Immagine che contiene testo

Descrizione generata automaticamente

Sistema di voto elettronico

Progetto di Ingegneria del software

Autori:

Umberto Pirovano

Mattia Garavaglia 923482

A.A. 2021/2022

Sommario

[1 Descrizione del problema 4](#_Toc114144340)

[1.1 Analisi dei requisiti 4](#_Toc114144341)

[1.1.1 Requisiti funzionali 4](#_Toc114144342)

[1.1.2 Requisiti non funzionali 6](#_Toc114144343)

[1.2 Glossario 6](#_Toc114144344)

[2 Progettazione del sistema 8](#_Toc114144345)

[2.1 Diagramma dei casi d’uso 8](#_Toc114144346)

[2.2 Descrizione degli scenari 9](#_Toc114144347)

[2.3 Diagrammi di sequenza 13](#_Toc114144348)

[2.4 Diagrammi delle attività 18](#_Toc114144349)

[2.5 Macchine di stato 20](#_Toc114144350)

[2.6 Diagramma dei componenti 21](#_Toc114144351)

[3 Implementazione del sistema 21](#_Toc114144352)

[3.1 Diagramma delle classi 21](#_Toc114144353)

[3.1.1 Discussione dei design pattern utilizzati 24](#_Toc114144354)

[3.2 Gestione dei dati persistenti 25](#_Toc114144355)

[3.3 Descrizione dell’interfaccia grafica 27](#_Toc114144356)

[3.4 Diagramma di deployment 36](#_Toc114144357)

[3.5 Specifica e verifica dei vincoli 37](#_Toc114144358)

[3.6 Descrizione del testing 39](#_Toc114144359)

Elenco delle figure

[Figura 1 Utente effettua login 13](#_Toc114144360)

[Figura 2 Utente effettua logout 14](#_Toc114144361)

[Figura 3 Elettore vota 14](#_Toc114144362)

[Figura 4 Amministratore visualizza risultati votazione 15](#_Toc114144363)

[Figura 5 Amministratore crea nuovo utente 16](#_Toc114144364)

[Figura 6 Amministratore crea nuova votazione 17](#_Toc114144365)

[Figura 7 Utente effettua login 18](#_Toc114144366)

[Figura 8 Utente effettua logout 18](#_Toc114144367)

[Figura 9 Elettore vota 19](#_Toc114144368)

[Figura 10 Amministratore visualizza i risultati di una votazione terminata 19](#_Toc114144369)

[Figura 11 Macchina di stato di un utente 20](#_Toc114144370)

[Figura 12 Macchina di stato di una votazione 20](#_Toc114144371)

[Figura 13 Diagramma delle classi modello 22](#_Toc114144372)

[Figura 14 Diagramma delle classi Observer 23](#_Toc114144373)

[Figura 15 Diagramma delle classi DAO 24](#_Toc114144374)

[Figura 16 Tabella user 25](#_Toc114144375)

[Figura 17 Tabella poll 25](#_Toc114144376)

[Figura 18 Tabella vote 26](#_Toc114144377)

[Figura 19 Tabella vote\_register 26](#_Toc114144378)

[Figura 20 Tabella party 26](#_Toc114144379)

[Figura 21 Tabella candidate 26](#_Toc114144380)

[Figura 22 Tabella log 27](#_Toc114144381)

[Figura 23 Schermata di login 27](#_Toc114144382)

[Figura 24 Schermata di selezione votazione per l' elettore 28](#_Toc114144383)

[Figura 25 Schermata di voto per un referendum 29](#_Toc114144384)

[Figura 26 Schermata dei menù dell'amministratore 30](#_Toc114144385)

[Figura 27 Schermata per la gestione degli utenti 30](#_Toc114144386)

[Figura 28 Schermata di gestione delle votazioni 31](#_Toc114144387)

[Figura 29 Schermata dei risultati di una votazione 32](#_Toc114144388)

[Figura 30 Schermata di configurazione per un referendum 32](#_Toc114144389)

[Figura 31 Schermata di configurazione per una votazione ordinale 33](#_Toc114144390)

[Figura 32 Schermata di inserimento partiti e candidati per votazione ordinale o categorica 34](#_Toc114144391)

[Figura 33 Schermata di configurazione di una votazione categorica 34](#_Toc114144392)

[Figura 34 Schermata dei log 35](#_Toc114144393)

[Figura 35 Schermata di notifica di un errore 36](#_Toc114144394)

[Figura 36 Diagramma di deployment 36](#_Toc114144395)

# Descrizione del problema

Questo sistema è una piattaforma completamente imparziale che permette di esprimere il proprio voto elettronicamente in maniera del tutto anonima. La sua adattabilità in termini di modalità previste gli permette di essere adeguatamente applicato anche in contesti di voto diversi ed è infatti previsto il supporto a: votazioni ordinali, categoriche, categoriche con preferenze e referendum. A questo si aggiunge la possibilità di definire criteri di valutazione del vincitore differenti quali: maggioranza, maggioranza assoluta, referendum senza quorum e referendum con quorum.

L’accesso è garantito, tramite identità digitale, a tutti gli aventi diritto al voto. A questi, una vota effettuato il login, viene offerta un’esperienza il più possibile analoga a quella del voto tradizionale senza alcun tipo di influenza e senza che sia possibile risalire all’associazione tra elettore e dati sensibili quali la preferenza espressa.

Per gli attori autorizzati dalla legge o dagli organi ufficiali incaricati dell’organizzazione di voto è inoltre possibile accedere a funzionalità privilegiate che permettano la creazione e la gestione di votazioni.

## Analisi dei requisiti

### Requisiti funzionali

#### Attori

**Amministratore**: individuo o organizzazione che si occupa della manutenzione e dell’aggiornamento del sistema creando e gestendo utenti, votazione e i partiti con relativi i candidati. Può effettuare login al sistema tramite la modalità amministratore.

**Elettore**: è un cittadino con diritto di voto. Una volta effettuato il login in modalità utente può consultare le elezioni disponibili e esprimere la propria preferenza nelle votazioni in corso alle quali non ha ancora preso parte.

**Utente non loggato**: si tratta di un individuo a cui non è stato ancora attribuito un ruolo nel sistema, il quale gli verrà associato in fase di login.

#### Requisiti utente

Utente non loggato:

1. Effettuare il login
2. Permettere l’accesso al sistema in un seggio elettorale. La fase di riconoscimento sarà delegata agli addetti del seggio secondo le tradizionali modalità (CI) e non verranno richieste le credenziali di accesso all’utente.

Amministratore:

1. Creare una nuova votazione.
2. Eliminare le votazioni già esistenti, ma non ancora in corso.
3. Gestire gli utenti del sistema.
4. Visualizzare l’esito di una votazione terminata.
5. Aggiungere partiti e i loro candidati ad una votazione.
6. Effettuare il logout.

Elettore:

1. Selezionare una tra le elezioni in corso.
2. Effettuare la votazione.
3. Effettuare il logout.

#### Requisiti di sistema

1. Il sistema mette a disposizione tramite la sua interfaccia utente di login una form nella quale l’utente non loggato può inserire le credenziali per l’accesso.
2. Il sistema mette a disposizione dei meccanismi tramite i quali è in grado di acquisire da un amministratore e memorizzare informazioni riguardanti una votazione, quali nome, modalità di voto, modalità di proclamazione del vincitore, date di inizio e fine etc.
3. Il sistema mette a disposizione dei meccanismi tramite i quali un amministratore è in grado di eliminare una votazione non ancora iniziate.
4. Il sistema dà la possibilità ad un amministratore di creare nuovi utenti.
5. Il sistema dà la possibilità ad un amministratore di eliminare utenti dal sistema.
6. Il sistema deve tenere traccia dei voti espressi dagli elettori, in modo da poter mostrare, a votazione conclusa, l’esito.
7. Il sistema mette a disposizione un modo per uscire dallo stesso.
8. Il sistema permette, all’atto della creazione di una votazione, l’aggiunta di partiti e i loro candidati ad essa.
9. Il sistema deve mostrare all’elettore la lista delle votazioni in corso.
10. Il sistema non deve permettere ad un elettore di votare per una votazione a cui ha già partecipato.
11. Il sistema deve permettere all’elettore di selezionare una delle votazioni mostrate.
12. Nel caso di voto ordinale, il sistema deve permettere all’elettore di stabilire un ordine di preferenze dei candidati.
13. Nel caso di voto categorico, il sistema deve permettere all’elettore di scegliere uno tra i candidati presenti.
14. Nel caso di voto categorico con preferenze, il sistema deve permettere all’elettore di scegliere un gruppo/partito e degli eventuali candidati.
    1. Il sistema deve permettere all’elettore di selezionare candidati solo dal gruppo/partito che ha scelto.
    2. Il sistema deve permettere all’elettore di scegliere al massimo due candidati.

### Requisiti non funzionali

#### Requisiti di sicurezza

Il sistema deve essere dotato di una modalità di **autenticazione** username – password, atta a identificare l’utente dopo che quest’ultimo ha selezionato il suo ruolo nel sistema (elettore o amministratore).

All’atto di creazione di un utente, la sua **password deve essere cifrata** e memorizzata nel database in modo da poter essere riutilizzata per i confronti in fase di login. Per quanto riguarda la **segretezza del voto**, essi devono essere memorizzati sul database in modo disgiunto rispetto all’elettore che li ha espressi, memorizzando l’associazione votazione – elettore e voto – votazione in due tabelle diverse.

Il sistema deve implementare inoltre un meccanismo di **logging** per registrare le attività svolte sul sistema da amministratori ed elettori.

#### Requisiti di qualità del software

Il software deve essere scritto garantendo un certo livello di **leggibilità** e deve essere **documentato adeguatamente** in modo da facilitarne la comprensione e la manutenibilità da parte di team diversi da quello che lo ha sviluppato inizialmente. Oltre a questo, si aggiunge anche la **modularità** su cui dovrebbe essere basata la sua struttura e che ne garantisce verificabilità, riparabilità, estensibilità e riusabilità.

