

# Sistema di voto e scrutinamento Elettronico

**AUTORE:** Alessandro Riemer Barboza

**MATRICOLA:** 941878

**GITHUB LINK:**

<https://github.com/FriendlyWizard23/SistemaVotoScrutinamentoElettronico>

## 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

Colui che utilizza il sistema. Può essere un elettore oppure un gestore del sistema.

#### Gestore

Utente del sistema che ha il ruolo di configurare e gestire la sessione di voto. I suoi privilegi consistono nella creazione della sessione di voto e anche nella possibilità di avviare la fase di scrutinamento e visualizzazione dei risultati.

#### Elettore

l'elettore è un utente del sistema che ha il **diritto di voto**

#### Scheda

Strumento con cui l'elettore esprime la propria preferenza all'interno di una **sessione di voto**.

#### Applicazione

**programma** utilizzabile su un computer attraverso la quale è possibile accedere alla sessione di voto. (Accessibile dal gestore per la gestione della sessione e dall'utente per effettuare la votazione)

## Quorum

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

## Scheda Bianca

**scheda elettorale** su cui l'**elettore** non ha volontariamente apposto nessuna preferenza

## Maggioranza

Tipologia di votazione per la quale un partito/candidato/lista ha ottenuto il maggior numero di voti

## Maggioranza assoluta

tipologia di calcolo del vincitore per la quale vince chi ottiene i 50% + 1 dei voti espressi.

## Identificazione Elettore

Operazione avente come scopo quello di **identificare** l'elettore.

L'identificazione rappresenta il riconoscimento della persona che vuole votare. 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. Come prima cosa l'utente deve essere identificato tramite operatore umano oppure tramite la fase di login nel caso di una votazione da remoto.

3. L'elettore deve potere esprimere il suo voto di persona, utilizzando un totem (in un seggio elettorale), o a distanza, utilizzando l'applicazione su un dispositivo elettronico.
4. Il gestore di sistema deve avere la possibilità di configurare una sessione di voto, selezionare le liste dei candidati / selezionare i partiti / selezionare i candidati singolarmente / inserire la domanda del referendum e far partire oppure avviare la fase di scrutinio e visualizzare l'esito del voto. Inoltre il gestore deve avere la possibilità di selezionare la modalità di votazione tra le seguenti:

- Voto Ordinale
- Voto Categorico
- Voto Categorico con Preferenze
- Referendum

Il gestore deve avere la possibilità di selezionare anche una tra le seguenti strategie di voto:

- Maggioranza
- Maggioranza Assoluta
- Referendum senza Quorum
- Referendum con Quorum

5. L'elettore deve avere la possibilità di esprimere il proprio voto tramite **scheda bianca**

### 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 dell' applicazione in base al tipo di utente che effettua il login
- 2
  1. Nel caso di una votazione a distanza, il sistema deve implementare un processo di **login** per la fase d'**identificazione** per l'elettore.
- 3

1. Il sistema deve permettere al gestore di sistema di avviare e di fermare la fase di voto.
  2. Il sistema deve implementare un modo con cui il gestore di sistema ha la possibilità di inserire le liste dei candidati/partiti/candidati all'interno della sessione di voto.
  3. Il sistema deve implementare un modo con cui il gestore di sistema ha la possibilità di inserire la domanda del referendum.
  4. Il sistema deve visualizzare automaticamente, dopo la fase di scrutinamento, l'esito del voto.
  5. Il sistema deve implementare un'apposita interfaccia grafica che permette al gestore di scegliere la modalità di voto e la strategia di calcolo del vincitore.
- 4.1
    1. Il sistema deve richiedere all'elettore di scegliere una preferenza per il candidato (o gruppo/partito) presente nella scheda.
  - 4.2
    1. Il sistema deve richiedere all'elettore di ordinare i candidati (o gruppi/partiti) presenti nella scheda in base alle proprie preferenze.
  - 4.3
    1. Il sistema deve richiedere all'elettore di scegliere una preferenza per un una lista di partito presente nella scheda, e in seguito permettere di indicare una o piu preferenze tra i candidati della lista selezionata selezionato che compongono quella lista
    2. una lista dunque è un sottoinsieme di candidati appartenenti ad uno stesso partito. Una lista può essere utilizzata in piu' sessioni di voto e una sessione di voto contiene piu' liste
  - 4.4
    1. Il sistema deve presentare all'elettore una domanda, con la quale si chiede l'espressione di un foto favorevole o non favorevole.
  - 5
    1. Il sistema deve impedire al gestore di sistema di selezionare altre strategie di voto al di fuori di Referendum senza Quorum e Referendum con Quorum

quando la modalità di voto selezionata e' referendum

2. Il sistema deve impedire al gestore di sistema di selezionare Referendum senza Quorum e Referendum con Quorum come strategia di voto quando la modalità di voto non è Referendum.

- 5.1

1. Nel caso della selezione di maggioranza come strategia di voto, dopo la chiusura della fase di voto, il sistema deve considerare vincitore il candidato che ha ottenuto il maggior numero di voti.
2. In seguito alla chiusura della fase di voto, nel caso ci sia pareggio, il sistema deve mostrare un messaggio significativo che identifichi questa situazione.

- 5.2

1. Nel caso della selezione di maggioranza assoluta come strategia di voto, 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. Quando si parla di votazione senza quorum, 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. Quando si parla di votazione con quorum, 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 permettere all'utente di esprimere un voto tramite scheda bianca

## **1.3.2 Requisiti Non Funzionali**

### **1.3.2.1 Requisiti di sicurezza**

1. il sistema deve permettere all'utente di votare all'interno di una sessione di voto, senza che il voto che ha espresso sia riconducibile a lui. Il voto quindi non deve per alcuna ragione essere vincolato a colui che lo ha espresso.
2. il sistema deve prevedere un meccanismo di auditing che funzioni tramite log, per verificare il corretto funzionamento del programma e tenere traccia di eventuali errori.
3. Una volta fermata la sessione, essa non può essere riattivata dal gestore per evitare invalidazioni del vincitore da parte anche del gestore stesso.

### **1.3.2.2 Requisiti di sviluppo**

1. Il sistema dev'essere implementato in Java.
2. L'interfaccia grafica del sistema deve essere implementata usando JavaFX.
3. Il progetto contiene i seguenti Design Pattern:
  - a. MVC
  - b. DAO
  - c. Singleton
  - d. Abstract Factory

### **1.3.2.3 Requisiti di affidabilità**

1. prima di ogni azione importante quale espressione del voto, conferma della creazione di una sessione oppure chiusura di una sessione di voto, il sistema

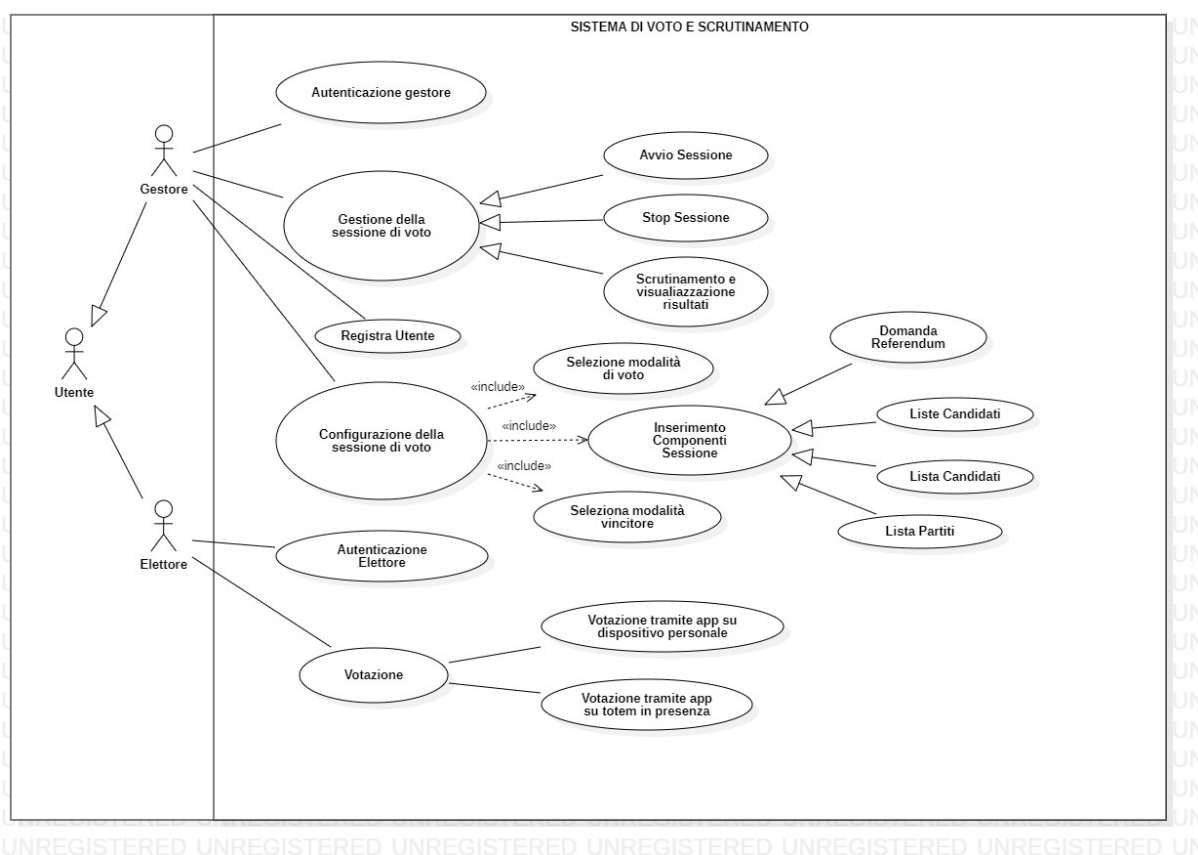
deve chiedere conferma all'utente.

### 1.3.2.4 Requisiti normativi

1. un elettore può votare una volta sola la sessione di voto.

## 2 Progettazione del sistema

### 2.1 diagrammi dei casi d'uso



Di seguito definita una breve descrizione per ogni possibile scenario:

- **Autenticazione** : L'utente può autenticarsi nel sistema tramite applicazione usando le sue credenziali definite in previa fase di registrazione.

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

- **Gestione Sessione di Voto:** Il gestore di sistema, in seguito alla fase di autenticazione (login) nel sistema come gestore, può gestire le sessioni esistenti e quindi effettuare operazioni di avvio sessione, stop sessione e di avvio scrutinamento e visualizzazione risultati.

*Requisito funzionale associato:* Il gestore di sistema deve avere la possibilità di configurare una sessione di voto, selezionare le liste dei candidati / selezionare i partiti / selezionare i candidati singolarmente / inserire la domanda del referendum e far partire oppure avviare la fase di scrutinio e visualizzare l'esito del voto.

- **Registra Utente:** Il gestore deve poter inserire all'interno del sistema un nuovo utente cosicché quest'ultimo possa effettuare il login per votare future sessioni di voto.

*Requisito funzionale associato:* il sistema deve permettere al gestore di iscrivere un nuovo utente

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

*Requisito funzionale associato:* Il gestore di sistema deve avere la possibilità di configurare una sessione di voto, selezionare le liste dei candidati / selezionare i partiti / selezionare i candidati singolarmente / inserire la domanda del referendum e far partire oppure avviare la fase di scrutinio e visualizzare l'esito del voto.

