



UNIVERSITÀ DEGLI STUDI DI MILANO
FACOLTÀ DI SCIENZE E TECNOLOGIE

Corso di Laurea in Informatica

Sistema di Voto Elettronico (Project Krateos)

di

Andrea Borghesi - 952944
Andrei Daniel Balanica - 941778

Anno Accademico 2021-2022

1	INTRODUZIONE	3
1.1	GLOSSARIO	3
2	SPECIFICA DEI REQUISITI	5
2.1	REQUISITI FUNZIONALI	5
2.1.1	<i>Gruppi di credenziali e autenticazione</i>	5
2.1.2	<i>Sessione di Voto</i>	5
2.1.3	<i>Modalità di scelta del vincitore</i>	6
2.2	REQUISITI NON FUNZIONALI	6
2.2.1	<i>Requisiti di sicurezza</i>	6
2.2.2	<i>Requisiti di Disponibilità</i>	7
2.2.3	<i>Requisiti di Osservabilità</i>	7
3	PROGETTAZIONE DEL SISTEMA	8
3.1	CASI D'USO	8
3.1.1	<i>Diagrammi</i>	8
3.1.2	<i>Descrizione degli Scenari</i>	9
3.2	ARCHITETTURA	14
3.2.1	<i>Descrizione</i>	14
3.2.2	<i>Diagramma delle Componenti</i>	14
3.2.3	<i>Server APIs</i>	15
3.3	DIAGRAMMI DELLE CLASSI DI PROGETTO	17
3.3.1	<i>Server</i>	17
3.3.2	<i>Client Elector</i>	18
3.3.3	<i>Client Manager</i>	19
3.4	DIAGRAMMI DI SEQUENZA	20
3.4.1	<i>Creazione Sessione di Voto – lato server</i>	20
3.4.2	<i>Creazione Sessione di Voto – lato client manager</i>	21
3.4.3	<i>Aggiunta Opzione ad una Sessione di Voto</i>	22
3.4.4	<i>Aggiunta Gruppo di Elettori ad una Sessione di Voto</i>	22
3.4.5	<i>Attivazione di una Sessione di Voto</i>	23
3.4.6	<i>Espressione Voto – lato server</i>	24
3.4.7	<i>Espressione Voto – lato client elector</i>	25
3.4.8	<i>Determinazione Vincitori di una Sessione di Voto</i>	26
3.5	DIAGRAMMA DELLE ATTIVITÀ	27
3.5.1	<i>Creazione di una sessione di voto (Client)</i>	27
3.5.2	<i>Determinazione Vincitori di una Sessione di Voto</i>	28
3.6	MACCHINE DI STATO	29
3.6.1	<i>Stati della Sessione di Voto</i>	29
4	IMPLEMENTAZIONE DEL SISTEMA	30
4.1	DIAGRAMMI DELLE CLASSI DI PROGRAMMA	30
4.1.1	<i>Server</i>	30
4.1.2	<i>Client Elector</i>	30
4.1.3	<i>Client Manager</i>	30
4.2	DISCUSSIONE DEI PATTERN UTILIZZATI	31
4.3	AUTENTICAZIONE E AUTORIZZAZIONE	32
4.4	LOGGING	32
4.5	TESTING	32
4.6	GESTIONE DATI PERSISTENTI	33
4.6.1	<i>Struttura database</i>	33
4.7	DESCRIZIONE DELL'INTERFACCIA GRAFICA	34
4.7.1	<i>Interfaccia grafica Client Manager</i>	34
4.7.2	<i>Interfaccia grafica Client Elector</i>	45
4.8	DIAGRAMMA DI DEPLOYMENT	52
4.9	SPECIFICA E VERIFICA DEI VINCOLI	53
5	APPENDICE A: GUIDA ALL'INSTALLAZIONE	55
6	APPENDICE B: SVILUPPI FUTURI	56

1 Introduzione

Kratéós è un sistema di voto e scrutinio elettronico che ha come obiettivo il permettere lo svolgimento di votazioni tramite seggi elettronici o da remoto attraverso la rete.

Mette a disposizione diverse modalità di voto (categorico, categorico con preferenza, ordinale e referendum) e di scelta del vincitore (maggioranza assoluta e quorum). È quindi una soluzione generica: è facilmente adattabile ad ogni contesto in cui è richiesta una votazione.

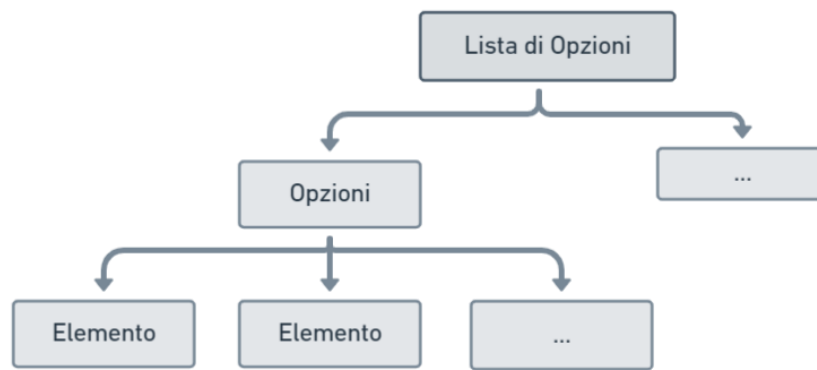
L'intero sistema è fruibile attraverso un'interfaccia grafica di semplice utilizzo.

Kratéós si pone anche l'obiettivo di garantire la segretezza e l'anonimato del voto, oltre che a rispettare i vincoli di sicurezza delle informazioni sensibili e voti degli elettori.

1.1 Glossario

- **Utente:** un qualsiasi individuo che interagisce con il sistema
- **Elettore:** un utente con delle credenziali valide
- **Gestore (del sistema):** detto anche Manager, è colui che imposta la sessione di voto, decide quali saranno le modalità di voto, le procedure che definiscono il vincitore, crea i gruppi di elettori e le credenziali per gli elettori
- **Credenziali:** Una coppia di stringhe dette username e password
- **Credenziali valide:** Credenziali che passano l'autenticazione
- **Gruppo di elettori:** un gruppo di elettori identificato da un nome univoco
- **Voto valido:** un voto espresso e confermato da un elettore
- **Voto non confermato:** un voto che non ha ricevuto la conferma dall'elettore
- **Quesito booleano:** proposizione che ammette come risposta solo "sì" o "no"
- **Ballottaggio:** sessione di voto che avviene in caso di pareggio tra due o più opzioni o elementi
- **Sede elettorale:** luogo fisico in cui sono presenti computer connessi al sistema che consentono la votazione di persona
- **Storico:** Insieme di sessioni concluse, con le rispettive conclusioni per ognuna.
- **Sessione di voto annullata:** sessione di voto terminata dal suo gestore che non prevede alcun vincitore
- **Voto ordinale:** all'elettore è richiesto di ordinare le opzioni presenti nella scheda in base alle proprie preferenze
- **Voto categorico:** l'elettore inserisce una preferenza per un'opzione
- **Voto categorico con preferenze:** l'elettore inserisce una preferenza per un'opzione e ha la possibilità di indicare una preferenza tra gli elementi che lo compongono (niente voto disgiunto)
- **Referendum:** consiste in un quesito booleano posto all'elettore con la quale si chiede se si sia favorevoli o contrari
- **Maggioranza:** il vincitore è l'opzione o elemento che ha ottenuto il maggior numero di voti
- **Maggioranza assoluta:** il vincitore è l'opzione o elemento che ha ottenuto il 50% + 1 dei voti espressi
- **Quorum:** requisito che alla sessione abbia votato almeno il 50% + 1 aventi diritto perché questa sia valida

- **Sessione di voto:** Insieme di opzioni, di una modalità di determinazione del vincitore (referendum, categorica, categorica con preferenza, ordinale) e di un intervallo di tempo



- **Access Token:** un token JWT responsabile della persistenza di autenticazione di un utente

2 Specifica dei requisiti

2.1 Requisiti funzionali

2.1.1 Gruppi di credenziali e autenticazione

- 1 Il gestore deve essere in grado di generare le credenziali per identificare ciascun utente abilitato al voto
 - 1.1 Il sistema non deve ammettere due username uguali
 - 1.2 Il gestore può inserire manualmente nome e cognome di un nuovo utente
- 2 Il gestore può revocare le credenziali di uno o più elettori
- 3 Il gestore deve essere in grado di creare uno o più gruppi di elettori
 - 3.1 Il gestore deve assegnare a ciascun insieme di elettori un nome identificativo univoco
 - 3.2 Il gestore può assegnare ogni utente a uno o più insiemi di elettori

