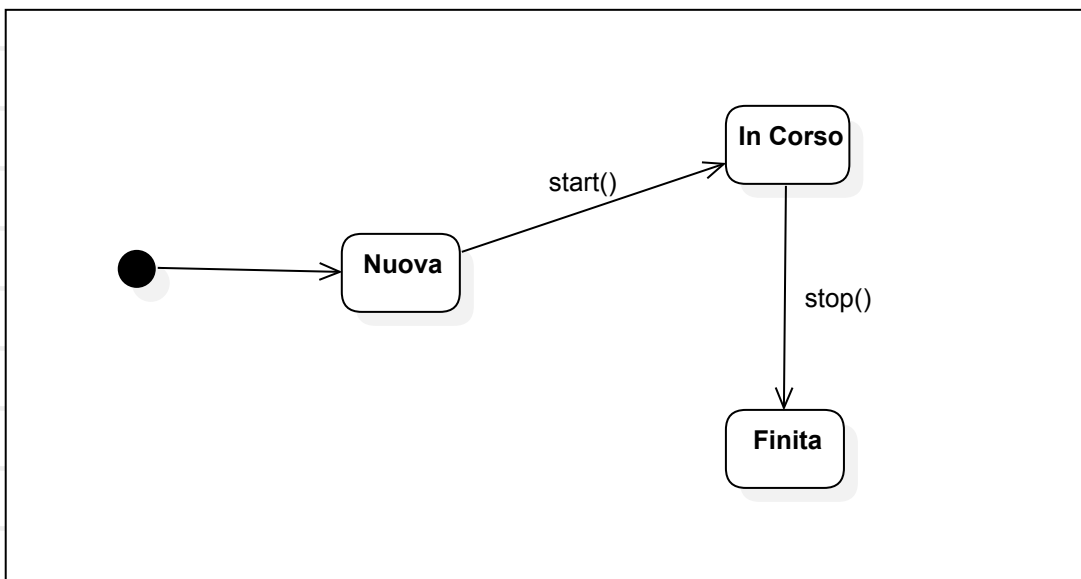


2.5.3 Voto (Categorico, Categorico con Preferenze, Ordinale)

2.6 Macchine di stato

2.6.1 Sessione di Voto

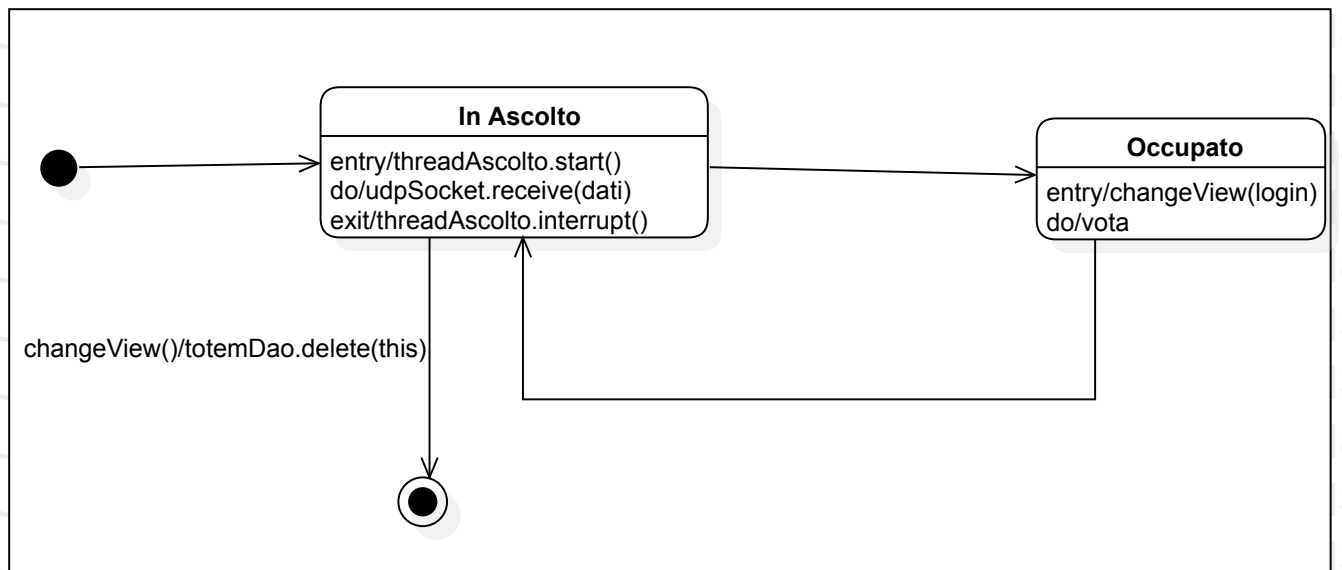
L'aggiornamento del stato di un'oggetto non avviene necessariamente attraverso metodi d'istanza, ma attraverso l'aggiornamento da parte della base dati. Spiegato in modo più concreto: quando si preme Start o Stop su una sessione, questa cambia stato all'interno del database, ma per cambiare anche il stato dell'oggetto all'interno del programma, bisogna fare un nuovo get, per avere la nuova versione di questa sessione. Oltre al stato non cambia nulla, ma nel caso in cui l'applicazione fosse usata da più di un gestore allo stesso tempo, si preferisce che le Sessioni di ogni gestore siano sincronizzate tra di loro, quindi si fa più affidamento sui dati nella base dati che sugli oggetti che abbiamo localmente.



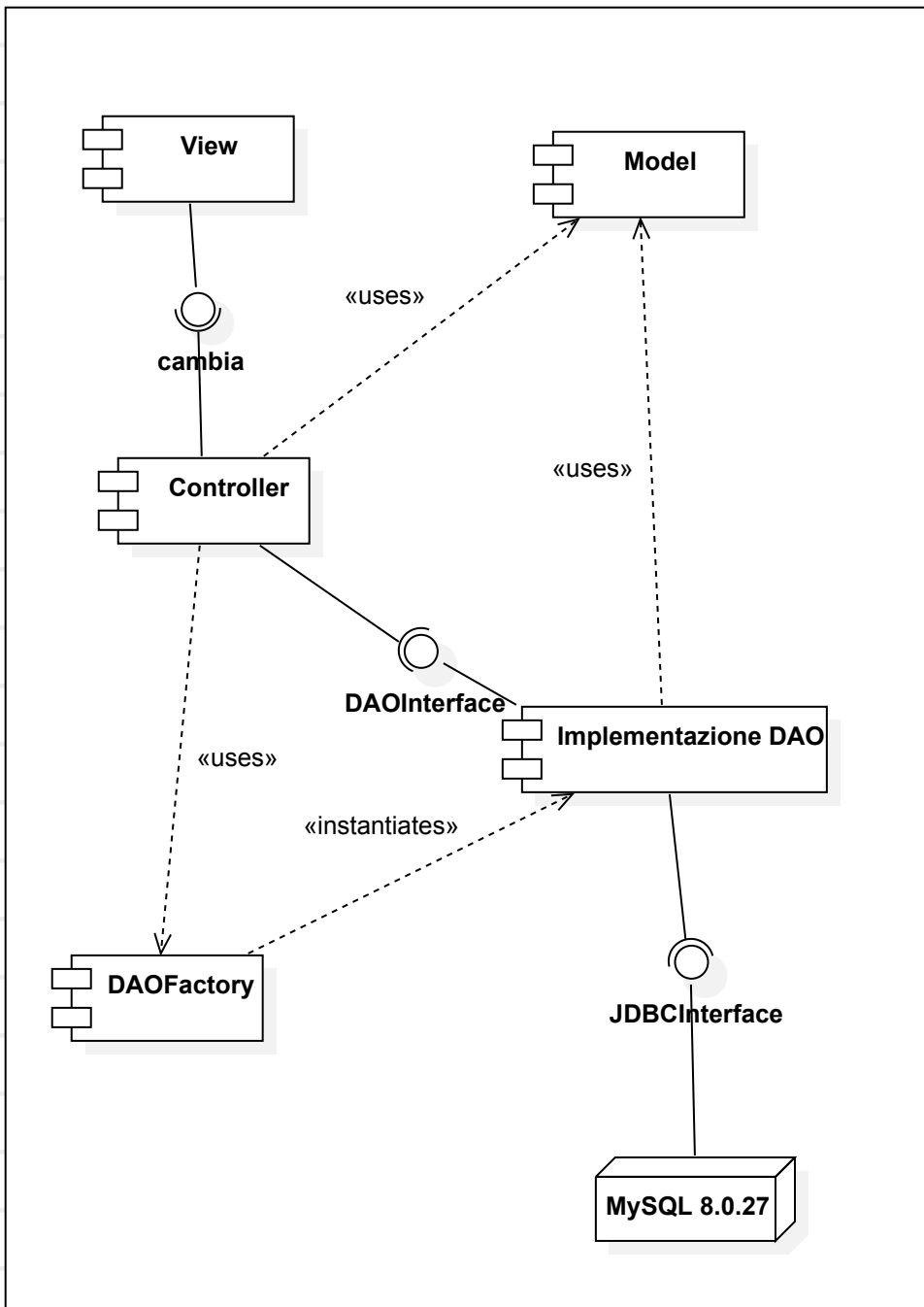
2.6.2 Totem di Voto

Il totem sta In Ascolto fino a quando una delle due cose succede:

- si cambia schermata e quindi non ci si trova più nella schermata di Login
- si fa una transizione di completamento, e quindi la socket ha ricevuto un segnale, e si passa alla schermata di Voto