In particolare, il sistema viene implementato tramite un linguaggio che segue il paradigma **object oriented**. È perciò necessario che vengano rispettati i principi di **alta coesione** e **lasco accoppiamento** tra le classi del sistema, al fine di ottenere un maggior controllo e tracciabilità di chi può svolgere determinate operazioni e chi no.

#### Requisiti di portabilità

Deve essere possibile accedere al sistema da smartphone, tablet, laptop e dai device abitualmente utilizzati. Al fine di permettere che ciò avvenga è necessario scegliere un linguaggio che garantisca la portabilità su sistemi diversi, come Java.

#### Opensource

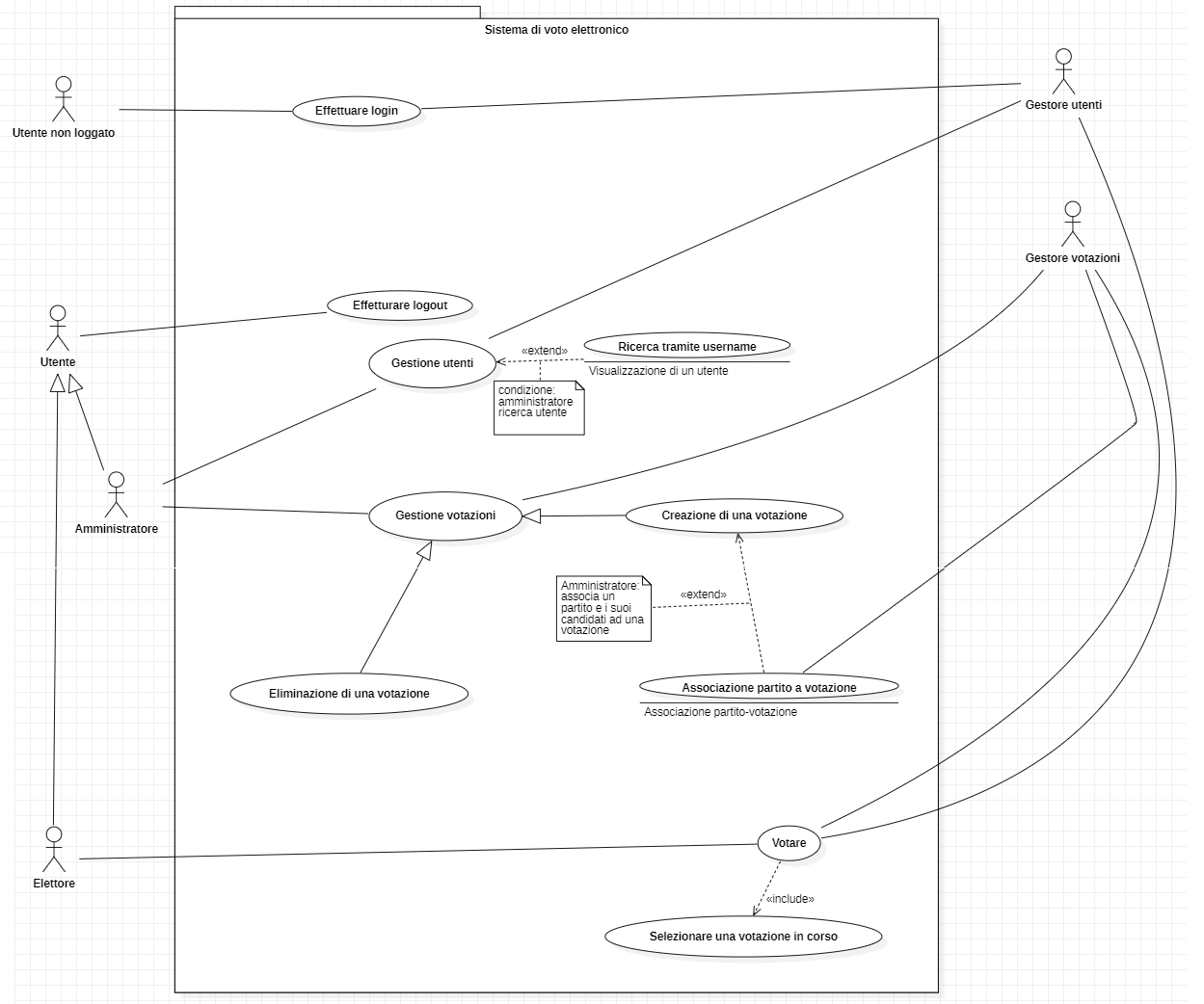
Al fine di poter verificare le caratteristiche precedentemente esposte, è necessario rendere disponibile il codice sorgente del sistema in modo pubblico. Ciò permette anche l’individuazione di eventuali falle nella sicurezza del sistema e la loro tempestiva segnalazione, con conseguente correzione.

## Glossario

* Audit: procedimento che registra le attività significative svolte in un sistema da parte dei vari attori che possono utilizzarlo.
* CI: abbreviazione per carta d’identità.
* CF: abbreviazione di codice fiscale.
* Form: è una componente dell’interfaccia utente. Nel nostro caso utilizziamo un form di login, in cui l’utente va ad inserire username e password, ma anche form nella fase di creazione votazioni, partiti, candidati o utenti da parte di un amministratore del sistema.
* Log: lista degli accessi e delle operazioni svolte sulla piattaforma da parte di membri del seggio, amministratori ed elettori.
* Maggioranza assoluta: il vincitore è il candidato che ha ottenuto la maggioranza assoluta dei voti, cioè il 50% + 1 dei voti espressi.
* Maggioranza (semplice): il vincitore è il candidato che ha ottenuto il maggior numero di voti.
* Menù a tendina: componente dell’interfaccia utente. Se premuto permette una visualizzazione a cascata di vari elementi dei quali solamente uno può essere selezionato. Ad esempio, nel caso di un amministratore che deve andare a selezionare la modalità di voto il menù a tendina conterrà: voto ordinale, voto categorico, voto categorico con preferenza.
* Modalità tradizionale: voto esercitato in presenza presso il proprio seggio elettorale.
* Referendum: consiste in una domanda fatta all’elettorato con la quale si chiede se si sia favorevoli o contrari a un determinato quesito.
* Referendum con quorum: si procede al conteggio dei voti espressi solo nel caso in cui abbia partecipato alla consultazione la maggioranza degli aventi diritto al voto.
* Referendum senza quorum: si procede al conteggio dei voti indipendentemente se abbia partecipato o meno alla consultazione la maggioranza degli aventi diritto al voto.
* Voto categorico: l’elettore inserisce una preferenza per un candidato (o gruppo/partito).
* Voto categorico con preferenze: l’elettore inserisce una preferenza per un gruppo/partito e ha la possibilità di indicare una o più preferenze tra i candidati del gruppo/partito selezionato (niente voto disgiunto).
* Voto ordinale: all’elettore è richiesto di ordinare i candidati (o gruppi/partiti) presenti nella scheda in base alle proprie preferenze.
* Server: con server si intende un’entità software che si occupa di fornire servizi richiesti da terzi, detti client. Nel nostro caso i client si dovranno mettere in contatto con il server per comunicargli i propri voti e permettere così al sistema di ottenere i risultati della votazione.
* Stato delle votazioni: insieme di dati e statistiche che descrivono una votazione (es: data e ora inizio, data e ora fine) e il suo andamento (es: percentuale di votanti).
* Identità digitale: credenziali che possono essere username con password atte ad autenticare un utente in un sistema. Nel nostro caso l’identità digitale è composta di codice fiscale e password per l’elettore, mentre da username e password per l’amministratore.
* Autenticazione: processo tramite il quale il sistema verifica l’identità del soggetto che vuole accedere al sistema. Nel caso in cui esso venga riconosciuto come utente del sistema, gli verranno associati permessi operativi e le limitazioni del caso.
* Database: un database è un insieme di informazioni (o dati) strutturate in genere archiviate elettronicamente in un sistema informatico.
* Tabella: con riferimento ad un database, rappresenta una struttura dati contenuta in un database caratterizzata da diversi campi (colonne) e record (righe). Ogni riga della tabella è univoca, ovvero ha valori diversi nei campi che sono indicati come identificatori di quella specifica tabella (chiavi primarie).
* Object oriented: paradigma di programmazione che si basa sugli oggetti, entità che possono descriverne altre (reali o astratte) attraverso uno stato e dei comportamenti.
* Classe: con riferimento alla programmazione object oriented, si riferiscono a elementi nei quali è strutturato un progetto object oriented.
* Alta coesione: una classe è altamente coesa se contiene operazioni e conoscenze strettamente correlate tra di loro.
* Lasco accoppiamento: due o più classi rispettano il principio di lasco di accoppiamento se la conoscenza dell’implementazione di una classe da parte di un'altra è minima o addirittura nulla.
* DB: abbreviazione di database.