2.1.2 Sessione di Voto

- 1 Il gestore del sistema può creare una o più sessioni di voto
 - 1.1 Il gestore del sistema deve assegnare un nome univoco alla sessione di voto
 - 1.2 Il gestore del sistema deve associare uno o più insiemi di elettori (detti gruppi) alla sessione di voto
 - 1.2.1 Solo l'elettore facente parte di almeno uno dei gruppi selezionati ha il diritto di voto in tale sessione
 - 1.3 La sessione di voto può terminare secondo due modalità
 - 1.3.1 La modalità automatica, che richiede l'inserimento di una data di scadenza della sessione di voto
 - 1.3.1.1 L'elettore deve sottomettere il suo voto prima della scadenza della sessione di voto.
 - 1.3.2 La modalità manuale permette al gestore del sistema di terminare la sessione di voto in un qualsiasi momento
 - 1.4 La sessione di voto può essere annullata dal gestore, terminandola senza effettuare alcun conteggio dei voti o invalidandone il risultato se questo è già stato effettuato
 - 1.5 Il gestore del sistema deve selezionare una ed una sola modalità di voto tra le 4 disponibili
 - 1.5.1 Nel caso di voto ordinale, categorico e categorico con preferenza il gestore del sistema dovrà inserire la lista delle opzioni
 - 1.5.1.1 La lista delle opzioni non può contenere due opzioni uguali
 - 1.5.1.2 Un'opzione non può contenere due elementi uguali
 - 1.5.2 Nel caso di selezione di referendum il gestore deve inserire un quesito booleano.
 - 1.6 Il gestore può decidere se la sessione di voto utilizzerà il quorum e/o la maggioranza assoluta per determinare il vincitore
 - 1.7 Il gestore del sistema può attivare una sessione di voto
 - 1.8 Il gestore del sistema può eliminare una sessione di voto inattiva
- 2 In seguito all'attivazione della sessione di voto non deve essere più possibile modificare gli elettori che possono votare in tale sessione
- 3 Il gestore del sistema può visualizzare tutte le sessioni di voto
- 4 L'elettore autenticato deve poter votare solo alle sessioni alle quali è stato ammesso
 - 4.1 Il voto può essere espresso solo quando la sessione utente è attiva
 - 4.2 Un elettore autenticato non deve poter votare più di una volta alla stessa sessione di voto

- 5 L'elettore autenticato deve confermare il voto espresso perché questo sia valido
 - 5.1 Il voto valido deve essere registrato
- 6 L'elettore può accedere al sistema, autenticarsi e votare dal proprio dispositivo ovunque si trovi
- 7 Il conteggio dei voti deve avvenire solo al termine della sessione di voto

2.1.3 *Modalità di scelta del vincitore*

- 1 Al termine del conteggio dei voti viene definito il vincitore
- 2 Gli elettori possono visualizzare lo storico delle sessioni di voto a cui hanno partecipato
- 3 Il gestore del sistema può visualizzare lo storico di tutte le sessioni di voto

2.2 *Requisiti non funzionali*

- 1 Ogni voto espresso deve essere anonimo: non deve essere possibile collegare un voto all'elettore che l'ha espresso
- 2 Ogni voto espresso deve essere segreto: non deve essere possibile collegare l'elettore ai voti che ha espresso
- 3 Ciascun elettore deve essere in grado di votare in autonomia in seguito alla visualizzazione di un breve tutorial

2.2.1 *Requisiti di sicurezza*

1. Per eseguire l'accesso al sistema e alle sue singole componenti, ciascun utente deve possedere delle credenziali valide
2. Devono essere realizzati due software separati e indipendenti per la gestione delle votazioni e per l'esecuzione delle votazioni
 - a. Nessuno dei due software deve memorizzare alcuna informazione sulla macchina in cui sono in esecuzione
3. A ciascuna credenziale valida deve essere associato un ruolo
 - a. Solo credenziali associate al ruolo di gestore devono consentire l'accesso al software di gestione delle votazioni
 - b. Solo credenziali associate al ruolo di elettore devono consentire l'accesso al software di esecuzione delle votazioni
4. I processi di autenticazione e autorizzazione devono essere gestiti in modo centralizzato
 - a. L'autenticazione e l'autorizzazione di un utente devono dipendere soltanto dalle credenziali valide utilizzate per effettuare l'accesso
5. La persistenza della sessione dell'utente autenticato deve avvenire attraverso token JWT
 - a. In seguito all'autenticazione tramite credenziali valide deve essere rilasciato un Access Token
 - b. Tutte le operazioni svolte da un utente autenticato devono essere corredate con l'Access Token
 - c. Operazioni richieste senza un Access Token valido non devono essere eseguite.
 - d. La validità dell'Access Token deve essere di 20 minuti
 - i. Il possesso di un Access Token scaduto implica il logout forzato dagli applicativi del sistema e l'impossibilità di svolgere qualsiasi operazione (escluso il login)
6. Tutte le operazioni dedicate alla gestione delle sessioni di voto, degli elettori e dei gruppi di elettori devono essere accessibili solo attraverso credenziali con ruolo di gestore
7. Tutte le informazioni sensibili trattate da qualsiasi componente del sistema e che necessitano persistenza nel tempo, devono essere criptate, prima di essere memorizzate nel database

- a. La criptazione delle password deve avvenire attraverso un algoritmo che prevede salt
- 8. Ogni voto valido deve essere memorizzato senza essere modificato

2.2.2 Requisiti di Disponibilità

- 1. Il sistema deve essere sempre disponibile
- 2. In caso di interruzioni del funzionamento del sistema, ciascun utente che lo sta utilizzando al momento dell'interruzione, deve essere riportato alla schermata di Login e ciascuna operazione in corso deve essere annullata
- 3. Quando il sistema o una sua parte torna disponibile, in seguito ad una interruzione di servizio, deve consentire a ciascun utente di riprendere la propria attività e di rieseguire le eventuali operazioni annullate

2.2.3 Requisiti di Osservabilità

- 1. Ciascuna componente del sistema deve disporre di un sistema di log che consenta di ricostruire lo storico e la sequenza di tutte le operazioni effettuate
- 2. Nei log non devono comparire informazioni sensibili di un qualsiasi utente del sistema