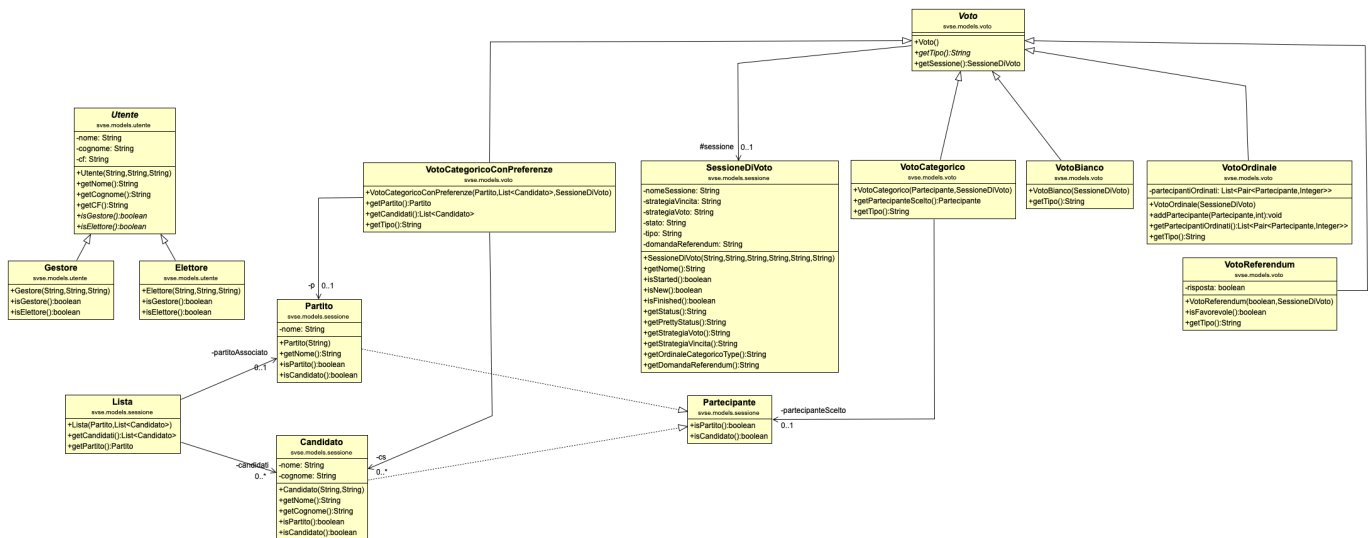
2.7 Diagramma dei componenti



3 Implementazione del sistema

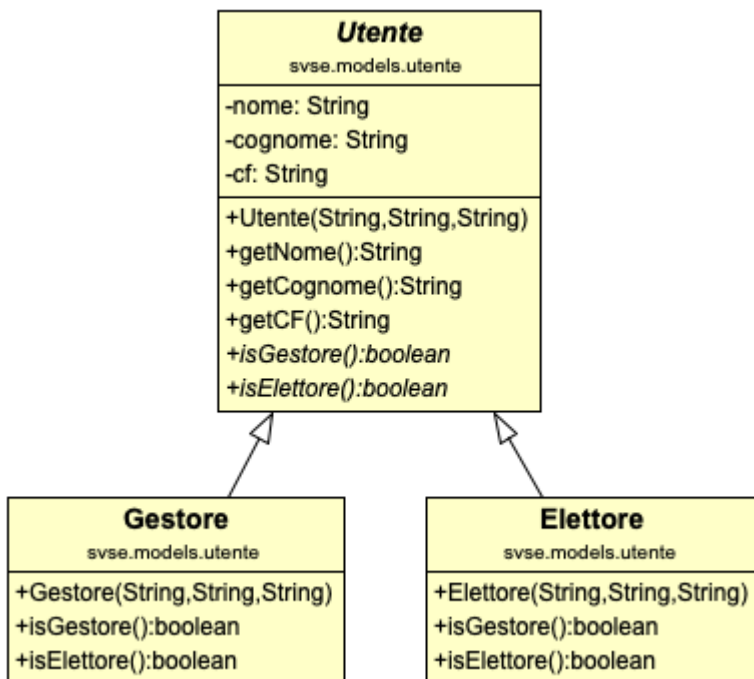
3.1 Diagramma delle classi

3.1.1 Modelli



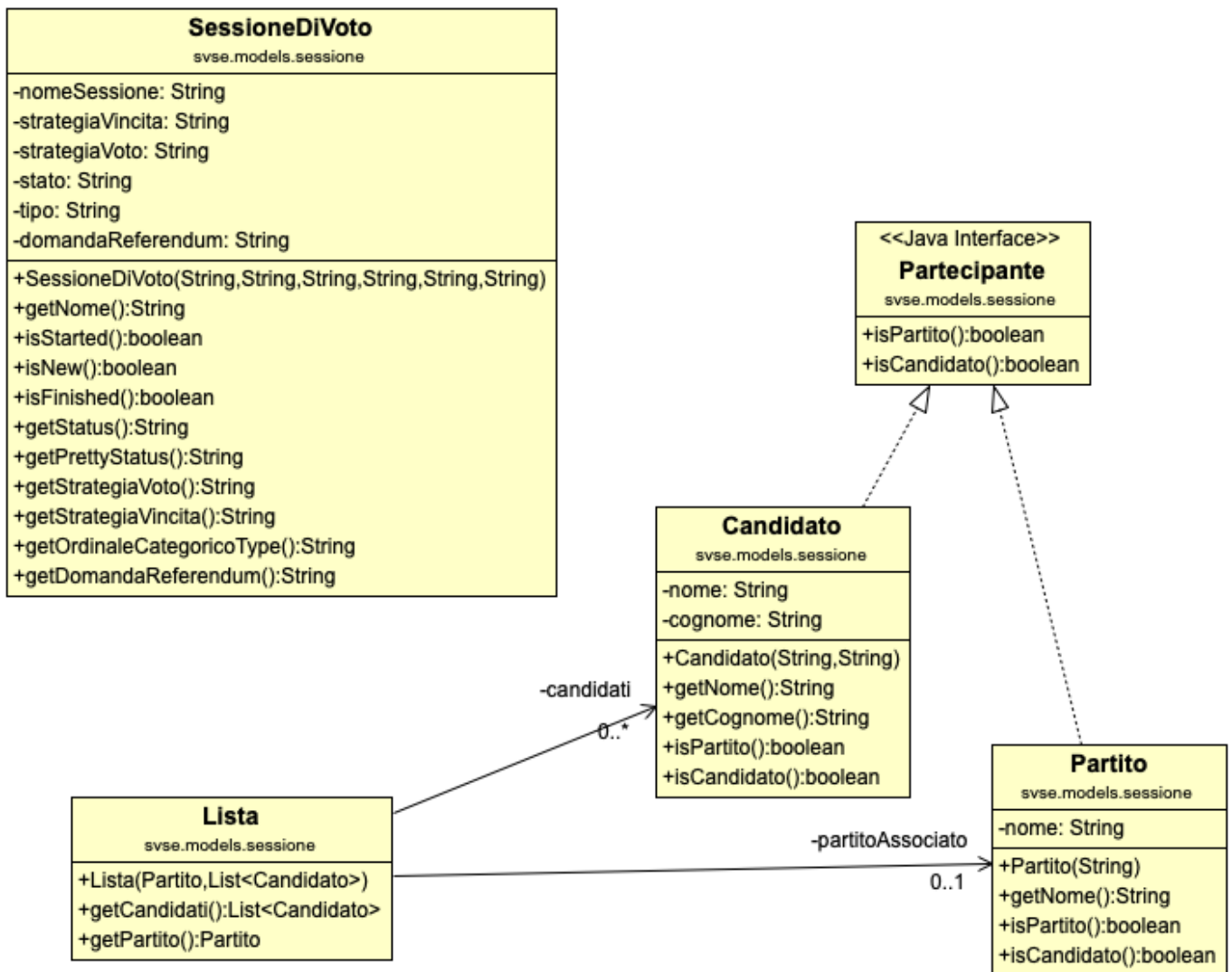
3.1.1.1 Utente

È stato scelto di implementare l'Elettore e il Gestore in questo modo per facilitare l'associazione della classe con la tabella del database.