# Progettazione del sistema

## Diagramma dei casi d’uso



* **Effettuare login**: caso d’uso nel quale l’utente non loggato presenta le proprie credenziali al sistema con lo scopo di accedere ai servizi da esso forniti. Il sistema fa affidamento sul gestore degli utenti per validare le credenziali fornite.
* **Effettuare logout**: caso d’uso nel quale un utente che ha accesso ai servizi del sistema decide di uscire dalla sessione in corso.
* **Votare**: caso d’uso nel quale un elettore esprime la propria preferenza su una delle votazioni corso. Il gestore delle votazioni, insieme al gestore degli utenti, terrà traccia dell’avvenuta votazione da parte di un elettore.
* **Gestione utenti**: caso d’uso nel quale un amministratore crea o elimina gli utenti del sistema.
* **Gestione votazioni**: caso d’uso nel quale un amministratore crea o elimina votazioni dal sistema.
* **Selezionare una votazione in corso**: caso d’uso nel quale un elettore seleziona una votazione dalla lista delle votazioni in corso per poter esprimere la sua preferenza. Le votazioni in corso vengono fornite al sistema dal gestore delle votazioni.
* **Ricerca tramite username**: caso d’uso nel quale un amministratore cerca un utente al sistema tramite lo username di quest’ultimo per poterlo eliminare dal sistema**.**
* **Associazione partito a votazione**: caso d’uso nel quale un amministratore aggiunge ad una votazione non ancora iniziata partiti e/o candidati.
* **Creazione di una votazione**: caso d’uso nel quale un amministratore crea una nuova votazione a patto che questa non sia già stata registrata dal sistema.
* **Eliminazione di una votazione**: caso d’uso nel quale un amministratore elimina una votazione dal sistema a patto che questa non sia già iniziata o terminata.

## Descrizione degli scenari

|  |  |
| --- | --- |
| Nome | Effettua login |
| Scopo | Accedere al sistema |
| Attore | Utente non loggato |
| Pre-condizioni | L’utente non deve essere loggato nel sistema. |
| Post-condizioni | L’utente accede al sistema. |
| Errori | Le credenziali fornite dall’utente non vengono riconosciute dal sistema. |
| Comportamento del sistema in caso d’errore | Il sistema non permette l’accesso e notifica l’errore all’utente non loggato. |
| Trigger | L’utente non loggato cerca di accedere al sistema. |
| Descrizione | Permette all’utente non loggato di accedere ai servizi del sistema. |
| Sequenza eventi | L’utente inserisce le credenziali e gli viene concesso l’accesso al sistema. |
| Alternativa | L’utente non loggato si reca ad un seggio, dove viene riconosciuto come elettore e gli viene concesso l’accesso ai servizi del sistema. |

Requisiti:

* Utente: Utente non logga nr. 1,2.
* Sistema: nr. 1.

|  |  |
| --- | --- |
| Nome | Votare |
| Scopo | Permettere all’elettore di esprimere la propria preferenza nella votazione selezionata |
| Attore | Elettore |
| Pre-condizioni | L’elettore non deve aver già votato per la votazione selezionata. |
| Post-condizioni | La preferenza espressa dall’utente viene registrata dal sistema. |
| Errori | L’elettore ha già espresso la sua preferenza nella votazione selezionata. |
| Comportamento del sistema in caso d’errore | Il sistema non permette l’accesso da parte di un elettore alla votazione per la quale è già stata espressa la preferenza. |
| Trigger | L’elettore vuole esprimere la propria preferenza in una votazione. |
| Descrizione | Permettere all’elettore di esprimere la preferenza nella votazione selezionata. |
| Sequenza eventi | * L’utente non loggato effettua l’accesso come elettore * Seleziona la votazione tra quelle disponibili * Esprime la propria preferenza |
| Alternativa | L’elettore si reca in un seggio elettorale dove verrà riconosciuto come elettori dagli addetti del seggio, i quali, oltre a garantirgli l’accesso al sistema, si assicureranno anche che l’elettore non abbia già espresso il voto per la votazione indicata. |

Requisiti:

* Utente: (Elettore) nr. 1, 2.
* Sistema: nr. 10, 11, 12, 13, 14.

|  |  |
| --- | --- |
| Nome | Effettua il logout |
| Scopo | Effettuare il logout dal sistema |
| Attore | Utente |
| Pre-condizioni | L’utente deve essere loggato nel sistema. |
| Post-condizioni | L’utente torna allo stato di utente non loggato ed alla fase di login. |
| Trigger | L’utente vuole scollegarsi dal sistema. |
| Descrizione | Permette all’utente di effettuare il logout dal sistema. |
| Sequenza eventi | * L’utente non loggato accede al sistema * Effettua le operazioni richieste * Seleziona dalla gui la possibilità di fare logout * Si scollega dal sistema |
| Alternativa | Nessuna. |

Requisiti:

* Utente:
  + Elettore: nr. 3.
  + Amministratore: nr. 6.
* Sistema: nr. 7.

|  |  |
| --- | --- |
| Nome | Creare una votazione |
| Scopo | Registrare nel sistema una nuova votazione. |
| Attore | Amministratore |
| Pre-condizioni | Nel sistema non deve essere presente una votazione con lo stesso nominativo di quella che si vuole inserire. Inoltre, la data d’inizio della nuova votazione deve essere successiva a quella attuale. |
| Post-condizioni | La votazione viene registrata dal sistema. |
| Errori | * Una votazione con lo stesso nominativo è già stata registrata dal sistema. * La data d’inizio della votazione non successiva a quella attuale. |
| Comportamento del sistema in caso d’errore | In entrambi i casi d’errore, viene impedito l’inserimento della votazione e all’amministratore viene notificata la causa dell’errore. |
| Trigger | Un amministratore vuole inserire una nuova votazione. |
| Descrizione | Permettere ad un amministratore di registrare una nuova votazione nel sistema. |
| Sequenza eventi | * L’utente non loggato accede come amministratore. * L’amministratore crea la votazione dopo aver inserito nome, data inizio e fine tipologia di votazione. |
| Alternativa | Nessuna |

Requisiti:

* Utente: (Amministratore) nr. 1.
* Sistema: nr. 2.

|  |  |
| --- | --- |
| Nome | Eliminare una votazione |
| Scopo | Eliminare una votazione registrata sul. |
| Attore | Amministratore |
| Pre-condizioni | La votazione che si vuole eliminare non deve essere iniziata. |
| Post-condizioni | La votazione viene eliminata dal sistema. |
| Errori | La votazione è già iniziata. |
| Comportamento del sistema in caso d’errore | Viene impedita l’eliminazione della votazione e all’amministratore viene notificata la causa dell’errore. |
| Trigger | Un amministratore vuole eliminare una votazione. |
| Descrizione | Permettere ad un amministratore di eliminare una votazione dal sistema. |
| Sequenza eventi | * L’utente non loggato accede come amministratore. * L’amministratore elimina la votazione dal sistema. |
| Alternativa | Nessuna |

Requisiti:

* Utente: (Amministratore) nr. 2.
* Sistema: nr. 3.

|  |  |
| --- | --- |
| Nome | Visualizza votazioni in corso |
| Scopo | Visualizzare le votazioni corso nel sistema. |
| Attore | Elettore |
| Post-condizioni | L’elettore visualizza l’elenco elenco delle votazioni in corso nel sistema. |
| Trigger | L’elettore vuole visualizzare la lista delle votazioni in corso nel sistema. |
| Descrizione | Permette all’elettore di visualizzare una lista delle votazioni in corso nel sistema. |
| Sequenza eventi | * L’utente non loggato effettua l’accesso come elettore * Seleziona tramite GUI la sezione relativa alle votazioni in corso * Visualizza l’elenco delle votazioni in corso |
| Alternativa | Nessuna |

Requisiti:

* Utente: (Elettore) nr. 1.
* Sistema: nr. 9.

|  |  |
| --- | --- |
| Nome | Gestione degli utenti |
| Scopo | Gestire gli utenti registrati nel sistema, permettendone l’eliminazione o l’aggiunta. |
| Attore | Amministratore |
| Precondizioni | In caso di aggiunta di un nuovo utente, nel sistema non deve essere presente un altro utente con lo stesso nominativo. |
| Post-condizioni | A seconda del caso viene aggiunto o eliminato un utente. |
| Errori | L’utente che si vuole aggiungere è già presente nel sistema. |
| Comportamento del sistema in caso d’errore | Non viene permesso l’inserimento dell’utente e all’amministrato viene notificato il motivo dell’errore. |
| Trigger | L’amministratore vuole aggiungere/rimuovere un utente dal sistema. |
| Descrizione | Permette all’amministratore di aggiungere o eliminare utenti dal sistema. |
| Sequenza eventi | * L’utente non loggato effettua l’accesso come amministratore * Inserisce un utente nel sistema dopo aver specificato le informazioni ad esso relative. * Effettua una ricerca per l’utente che è interessato a cancellare. * Elimina l’utente dal sistema |
| Alternativa | Nessuna |

Requisiti:

* Utente: (Amministratore) nr. 3.
* Sistema: nr. 4, 5.

|  |  |
| --- | --- |
| Nome | Associare un partito o un candidato ad una votazione. |
| Scopo | Associare un partito ad una votazione. |
| Attore | Amministratore |
| Precondizioni | La votazione non deve essere iniziata. |
| Post-condizioni | Il partito o il candidato viene associato ad una votazione. |
| Errori | La votazione è già iniziata. |
| Comportamento del sistema in caso d’errore | Il sistema non permette di selezionare le votazioni in corso tra quelle a cui è possibile associare un partito. |
| Trigger | Un amministratore vuole associare un partito ad una votazione. |
| Descrizione | Permettere ad un amministratore di associare un partito e i suoi candidati ad una votazione. |
| Sequenza eventi | * L’utente non loggato effettua l’accesso come amministratore * L’amministratore seleziona una tra le votazioni non in corso e non terminate. * L’amministratore seleziona un partito tra quelli registrati nel sistema. * L’amministrazione procede ad associare partito e votazione. |
| Alternativa | Nessuna |

Requisiti:

* Utente: (Amministratore) nr. 5.
* Sistema: nr. 8.

## Diagrammi di sequenza

I sei diagrammi di sequenza raffigurati di seguito descrivono la sequenza dei principali eventi che concernono i principali casi d’uso elencati precedentemente.

Al fine di ottenere una rappresentazione più leggibile dei digrammi, sono state omesse da essi le classi Controller, che si interfacciano con gli elementi della gui, e le classi DAO, che comunicano con il DB. I messaggi tra le classi presenti nei diagrammi e quelle non presenti sono rappresentati come dei found message (ricezione) e dei lost message (invio).