3 Progettazione del sistema

3.1 Casi d'Uso

3.1.1 Diagrammi

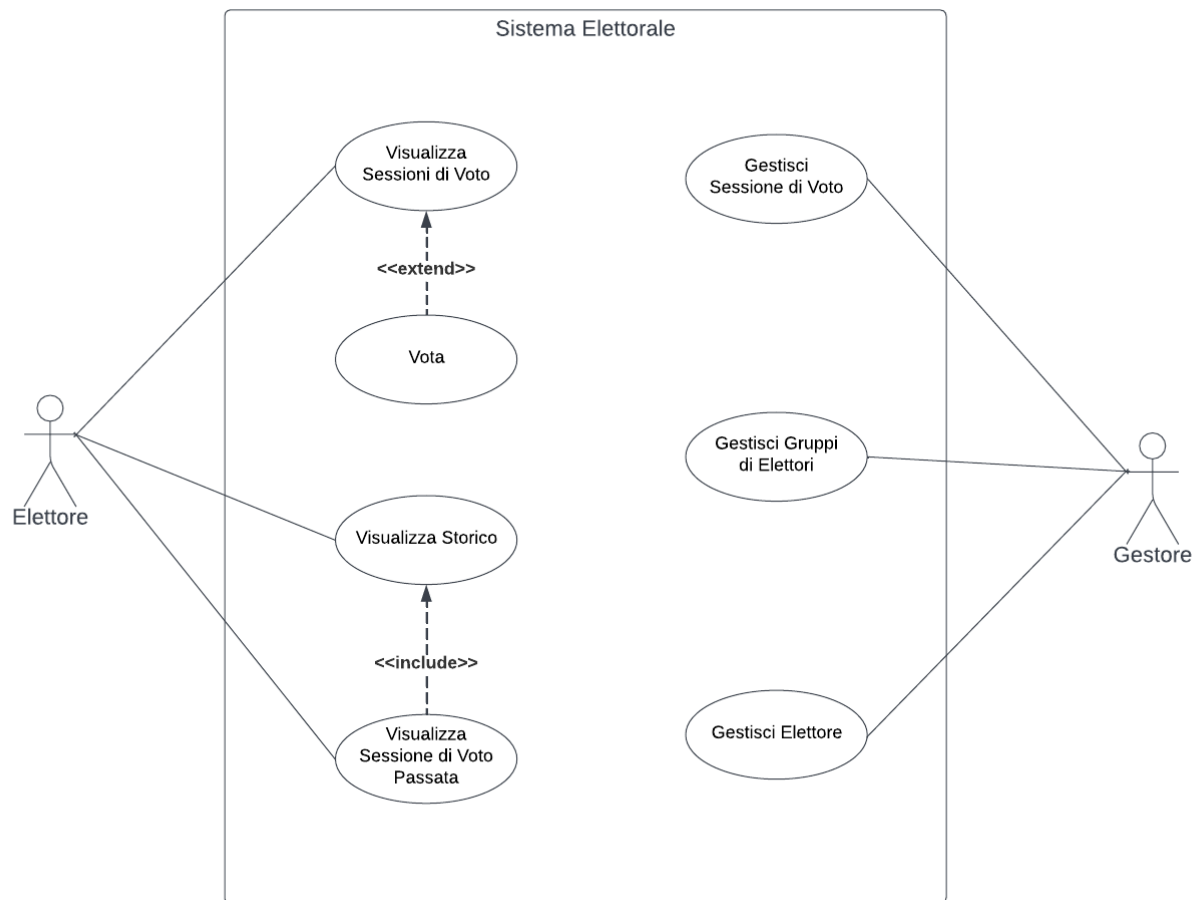


Figure 1 Diagramma dei casi d'uso del sistema elettorale

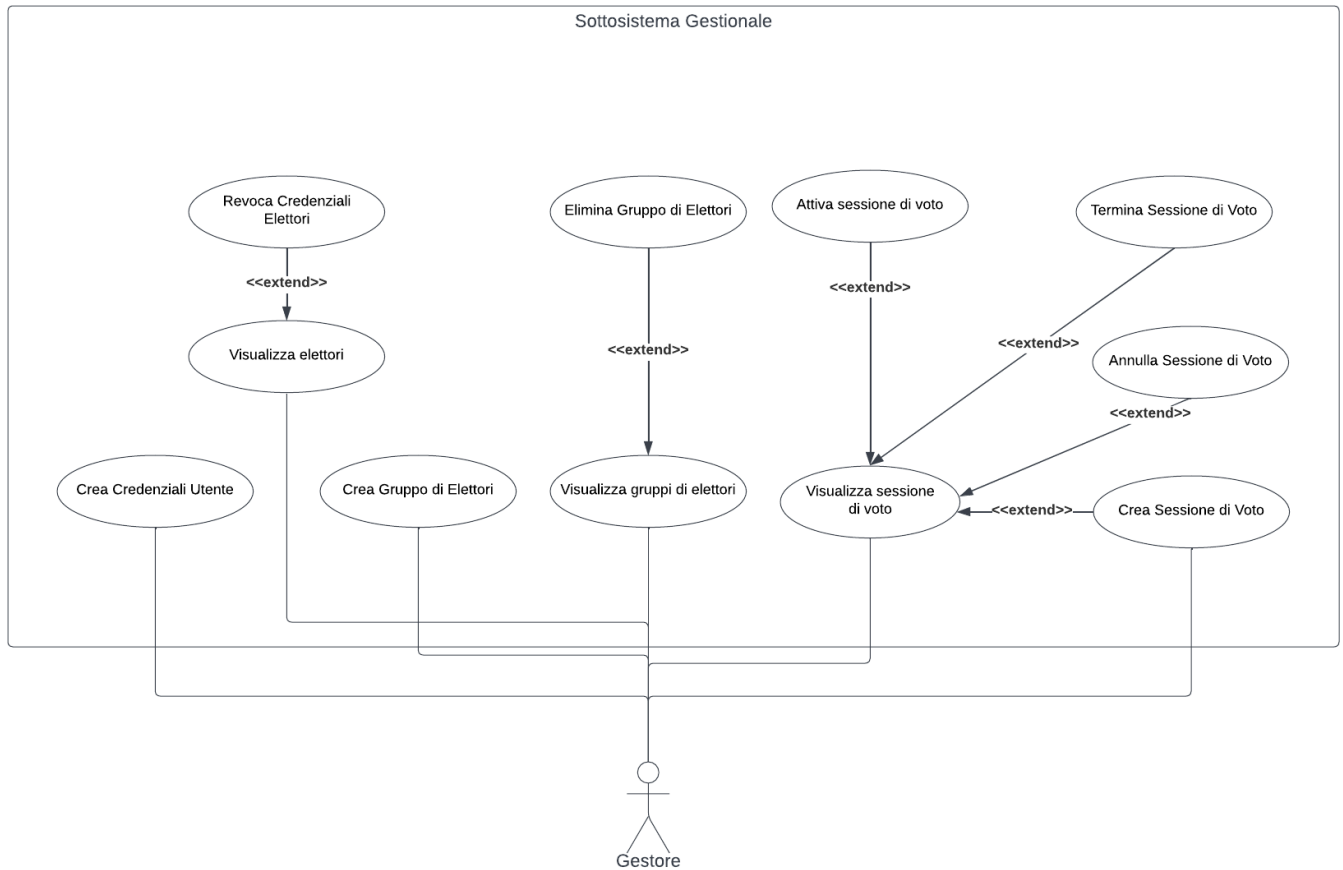


Figure 2 Diagramma dei casi d'uso del sottosistema gestionale

3.1.2 Descrizione degli Scenari

Nome	Crea sessione di voto
Scopo	A richiesta del gestore viene creata una nuova sessione di voto
Attore/i	Gestore
Pre-condizioni	Il gestore è loggato
Casi d'uso correlate	-
Trigger	Il gestore richiede la creazione della sessione di voto
Descrizione sequenza di eventi	<ol style="list-style-type: none"> 1. Inserisci il nome della sessione di voto 2. Definisci la data di termine 3. Scegli il tipo di sessione di voto (categorica, ordinale, categorica con preferenza o referendum) 4. Seleziona quorum e/o maggioranza assoluta se desiderato 5. Aggiungi le opzioni di voto 6. Scegli i gruppi di elettori che parteciperanno alla sessione di voto
Alternativa/e	-
Post-condizioni	È presente una nuova sessione di voto con le caratteristiche date
Requisiti	Sessione di voto 1.*

Nome	Crea gruppo di elettori
Scopo	Crea un nuovo gruppo di elettori
Attore/i	Gestore
Pre-condizioni	Il gestore è loggato
Casi d'uso correlate	-
Trigger	Il gestore richiede la creazione di un nuovo gruppo di elettori
Descrizione sequenza di eventi	<ol style="list-style-type: none"> 1. Inserisci il nome del nuovo gruppo di elettori 2. Seleziona gli elettori che faranno parte del gruppo
Alternativa/e	-
Post-condizioni	È presente un nuovo gruppo di elettori con le specifiche date
Requisiti	Gruppi di credenziali e autenticazione: 4

Nome	Elimina gruppo di elettori
Scopo	Rimuovere un gruppo di elettori
Attore/i	Gestore
Pre-condizioni	Il gestore è loggato
Casi d'uso correlate	Estende <i>Visualizza gruppo di elettori</i>
Trigger	Il gestore richiede la rimozione di un gruppo di elettori dal sistema
Descrizione sequenza di eventi	<ol style="list-style-type: none"> 1. Seleziona il gruppo da rimuovere 2. Conferma la rimozione del gruppo selezionato
Alternativa/e	-
Post-condizioni	Il gruppo selezionato non è più presente nel sistema
Requisiti	Gruppi di credenziali e autenticazione: 4

Nome	Annulla sessione di voto
Scopo	Annullare una sessione di voto terminandola senza effettuare il conteggio o invalidarne il risultato nel caso sia già stato effettuato.
Attore/i	Gestore
Pre-condizioni	Il gestore è loggato
Casi d'uso correlate	Estende <i>Visualizza sessioni di voto</i> (Gestore)
Trigger	Il gestore richiede l'annullamento di una sessione di voto
Descrizione sequenza di eventi	<ol style="list-style-type: none"> 1. Seleziona la sessione da annullare 2. Conferma l'annullamento della sessione
Alternativa/e	-
Post-condizioni	Viene annullata una sessione di voto
Requisiti	Sessione di voto: 1.4

Nome	Termina sessione di voto
Scopo	Terminare una sessione di voto
Attore/i	Gestore
Pre-condizioni	Il gestore è loggato
Casi d'uso correlate	Estende <i>Visualizza sessioni di voto</i> (Gestore)
Trigger	Il gestore richiede la terminazione di una sessione di voto
Descrizione sequenza di eventi	<ol style="list-style-type: none"> 1. Seleziona la sessione da terminare 2. Seleziona la terminazione della sessione
Alternativa/e	Si aspetta la data di termine della sessione di voto
Post-condizioni	Viene terminata una sessione di voto e presentati i risultati della votazione
Requisiti	Sessione di voto: 1.3.*

Nome	Visualizza sessioni voto
Scopo	Visualizzare tutte le sessioni di voto attualmente attive alla quale l'elettore ha diritto di voto
Attore/i	Elettore
Pre-condizioni	L'elettore è loggato
Casi d'uso correlati	Esteso da <i>Vota</i>
Trigger	L'utente richiede la visualizzazione delle sessioni di voto
Descrizione sequenza di eventi	-
Alternativa/e	-
Post-condizioni	Vengono visualizzate tutte le sessioni di voto attualmente attive alla quale l'elettore ha diritto di voto
Requisiti	Sessione di voto: 3

Nome	Vota
Scopo	Esprimere un voto nella sessione di voto selezionata
Attore/i	Elettore
Pre-condizioni	L'elettore è loggato, l'elettore ha diritto di voto alla sessione di voto selezionata, la sessione di voto è attiva, l'elettore non ha ancora espresso il voto alla sessione di voto selezionata
Casi d'uso correlati	Estende <i>Visualizza sessioni voto</i> (Elettore)
Trigger	L'utente ha selezionato una sessione di voto
Descrizione sequenza di eventi	<ol style="list-style-type: none"> 1. Esprime le preferenze di voto 2. Conferma il voto
Alternativa/e	-
Post-condizioni	Viene espresso un voto alla sessione di voto selezionata
Requisiti	Sessione di voto: 3

Nome	Visualizza sessioni voto
Scopo	Visualizzare tutte le sessioni di voto attualmente attive alla quale l'elettore ha diritto di voto
Attore/i	Gestore
Pre-condizioni	Il gestore è loggato
Casi d'uso correlati	Esteso da <i>Termina sessione di voto</i> Esteso da <i>Annulla sessione di voto</i> Esteso da <i>Attiva sessione di voto</i>
Trigger	L'utente richiede la visualizzazione delle sessioni di voto
Descrizione sequenza di eventi	-
Alternativa/e	-
Post-condizioni	Vengono visualizzate tutte le sessioni di voto attualmente attive alla quale l'elettore ha diritto di voto
Requisiti	Sessione di voto: 2

Nome	Attiva sessione di voto
Scopo	Attiva una sessione di voto
Attore/i	Gestore
Pre-condizioni	Il gestore è loggato
Casi d'uso correlate	Estende <i>Visualizza sessioni di voto</i> (Gestore)
Trigger	Il gestore richiede l'attivazione di una sessione di voto
Descrizione sequenza di eventi	<ol style="list-style-type: none"> 1. Seleziona la sessione di voto 2. Attiva la sessione di voto selezionata
Alternativa/e	-
Post-condizioni	Viene attivata la sessione di voto selezionata
Requisiti	Sessione di voto: 1.7

3.2 Architettura

3.2.1 Descrizione

Per la realizzazione del sistema di voto sono previsti tre componenti software:

- Il client con GUI utilizzato dall'elettore per votare
- Il client con GUI utilizzato dal gestore per gestire le sessioni di voto
- Il server: si occupa di gestire l'autenticazione, l'autorizzazione, la persistenza dei dati e la validazione tutte le operazioni eseguite. Il server è autoritativo su tutte le operazioni eseguite nel contesto del sistema di voto.