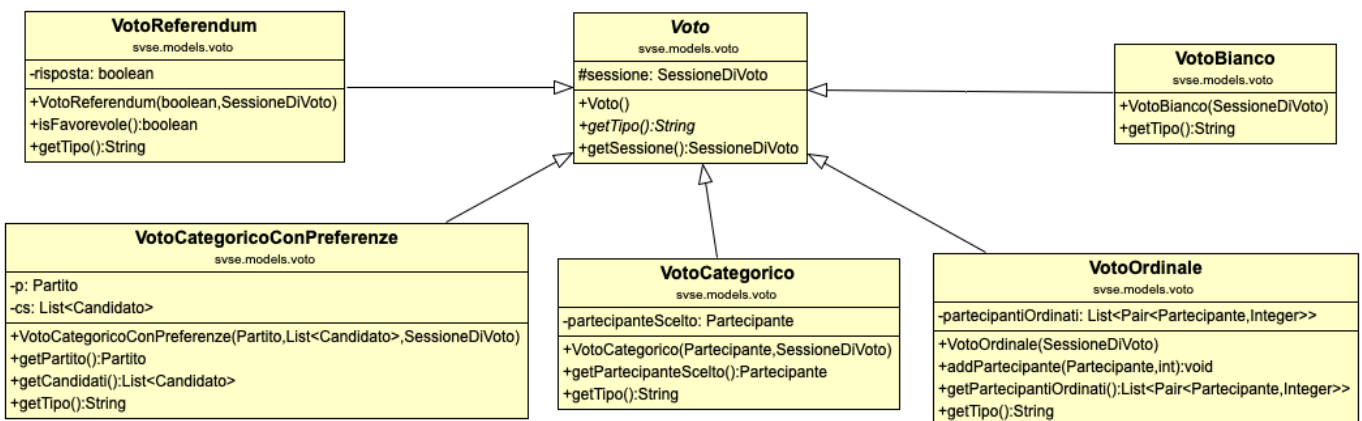


3.1.1.2 Sessione

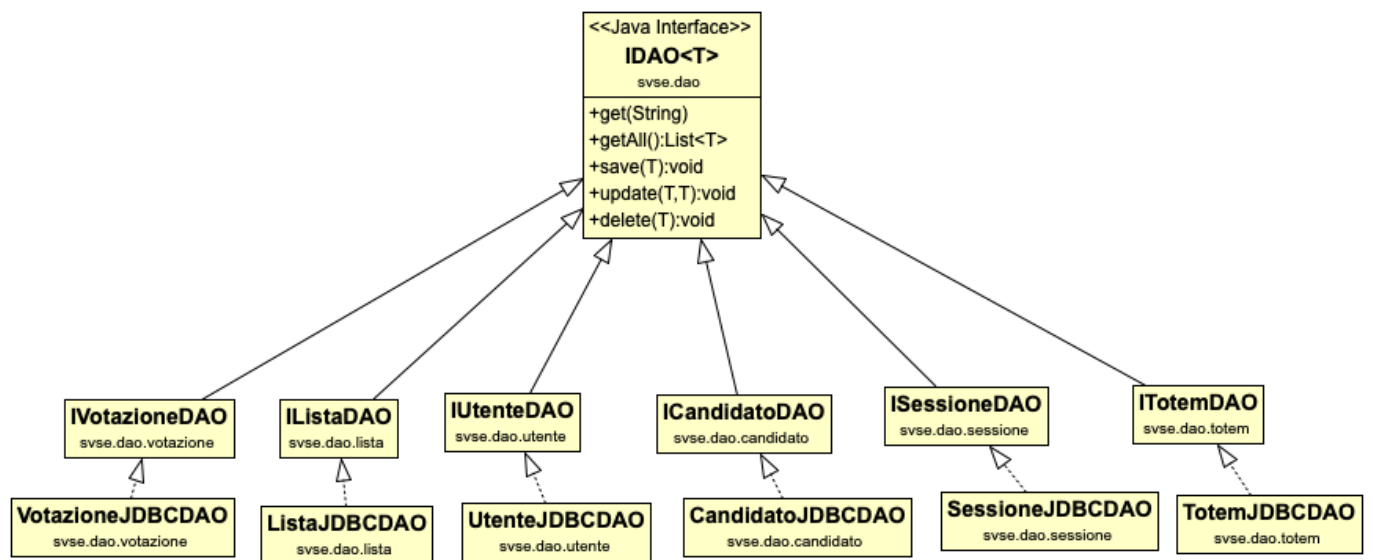
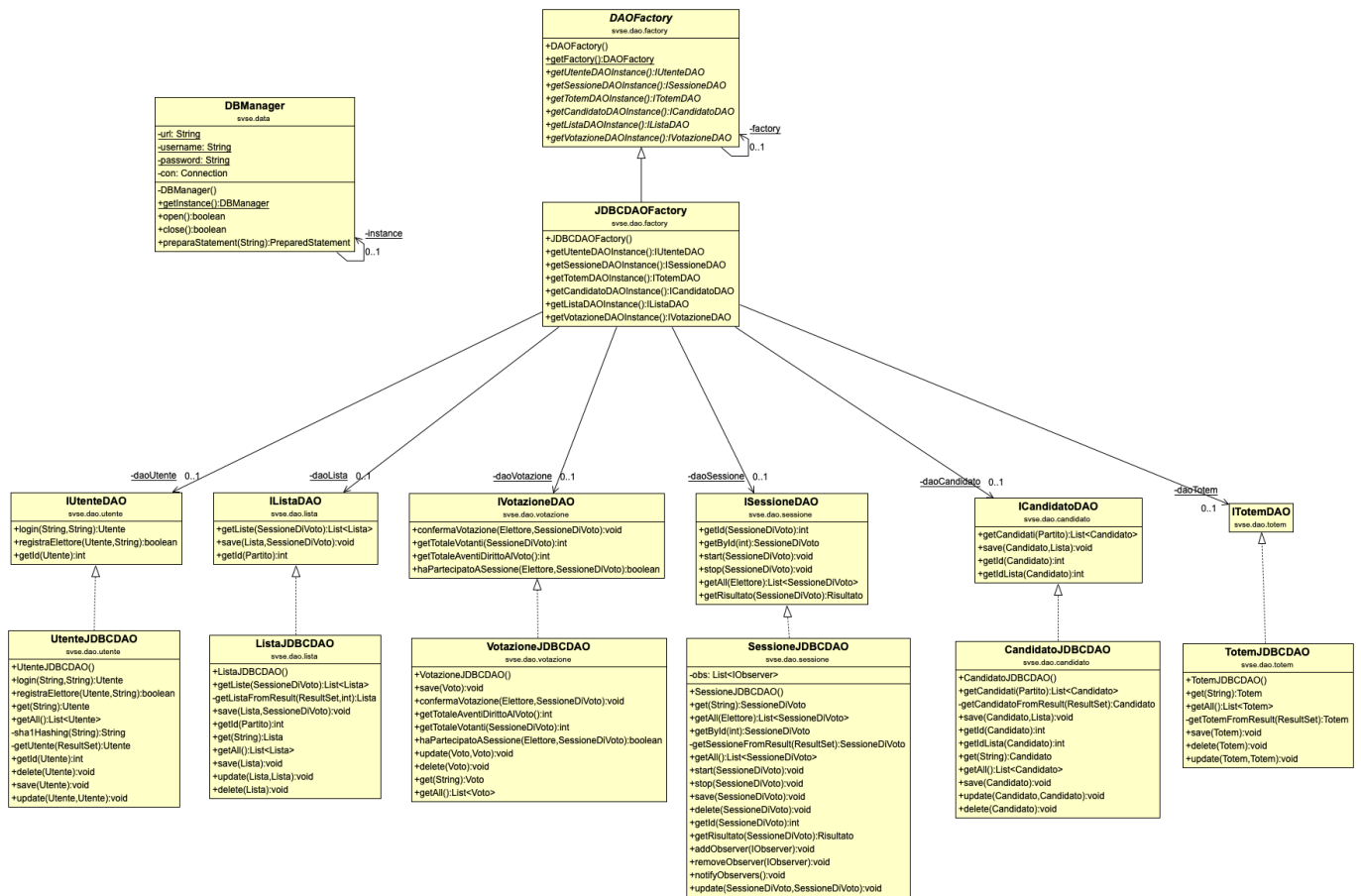
È stato scelto di considerare il Candidato e il Partito entrambi come Partecipante per facilitare le modalità di Voto dove si possono scegliere o Candidati o Partiti. L'uso di questa generalizzazione si può vedere nel diagramma 3.1.1.3, nel VotoOrdinale e VotoCategorico.