Nome	Autenticazione
Scopo	Accedere al sistema in modo da poter partecipare ad una sessione di voto / Accedere al sistema in modo da poter modificare / inserire una sessione
Attore	Elettore, Gestore



Pre Condizioni	Elettore registrato, Gestore registrato
Trigger	Utente vuole entrare nella sessione
Descrizione sequenza eventi	1) Utente inserisce le sue credenziali nella sua sezione specifica di Login. 2) Utente preme il pulsante Autentifica.
Alternative	
Post Condizioni	L'utente risulta autenticato correttamente nella applicazione

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

*Requisito funzionale associato: L'elettore deve potere esprimere il suo voto di persona, utilizzando un totem (in un seggio elettorale), o a distanza, utilizzando l'applicazione su un dispositivo elettronico.*

## 2.2 Descrizione degli scenari

Nome	<b>Gestione Sessione di voto</b>
Scopo	Avviare, fermare o visualizzare l'esito di una sessione di voto.
Attore	Gestore
Pre Condizioni	Gestore autenticato
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 su uno dei tasti che permettono di avviare / fermare o visualizzare esiti di una sessione di voto
Alternative	
Post Condizioni	La sessione di voto viene avviata, fermata, modificata o l'esito viene visualizzato.

Nome	<b>Registra Utente</b>
Scopo	registrare un nuovo utente nel sistema per permettergli di votare
Attore	Gestore
Pre Condizioni	Gestore deve essere autenticato nel sistema

Trigger	Gestore decide di registrare un utente nel sistema
Descrizione sequenza eventi	1) il gestore inserisce le informazioni del nuovo utente da registrare 2) il gestore preme invio 3) le informazioni vengono inserite nella base dati
Alternative	
Post Condizioni	Il gestore inserisce il nuovo utente nella base dati

Nome	<b>Configurazione Sessione di Voto</b>
Scopo	Creare una sessione di voto con determinate caratteristiche
Attore	Gestore
Pre Condizioni	Gestore deve essere autenticato nel sistema
Trigger	il gestore decide di creare una nuova sessione
Descrizione sequenza eventi	1) sceglie il codice della sessione. 1.1) il sistema verifica che il codice inserito sia disponibile e valido. 2) il gestore seleziona il tipo di votazione che vuole creare. 3) Il gestore sceglie la modalità di scelta del vincitore. 4) Il gestore inserisce le liste dei candidati/la domanda del referendum. 5) Il gestore preme crea e appare un alert nel quale viene richiesto se davvero vuole proseguire e creare una sessione 6) Il gestore preme il pulsante OK.
Alternative	1.2) la sessione non e' valida poiche' esiste e quindi si riparte dal punto 1 6.1) il gestore non accetta la creazione della sessione e quindi modifica i dati e ritorna al punto 6
Post Condizioni	viene creata la nuova sessione

Nome	<b>Votazione</b>
Scopo	Permettere all'elettore di esprimere il suo voto tramite l'applicazione
Attore	Elettore
Pre Condizioni	Elettore deve essere autenticato nel sistema
Trigger	L'elettore decide di esprimere il proprio voto in una sessione di voto
Descrizione sequenza eventi	1) L'elettore inserisce il codice della sessione di voto e preme avanti 2) L'elettore esprime 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 è sicuro, preme OK e gli compare un>alert che conferma la registrazione del voto. 5. L'elettore ritorna alla pagina di login.
Alternative	1.2) la sessione non è valida poichè non esiste / è stata chiusa / è già stata votata dall'elettore in questione 2.1) L'elettore decide di avvalersi

	del diritto della <b>scheda bianca</b> . 4.2) Dopo aver visto che il voto è sbagliato, l'elettore preme Cancel.
Post Condizioni	viene creata la nuova sessione