Il server segue il paradigma http REST e, conseguentemente, ogni comunicazione tra i client ed il server avviene mediante richieste http.

Seguendo il paradigma REST, si identificano le seguenti risorse: elettore, gruppo di elettori, sessione di voto, opzione di voto. Ognuna di queste ha un identificativo univoco che viene utilizzato nelle chiamate http per fare riferimento alla specifica risorsa ed effettuare operazioni su di essa.

3.2.2 Diagramma delle Componenti

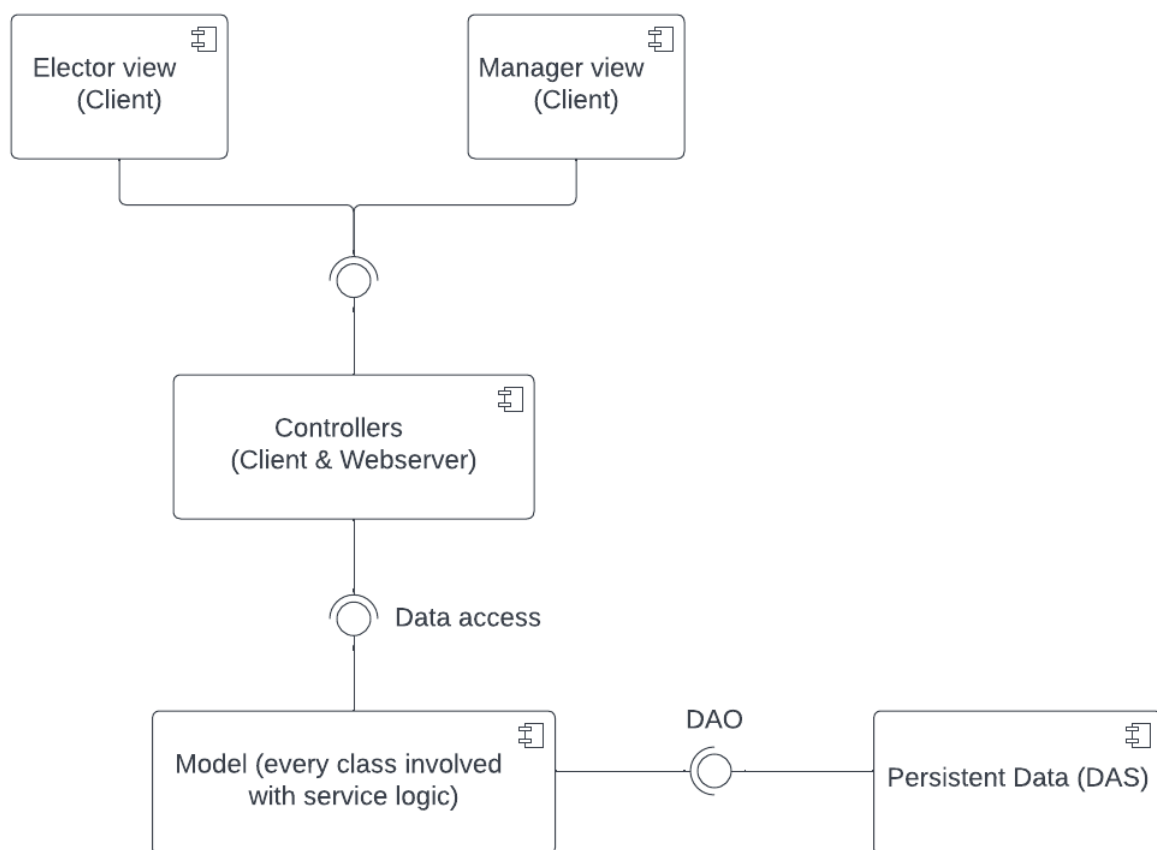


Figure 3 Diagramma delle component del sistema

3.2.3 Server APIs

Le API messe a disposizione dal server sono le seguenti, suddivise per risorsa.

Si indicano con un asterisco “*” le API accessibili dal client dell’elettore. Quelle non contrassegnate sono accessibili soltanto dal gestore. L’unica API non accessibile al gestire è quella per l’espressione di un voto ed è indicata con “-”.

3.2.3.1 Login

- **[*] POST /login**
Consente di effettuare l’accesso e ottenere un token JWT, date delle credenziali valide

3.2.3.2 Elettore

- **GET /elector**
Restituisce i dettagli di tutti gli elettori registrati nel sistema
- **POST /elector**
Consente l’inserimento di un nuovo elettore
- **GET /elector/{electorId}**
Restituisce i dettagli di uno specifico elettore, dato il suo id
- **DELETE /elector/{electorId}**
Consente la rimozione di un elettore dal sistema, dato il suo id

3.2.3.3 Gruppo di Elettori

- **GET /group**
Restituisce la descrizione di tutti i gruppi registrati nel sistema
- **POST /group**
Consente la creazione di un nuovo gruppo di elettori
- **GET /group/{groupId}**
Restituisce la descrizione di uno specifico gruppo, dato il suo id
- **DELETE /group/{groupId}**
Consente la cancellazione di un gruppo dal sistema, dato il suo id
- **GET /group/{groupId}/elector**
Restituisce i dettagli di tutti gli elettori facenti parti del gruppo di cui è dato l’id
- **PUT /group/{groupId}/elector/{electorId}**
Consente di inserire l’elettore identificato dall’electorId nel gruppo identificato dal groupId
- **DELETE /group/{groupId}/elector/{electorId}**
Consente di rimuovere l’elettore identificato dall’electorId dal gruppo identificato dal groupId

3.2.3.4 Sessione di Voto

Le opzioni di voto sono considerate sotto-risorsa della sessione di voto, essendo strettamente dipendenti da quest'ultima.

Gestione sessione:

- **[*] GET /session**
Restituisce la descrizione delle sessioni registrate nel sistema. Il contenuto restituito varia in base al ruolo del chiamante. Ad un elettore verranno restituite solo le sessioni a cui è stato assegnato, mentre al gestore verranno restituite tutte.
- **POST /session**
Consente la creazione di una nuova sessione di voto
- **[*] GET /session/{sessionId}**
Restituisce la descrizione di una specifica sessione di voto, dato il suo id
- **DELETE /session/{sessionId}**
Consente la cancellazione di una specifica sessione di voto, dato il suo id
- **PATCH /session/{sessionId}/state/{newStateString}**
Consente di cambiare lo stato della sessione di voto identificata dal sessionId a quello specificato come newStateString.
- **[-] POST /session/{sessionId}/vote**
Consente di registrare un voto per la sessione di voto identificata dal sessionId
- **GET /session/{sessionId}/result/option**
Restituisce, per la sessione identificata dal sessionId, le opzioni con il relativo numero di voti ottenuti
- **[*] GET /session/{sessionId}/result/winner**
Restituisce, per la sessione identificata dal sessionId, l'opzione che ha vinto l'elezione (e l'elemento nel caso di votazione categorica con preferenze)

Gestione dei gruppi assegnati alla sessione:

- **GET /session/{sessionId}/group**
Restituisce i dettagli dei gruppi assegnati alla sessione di voto identificata da sessionId
- **PUT /session/{sessionId}/group/{groupId}**
Consente di inserire il gruppo identificato dal groupId nella sessione identificata dal sessionId
- **DELETE /session/{sessionId}/group/{groupId}**
Consente di rimuovere il gruppo identificato dal groupId dalla sessione identificata dal sessionId

Gestione delle opzioni della sessione:

- **[*] GET /session/{sessionId}/option**
Restituisce i dettagli delle opzioni della sessione di voto identificata da sessionId
- **PUT /session/{sessionId}/option**
Consente di creare una nuova opzione per la sessione di voto identificata da sessionId
- **PUT /session/{sessionId}/option/{optionId}/suboption**
Consente di creare un nuovo elemento come figlio dell'opzione identificata dal optionId, per la sessione di voto identificata da sessionId
- **DELETE /session/option/{optionId}**
Consente di rimuovere una opzione o elemento dal sistema