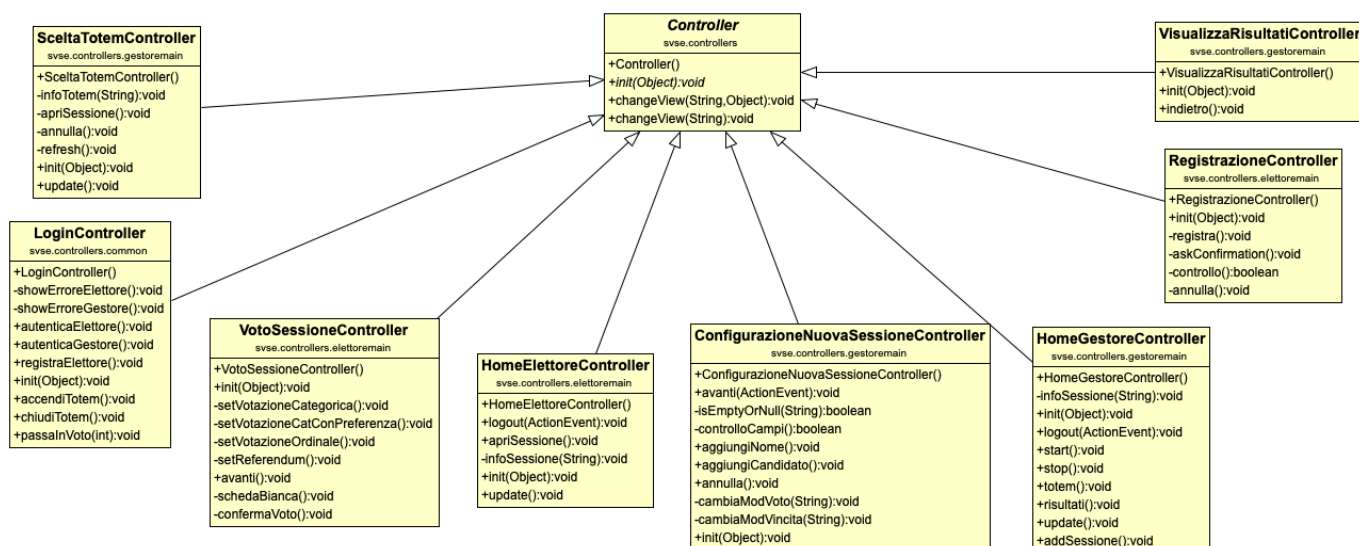
3.1.1.3 Voto



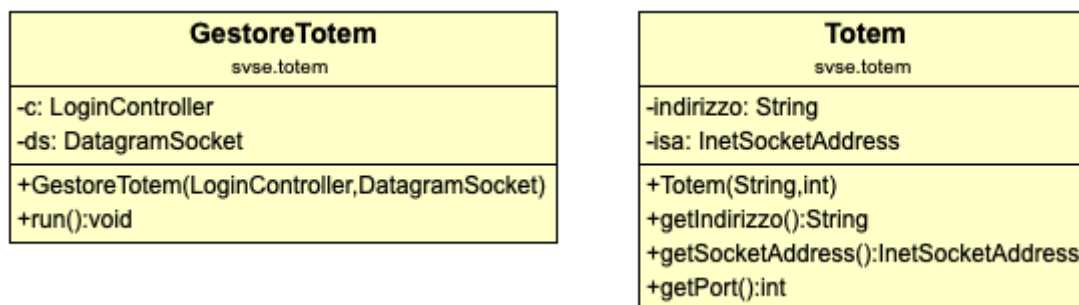
3.1.2 DAO



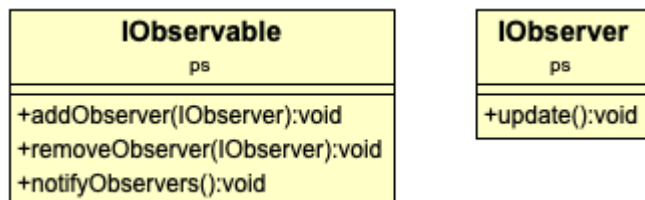
3.1.3 Controller



3.1.4 Totem



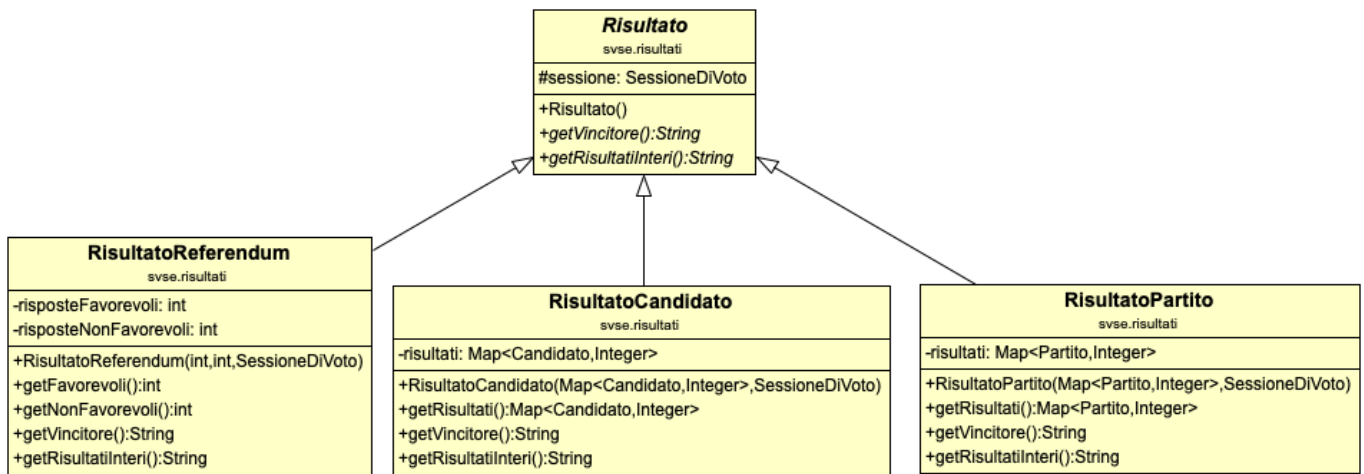
3.1.5 Publish Subscribe



3.1.6 Risultati

Questa gerarchia viene usata per memorizzare i risultati di una sessione. Il metodo **getRisultato** di SessioneDAO restituisce un'oggetto di questo tipo che può essere usato ulteriormente per mostrare il risultato di una sessione.

Sono stati implementati diversi tipi di risultato, che possono essere associati a Votazioni di Candidati, Votazioni di Partiti e Referendum.



3.1.7 Design Pattern Utilizzati

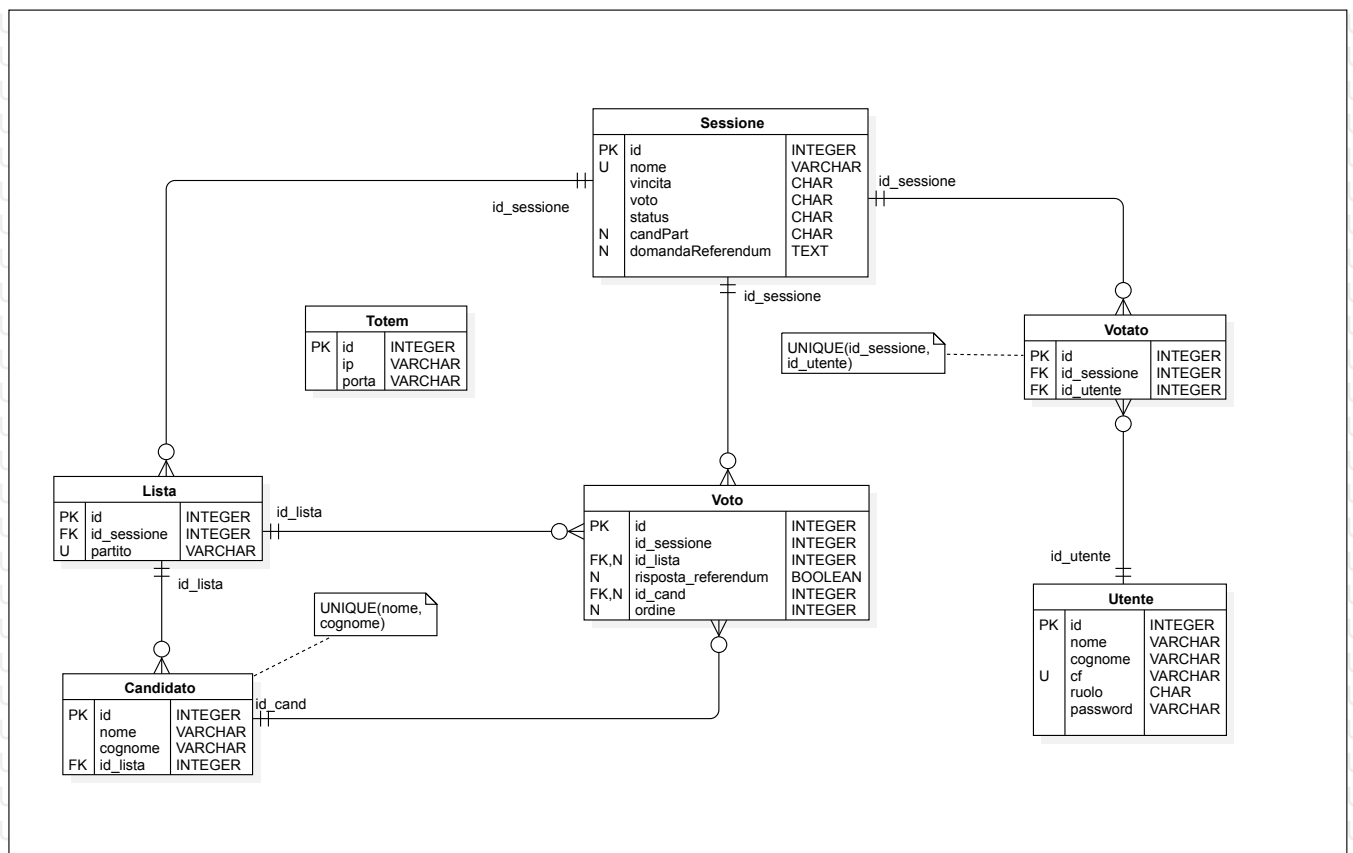
- **MVC:** L'applicazione è stata divisa in modelli, controllers e view. Questo si può vedere dalla struttura delle cartelle: **svse.models**, **svse.controllers** e **svse.views**.
- **DAO:** Il Data Access Object astrae il database e l'implementazione in questo progetto si può trovare in **svse.dao**. I DAO sono stati implementati assieme al design pattern Abstract Factory che facilita l'aggiunta ulteriore di **altre implementazioni** per il Database e l'accesso ai DAO **senza conoscere l'implementazione** concreta che si sta usando. Il diagramma delle classi 3.1.2 mostra in modo chiaro l'utilizzo di questo pattern.
- **Singleton:** Questo pattern viene usato per esempio in **DBManager**, che viene usata come un'intermediario tra la base dati e il resto dell'applicazione. Per tutta l'esecuzione, non abbiamo bisogno di più di una connessione con la base dati, e il DBManager implementato come Singleton aiuta a fare proprio questo. Il diagramma delle classi 3.1.2 mostra il DBManager con un costruttore privato, un metodo factory statico getInstance e un'istanza privata di se stesso.
- **Observer:** JavaFX usa questo pattern per la gestione degli eventi provenienti dall'interazione con la UI. Un'altro posto dove uso il pattern Observer/Publish-Subscribe e la DAO della Sessione, che, quando vengono fatte modifiche alle sessioni (start/stop di una sessione, aggiunta di una nuova sessione), si annunciano tutti i controller che sono attached a questo Observable di questo cambiamento, in modo che possano aggiornarsi col nuovo stato. Il diagramma delle classi 3.1.5 mostra le interfacce che ho usato per fare questo design pattern.

- **Abstract Factory:** Questo pattern viene usato per facilitare l'aggiunta di nuove implementazioni per le DAO e per poter usare in modo facilitato le DAO all'interno del codice, nascondendo in questo modo l'implementazione concreta della DAO che stiamo usando. In **svse.dao** si trovano le gerarchie di tutte le famiglie di oggetti (famiglie di DAO), che nel mio caso sono correlate dal tipo di Database usato. Il diagramma delle classi 3.1.2 mostra la struttura tipica di un Abstract Factory Pattern, dove abbiamo la Factory e la ConcreteFactory, insieme alle famiglie di Products coi loro ConcreteProducts.

3.2 Gestione dei dati persistenti

Per la gestione dei dati persistenti è stato usato un DB MySQL (versione 8.0.27). Se si volesse usare un'altro DB, dovrebbe essere aggiunta una nuova implementazione per ogni interfaccia DAO (le interfacce che estendono IDAO) e dovrebbe essere modificata la DAOFactory in modo che si occupi anche della creazione di questa nuova implementazione.

3.2.1 Struttura Database



3.2.2 Descrizione Database

Totem - tabella contenente le informazioni necessarie per configurare un Totem nel caso di una votazione in persona.

Sessione - tabella contenente le informazioni di base che riguardano una sessione di voto, come il **nome**, la **modalita di vincita**, la **modalita di voto**, la **domanda del referendum** (nel caso abbia la modalita di voto come referendum) e il **status** (stato in cui si puo trovare una sessione. Si veda 2.6.1).

Lista - tabella contenente le informazioni che riguardano una lista di una sessione di voto. Una sessione di voto può avere piu liste. Viene rappresentata da un partito.

Candidato - tabella contenente le informazioni che riguardano la composizione di una lista candidati. Una lista puo avere piu candidati.