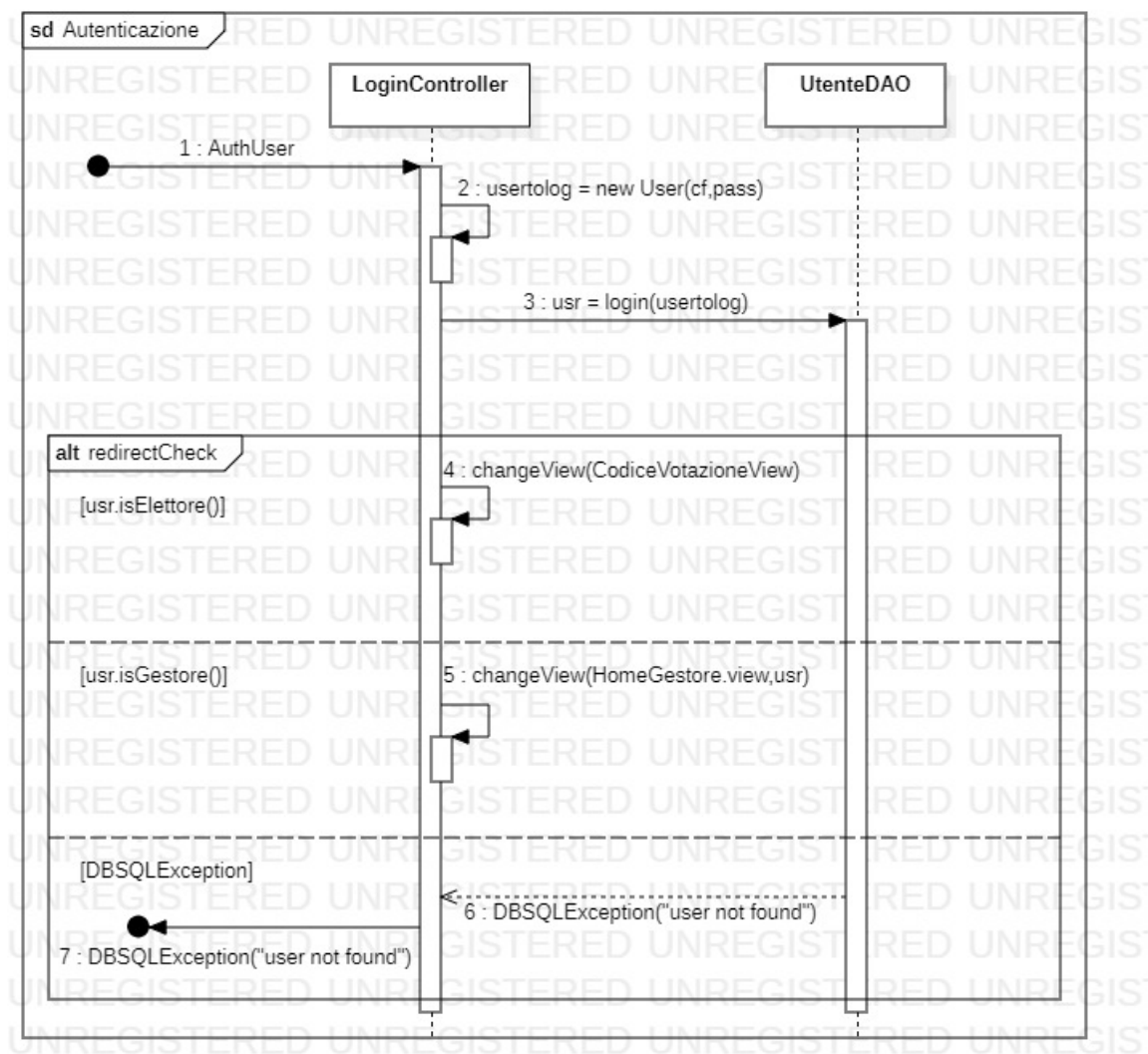
## 2.3 Diagramma delle classi

si veda sezione 3.1

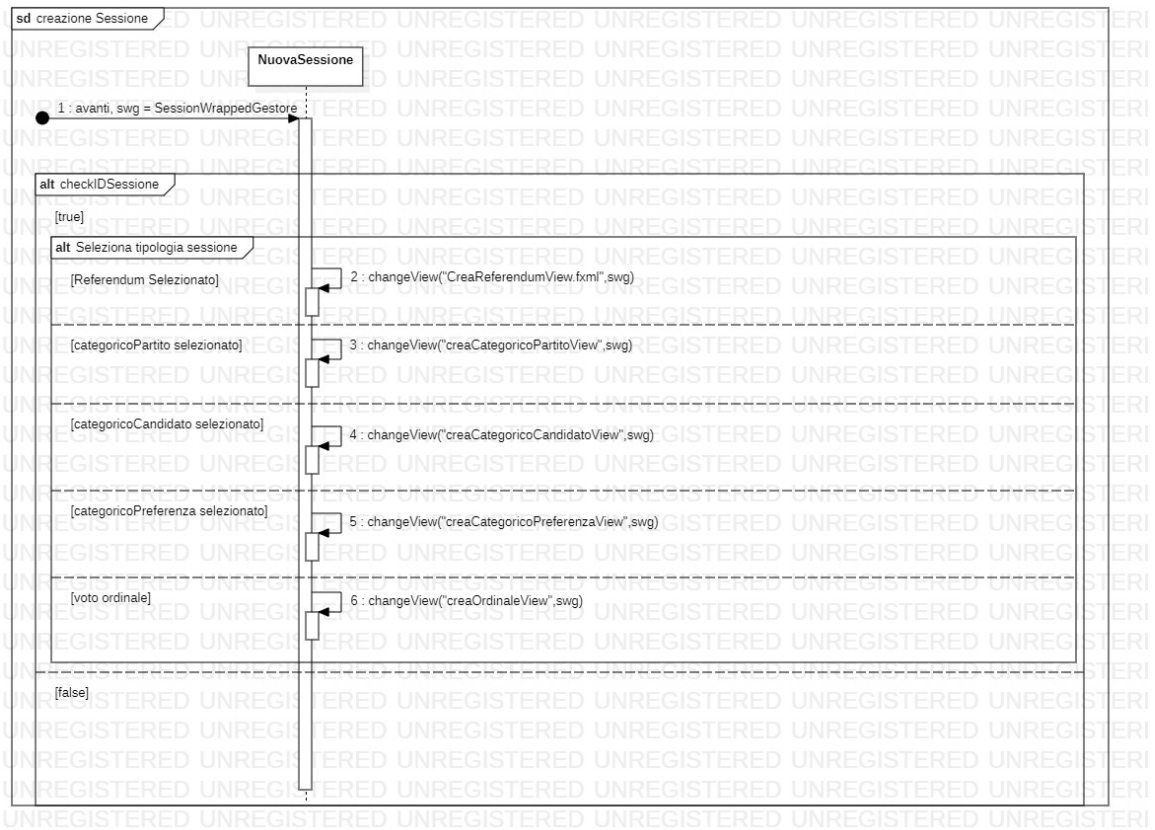
## 2.4 Diagramma di sequenza

Vengono di seguito illustrati alcuni diagrammi di sequenza

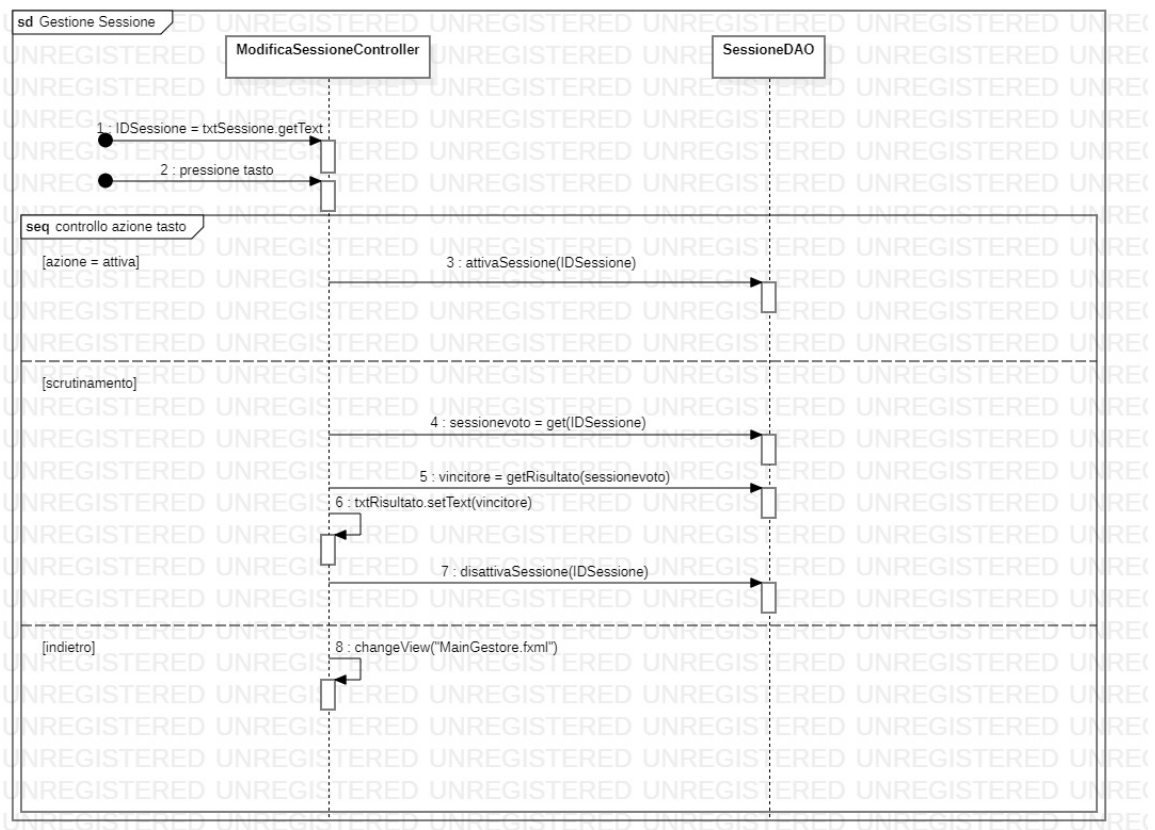
### 2.4.1 Autenticazione Elettore



### 2.4.2 Configurazione Sessione di Voto

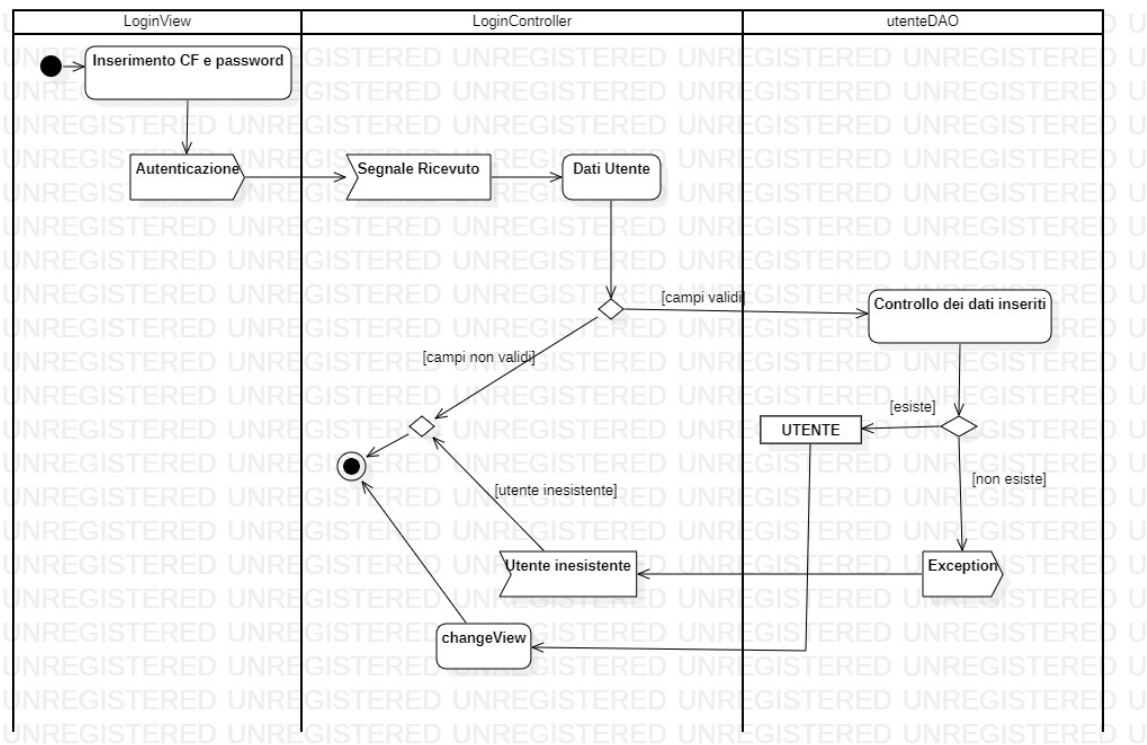


## 2.4.3 Gestione Sessione di voto



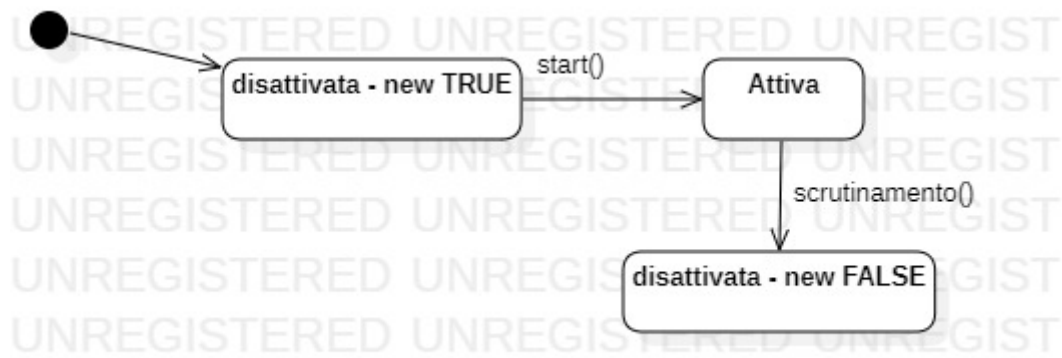
## 2.5 Diagramma delle attività

### 2.5.1 Autenticazione / login



### 2.5.2 Votazione





importante dire che la macchina a stato nella rappresentazione concreta del concetto viene gestita tramite la base dati. I flag **new** e **status** sono infatti degli attributi dell'entità `Sessione_voto` della base dati e vengono aggiornati in base alle operazioni svolte.

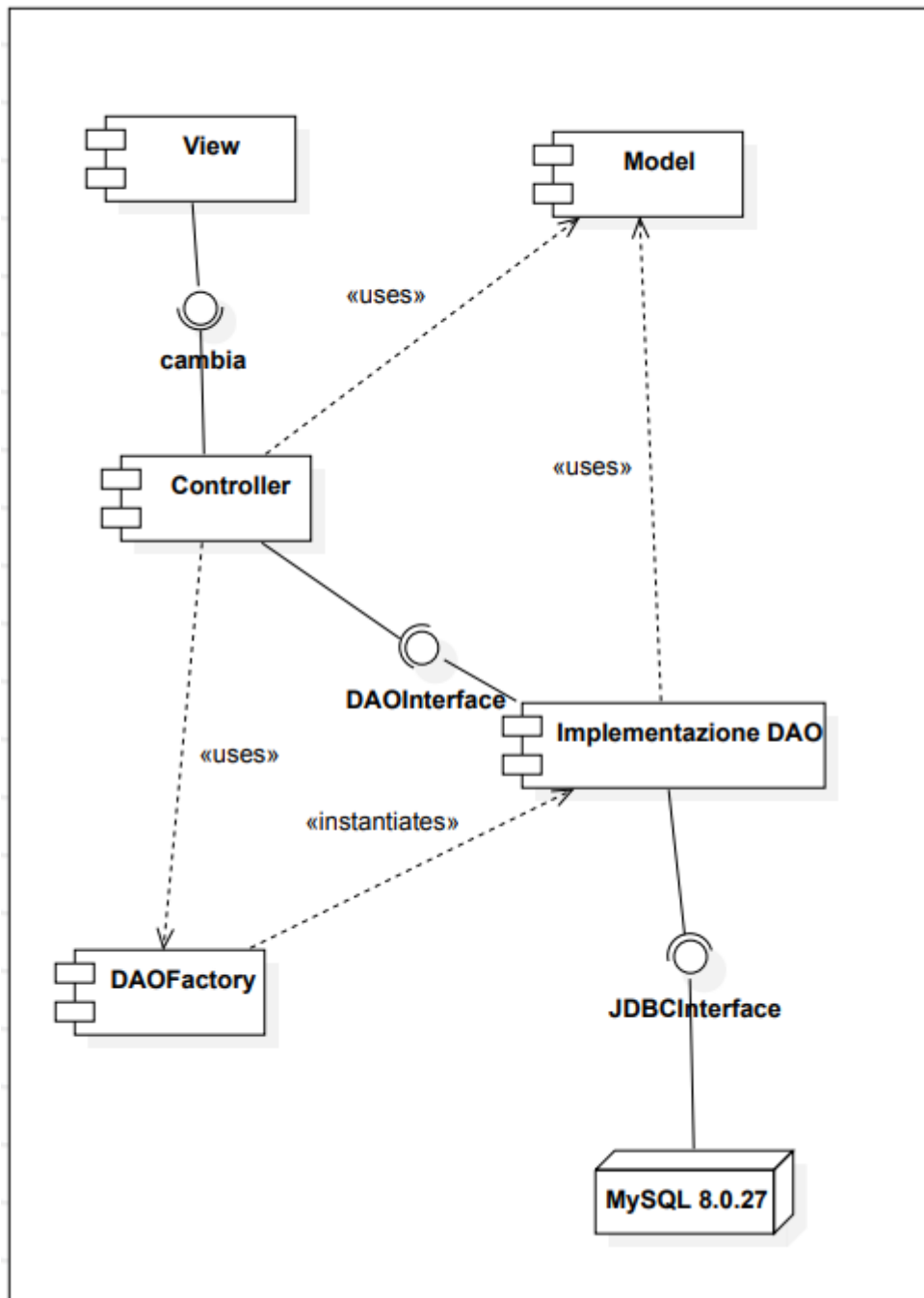
## 2.7 Diagramma dei componenti

per definizione

*I diagrammi dei componenti sono utilizzati per visualizzare l'organizzazione dei componenti del sistema e le relazioni di dipendenza tra di essi*

Di seguito il diagramma che mostra l'interazione tra le varie componenti del sistema.