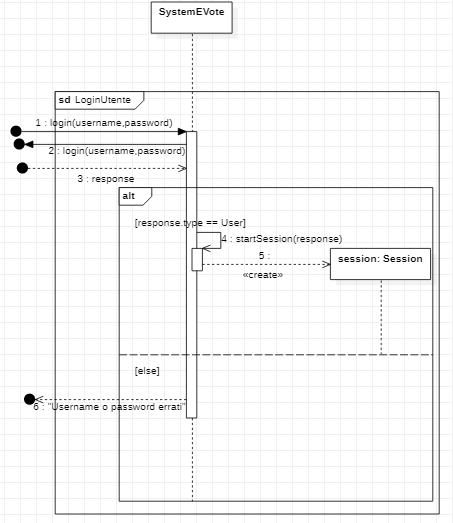


Figura Utente effettua login

In questo diagramma di sequenza l’utente non loggato inserisce le credenziali tramite la gui, le quali vengono inviate al sistema. Esso le invia a sua volta al DB per verificare la loro correttezza. A seconda della risposta che si riceve dal DB, viene inizializzata una nuova sessione oppure viene notificato all’utente un errore nelle credenziali.

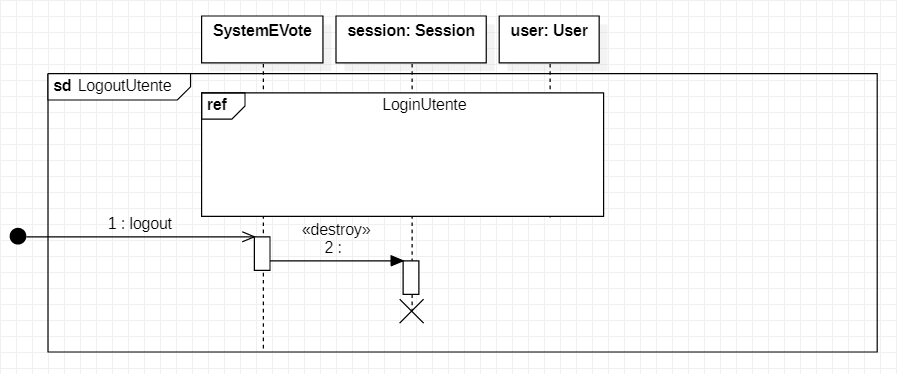


Figura Utente effettua logout

Un utente, che è loggato nel sistema, interagisce con elementi della gui per uscire da esso. Questo evento dà il via alla procedura da logout che comprende la “distruzione” della sessione corrente.

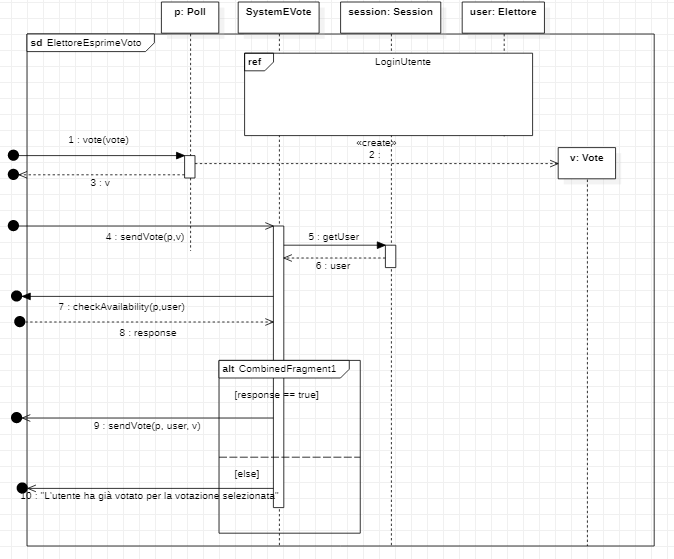


Figura Elettore vota

Un utente che ha effettuato l’accesso al sistema ed è stato riconosciuto come elettore seleziona dalla gui una delle votazioni in corso (p) ed esprime le proprie preferenze (vote). Viene creato il voto. Il sistema verifica se l’elettore ha già partecipato alla votazione selezionata: in caso affermativo la procedura si interrompe senza che il voto venga registrato dal database. Altrimenti, il sistema invia alla classe DAO voto, votazione ed elettore, per registrare il voto e la partecipazione dell’elettore alla votazione sul DB.

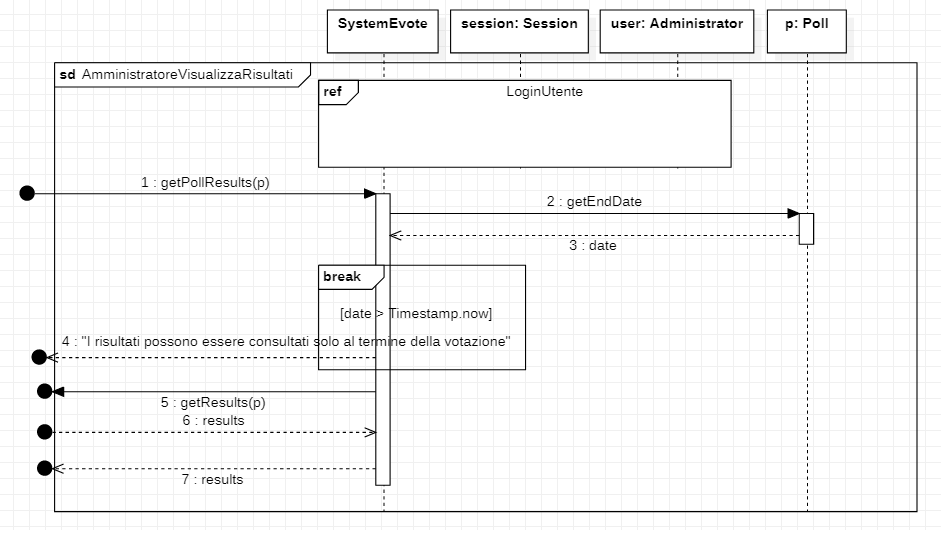


Figura Amministratore visualizza risultati votazione

Un utente che ha ottenuto l’accesso al sistema come amministratore richiede i risultati della votazione p, selezionandola tramite la gui. Il sistema, ricevuta la votazione dall’apposita classe Controller, verifica lo stato della votazione: se non è ancora terminata, non permette all’amministratore di visualizzare i risultati; altrimenti, tramite una classe DAO, recupera dal DB i risultati della votazione p e li invia al Controller che li aveva richiesti.

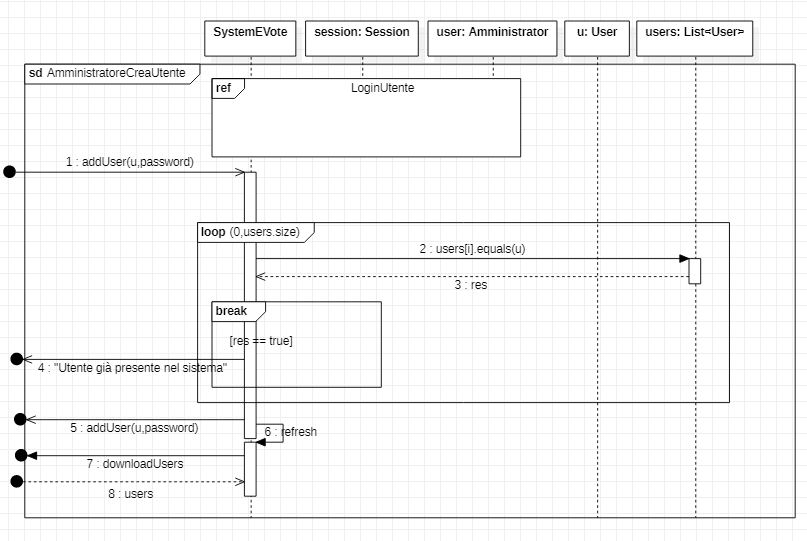


Figura Amministratore crea nuovo utente

Un utente che ha ottenuto l’accesso al sistema come amministratore inserisce tramite la gui le credenziali del nuovo utente che intende registrare nel sistema. Le credenziali vengono ricevute dal sistema, il quale verifica se un utente con quelle credenziali è già presente e in caso affermativo la procedura su interrompe notificando all’amministratore il problema. Altrimenti, il sistema invia ad una classe DAO le credenziali e quest’ultima procede a registrare l’utente sul DB. Poi procede ad aggiornare la collezione di utenti locali.

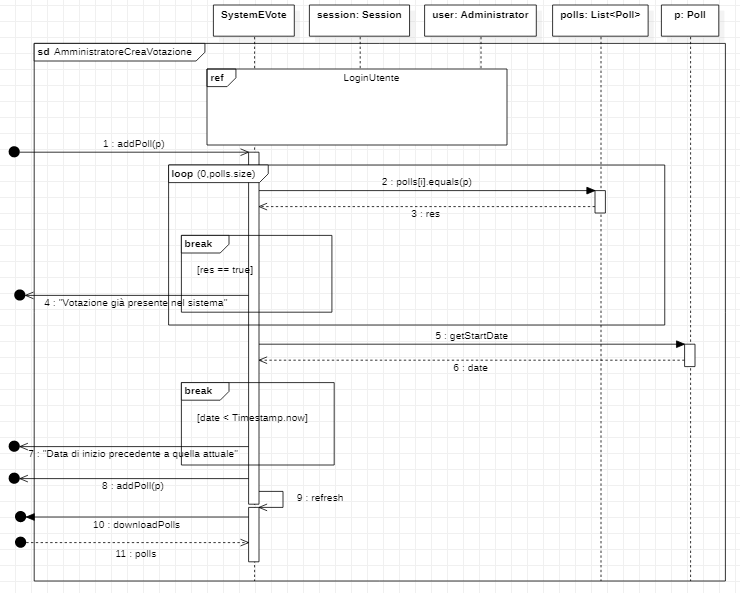


Figura Amministratore crea nuova votazione

Un utente che ha ottenuto l’accesso al sistema come amministratore inserisce tramite la gui le informazioni della votazione che intende creare. La votazione viene inviata al sistema, il quale verifica una votazione uguale a quella ricevuta è già presente e in caso affermativo la procedura su interrompe notificando all’amministratore il problema. Altrimenti, il sistema procede a verificare che la data d’inizio della votazione sia successiva a quella attuale. In caso negativo, la procedura si interrompe, notificando all’amministratore il problema. Altrimenti, il sistema invia ad una classe DAO le credenziali e quest’ultima procede a registrare l’utente sul DB. Poi procede ad aggiornare la collezione di votazioni locali.