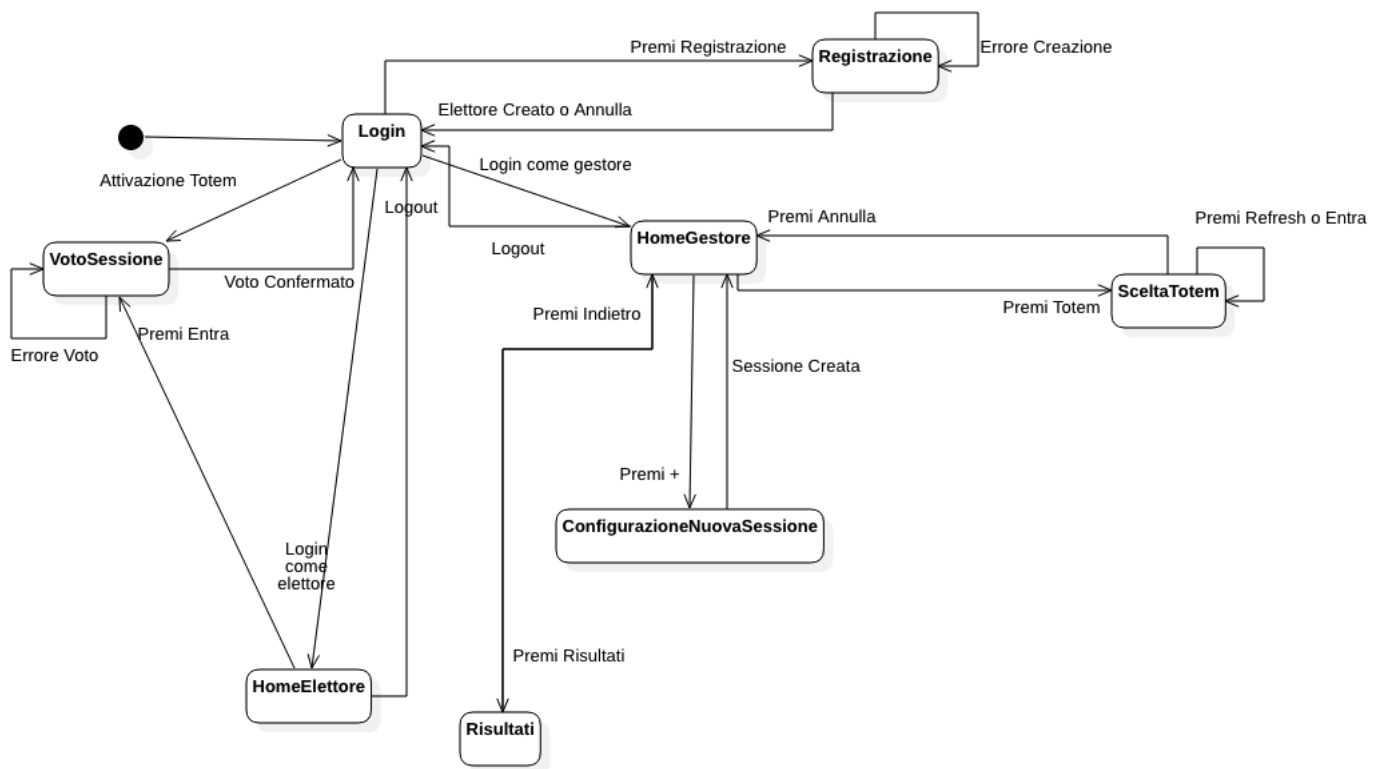
Utente - tabella contenente le informazioni che riguardano gli utenti che usano l'applicazione. Si memorizzano informazioni come: **ruolo**, **nome**, **cognome**, **codice fiscale** e **password**.

Votato - tabella contenente gli elettori che hanno votato insieme alle sessioni per cui hanno votato, in modo da capire che sessioni mostrare all'elettore e impedire che un'elettore voti per più di una volta nella stessa sessione. Un'elettore puo partecipare a piu sessioni di voto, ma non più di una volta per ognuna.

Voto - tabella contenente le informazioni riguardanti i voti. La tabella può contenere tutti i diversi tipi di voti siccome ha i campi neccessari per ognuno di essi. Una sessione può avere più voti. È stato scelto questo approccio per evitare la creazione di tabelle diverse per ogni tipo di voto.

3.3 Descrizione dell'interfaccia grafica

Qui abbiamo una Navigation Map per chiarire il modo in cui l'applicazione passa da una schermata all'altra:



3.3.1 Login

La schermata di Login presenta 2 sezioni separate per l'elettore e il gestore. Entrambe le sezioni contengono i stessi campi: Codice Fiscale + Password.

La sezione dell'elettore presenta 2 tasti: uno che permette di entrare nell'applicazione e l'altro che permette di registrarsi nel caso non si abbia un'account.

La sezione del gestore presenta un solo tasto che permette al gestore di entrare nell'applicazione.

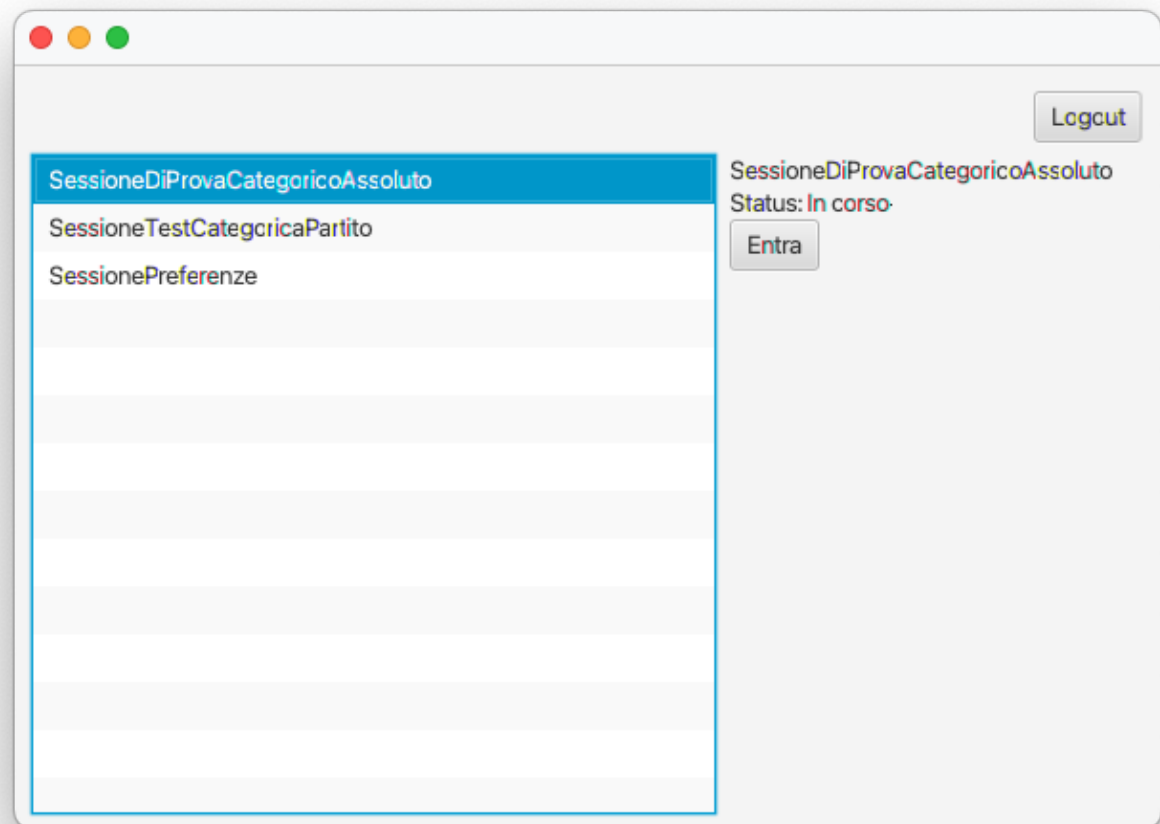
Se si prova a fare login con delle credenziali non esistenti si riceverà un messaggio d'errore.

The image shows a web application window with a light gray background and a standard macOS-style title bar with red, yellow, and green window control buttons. The window is divided into two main sections by a horizontal line. The top section is titled 'Elettore' in a small, dark font. It contains two input fields: 'Codice Fiscale' and 'Password', both with light gray placeholder text. Below these fields are two buttons: 'Autenticazione' (highlighted with a blue border) and 'Registrazione'. The bottom section is titled 'Gestore' in a small, dark font. It contains two input fields: 'Nome Utente' and 'Password', both with light gray placeholder text. Below these fields is a single button labeled 'Autenticazione'.

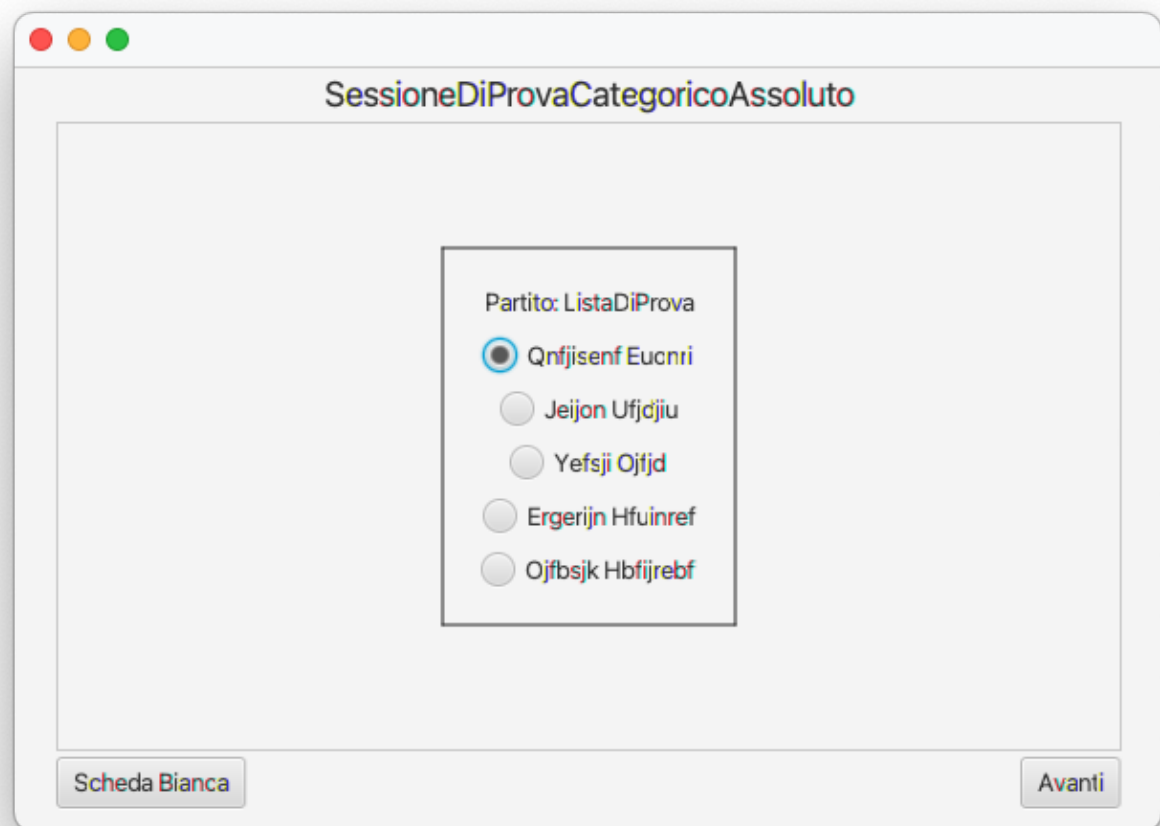
3.3.2 Home Elettore

La schermata principale dell'elettore presenta una sezione con le sessioni di voto disponibili per l'elettore (sessioni aperte e in cui non ha ancora votato) e una sezione con le informazioni riguardanti la sessione di voto selezionata. Nella sezione informazioni è presente un tasto Entra che permette all'elettore di partecipare alla sessione e di passare alla schermata di voto.