è possibile notare alcuni pattern architetturali che verranno definiti meglio nel paragrafo 3.1.5

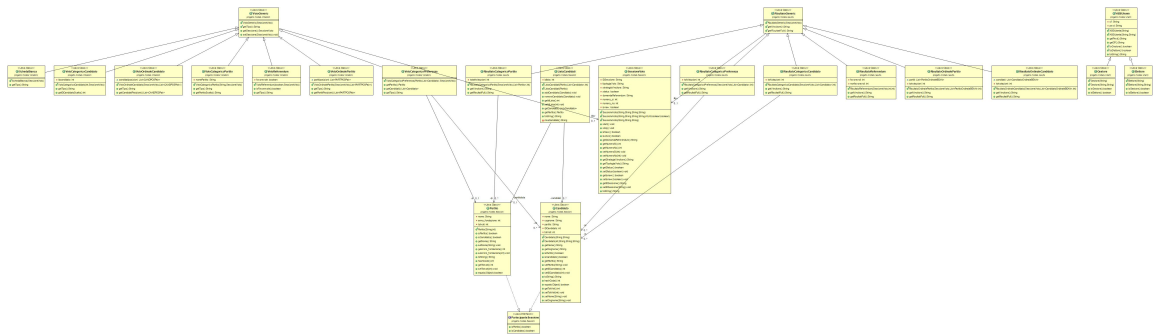


## 3 Implementazione del sistema

### 3.1 Diagramma delle classi

#### 3.1.1 Modelli



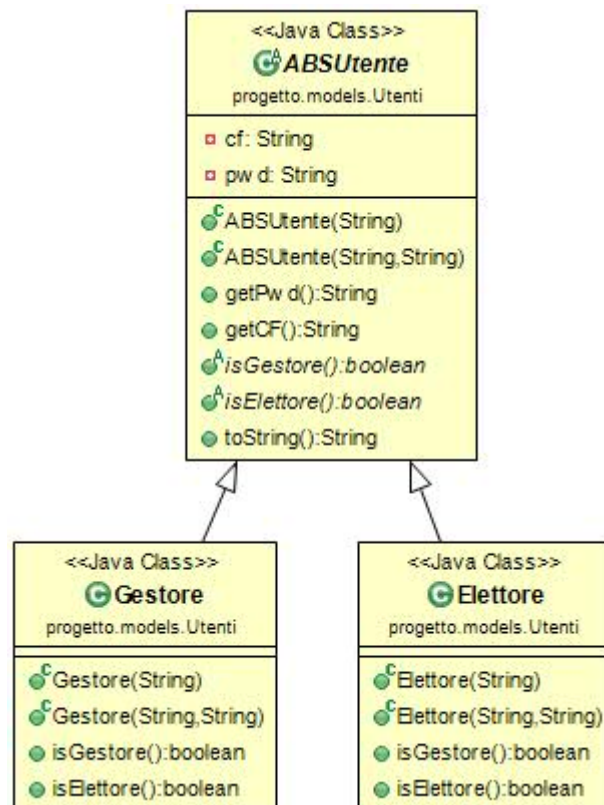


### 3.1.1.1 Utente

Come si può notare dal diagramma che segue, l'utente generico permette di gestire nel miglior modo le tue tipologie di utenti utili al sistema.

Entrambi infatti sono caratterizzati da un Codice Fiscale e una password ma poi vengono utilizzati in maniera diversa in base al contesto.

La tecnica che ci permette di traslare questa forma gerarchica su java si chiama **astrazione**.

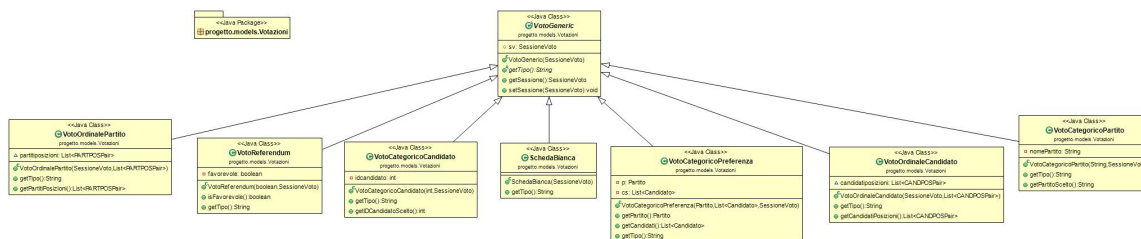


### 3.1.1.2 Votazioni

Come per gli elettori ho fatto utilizzo dell'astrazione.

Tutte le votazioni estendono VotoGeneric che contiene le caratteristiche generali

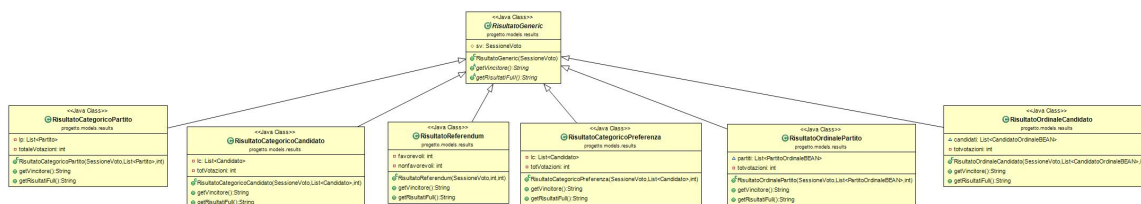
di un voto, quindi la sessione di voto e dei metodi che verranno poi sovrascritti da ciascuna classe che lo estende



### 3.1.1.3 Risultati

Come per le votazioni ho fatto utilizzo del concetto di astrazione che genera anche qui una forma di gerarchia.

Tutte le classi di risultati estendono RisultatoGenerico che contiene le caratteristiche generali di un risultato, quindi, ancora, la sessione di voto e dei metodi che verranno poi sovrascritti da ciascuna classe che lo estende

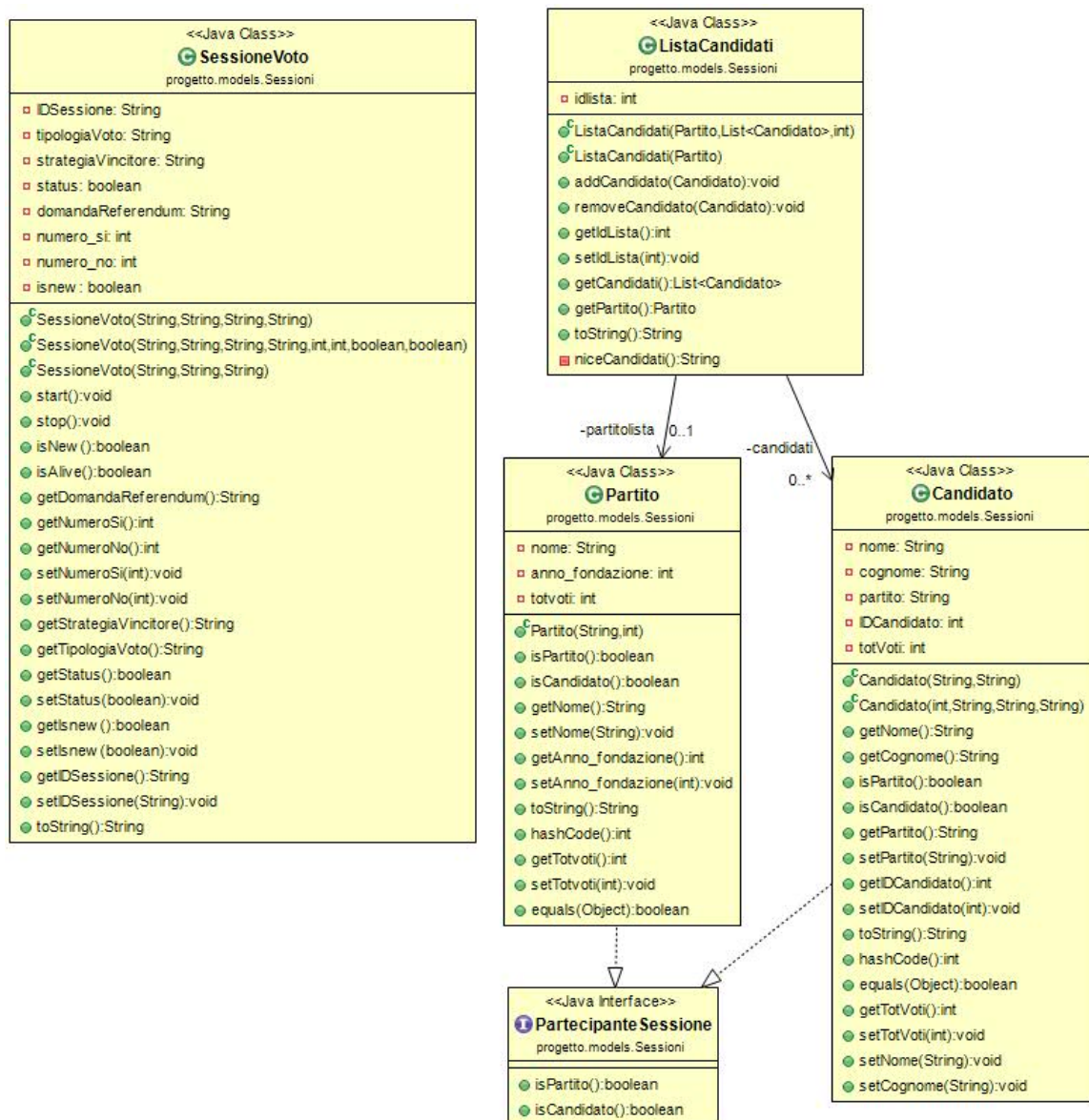


### 3.1.1.4 Sessioni

Come si può notare, il modello SessioneVoto contiene diversi costruttori per le diverse esigenze e comodità. Contiene inoltre diversi metodi tra cui svariati getter e setter, molto utili per le classi che definiscono oggetti di questo tipo.

Analizzando invece ListaCandidati, è possibile osservare che essa contiene oggetti di tipo Partito e Candidato i quali sono a loro volta una implementazione concreta di PartecipanteSessione.

La comodità dell'astrazione che ci permette la programmazione ad oggetti viene ancora una volta utilizzata per rendere la struttura gerarchica migliore e più comprensibile.



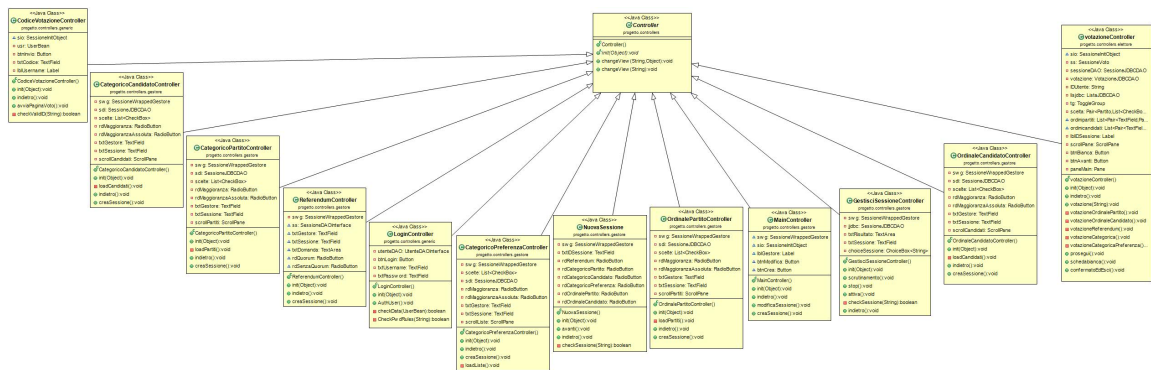
### 3.1.2 Controllers

I controllers sono la parte cruciale di tutto il sistema. Essi manipolano le view e giocano il ruolo fondamentale nel modello **MCV** (paragrafo [3.1.5](#)).

Come suggerisce il nome, il controller contiene tutto il necessario per permettere al codice di interagire facilmente con l'utente tramite l'interfaccia grafica.