## Diagrammi delle attività

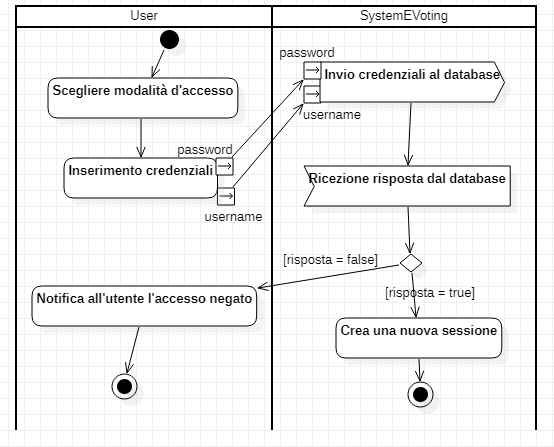


Figura Utente effettua login

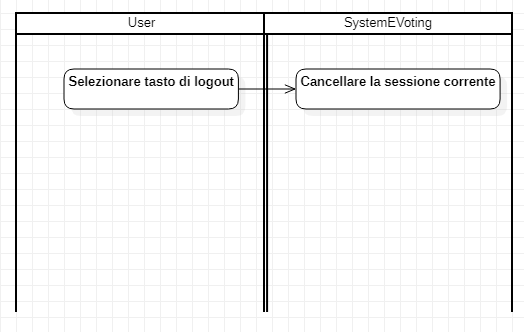


Figura Utente effettua logout

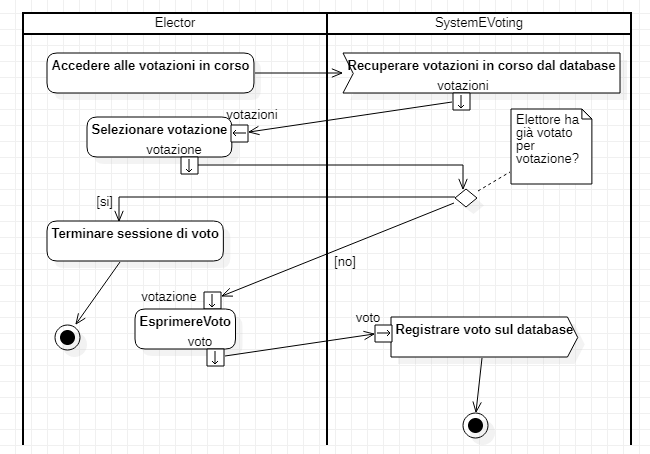


Figura Elettore vota

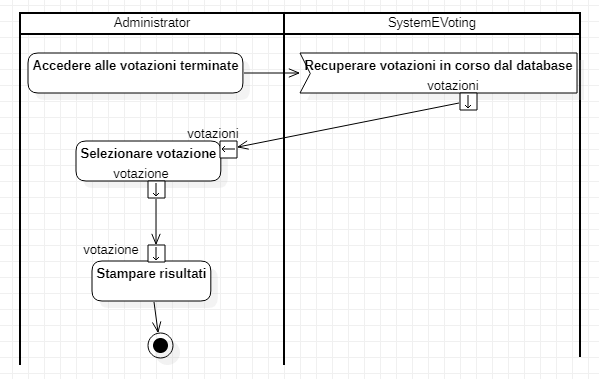


Figura Amministratore visualizza i risultati di una votazione terminata

## Macchine di stato

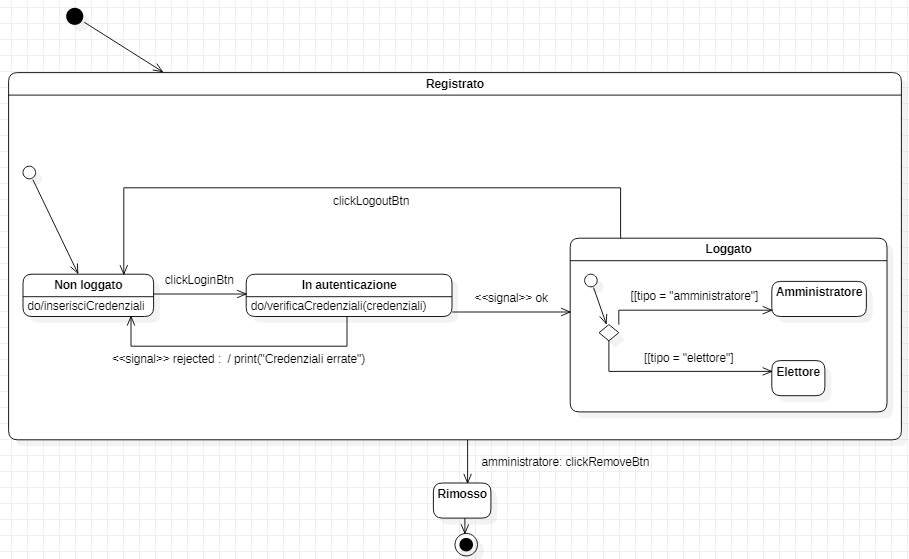


Figura Macchina di stato di un utente

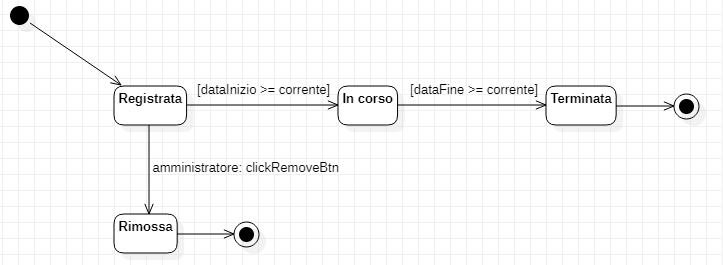
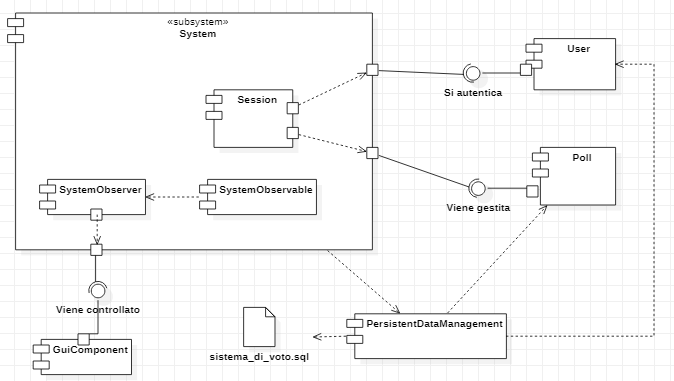


Figura Macchina di stato di una votazione

## Diagramma dei componenti



# Implementazione del sistema

## Diagramma delle classi

In questa sezione viene riportato il diagramma delle classi. Per ottenere una miglior leggibilità, è stato suddiviso in tre diagrammi:

* Classi di modello.
* Classi che implementano il pattern Observer.
* Classi che implementano l’interazione tra DB e modello (pattern DAO).