C'è presente anche un tasto di Logout che permette all'elettore di andare alla schermata di Login.



3.3.3 Sessione con Voto Categorico su Candidati



3.3.4 Sessione con Voto Categorico su Partiti

SessioneCategoricaPartito

Partito: Partito Prova 1	Partito: Partito Prova 2
<input type="radio"/> Candidato Prova Uno	<input checked="" type="radio"/> Candidato Prova 2 Primo
Candidato Prova Secondo	Candidato Prova 2 Secondo

Scheda Bianca

Avanti

3.3.5 Sessione con Voto Ordinale su Candidati

SessioneOrdinaleCandidato

Partito: Partito Prova 1

Candidato Prova Uno

Candidato Prova Secondo

Partito: Partito Prova 2

Candidato Prova 2 Primo

Candidato Prova 2 Secondo

Scheda Bianca Avanti

3.3.6 Sessione con Voto Ordinale su Partiti

SessioneOrdinalePartito

2	Partito: Partito Prova 1 Candidato Prova Uno Candidato Prova Secondo
Ordine	Partito: Partito Prova 2 Candidato Prova 2 Primo Candidato Prova 2 Secondo

Scheda Bianca

Avanti

3.3.7 Sessione con Voto Categorico con Preferenze

SessioneCategoricaConPreferenze

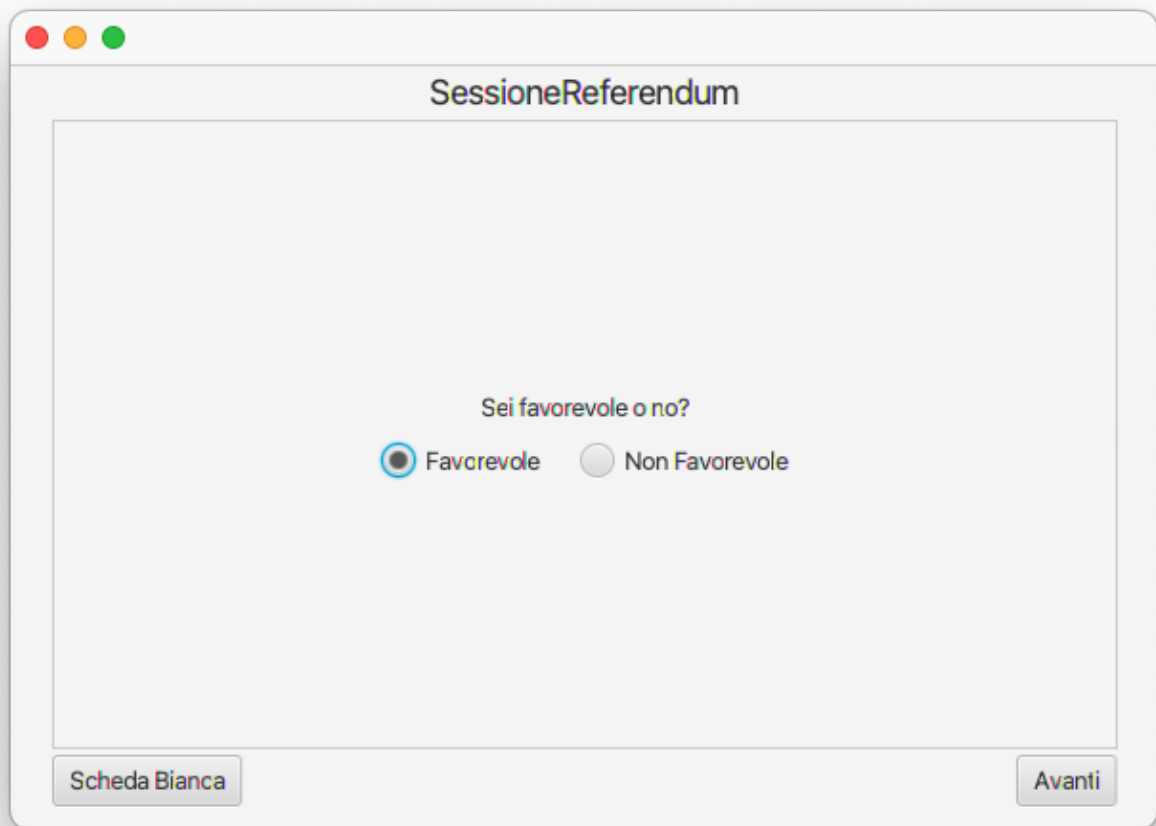
☐ Partito Prova 1 ☒ Partito Prova 2

☒ Candidato Prova 2 Primo

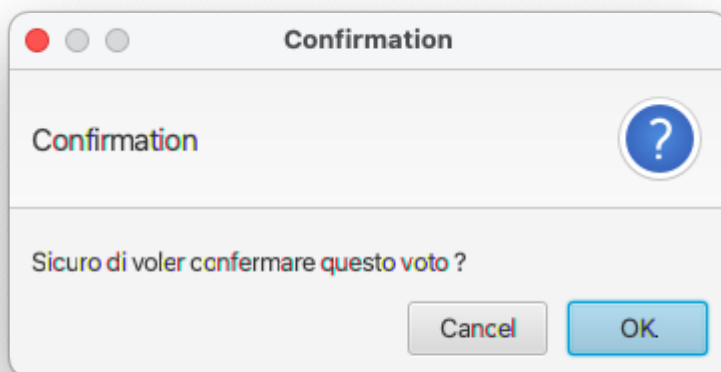
☐ Candidato Prova 2 Secondo

Scheda Bianca Avanti

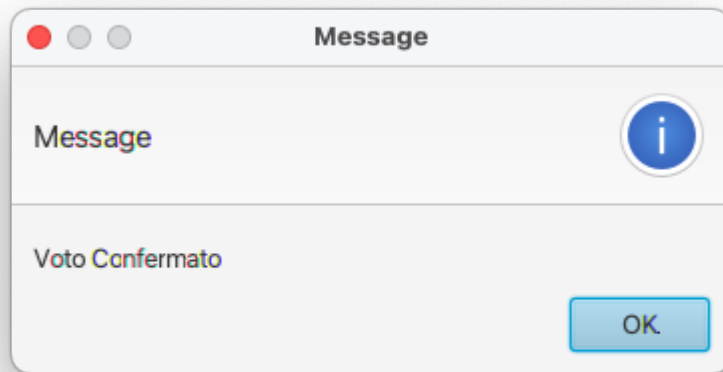
3.3.8 Sessione con Referendum



3.3.9 Richiesta Conferma Voto



3.3.10 Conferma Voto

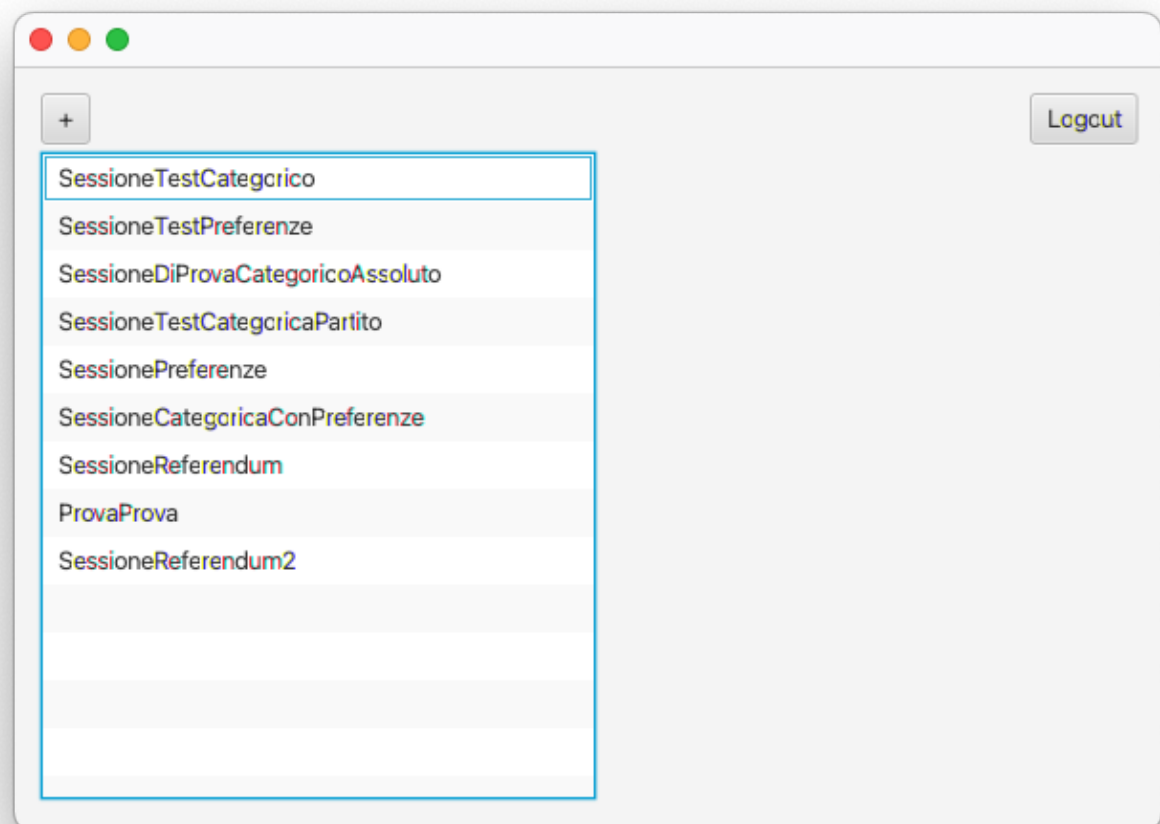


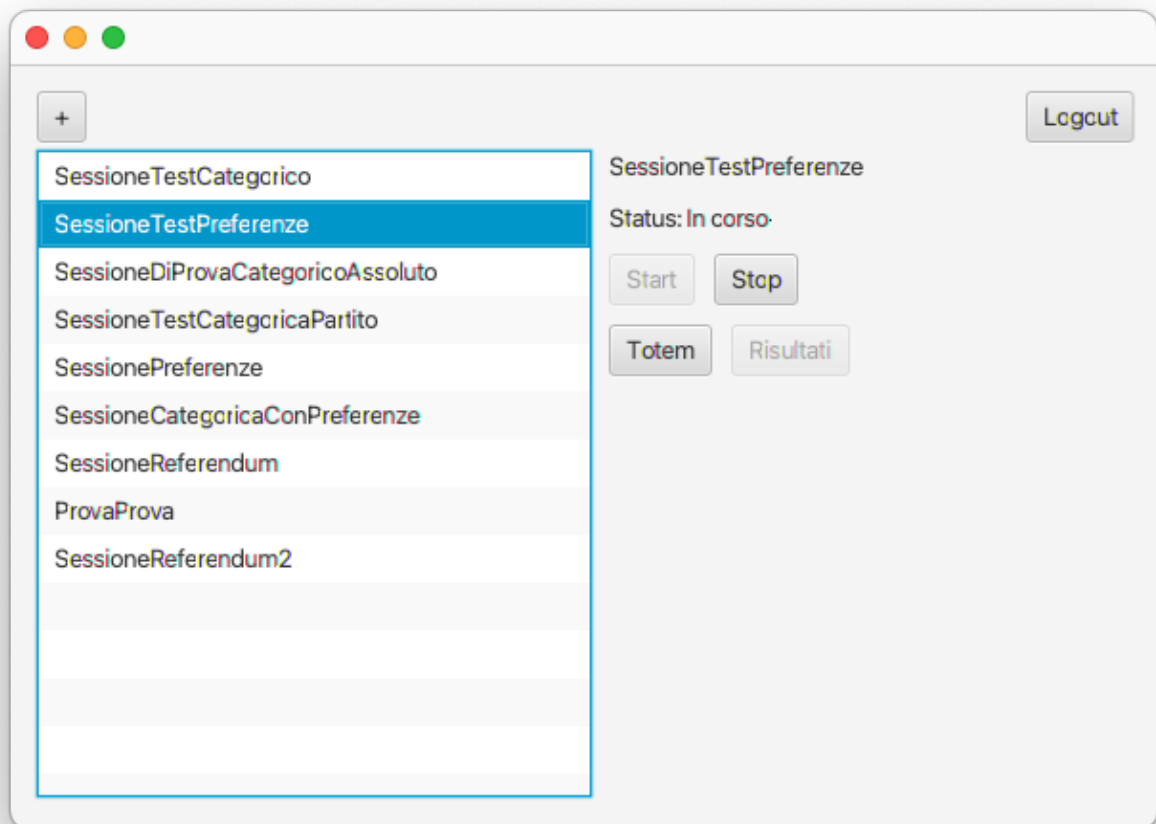
3.3.11 Home Gestore

La schermata principale del gestore presenta una sezione con le sessioni di voto esistenti e una sezione con le informazioni riguardanti la sessione di voto selezionata. Nella sezione informazioni sono presenti 4 tasti che sono attivi e non rispetto al stato della sessione.

- Il tasto Start permette di far iniziare la sessione di voto selezionata. (Stato in cui questo tasto è attivo: Sessione Nuova)
- Il tasto Stop permette di finire la sessione di voto selezionata. (Stato in cui questo tasto è attivo: Sessione In Corso)
- Il tasto Totem permette di attivare un Totem esterno, per un'elettore, con la sessione di voto selezionata. (Stato in cui questo tasto è attivo: Sessione In Corso)