3.3 Diagrammi delle Classi di Progetto

3.3.1 Server



Figure 4 Diagramma delle classi - Server

3.3.2 Client Elector

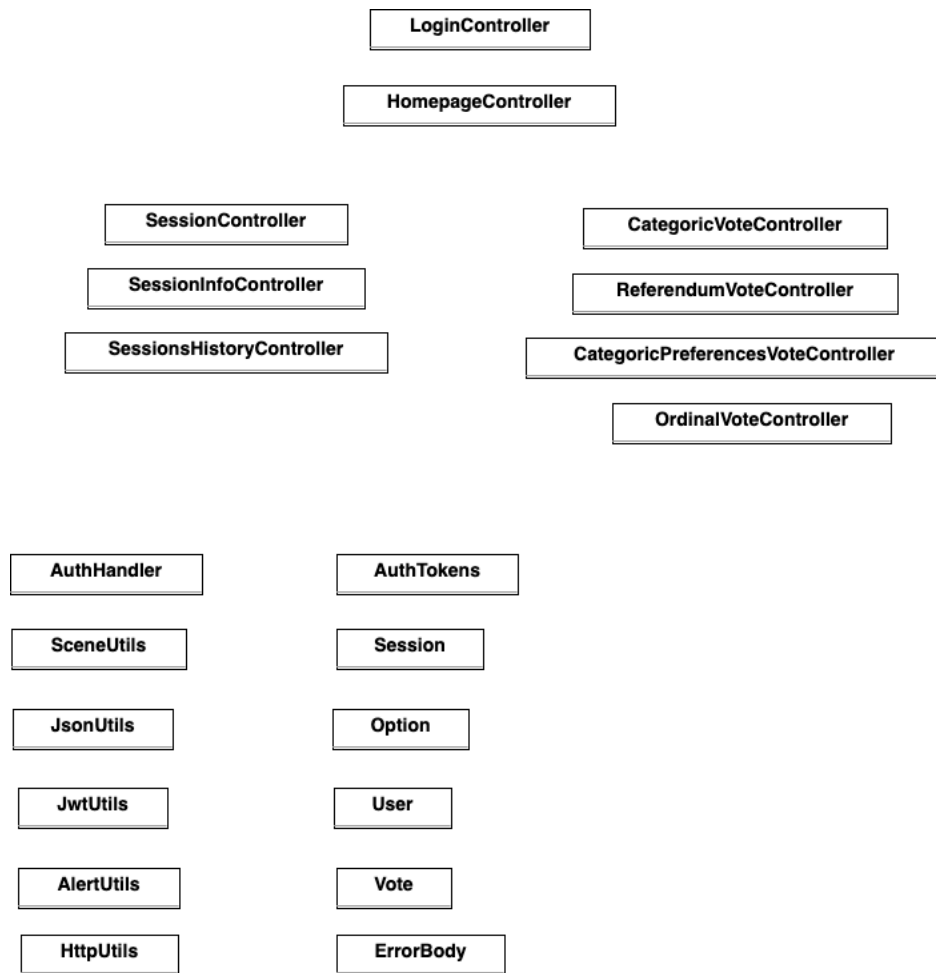


Figure 5 Diagramma delle classi - client elector

3.3.3 Client Manager

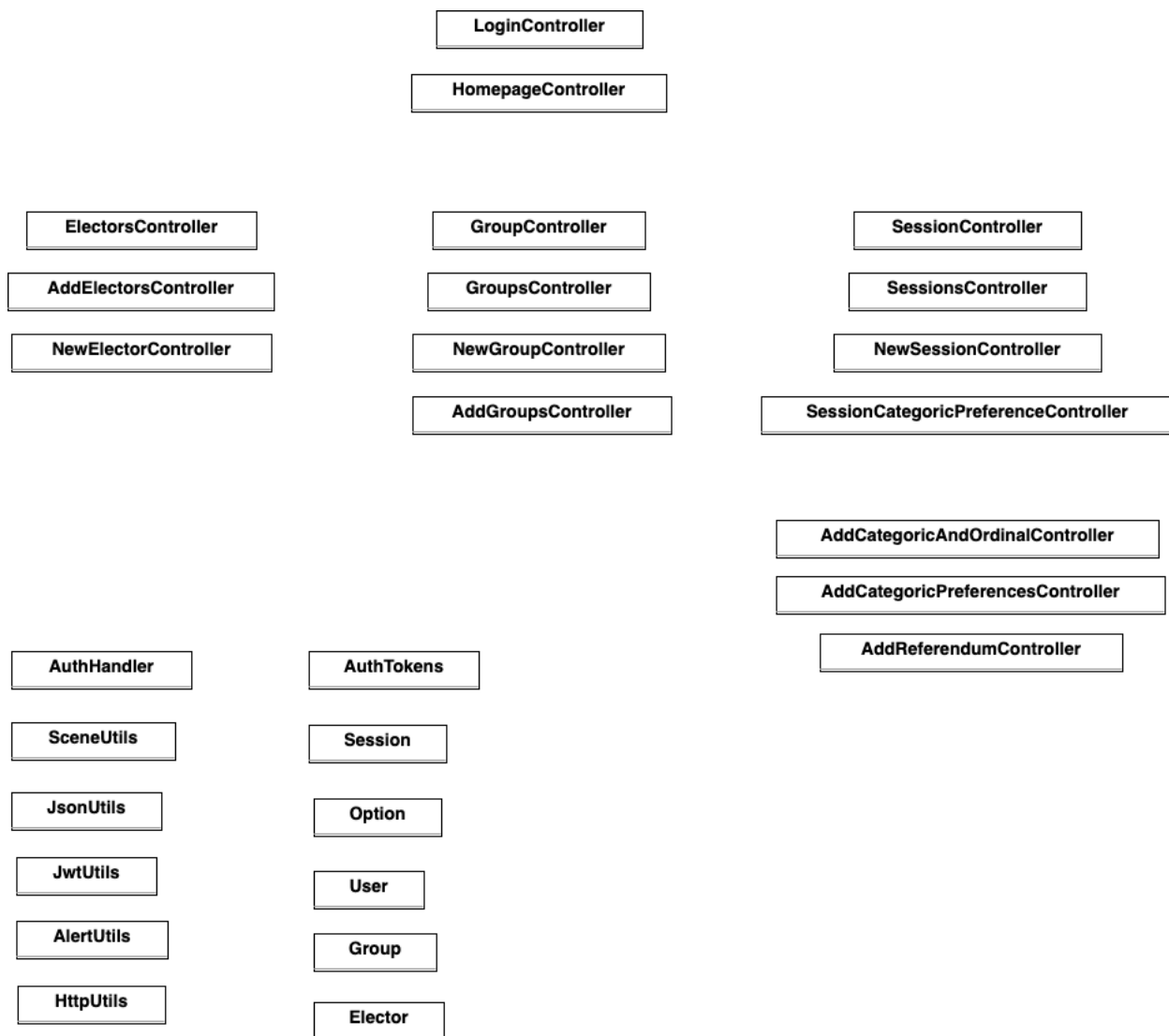


Figure 6 diagramma delle classi - client manager

3.4 Diagrammi di sequenza

3.4.1 Creazione Sessione di Voto – lato server

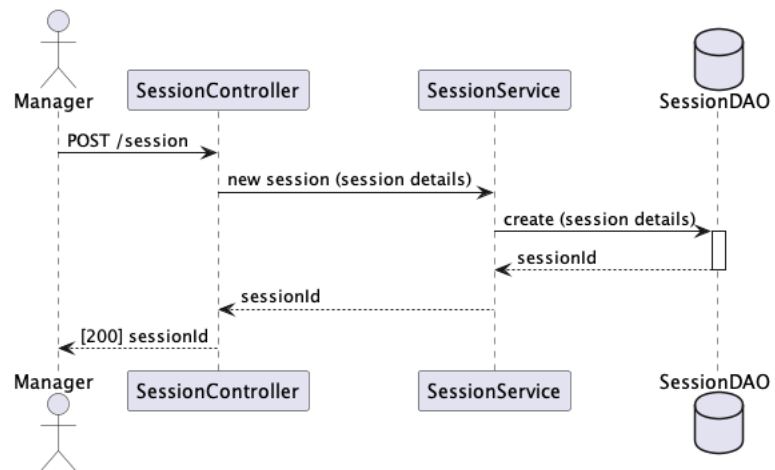


Figure 7 Creazione di una Sessione di Voto, lato server