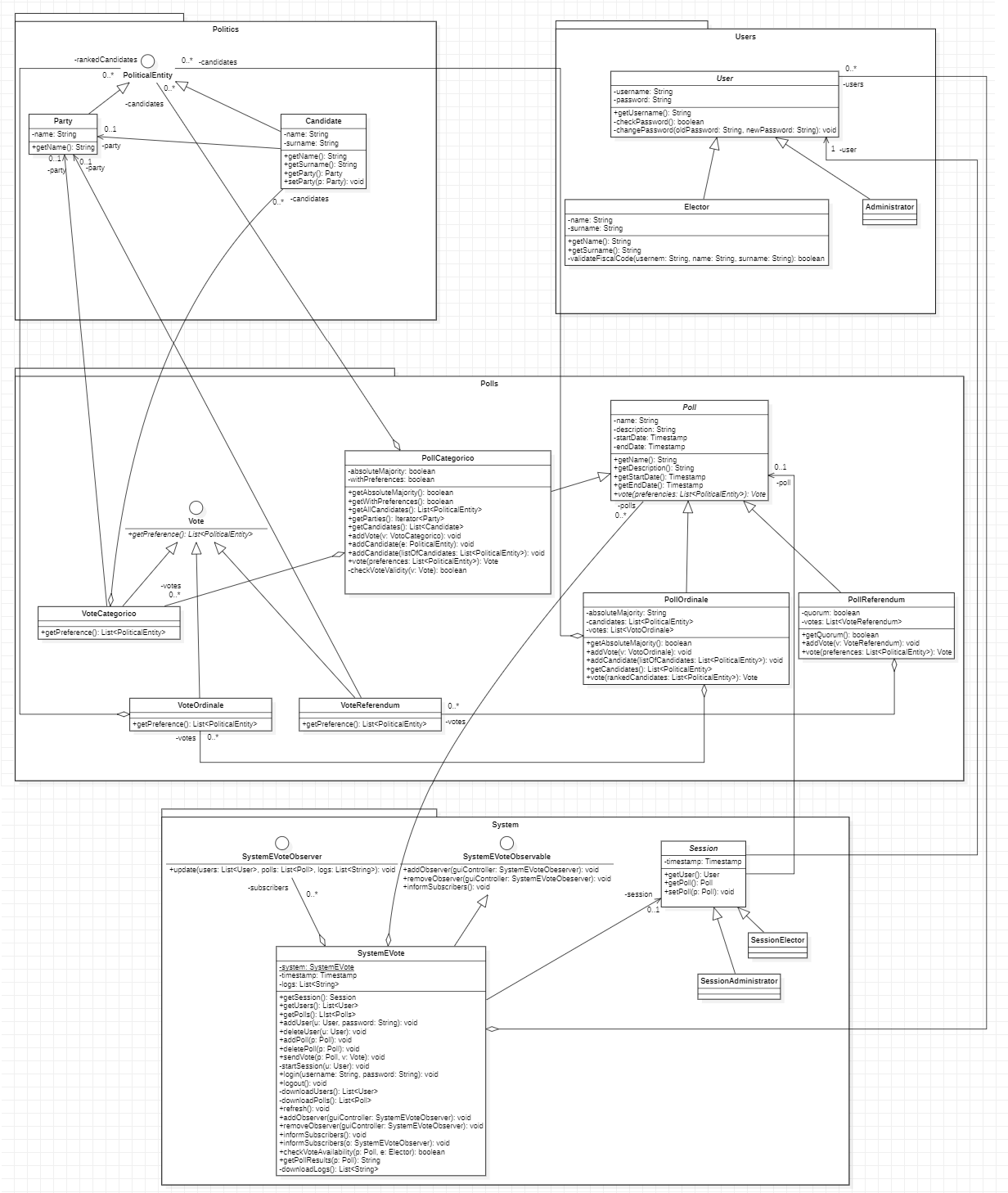


Figura Diagramma delle classi modello

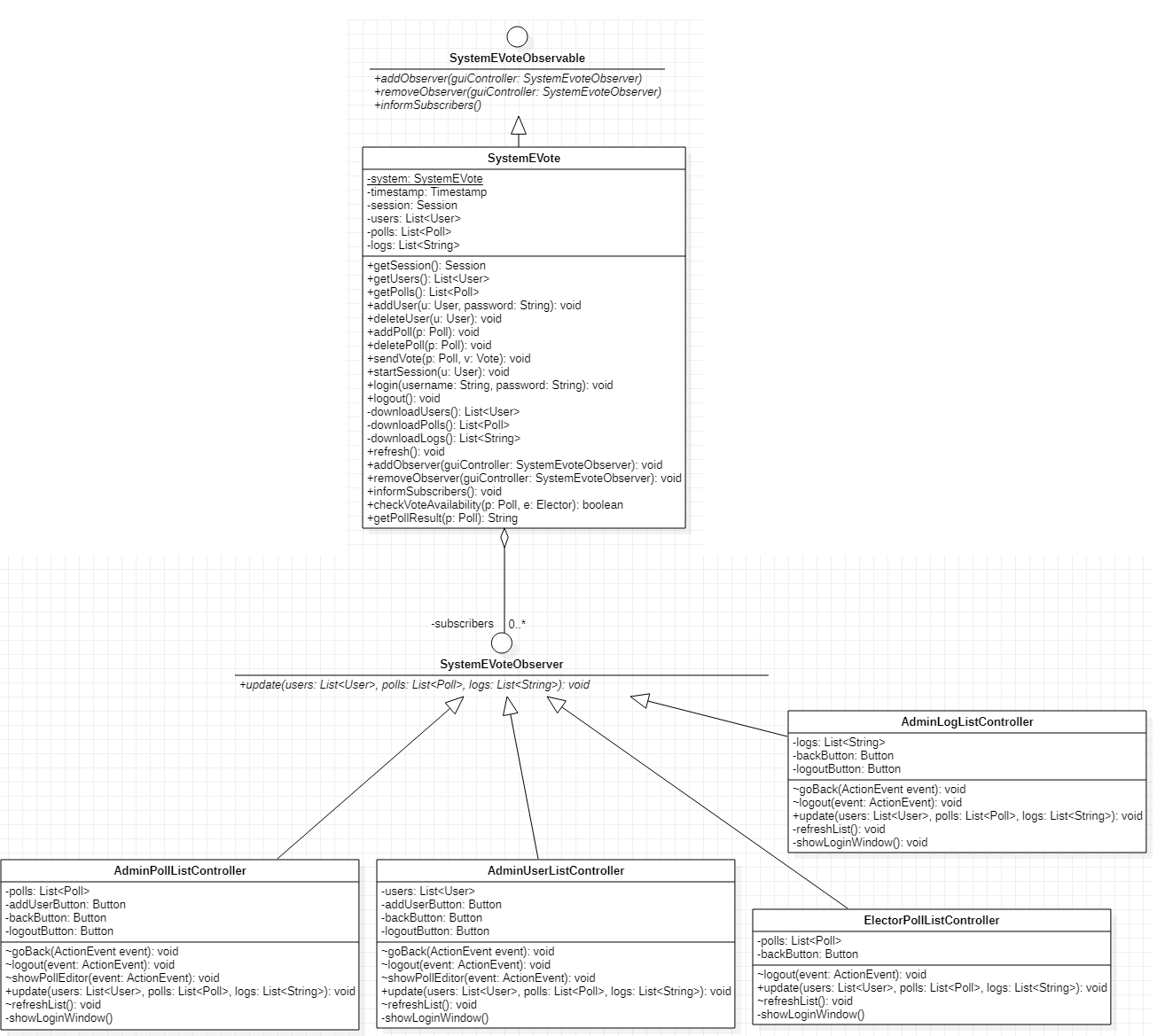


Figura Diagramma delle classi Observer

Immagine che contiene tavolo

Descrizione generata automaticamente

Figura Diagramma delle classi DAO

### Discussione dei design pattern utilizzati

I design pattern utilizzati in questo progetto sono Observer, MVC, DAO e Singleton.

Observer è caratterizzato da una relazione uno a molti, dove l’uno è la classe detta Observable e i molti sono le classi Observer. Quando la classe Observable rileva un certo evento, va ad avvisare le classi Observer che sono registrate presso tale Observer tramite un loro metodo apposito, update. Nel sistema il pattern Observer è stato utilizzato per permettere l’aggiornamento di utenti, votazioni e log presso alcune classi GUI Observer, mentre l’interfaccia Observable è implementata dalla classe SystemEVote, che, com’è possibile vedere, contiene un riferimento alle classi Observer che attendono aggiornamenti.

MVC viene utilizzato per la gestione dell’interazione tra interfacce grafiche (View) e le classi di modello (Model) tramite l’introduzione di un terzo componente: una classe Controller che si occupa di gestire le interazioni dell’utente con la GUI e, attraverso dei metodi, andare a svolgere determinate operazioni che possono apportare cambiamenti alle classi Model. Nel caso di questo progetto, viene utilizzata una GUI implementata tramite JavaFX utilizzando dei file fxml per descrivere cosa dovrà contenere la GUI. Essi rappresentano la componente View del pattern MVC. A ciascun file fxml è associata una classe Controller, che andrà a svolgere la funzione di interprete delle interazioni dell’utente con la GUI e ad apportare gli opportuni cambiamenti alle classi di Modello.

DAO ha l’obiettivo di separare la logica implementata dalle classi del modello dalla logica per la gestione dei dati persistenti. Questo viene fatto tramite la creazione di interfacce e relative classi che forniscono metodi per memorizzare o recuperare dati da un DB. I metodi ricevono in input e restituiscono in output oggetti che sono istanze delle classi del modello. In questo sistema le interfacce – classi DAO si dividono in gestione degli utenti e gestione delle votazioni.

Singleton permette di realizzare classi con un’unica istanza. Ciò può essere vantaggioso nei casi in cui è logico prevedere che ci sarà solo un’entità di un determinato tipo. Nel sistema qui descritto, Singleton risulta utile a rappresentare la classe SystemEVote, che rappresenta il sistema centrale ed è quindi logico averne un’unica istanza. Un altro caso, sempre all’interno di questo sistema, nel quale risulta utile il pattern Singleton è nelle classi DAO: il database con cui comunicare non varia nel tempo; perciò, risulta inutile andare a re-istanziare più volte oggetti di tali classi.

## Gestione dei dati persistenti

Il sistema salva i dati persistenti su un database relazionale che utilizza il DBMS MariaDB.

Di seguito vengono riportate le tabelle che compongono il DB con una breve descrizione.

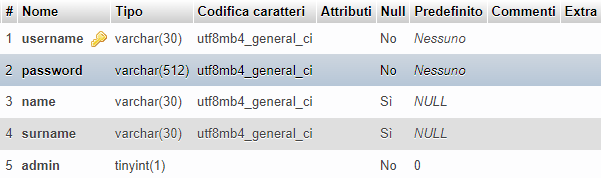


Figura Tabella user

Tiene traccia delle informazioni di un utente, che sia elettore o amministratore. Il tinyInt admin viene utilizzato come flag: se settato a 1 indica che l’utente è un amministratore, altrimenti è un elettore.

Immagine che contiene testo

Descrizione generata automaticamente

Figura Tabella poll

Tiene traccia delle informazioni di una votazione. Il campo type contiene il tipo della votazione (“ORDINALE”,”CATEGORICO”,”REFERENDUM”).

Immagine che contiene testo

Descrizione generata automaticamente

Figura Tabella vote

Tiene traccia delle associazioni voto – votazione, al fine di permettere, a votazione conclusa di fornire i risultati. Il campo ranking viene utilizzato nel caso in cui il voto faccia riferimento ad una votazione ordinale, per la quale è quindi necessario stabilire un ordine dei vari partiti presenti nel singolo voto.