Ancora una volta l'astrazione viene in aiuto e permette di rendere più compatta la struttura del sistema. Tutti i controller presenti estendono la classe astratta Controller che contiene i metodi fondamentali di ogni controller ovvero:

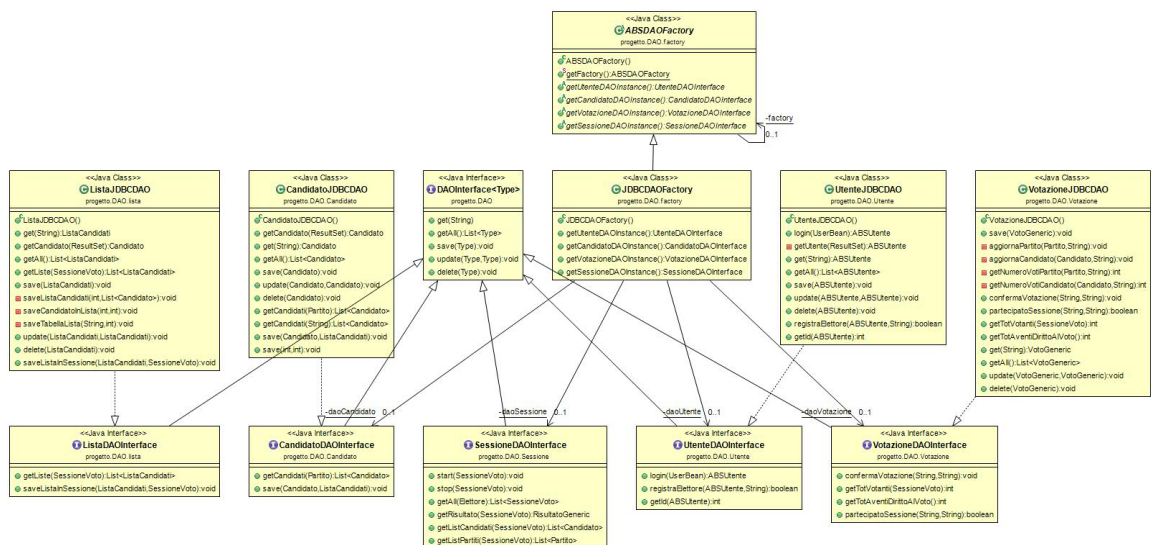
- **init** : permette di inizializzare il controller con dei parametri, utile per il passaggio di dati da una view ad un'altra
- **changeView** : permette di cambiare view



### 3.1.3 DAO

La struttura DAO è molto importante per l'esecuzione del programma, come sempre si utilizza l'astrazione e tutte le classi interagiscono tra di loro in una struttura simile a quella di una rete magliata.

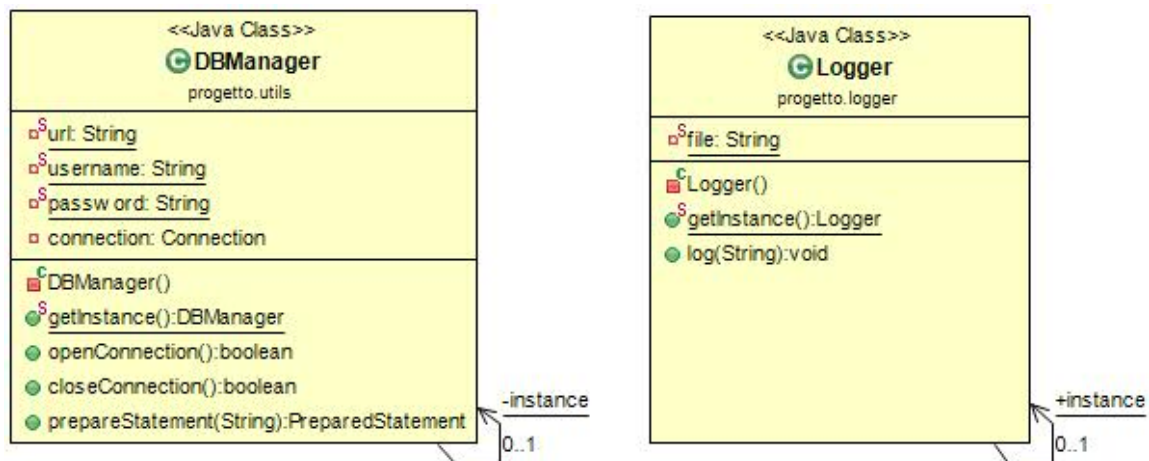
Maggiori informazioni sul pattern DAO al paragrafo [3.1.5](#)



### 3.1.4 DBManager e Logger

Queste due classi rientrano nelle utils del programma.

- **DBManager** viene utilizzata per gestire la connessione al database e si basa sul pattern singleton (paragrafo [3.1.5](#))
- **Logger** viene utilizzato come strumento di auditing per scrivere su un file di testo eventuali errori generati dal programma durante il suo utilizzo. Anch'esso si basa sul pattern singleton



### 3.1.5 Design Pattern Utilizzati

- **DAO:** Il Data Access Object viene utilizzato per astrarre tutte le operazioni su database tutti i DAO sono stati implementati assieme al design pattern Abstract Factory che facilita l'ulteriore aggiunta di eventuali future implementazioni per tutte le operazioni che coinvolgono l'utilizzo di database (e quindi l'accesso ai DAO) senza conoscere l'implementazione concreta che si sta usando. Per comprendere meglio il funzionamento del pattern si osservi la sezione [3.1.3](#)
- **MVC:** *"Il componente centrale del MVC, il modello, cattura il comportamento dell'applicazione in termini di dominio del problema, indipendentemente dall'interfaccia utente. Il modello gestisce direttamente i dati, la logica e le regole dell'applicazione. Una vista può essere una qualsiasi rappresentazione in output di informazioni, come un grafico o un diagramma."* Questa definizione è esplicita nel progetto dove queste componenti sono ben separate
- **SINGLETON:** All'interno di questo progetto, il pattern viene utilizzato nella classe DBManager e nel Logger questo perché per tutta l'esecuzione, non si necessita di più di una connessione con la base dati / di una apertura del file, e queste due classi implementate come Singleton aiutano a fare proprio questo. Per una concreta rappresentazione osservare il diagramma delle classi nel paragrafo [3.1.4](#)
- **ABSTRACT FACTORY:** Questo pattern viene usato per facilitare l'aggiunta di nuove implementazioni per le DAO e per semplificarne l'utilizzo all'interno del codice, mascherando la vera e propria implementazione che si intende usare. Nel paragrafo [3.1.3](#) è possibile osservare la struttura del pattern a livello concreto nel progetto.



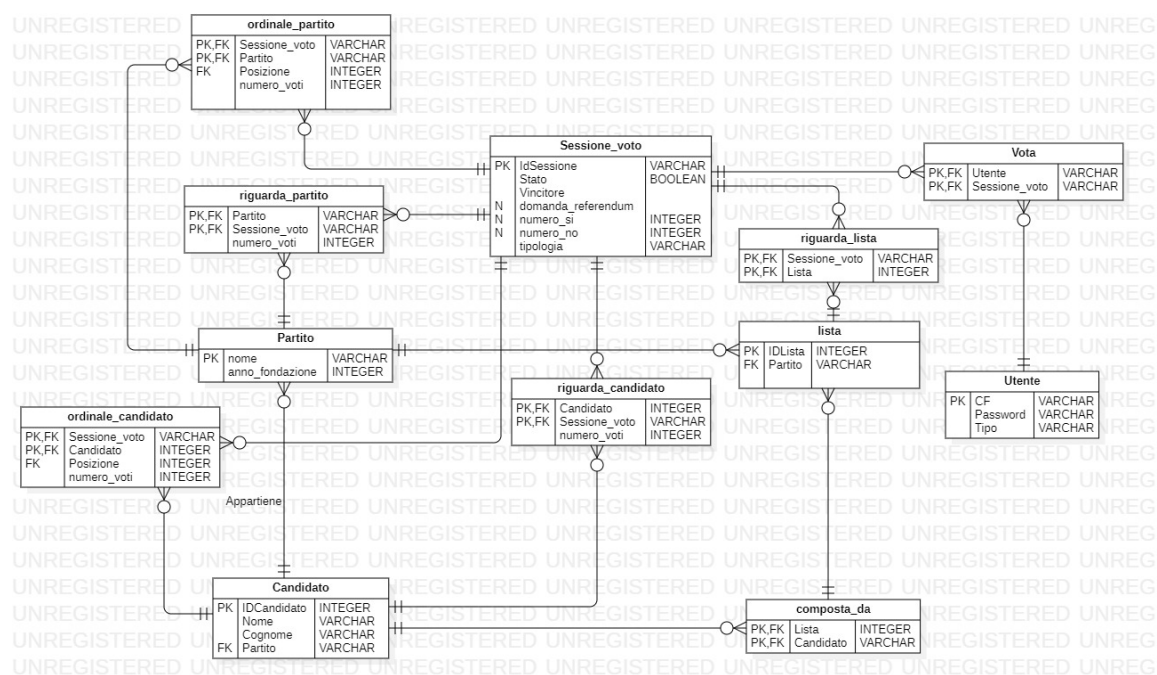
## 3.2 Gestione Dati Persistenti

Per la gestione dei dati persistenti si è deciso di utilizzare un DB MySQL versione 8.0.27.

Se si volesse usare un'altro DB, bisognerebbe aggiungere una nuova implementazione per ogni interfaccia DAO e ovviamente sarebbe necessario modificare la DAOFactory in modo che si occupi anche della generazione di oggetti di questa nuova implementazione.

### 3.2.1 Struttura Database

Di seguito una rappresentazione concettuale della base dati impiegata:



### Spiegazione entità diagramma ER:

#### ◦ Sessione\_voto

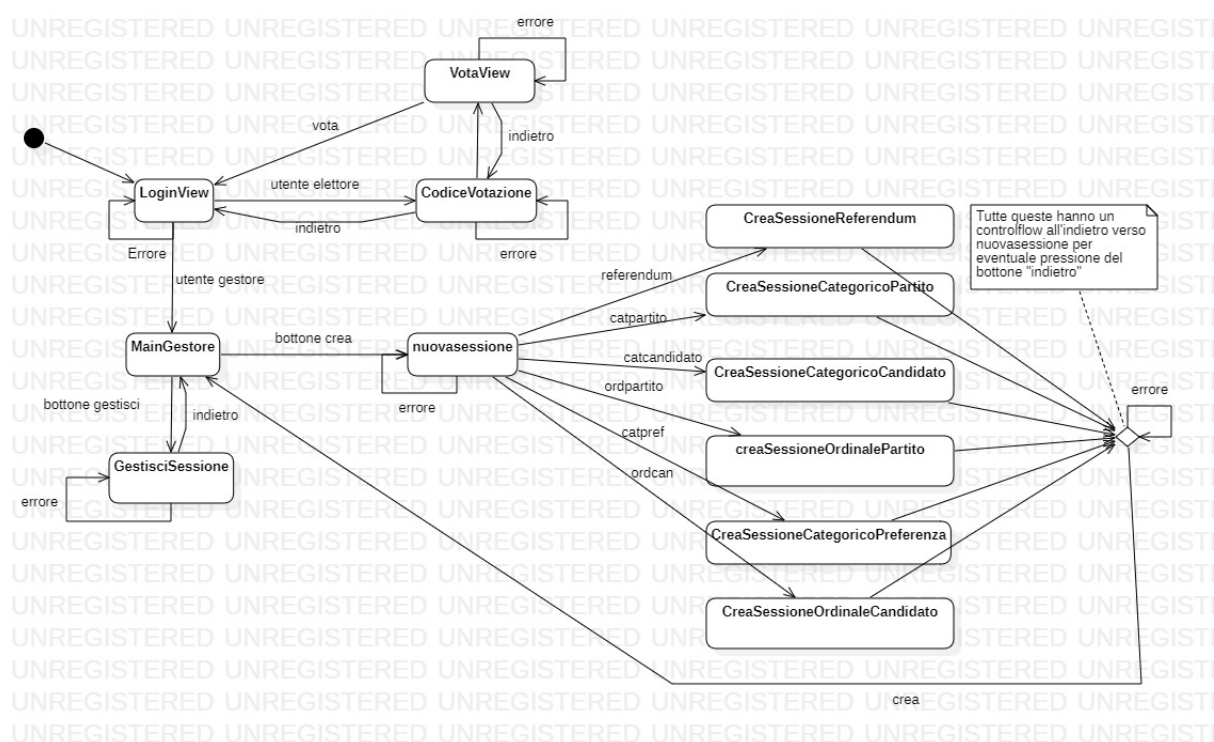
- Rappresenta la sessione di voto e tutte le sue caratteristiche.
- una sessione di voto può essere votata da uno o più utenti, essere composta da una o più liste (nel caso di una votazione di tipo categorica con preferenza) oppure essere composta da uno o più partiti oppure uno o più candidati
- dal punto precedente quindi risalta la necessità di creare delle tabelle di giunzione, ragion per cui ho definito le seguenti tabelle di giunzione: riguarda\_partito, riguarda\_lista, riguarda\_candidato