- Il tasto Risultati permette di visualizzare i risultati della sessione selezionata. (Stato in cui questo tasto è attivo: Sessione Finita)

C'è presente anche un tasto di Logout che permette all'elettore di andare alla schermata di Login.





3.3.12 Configurazione Nuova Sessione

In questa schermata il gestore può: dare un nome alla sessione, scegliere la modalità di voto e di vincità usando le box di scelta nell'angolo superiore destro, inserire delle liste e dei candidati usando il primo rettangolo della schermata o inserire una domanda per il referendum usando il secondo rettangolo della schermata.

The screenshot shows a web application window with a light gray background and standard macOS window controls (red, yellow, green buttons) in the top-left corner. The interface is divided into several sections:

- Top Section:** Contains a text input field labeled "Nome Sessione" with the value "Nuova Sessione". To its right are two buttons: "Categorico Con Prefer." and "Maggioranza Assoluta". Below these is a checkbox labeled "Solo Partiti".
- Middle Section:** Titled "Aggiungi alla lista: Partito 2". It contains two sub-sections:
 - Partito Lista:** A text input field with "Partito 2" and a button labeled "Scegli Lista".
 - Dettagli Candidato:** Two text input fields labeled "Nome3" and "Cognome3", followed by a button labeled "Aggiungi Candidato a Lista".
- Right Side:** A list of candidate entries:
 - Candidato: Nome1 Cognome1
 - Lista: Partito 1
 - Candidato: Nome2 Cognome2
 - Lista: Partito 1
 - Candidato: Nome3 Cognome3
 - Lista: Partito 2
- Bottom Section:** Titled "Referendum", it contains a large empty text input field.
- Footer:** Two buttons, "Annulla" on the left and "Avanti" on the right.

3.3.13 Scelta Totem

La schermata è divisa in tre sezioni:

- i totem disponibili (mostrati col loro ip + porta)
- la sezione col tasto Entra (che attiva il totem selezionato)
- la sezione in cui si deve inserire il codice fiscale dell'elettore che vuole votare in presenza, sul totem selezionato

Quando si vuole attivare un totem, viene fatto un controllo per vedere se l'elettore non ha già votato nella sessione selezionata. Nel caso non si è inserito un codice fiscale nella sezione apposita, viene mostrato un>alert che informa il gestore di ciò.