3.4.2 Creazione Sessione di Voto – lato client manager

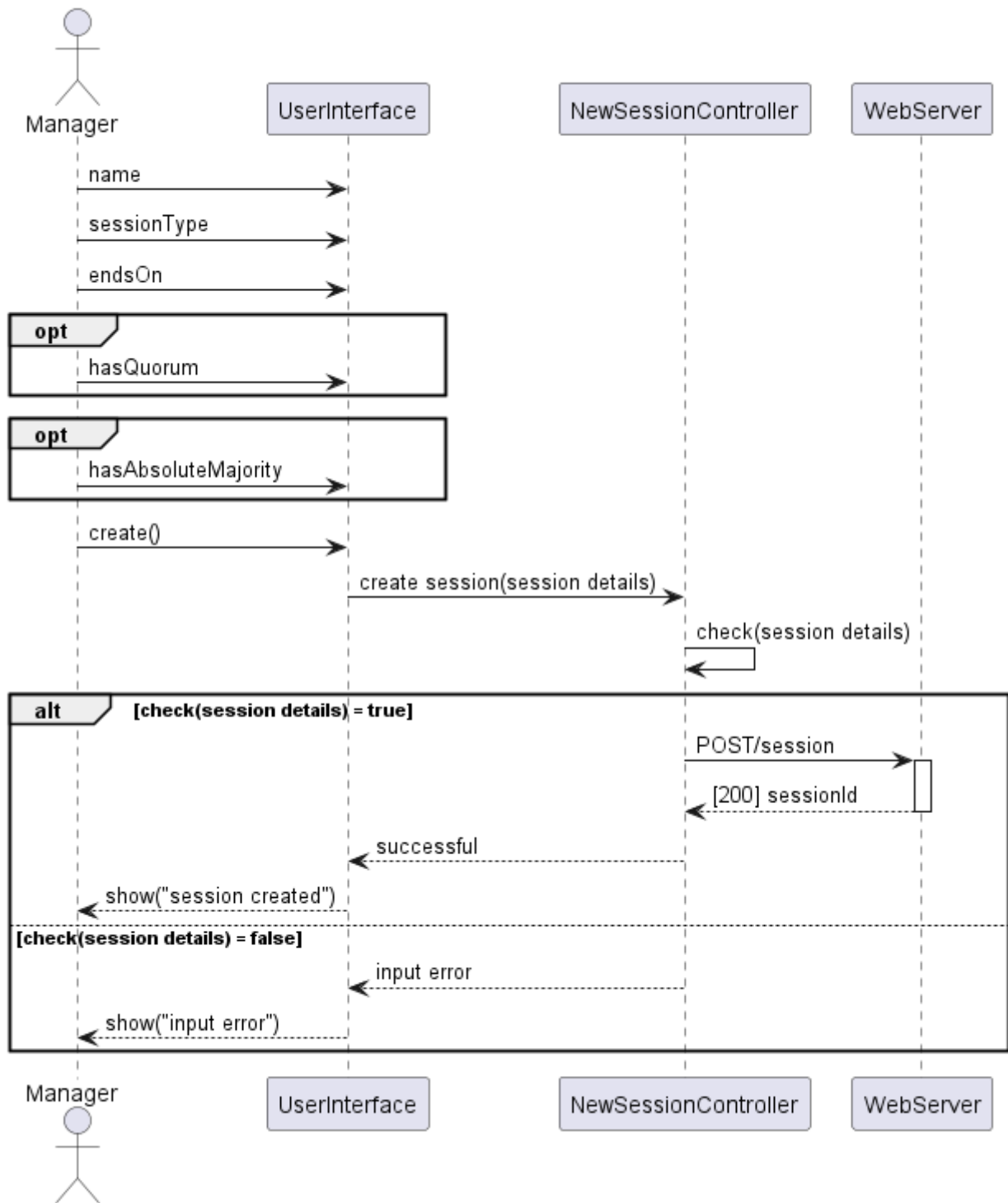


Figure 8 Creazione di una sessione di voto, lato client manager

3.4.3 Aggiunta Opzione ad una Sessione di Voto

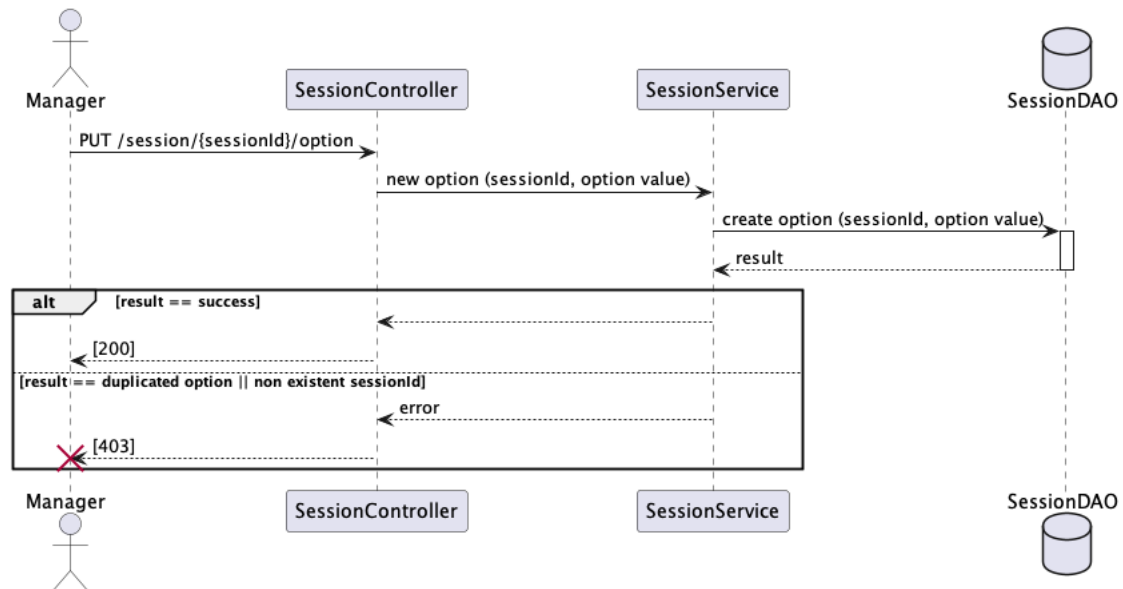


Figure 9 Aggiunta di un'opzione, lato server

3.4.4 Aggiunta Gruppo di Elettori ad una Sessione di Voto

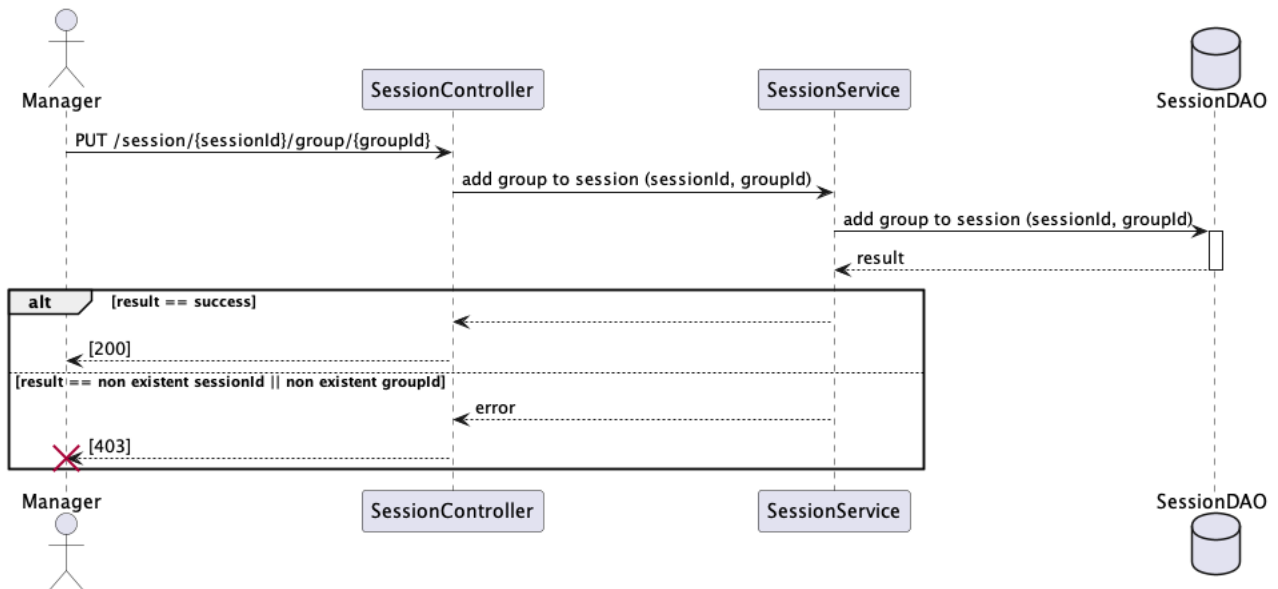


Figure 10 Aggiunta di un gruppo ad una sessione, lato server

3.4.5 Attivazione di una Sessione di Voto



Figure 11 Attivazione di una sessione di voto