- Queste tabelle di giunzione hanno pressochè gli stessi attributi poichè hanno il ruolo di collegare uno o più partiti/candidati/liste alla sessione.
  - Per la gestione di una votazione di tipo referendum invece ho inserito semplicemente un attributo domanda e numero\_si e numero\_no
  - infine, la sessione contiene anche la strategia per definire il vincitore, la tipologia di sessione e due flag, status e isnew usati per definire lo stato della sessione in un determinato momento
- **Utente**
    - Rappresenta l'utente del sistema.
    - Esso può essere un elettore oppure un gestore, fattore definito in base al flag tipo (1=gestore, 0=elettore)
    - Anche qua è presente una tabella di giunzione chiamata vota, utilizzata per poter analizzare quali sessioni sono state votate da quali elettori.
- **Partito**
    - rappresenta un partito, caratterizzato da nome e anno fondazione
    - Come spiegato nella sessione di voto, un partito può partecipare a diverse sessioni di voto per questo la tabella di giunzione riguarda\_partito.
- **Candidato**
    - Rappresenta il candidato ad una sessione di voto
    - Il candidato è caratterizzato da un nome, un cognome e un identificativo.
    - è presente anche una Foreign Key come attributo poichè un candidato appartiene ad un partito.
    - Come spiegato nella sessione di voto, un candidato può partecipare a diverse sessioni di voto per questo la tabella di giunzione riguarda\_candidato.
- **Lista**
    - La lista rappresenta un sottoinsieme di candidati appartenenti ad un partito.
    - Viene utilizzata la tabella di giunzione composta\_da per poter di fatto legare più un candidato a più liste e una lista a più candidati.

- Come attributi si è deciso di inserire semplicemente un identificativo e una Foreign Key al partito
- Come spiegato nella sessione di voto, una lista può partecipare a diverse sessioni di voto per questo la tabella di giunzione riguarda \_lista.
- **Ordinale\_partito & ordinale\_candidato**
  - Queste due tabelle non sono altro che delle tabelle di supporto definite per aiutare con la memorizzazione delle votazioni delle sessioni di tipo ordinale

### 3.3 Interfaccia Grafica

Di seguito un diagramma che spiega la navigazione grafica all'interno della applicazione



Nelle seguenti immagini saranno presenti esempi di partiti e nomi di politici.

La loro presenza è assolutamente di puro scopo illustrativo, **non vi è alcuna accezione / apologia concreta** in questo progetto.

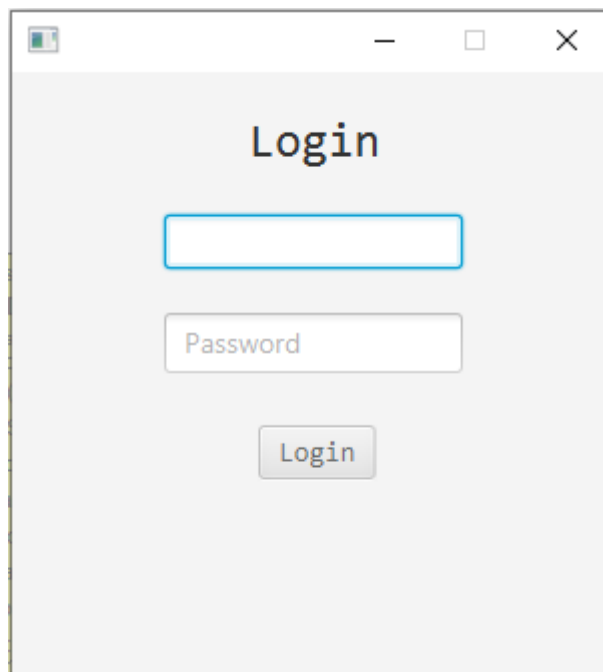


### 3.3.1 LoginView

Nella fase di login, come spiegato nei paragrafi precedenti, viene definito se un utente è un elettore oppure se è un gestore.

Questa fase risulta quindi cruciale per la definizione dei privilegi che un utente ha all'interno del sistema.

La schermata di login risulta quindi uguale per ambe le tipologie di utente, sarà poi il sistema ad attribuire all'utente che si sta loggando i privilegi, in base alle credenziali inserite.

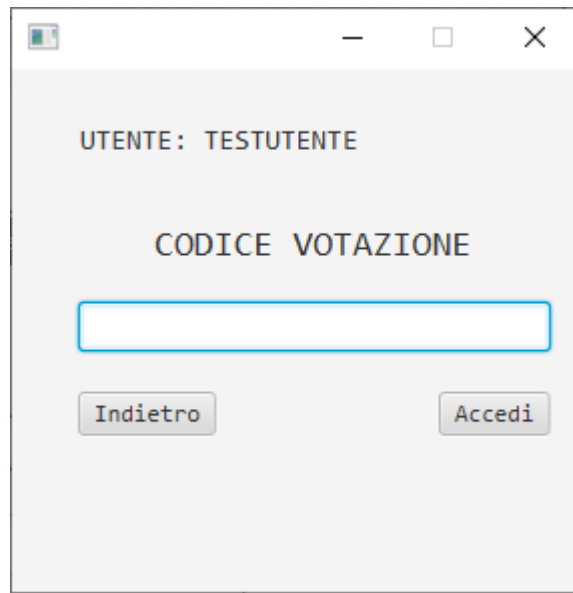
The image shows a screenshot of a web application window titled "Login". The window has a standard title bar with minimize, maximize, and close buttons. The main content area is light gray and contains the word "Login" in a large, bold, black font. Below the title, there are two input fields: the first is a simple text box, and the second is a text box with the placeholder text "Password" in a light gray font. Below these fields is a button labeled "Login" in a black font. The button has a slight shadow and a rectangular border.

### 3.3.2 CodiceVotazioneView

Questa view serve a permettere all'elettore di inserire il codice della votazione fornito.

il codice è composto da 12 caratteri numerici, nel caso di una sessione non valida l'utente verrà avvisato con un messaggio di errore. Stessa cosa vale nel caso di una sessione terminata o già votata dall'utente.

Nel caso queste condizioni non si verifichino, l'utente passerà alla schermata successiva dove potrà esprimere il suo voto.



UTENTE: TESTUTENTE

CODICE VOTAZIONE

Indietro Accedi

### 3.3.3 Vota view

Di seguito sono presenti le diverse view in base alla tipologia di sessione di voto in considerazione.

Come si può notare elementi comuni sono i tre bottoni indietro scheda bianca e avanti.

- **indietro** permette di tornare alla schermata di selezione sessione, non modifica in alcun modo lo stato della votazione. Ogni scelta effettuata, se si preme il bottone per tornare indietro verrà ignorata.
- **Scheda Bianca** permette all'utente di astenersi dal voto (vedi requisiti utente n. 5)
- **Avanti** permette di procedere con la votazione e quindi confermare il voto.



**Importante** dire che, come si può notare dallo schema relazionale della base dati, l'utente non è collegato in alcun modo al voto che esprime, mantenendo così segretezza, come da requisito di sicurezza ( paragrafo 1.3.2.1 )

#### 3.3.3.1 votaview (referendum)

The screenshot shows a web application window with a title bar containing standard window controls. The main content area has a light gray background. At the top, the text "ID SESSIONE: 000000000008" is displayed in a bold, black, sans-serif font. Below this, a white rectangular box contains the text "Referendum abrogativo: art. xyz-quater cod. pen." followed by the question "Favorevole all'abrogazione di questa legge?". Underneath the question, there are two radio button options: "Favorevole" (which is selected, indicated by a filled circle) and "Non Favorevole" (which is unselected, indicated by an empty circle). At the bottom of the window, there are three buttons: "Indietro" on the left, "SCHEDA BIANCA" in the center (highlighted with a blue border), and "AVANTI" on the right.

ID SESSIONE: 000000000008

Referendum abrogativo: art. xyz-quater cod. pen.  
Favorevole all'abrogazione di questa legge?

☒ Favorevole ☐ Non Favorevole

Indietro SCHEDA BIANCA AVANTI

### 3.3.3.2 votaview (categorica partito)

The screenshot shows a web application window with a title bar containing standard window controls. The main content area has a light gray background. At the top, the text "ID SESSIONE: 000000000003" is displayed in a bold, black, sans-serif font. Below this, there is a list of four political parties, each with a radio button: "EuropaVerde" (selected, indicated by a filled circle), "Forzitalia", "M5S", and "PartitoDemocratico". To the right of this list is a large, empty rectangular area with a vertical scrollbar on its right side. At the bottom of the window, there are three buttons: "Indietro" on the left, "SCHEDA BIANCA" in the center, and "AVANTI" on the right.

ID SESSIONE: 000000000003

☒ EuropaVerde  
☐ Forzitalia  
☐ M5S  
☐ PartitoDemocratico

Indietro SCHEDA BIANCA AVANTI

### 3.3.3.3 votaview (categorico candidato)

**ID SESSIONE: 000000000011**

<input type="radio"/> CANDIDATO: Matteo Salvini PARTITO: LegaNord	
<input type="radio"/> CANDIDATO: Jacopo Vignati PARTITO: LegaNord	
<input type="radio"/> CANDIDATO: Carlo Piastra PARTITO: LegaNord	
<input type="radio"/> CANDIDATO: Anakin Skywalker PARTITO: LegaNord	

Indietro **SCHEDA BIANCA** AVANTI

### 3.3.3.4 votaview (categorico con preferenza)

**ID SESSIONE: 000000000005**

☐ LegaNord ☒ M5S ☐ PRC

☐ Bruno Gaspare

☐ Guglielmo LaRosa

Indietro **SCHEDA BIANCA** AVANTI

### 3.3.3.5 votaview (ordinale partito)

**ID SESSIONE: 000000000024**

Ordine	EuropaVerde
Ordine	ForzItalia
Ordine	LegaNord
Ordine	MSS

Indietro SCHEDA BIANCA AVANTI

### 3.3.3.6 votaview (ordinale candidato)

**ID SESSIONE: 000000000025**

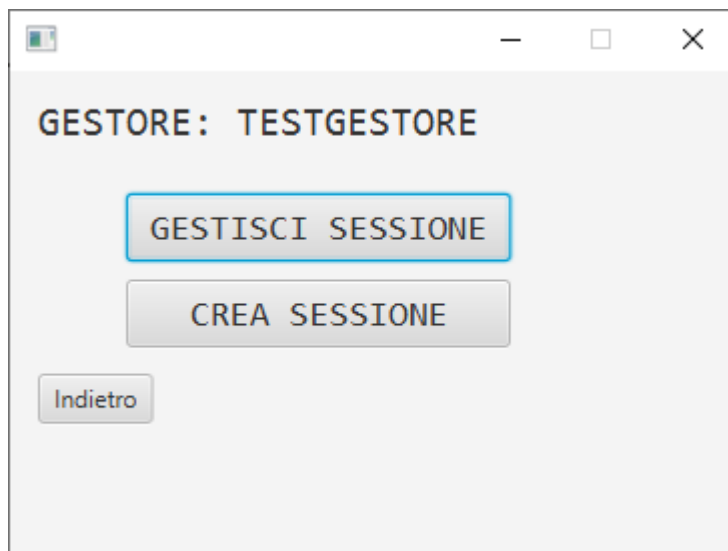
Ordine	Emma Bonino
Ordine	Leia Organa
Ordine	Antonio Tajani
Ordine	Santini Giovanni