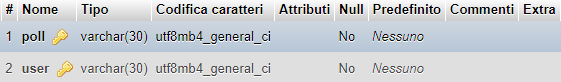


Figura Tabella vote\_register

Tiene traccia delle associazioni votazione – elettore, al fine di garantire che un elettore non possa esprimere più di un voto per una data votazione.

Immagine che contiene testo

Descrizione generata automaticamente

Figura Tabella party

Tiene traccia dei partiti e a quale votazione sono associati.

Immagine che contiene testo

Descrizione generata automaticamente

Figura Tabella candidate

Tiene traccia dei candidati, a quale partito sono associati e a quale votazione partecipano.

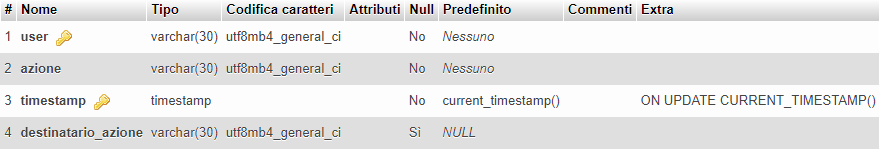


Figura Tabella log

Tiene traccia dei log. Il campo user indica l’utente che compie l’azione. Il campo destinatario\_azione può fare riferimento ad una votazione o a un altro utente, dipendentemente dal tipo di azione.

## Descrizione dell’interfaccia grafica

La GUI è stata implementata con JavaFX e tramite l’utilizzo di file fxml per rappresentare le componenti grafiche.

Di seguito vengono riportate alcune schermate della GUI con una breve descrizione.

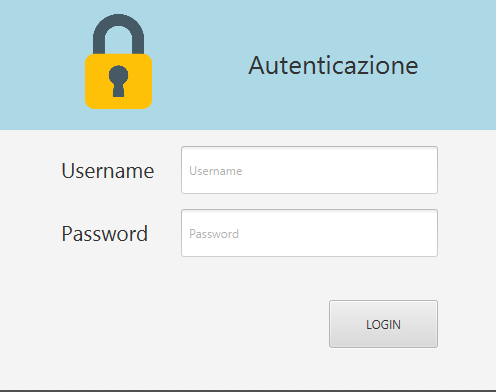


Figura Schermata di login

All’avvio del sistema, tale schermata viene presentata all’utente, che dovrà inserirvi username e password e successivamente premere LOGIN.

Immagine che contiene tavolo

Descrizione generata automaticamente

Figura Schermata di selezione votazione per l' elettore

Se l’utente che accede viene riconosciuto come elettore, viene reindirizzato a questa pagina, dove è presente una lista delle votazioni, con alcune informazioni a riguardo, tra cui il loro stato. Per un elettore è possibile interagire con i due tasti a lato della tabella per visualizzare la descrizione della votazione o per procedere ad esprimere la propria preferenza (questo solo nel caso di votazione in corso).



Figura Schermata di voto per un referendum

Se un elettore decide di partecipare ad una votazione in corso, supponendo che si tratti di un referendum, viene reindirizzato a questa pagina, dove potrà, dopo aver selezionato una tra le opzioni disponibili, confermare la sua scelta e tornare quindi alla schermata delle votazioni.

La stessa schermata sarà utilizzata nel caso di votazione categorica e ordinale. In quest’ultimo tipo di votazione però, dovranno essere selezionati tutti i candidati disponibili, in ordine di preferenza decrescente.

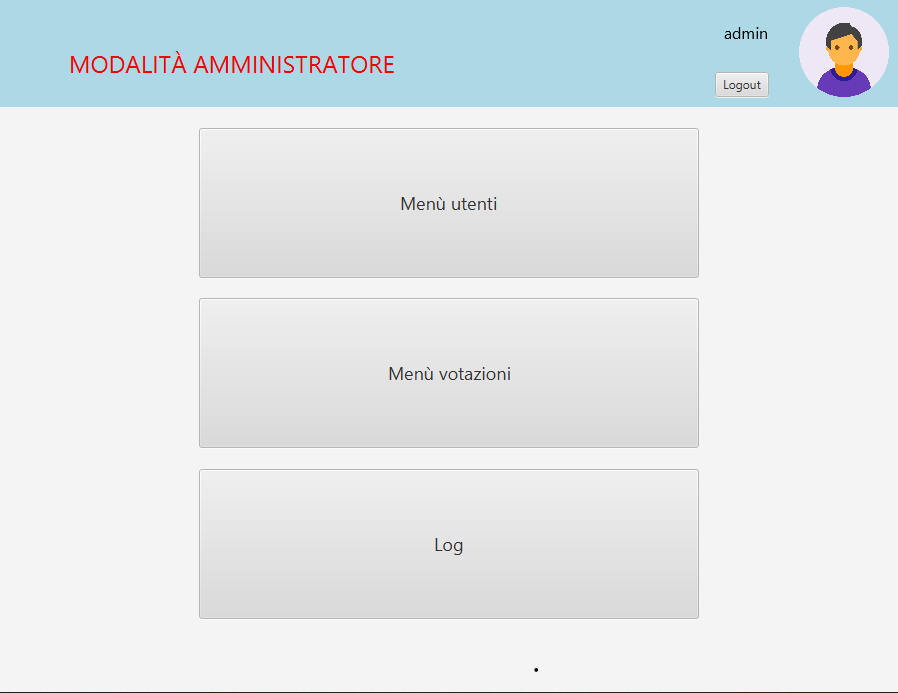


Figura Schermata dei menù dell'amministratore

Se l’utente loggato viene riconosciuto come amministratore viene reindirizzato a questa schermata. Da essa è possibile accedere alla lista degli utenti, delle votazioni o dei log.

Immagine che contiene testo

Descrizione generata automaticamente

Figura Schermata per la gestione degli utenti

Selezionando “Menù utenti” dalla schermata dei menù è possibile accedere a questa schermata. Qui sono elencati gli utenti del sistema con le loro informazioni principali. È possibile eliminare utenti (tranne l’amministratore in uso attualmente) o crearne di nuovi.

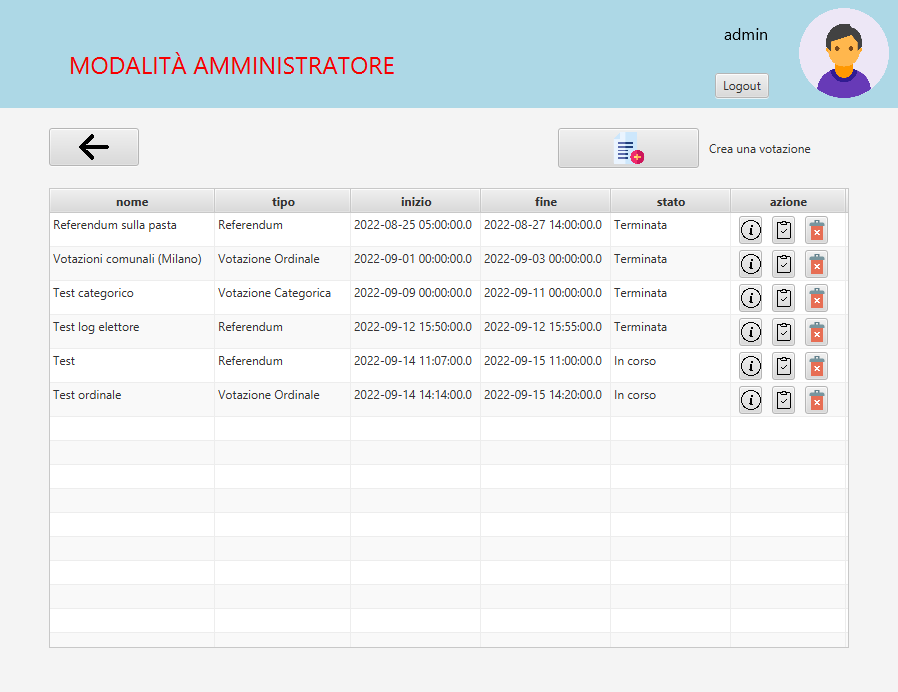


Figura Schermata di gestione delle votazioni

Se l’amministrazione seleziona “Menù votazioni” dalla schermata dei menù viene reindirizzato a questa schermata, dove elencate le votazioni registrate dal sistema, con le principali informazioni ad esse associate. È possibile, per le votazioni già registrate, consultarne la descrizione, visualizzarne i risultati (solo per le votazioni terminate) o eliminarle (solo per le votazioni in preparazione). Inoltre, è possibile creare nuove votazioni.

Immagine che contiene testo

Descrizione generata automaticamente

Figura Schermata dei risultati di una votazione

Se l’amministratore procede a visualizzare la schermata dei risultati di una votazione conclusa, verrà visualizzata una schermata simile a questa. In questo caso, si fa riferimento ad una votazione categorica a candidati: infatti viene visualizzato il partito seguito dal nome del candidato e dal numero di voti che ha ottenuto. Posizione non è settato in quanto viene utilizzato nelle votazioni ordinali. Seguono, nelle righe successive, il numero totale dei voti e quello dei votanti.