3.4.6 Espressione Voto – lato server

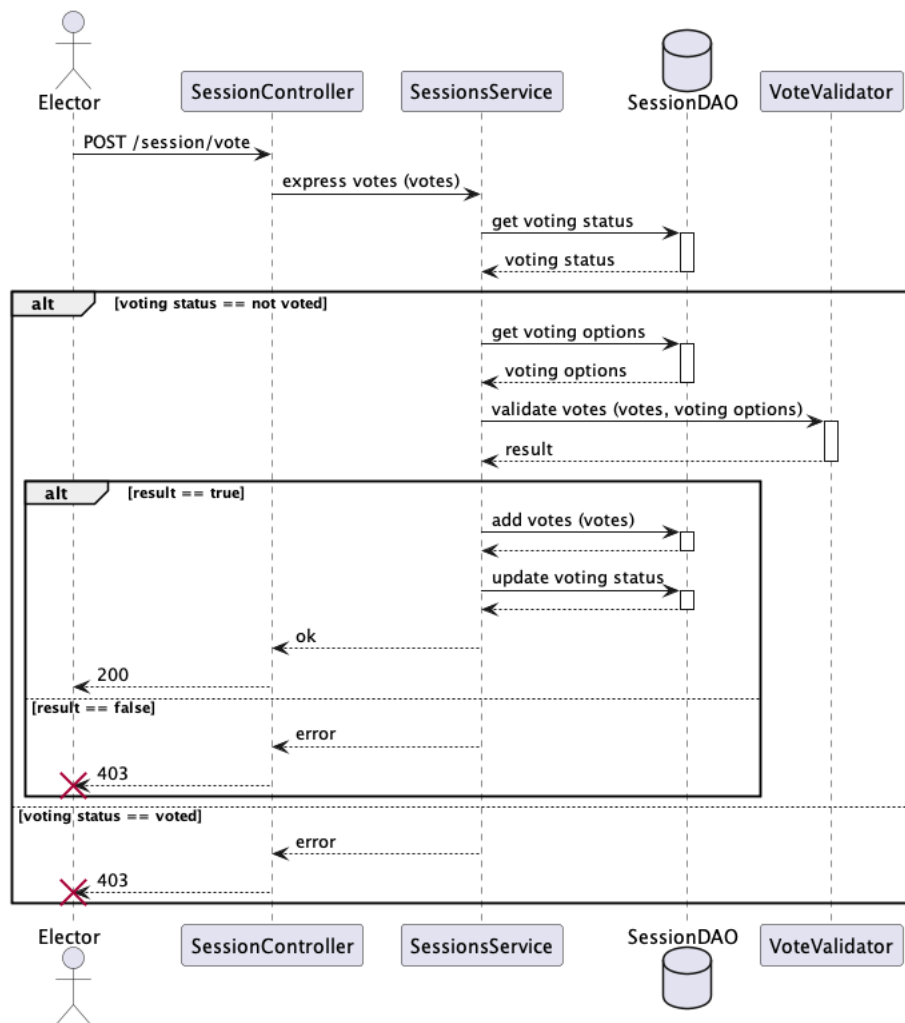


Figure 12 Espressione di un voto

3.4.7 Espressione Voto – lato client elector

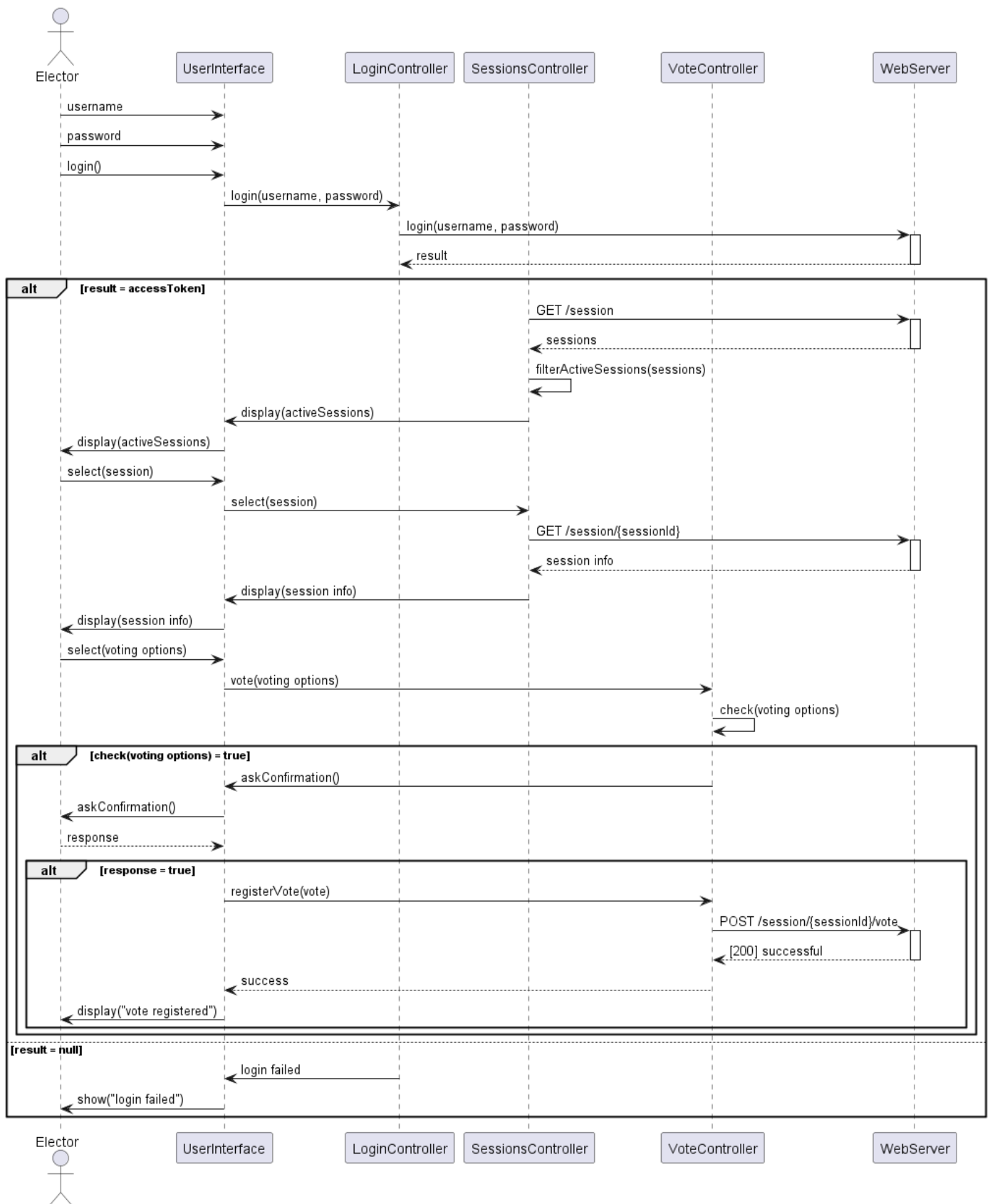


Figure 13 Espressione voto, lato client elector

3.4.8 Determinazione Vincitori di una Sessione di Voto

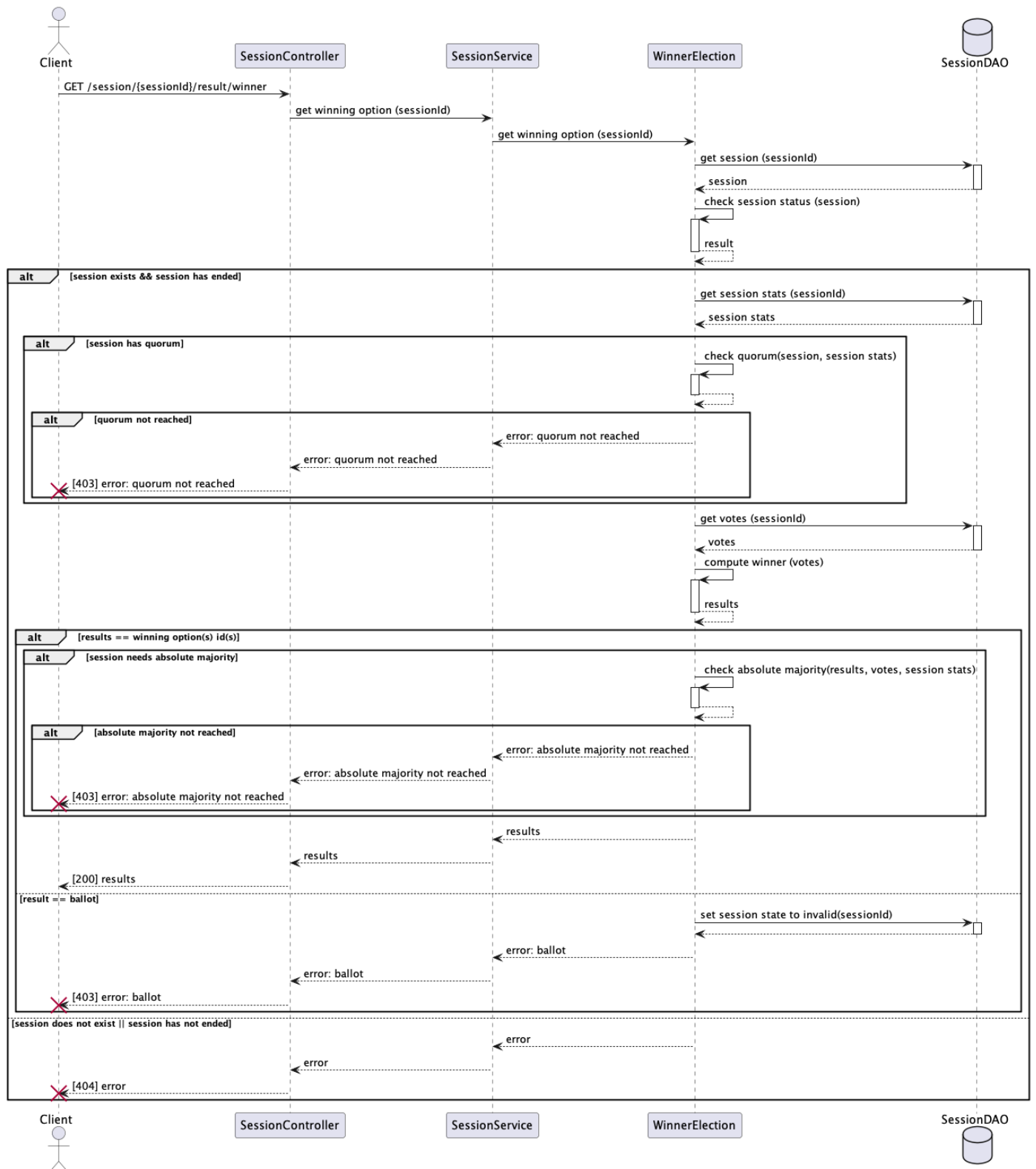


Figure 14 Determinazione Vincitori

3.5 Diagramma delle attività

3.5.1 Creazione di una sessione di voto (Client)

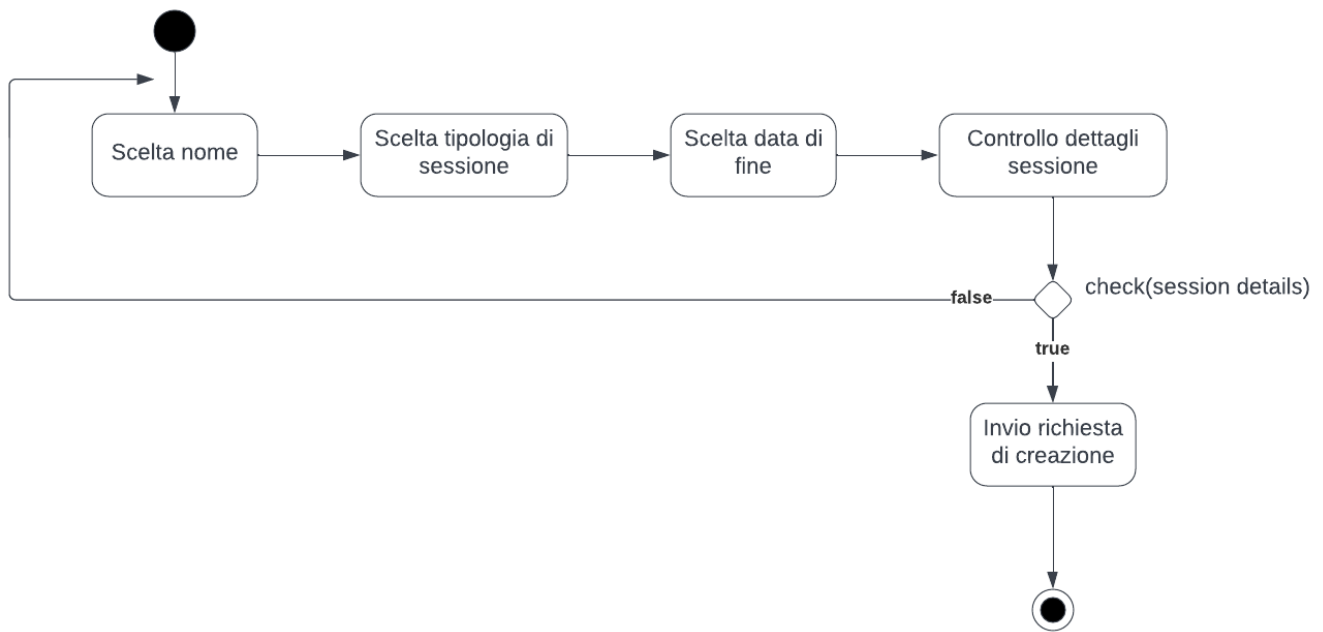