Indietro SCHEDA BIANCA AVANTI

### 3.3.4 MainGestore

Questa view permette al gestore di scegliere cosa fare all'interno del sistema. ovviamente come definito nei diagrammi precedenti, il gestore può

- Gestire una sessione esistente (avviarla, fermarla o scrutinare)
- Creare una nuova sessione



### 3.3.5 GestisciSessione (con risultati sessione di voto)

Di seguito si presenta la finestra di gestione della sessione, come si può osservare vi sono tre tasti principali:

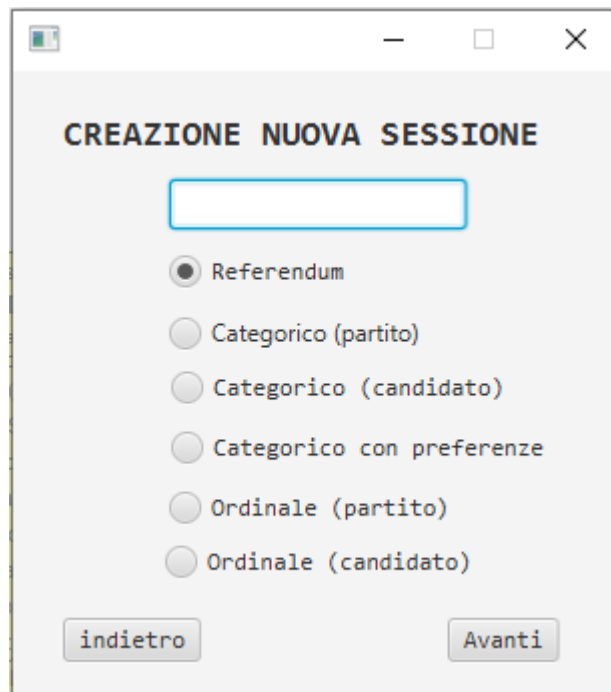
- **Ferma:** ferma la sessione. NB. la sessione non può essere fermata se prima non è stata attivata, questo viene gestito tramite i flag stato e isnew nella base dati. Nel caso si tentasse di terminare una sessione prima ancora di averla avviata, il sistema mostrerà un alert ed impedirà l'azione.
- **Attiva:** attiva la sessione. NB. la sessione non può essere attivata una volta che è stata fermata. Questa è una misura di sicurezza utilizzata per evitare che una eventuale sessione venga manomessa anche da un gestore. Come per il punto precedente, vengono utilizzati i flag stato e isnew della base dati per gestire questa problematica. Nel caso si verificasse una situazione del genere, il sistema mostra un alert ed impedisce l'azione.
- **Scrutinamento:** avvia la fase di scrutinamento e mostra i risultati nella box sottostante. NB. La fase di scrutinamento non può essere avviata se la sessione è ancora attiva. È necessario quindi prima chiudere la sessione e poi avviare la fase di scrutinamento. Nel caso si tentasse di scrutinare senza fermare la sessione, il sistema mostrerà un alert di errore ed impedirà l'azione.

The screenshot shows a software window titled "LISTA SESSIONI". At the top, there is a text input field containing the number "000000000001" with a dropdown arrow on the right. Below this are two buttons: "Ferma" and "Attiva". In the center, there is a button labeled "Scrutinamento" which is highlighted with a blue border. Below the buttons, the section is titled "Risultato". Inside this section, the text reads: "VINCITA: FAVOREVOLI", "[+] voti favorevoli: 3", and "[+] voti sfavorevoli: 2". At the bottom left of the window, there is a button labeled "Indietro". The window has standard OS window controls (minimize, maximize, close) in the top right corner.

### 3.3.6 NuovaSessione

Questa view permette al gestore di definire un nuovo codice e la tipologia di voto per la sessione che intende creare.

Nel caso la sessione fosse di un formato non valido o nel caso esistesse nella base dati, il sistema mostrerebbe al gestore un alert e successivamente impedirebbe l'azione.



**CREAZIONE NUOVA SESSIONE**

☒ Referendum

☐ Categorico (partito)

☐ Categorico (candidato)

☐ Categorico con preferenze

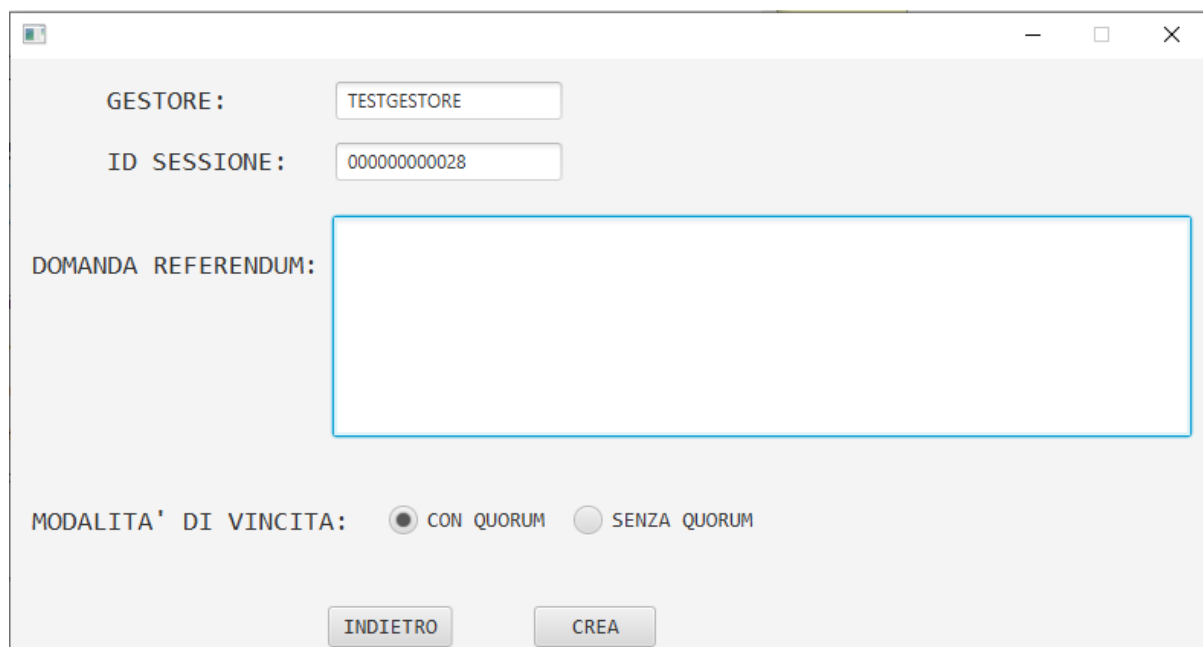
☐ Ordinale (partito)

☐ Ordinale (candidato)

### 3.3.7 NuovaSessione con tipologia selezionata

Una volta selezionata la tipologia e il codice della sessione, il gestore verrà presentato con una view diversa in base alla tipologia selezionata precedentemente.

#### 3.3.7.1 NuovaSessione (referendum selezionato)



**GESTORE:**

**ID SESSIONE:**

**DOMANDA REFERENDUM:**

**MODALITA' DI VINCITA:** ☒ CON QUORUM ☐ SENZA QUORUM

#### 3.3.7.2 NuovaSessione (Categorico Partito selezionato)



GESTORE: TESTGESTORE

ID SESSIONE: 000000000028

<input type="checkbox"/> Casapound	
<input type="checkbox"/> EuropaVerde	
<input type="checkbox"/> Forzitalia	
<input type="checkbox"/>	

modalità vincita ☒ maggioranza ☐ maggioranza assoluta

INDIETRO CREA

### 3.3.7.3 NuovaSessione (Categorico con preferenza selezionato)

GESTORE: TESTGESTORE

ID SESSIONE: 000000000028

<input type="checkbox"/> LISTA [ 3 ] ==> PARTITO=M5S, CANDIDATI={{Bruno Gaspere} [Guglielmo LaRosa] }	
<input type="checkbox"/> LISTA [ 4 ] ==> PARTITO=PartitoDemocratico, CANDIDATI={{Bibo Beggins} [Dario Parrini] [Tatjana Rojic] }	
<input type="checkbox"/> LISTA [ 2 ] ==> PARTITO=PRC, CANDIDATI={{Antonello Patta} [PennyWise TheClown] }	
<input type="checkbox"/>	

modalità vincita ☒ maggioranza ☐ maggioranza assoluta

INDIETRO CREA

### 3.3.7.4 NuovaSessione (Ordinale partito selezionato)

The screenshot shows a web application window with a title bar containing standard minimize, maximize, and close buttons. The main content area is divided into several sections:

- GESTORE:** A label followed by a text input field containing the value "TESTGESTORE".
- ID SESSIONE:** A label followed by a text input field containing the value "000000000028".
- Party Selection:** A vertical list of four items, each with an unchecked checkbox and a label:
  - ☐ ForzItalia
  - ☐ LegaNord
  - ☐ M5S
  - ☐ PartitoDemocratico
- Winning Mode:** A label "modalità vincita" followed by two radio button options:
  - ☒ maggioranza
  - ☐ maggioranza assoluta
- Buttons:** At the bottom, there are two buttons: "INDIETRO" (highlighted with a blue border) and "CREA".

### 3.3.7.5 NuovaSessione (Ordinale Candidato selezionato)

GESTORE: TESTGESTORE

ID SESSIONE: 000000000028

<input type="checkbox"/> Jeffrey Lannister
<input type="checkbox"/> Emma Bonino
<input type="checkbox"/> Elena Grandi

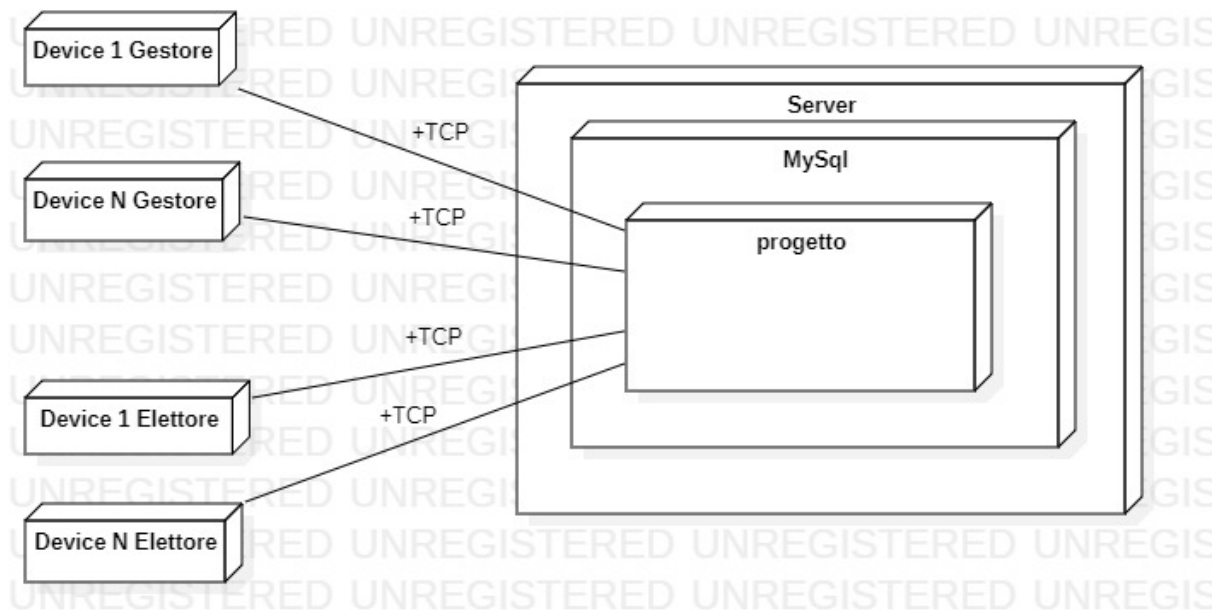
modalità vincita ☒ maggioranza ☐ maggioranza assoluta