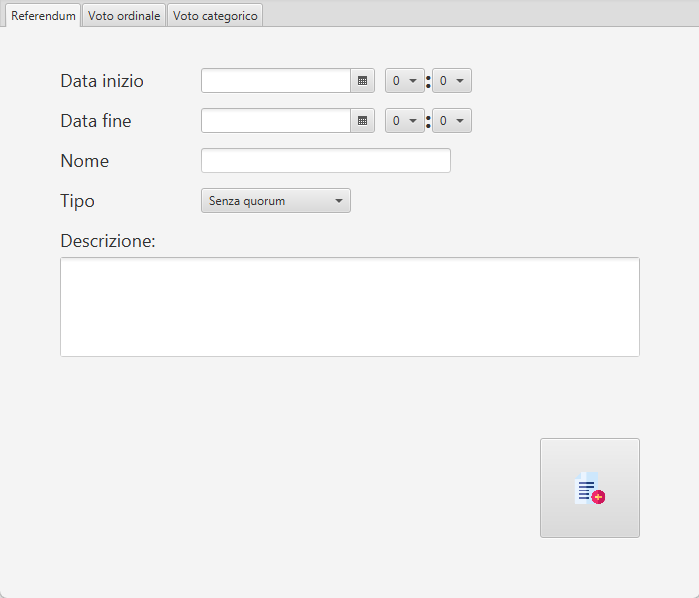


Figura Schermata di configurazione per un referendum

Se si procede con la creazione di una nuova votazione si viene reindirizzati a questa schermata (supponendo di selezionare il referendum). Qui è possibile inserire le principali informazioni riguardanti il referendum e stabilire sarà utilizzato o no il meccanismo del quorum. Si può quindi procedere con la creazione del referendum.

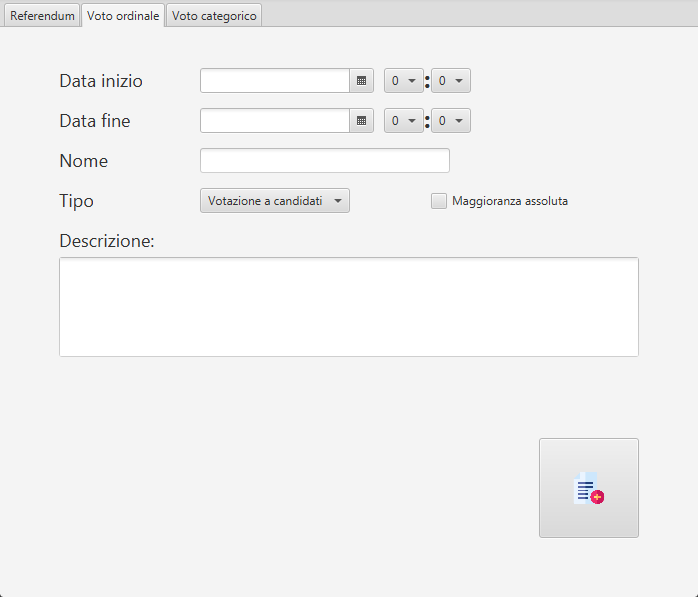


Figura Schermata di configurazione per una votazione ordinale

In caso si selezioni la votazione ordinale, questa è la schermata di configurazione nella quale vengono richieste le principali informazioni riguardanti la votazione. Tramite il menù a tendina è possibile specificare se la votazione riguarderà partiti o i singoli candidati.

Immagine che contiene tavolo

Descrizione generata automaticamente

Figura Schermata di inserimento partiti e candidati per votazione ordinale o categorica

Questa schermata riguarda il caso in cui il voto sia rivolto al candidato; in caso di votazione a partito non verrà chiesto all’amministratore di inserire nome e cognome del candidato. Prima di confermare, è possibile andare a rimuovere dei candidati o partiti dalla tabella per non associarli alla votazione.

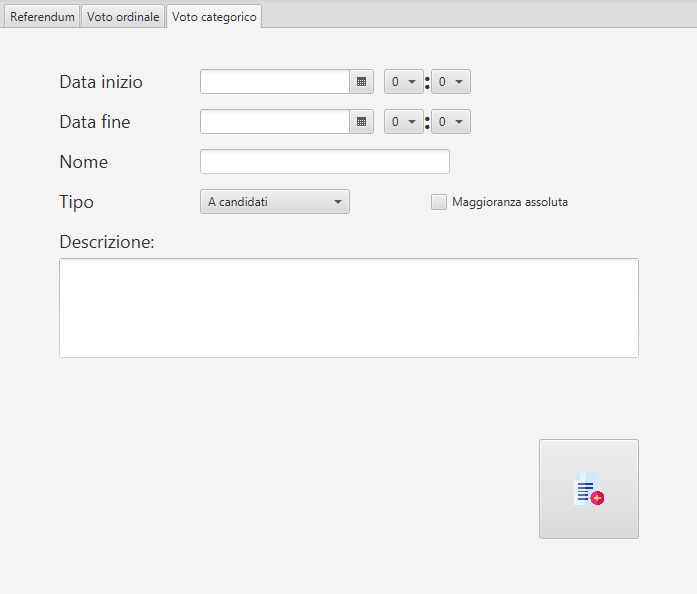


Figura Schermata di configurazione di una votazione categorica

In caso si selezioni la votazione categorica, questa è la schermata di configurazione nella quale vengono richieste le principali informazioni riguardanti la votazione. Tramite il menù a tendina è possibile specificare se la votazione riguarderà partiti, singoli candidati oppure sarà una votazione a partiti specificando dei candidati di quel partito.

Immagine che contiene testo

Descrizione generata automaticamente

Figura Schermata dei log

Se dalla schermata dei menù l’amministratore seleziona “Log”, questa è la schermata a cui verrà reindirizzato. Qui potrà consultare i log del sistema.

Immagine che contiene testo

Descrizione generata automaticamente

Figura Schermata di notifica di un errore

Gli eventuali errori vengono segnalati tramite schermate come quella raffigurata qui sopra, specificando il motivo del problema.

## Diagramma di deployment

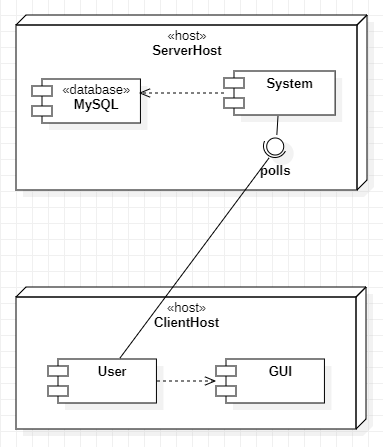


Figura Diagramma di deployment

## Specifica e verifica dei vincoli

I seguenti vincoli, insieme ad altri non riportati qui di seguito, espressi tramite OCL sono stati mappati nel codice in vincoli JML.

User

{context User

inv: self.username != null

inv: self.password != null

}

{context User::User(username: String, password: String)

pre: username != null and password != null

}

{context User::checkPassword(pwd: String): boolean

pre: pwd != null

post: !(self.password == pwd) || result

}

{context User::changePassword(oldPassword: String, newPassword: String)void

pre: newPassword != null and oldPassword == self.password

post: self.password = newPassword

Elettore

{context Elector

inv: self.name != null

inv: self.surname != null

}

{context Elector::Elector(username: String, password: String, name: String, surname: String)

pre: name != null and surname != null

}

Poll

{context Poll

inv: self.name != null

inv: self.description != null

inv: self.startDate != null

inv: self.endDate != null

}

{context Poll::Poll(name: String, description: String, startDate: Timestamp, endDate: Timestamp)

pre: name != null and description != null and startDate != null and endDate != null

}

PollCategorico

{context PollCategorico

inv: self.candidates != null

inv: self.votes != null

inv: self.candidates -> forAll(candidate | candidate != null)

inv: self.votes -> forAll(vote | vote != null)

}

{context PollCategorico::getParties():List<Party>

post: result -> forall(party| self.candidates -> exists(candidate| candidate == party))

}

{context PollCategorico::addVote(v: VoteCategorico): void

pre: v != null

post: self.votes -> exists(vote| vote == v)

}

{context PollCategorico::addCandidate(e: PolicalEntity): void

pre: e != null

post: self.candidates -> exists(candidate| candidate == e)

}

{context PollCategorico::vote(preferences: List<PoliticalEntity>): Vote

pre: preferences != null and preferences -> forall(preference| preference != null)

}

Candidate

{context Candidate

inv: self.name != null

inv: self.surname != null

}

{context Candidate::setParty(p: Party): void

pre: p != null

post: self.party == p

}

## Descrizione del testing

Test per classe Amministratore:

* testEquals(): testa il funzionamento del metodo equals di Administrator nel caso di confronto di due amministratori uguali.
* testNotEquals(): testa il funzionamento del metodo equals di Administrator nel caso di confronto di due amministratori diversi.

Test per classe Elettore:

* testEquals(): testa il funzionamento del metodo equals di Elector nel caso di confronto di due elettori uguali.
* testNotEquals(): testa il funzionamento del metodo equals di Elector nel caso di confronto di due elettori diversi.

Test per classe PollCategorico:

* testAddCandidate(): testa il funzionamento del metodo addCandidate di PollCategorico, controllando che il candidato venga registrato correttamente.
* testVote(): testa il funzionamento del metodo vote di PollCategorico, verificando che il voto restituito dal metodo sia uguale a quello atteso.

Test per classe PollReferendum:

* testVote(): testa il funzionamento del metodo vote di PollReferendum, verificando che il voto restituito dal metodo sia uguale a quello atteso.

Test per classe PollOrdinale:

* testAddCandidate(): testa il funzionamento del metodo addCandidate di PollOrdinale, controllando che il candidato venga registrato correttamente.
* testVote(): testa il funzionamento del metodo vote di PollOrdinale, verificando che il voto restituito dal metodo sia uguale a quello atteso.

Test per la classe Candidate:

* testSetParty(): testa il funzionamento del metodo setParty di Candidate, controllando che il partito di appartenenza del candidato sia uguale a quello atteso.
* testEquals(): testa il funzionamento del metodo equals di Candidate nel caso di confronto di due candidati uguali.
* testNotEquals(): testa il funzionamento del metodo equals di Candidate nel caso di confronto di due candidati diversi.

Test per la classe Party:

* testEquals(): testa il funzionamento del metodo equals di Party nel caso di confronto di due partiti uguali.
* testNotEquals(): testa il funzionamento del metodo equals di Party nel caso di confronto di due partiti diversi.