127.0.0.1 52110

Entra

Codice Fiscale Elettore

TEST

Annulla Refresh

3.3.14 Attivazione Totem

SessioneCategoricaConPreferenze

Partito Prova 1 Partito Prova 2

Scheda Bianca Avanti

127.0.0.1 58733

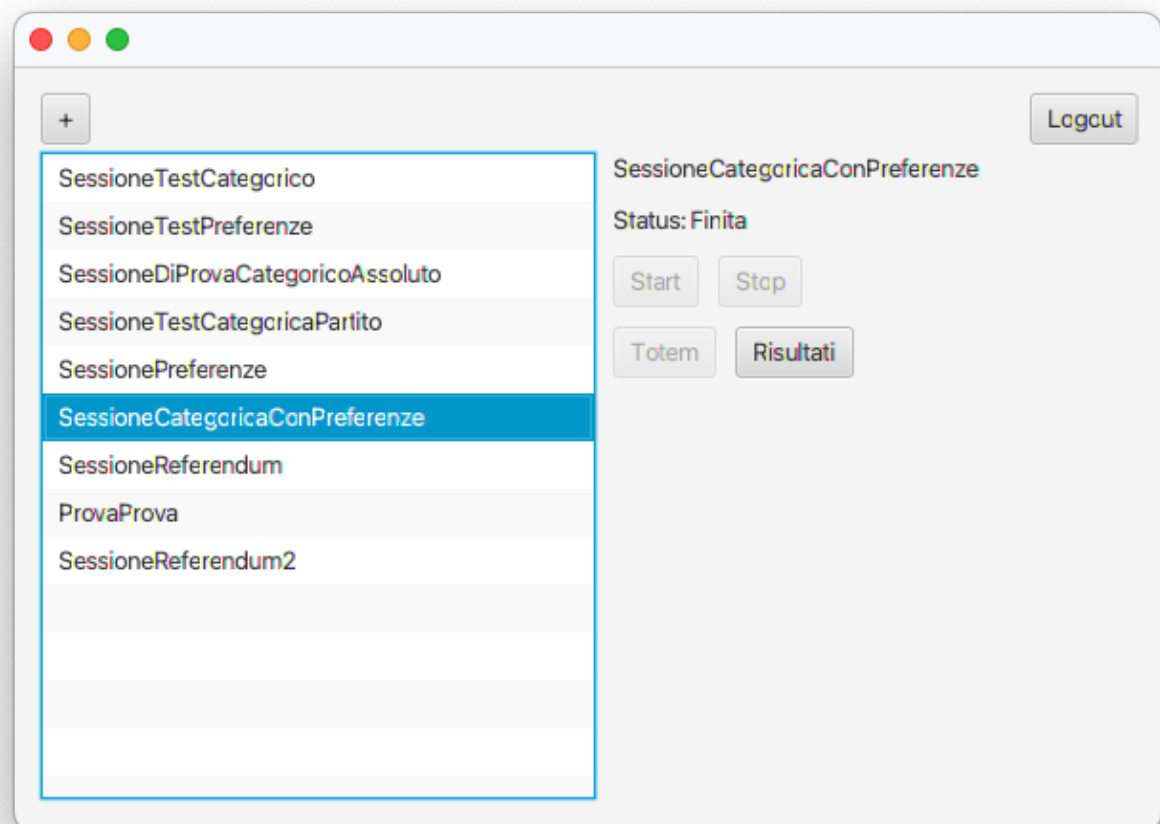
Entra

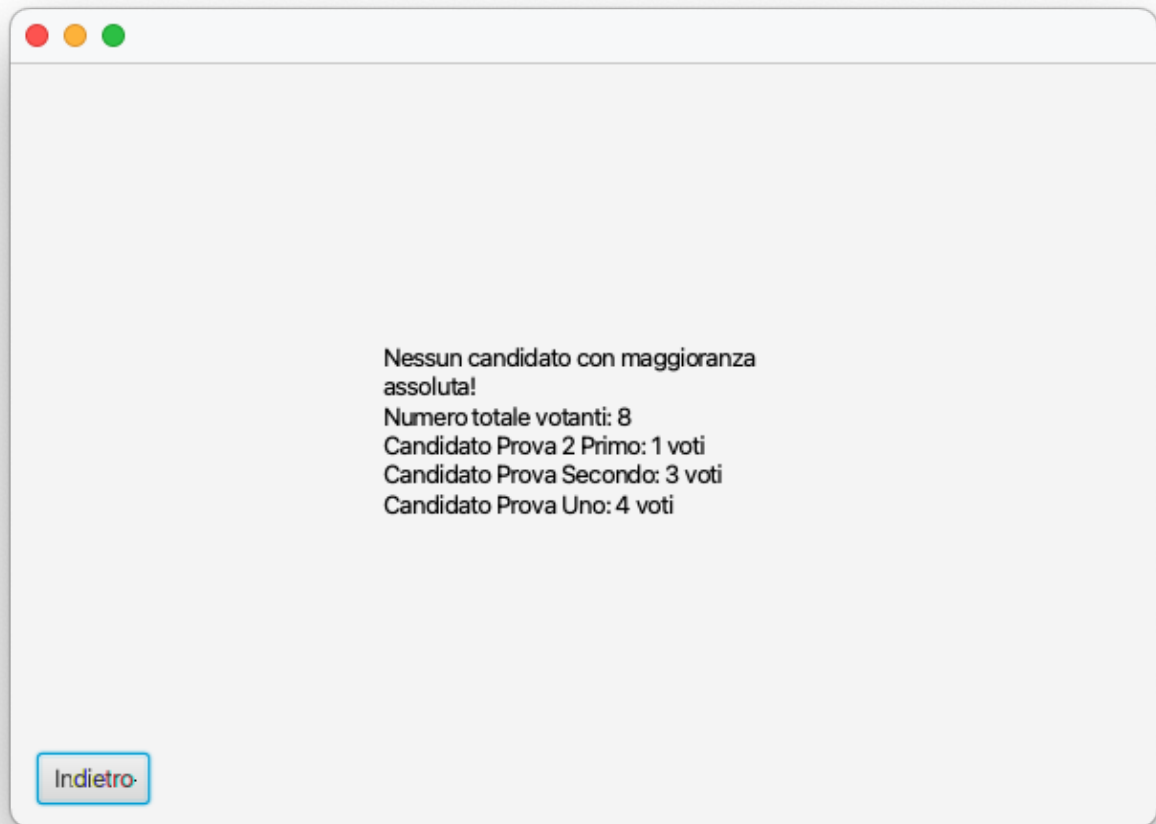
Codice Fiscale Elettore

TEST2

Annulla Refresh

3.3.15 Visualizza Risultati





3.3.16 Registrazione Elettore

Quando si preme Registra, viene fatto un controllo sugli input inseriti e, se c'è qualche campo che non è valido, questa informazione viene mostrata all'elettore non registrato tramite un>alert.

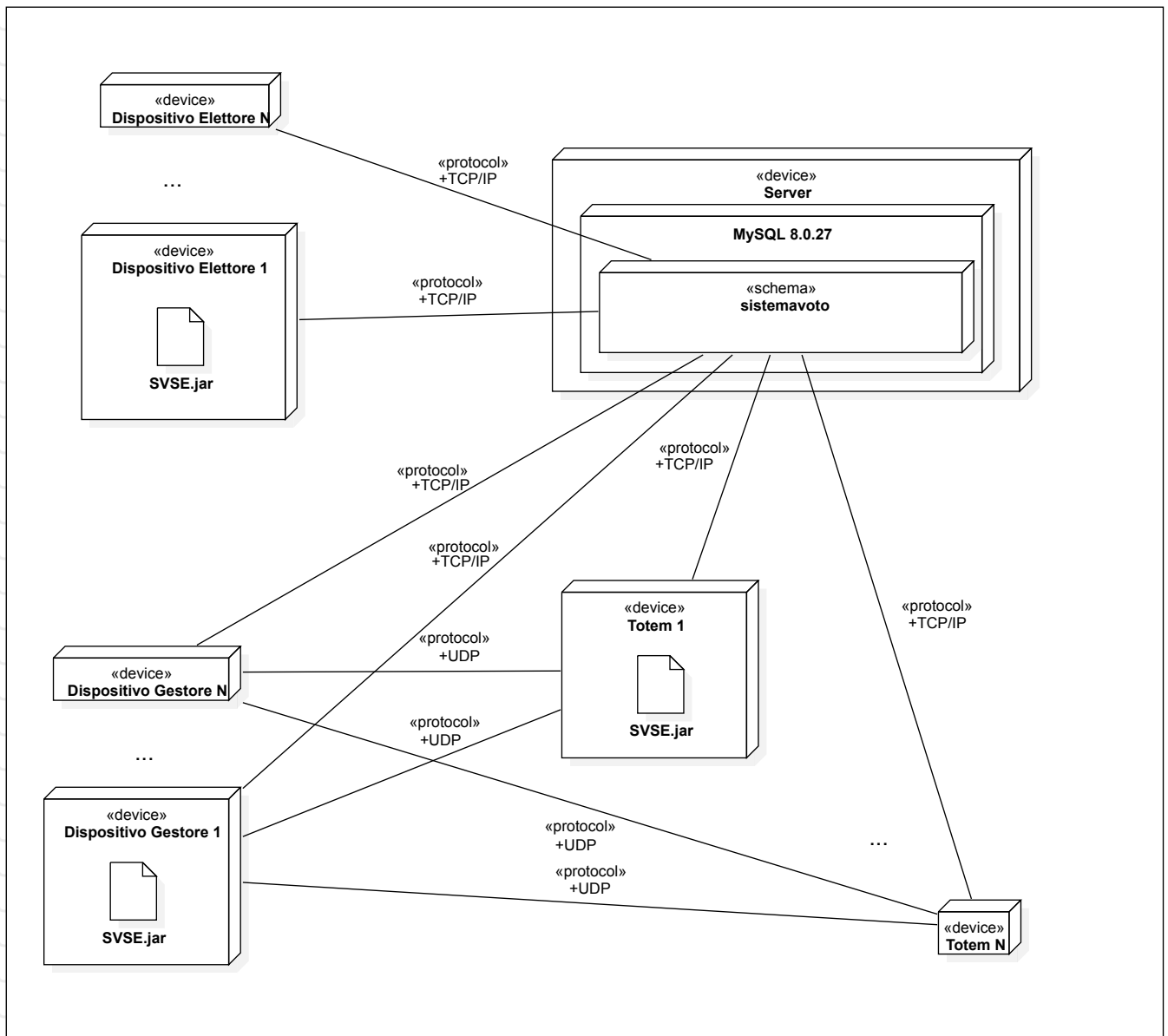
Registration form interface:

- Input field (empty)
- Input field: Cognome
- Input field: Codice Fiscale
- Input field: Eta
- Input field: Password
- Input field: Ripeti Password
- Buttons: Annulla, Registra

3.4 Diagramma di deployment

Il diagramma di deployment modella:

- dispositivi degli elettori e dei gestori che si connettono ad un backend contenente la base dati
- dispositivi dei gestori che comunicano con i totem per poter attivarli nel caso di votazione in presenza



3.5 Specifica e verifica dei vincoli

La mappatura JML si può trovare nelle rispettive classi.

3.5.1 UtenteJDBCDAO

```
{context UtenteJDBCDAO::login(cf: String, password: String)
```

pre:

```
cf <> null and password <> null}
```

```
{context UtenteJDBCDAO::sha1hashing(password: String)
```

pre:

```
cf <> null
```


post:

result <> password}

{context UtenteJDBCDAO::registraElettore(t: Utente, pass: String)

pre:

t <> null and pass <> null and getId(t) == 0

post:

result == true}

3.5.2 VotazioneJDBDDAO

{context VotazioneJDBCDAO::confermaVotazione(e: Elettore, s: SessioneDiVoto)

pre:

e <> null and s <> null

post:

haPartecipatoASessione(e, s) == true}

3.5.3 SessioneDiVoto

{context SessioneDiVoto

inv: self.strategiaVoto.equals("o") implies self.tipo <> null

inv: self.strategiaVoto.equals("c") implies self.tipo <> null

inv: self.strategiaVoto.equals("r") implies self.domandaReferendum <> null

inv: self.nomeSessione <> null

inv: self.strategiaVincita <> null

inv: self.strategiaVoto <> null

inv: self.stato <> null}

3.6 Descrizione del testing

I test sono generati da JUnit e sono stati implementati i seguenti:

- Test riguardanti il database: test su login di un'account esistente, test inizio sessione, test fermata sessione.
- Test riguardanti il dominio: test inizio sessione, test fermata sessione.

Viene fornito anche un dump della base dati in modo da poter testare l'applicazione con un gestore ed alcuni elettori già esistenti.

3.7 Note per l'installazione e l'utilizzo

L'ambiente di sviluppo è stato Eclipse 2020-12 (4.18.0) su sistema operativo MacOS con OpenJDK 11 (2018-09-25).

Per avviare il programma:

Main Class: svse.App

VM Options: --module-path "%PATH_TO_FX%;%PATH_TO_FX%\lib" --add-modules javafx.controls,javafx.fxml,javafx.graphics

Da Eclipse: la configurazione di run deve avere come goal javafx:run. Tutto il resto è configurato nel file **pom**.

3.7.1 Credenziali Utenti

CodiceFiscale: TEST

Tipo Utente: elettore

Password: 1234

CodiceFiscale: TEST2

Tipo Utente: elettore

Password: 1234

CodiceFiscale: TEST3

Tipo Utente: elettore

Password: 1234

CodiceFiscale: TEST5

Tipo Utente: gestore

Password 1234