INDIETRO CREA

### 3.4 Diagramma di Deployment

*“descrivere un sistema in termini di risorse hardware dette nodi e di relazioni fra di esse. Spesso si utilizza un diagramma che mostra come le componenti software siano distribuite rispetto alle risorse hardware disponibili sul sistema”*

Nel caso pratico del sistema di voto si vuole rappresentare come i dispositivi degli elettori e dei gestori di sistema comunichino con il sistema. Di seguito uno schema semplificato:



## 3.5 Specifica e verifica dei vincoli

*“**Java Modelling Language(JML)** è un linguaggio di specifica che permette di definire astrazioni procedurali su un modello di programmazione per contratto effettuando dei controlli sui parametri di ingresso di un metodo e sul suo valore di ritorno”*

Alcune specifiche JML sono presenti nei seguenti metodi:

### 3.5.1 UtenteJDBCDAO

1. **{context** UtenteJDBCDAO::login(usr:UserBean)  
**pre:** *usr.getCodiceFiscale() != null && usr.getPassword() != null && usr != null;*
2. **{context** UtenteJDBCDAO::get(id:String)  
**pre:** *requires id!=null*

### 3.5.2 VotazioneJDBCDAO

1. **{context** VotazioneJDBCDAO::save(t:VotoGeneric)  
**pre:** *ensures t!=null*
2. **{context** VotazioneJDBCDAO::aggiornaPartito(par:Partito,IDSession:String)

```
pre: par!=null && par.getNome()!=null&&IDSessione!=null}
```

### 3.5.3 SessioneJBCDAO

#### 1. {context SessioneDiVoto

**inv:** self.tipologiaVoto.equals("referendum") implies self.domandaReferendum  
<> null

**inv:** self.IDSessione <> null

**inv:** self.strategiaVincitore <> null

**inv:** self.tipologiaVoto <> null

**inv:** self.stato <> null

**inv:** self.isnew <> null

**inv:** self.status <> null }

---

## 3.6 Testing

Per quanto riguarda il testing, è possibile testare le applicazioni accedendo con le credenziali al sistema di voto presenti nel paragrafo 3.7.1

Sono stati inoltre effettuati dei test per quanto riguarda database e funzionamento generale dell'applicazione con JUNIT

### 3.6.1 Test Database

```

class DatabaseTest {
    @BeforeAll
    public static void setup() {
        DBManager.getInstance().openConnection();
    }

    @Test
    void testLogin() {
        UtenteJDBCDAO usr=new UtenteJDBCDAO();
        ABSUtente log=usr.login(new UserBean("TESTGESTORE",SHA256.hash("test1234")));
        assertEquals(true,log.isGestore());
    }
    @Test
    void testSessioneExists() {
        SessioneJDBCDAO sess=new SessioneJDBCDAO();
        assertNotNull(sess.get("000000000001"));
    }
    @Test
    void testCandidatiPartito() {
        SessioneJDBCDAO sess=new SessioneJDBCDAO();
        assertNotNull(sess.getListaCandidatiFromPartito(new Partito("EuropaVerde",123)));
    }
    @Test
    void creaSessioneEsistente() {
        SessioneJDBCDAO sess=new SessioneJDBCDAO();
        SessioneVoto sv=new SessioneVoto("000000000001", null, null, null, 0, 0, false, false);
        Exception exception = assertThrows(DBSQLException.class, () -> sess.save(sv));
        assertEquals("SQL ERROR", exception.getMessage());
    }
}

```

### 3.6.2 Test Generici

```

@Test
void UserTests() {
    ABSUtente abs=new Gestore("test","test");
    assertFalse(abs.isElettore());
    assertTrue(abs.isGestore());
}
@Test
void VotazioniTests() {
    VotoGeneric v=new VotoReferendum(true,null);
    System.out.println(v.getTipo());
    assertEquals("referendum",v.getTipo());
}
@Test
void cfparsertest() {
    CodiceFiscale cf=new CodiceFiscale("LMRJBI48R13H123M");
    assertTrue(cf.validate());
}

```

## 3.7 Installazione

### 3.7.1 Informazioni di sviluppo

- **IDE:** Eclipse 2021-09 (4.21.0)
- **O.S:** Windows10
- **JDK:** OpenJDK 11 (2018-09-25)
- **Opzioni VM:** --module-path "%PATH\_TO\_FX%;%PATH\_TO\_FX%\lib" --addmodules javafx.controls,javafx.fxml,javafx.graphics
- **Classe principale:** progetto.App

### 3.7.2 Esecuzione programma

Da **Eclipse** la configurazione di run deve avere come goal javafx:run.

Tutto il resto è configurato nel file pom.

### 3.7.3 credenziali di test

Di seguito le credenziali per accesso al sistema di voto:

- **GESTORE** —> username: TESTGESTORE  
password: test1234

- **ELETTORE** —> username: TESTELETTORE  
password: test1234

## 3.8 Informazioni generali sull'applicazione

Insieme al progetto viene allegato un file chiamato metrics.xml, esso permette di vedere tutte le informazioni ricavate dal progetto tramite il plugin metrics.

*This plug in calculates various metrics for your code during build cycles and warns you, via the Problems View, of 'range violations' for each metric. This allows you to stay continuously aware of the health of your code base.*

di seguito alcuni screenshot della *metric view* sull'app

Metric	Total	Mean	Std. Dev.	Maxim...	Resource causing Maximum	Method
> McCabe Cyclomatic Complexity (avg/max per method)		1,773	1,844	21	/Elezioni_Scrutinamenti_2022/src/main/java/progetto/DA...	save
> Number of Parameters (avg/max per method)		0,717	0,897	8	/Elezioni_Scrutinamenti_2022/src/main/java/progetto/mo...	SessioneVoto
> Nested Block Depth (avg/max per method)		1,36	0,878	5	/Elezioni_Scrutinamenti_2022/src/main/java/progetto/DA...	save
> Afferent Coupling (avg/max per packageFragment)		7,682	9,102	38	/Elezioni_Scrutinamenti_2022/src/main/java/progetto/mo...	
> Efferent Coupling (avg/max per packageFragment)		2,227	2,485	9	/Elezioni_Scrutinamenti_2022/src/main/java/progetto/con...	
> Instability (avg/max per packageFragment)		0,414	0,379	1	/Elezioni_Scrutinamenti_2022/src/main/java/progetto/con...	
> Abstractness (avg/max per packageFragment)		0,264	0,313	1	/Elezioni_Scrutinamenti_2022/src/main/java/progetto/DAO	
> Normalized Distance (avg/max per packageFragment)		0,329	0,345	1	/Elezioni_Scrutinamenti_2022/src/main/java/progetto/exc...	
> Depth of Inheritance Tree (avg/max per type)		1,486	0,89	4	/Elezioni_Scrutinamenti_2022/src/main/java/progetto/exc...	
> Weighted methods per Class (avg/max per type)	695	9,929	11,76	82	/Elezioni_Scrutinamenti_2022/src/main/java/progetto/DA...	
> Number of Children (avg/max per type)	40	0,571	1,871	12	/Elezioni_Scrutinamenti_2022/src/main/java/progetto/con...	
> Number of Overridden Methods (avg/max per type)	15	0,214	0,652	3	/Elezioni_Scrutinamenti_2022/src/main/java/progetto/mo...	
> Lack of Cohesion of Methods (avg/max per type)		0,249	0,349	1	/Elezioni_Scrutinamenti_2022/src/main/java/progetto/con...	
> Number of Attributes (avg/max per type)	155	2,214	2,971	15	/Elezioni_Scrutinamenti_2022/src/main/java/progetto/con...	
> Number of Static Attributes (avg/max per type)	18	0,257	0,805	4	/Elezioni_Scrutinamenti_2022/src/main/java/progetto/DA...	
> Number of Methods (avg/max per type)	374	5,343	4,855	31	/Elezioni_Scrutinamenti_2022/src/main/java/progetto/DA...	
> Number of Static Methods (avg/max per type)	18	0,257	0,966	7	/Elezioni_Scrutinamenti_2022/src/main/java/progetto/test...	
> Specialization Index (avg/max per type)		0,033	0,13	1	/Elezioni_Scrutinamenti_2022/src/main/java/progetto/App...	
> Number of Classes (avg/max per packageFragment)	70	3,182	3,025	12	/Elezioni_Scrutinamenti_2022/src/main/java/progetto/utis	

Metric	Total	Mean	Std. Dev.	Maxim...	Resource causing Maximum	Method
> Instability (avg/max per packageFragment)		0,414	0,379	1	/Elezioni_Scrutinamenti_2022/src/main/java/progetto/con...	
> Abstractness (avg/max per packageFragment)		0,264	0,313	1	/Elezioni_Scrutinamenti_2022/src/main/java/progetto/DAO	
> Normalized Distance (avg/max per packageFragment)		0,329	0,345	1	/Elezioni_Scrutinamenti_2022/src/main/java/progetto/exc...	
> Depth of Inheritance Tree (avg/max per type)		1,486	0,89	4	/Elezioni_Scrutinamenti_2022/src/main/java/progetto/exc...	
> Weighted methods per Class (avg/max per type)	695	9,929	11,76	82	/Elezioni_Scrutinamenti_2022/src/main/java/progetto/DA...	
> Number of Children (avg/max per type)	40	0,571	1,871	12	/Elezioni_Scrutinamenti_2022/src/main/java/progetto/con...	
> Number of Overridden Methods (avg/max per type)	15	0,214	0,652	3	/Elezioni_Scrutinamenti_2022/src/main/java/progetto/mo...	
> Lack of Cohesion of Methods (avg/max per type)		0,249	0,349	1	/Elezioni_Scrutinamenti_2022/src/main/java/progetto/con...	
> Number of Attributes (avg/max per type)	155	2,214	2,971	15	/Elezioni_Scrutinamenti_2022/src/main/java/progetto/con...	
> Number of Static Attributes (avg/max per type)	18	0,257	0,805	4	/Elezioni_Scrutinamenti_2022/src/main/java/progetto/DA...	
> Number of Methods (avg/max per type)	374	5,343	4,855	31	/Elezioni_Scrutinamenti_2022/src/main/java/progetto/DA...	
> Number of Static Methods (avg/max per type)	18	0,257	0,966	7	/Elezioni_Scrutinamenti_2022/src/main/java/progetto/test...	
> Specialization Index (avg/max per type)		0,033	0,13	1	/Elezioni_Scrutinamenti_2022/src/main/java/progetto/App...	
> Number of Classes (avg/max per packageFragment)	70	3,182	3,025	12	/Elezioni_Scrutinamenti_2022/src/main/java/progetto/utis	
> Number of Interfaces (avg/max per packageFragment)	7	0,318	0,466	1	/Elezioni_Scrutinamenti_2022/src/main/java/progetto/DAO	
> Number of Packages	22					
> Total Lines of Code	4132					
> Method Lines of Code (avg/max per method)	2389	6,094	10,966	156	/Elezioni_Scrutinamenti_2022/src/main/java/progetto/DA...	save