Figure 15 Diagramma di attività per la creazione di una sessione di voto – lato Client

Il processo descritto in figura 15 è implementato nella classe `NewSessionController` del client manager. L'invio della richiesta finale fa riferimento all'api `POST /session`.

3.5.2 Determinazione Vincitori di una Sessione di Voto

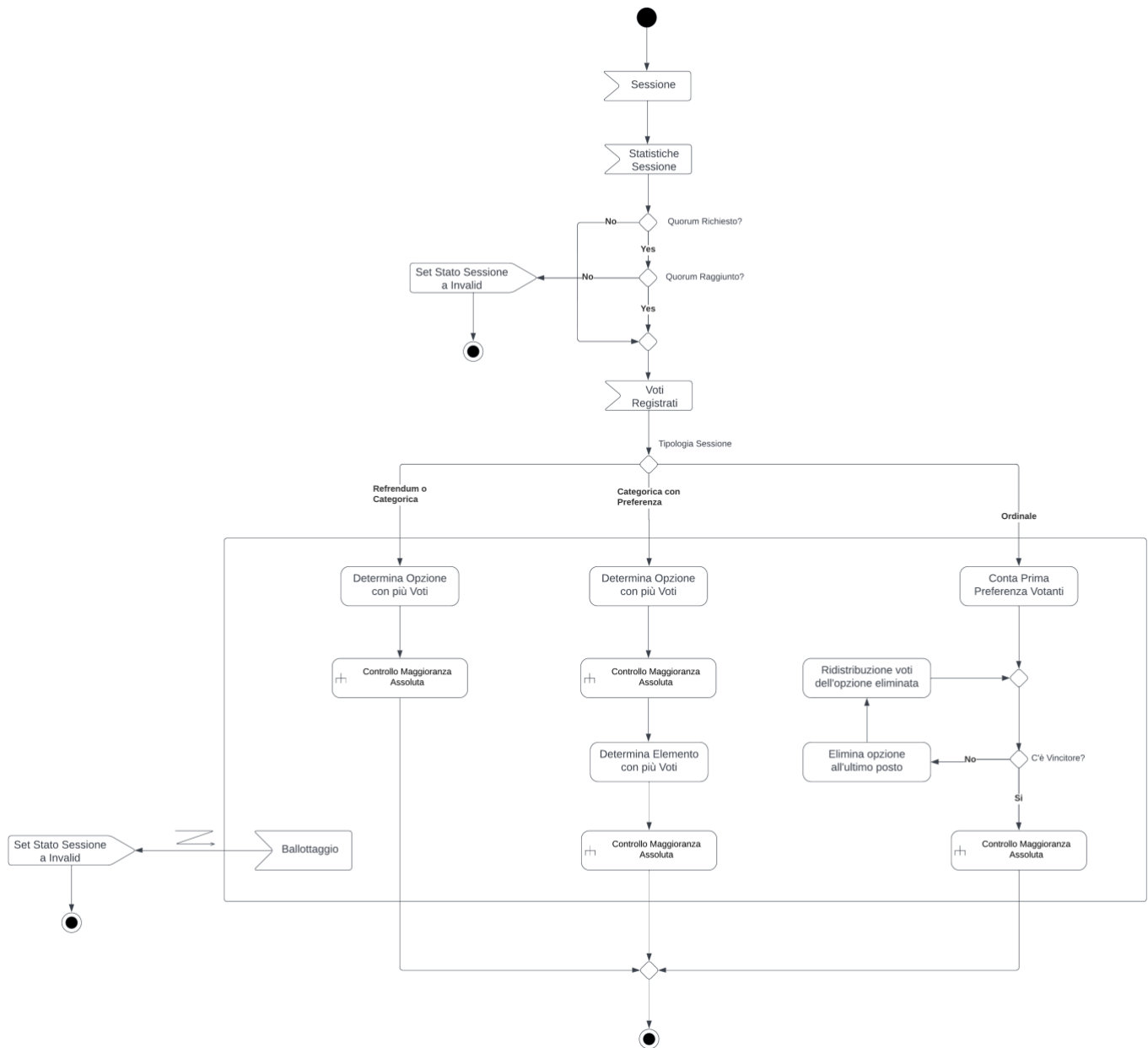


Figure 16 Determinazione del vincitore

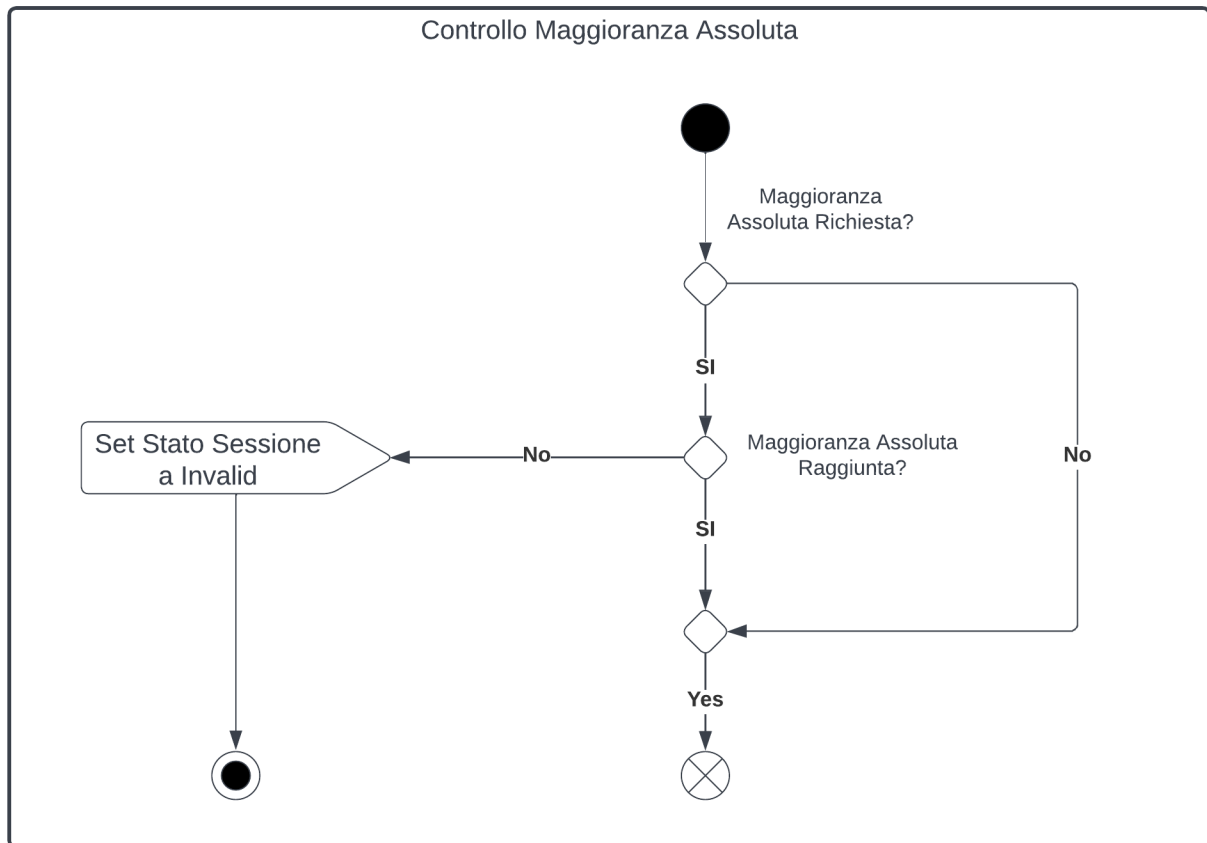


Figure 17 controllo del raggiungimento della maggioranza assoluta

Il processo descritto dai diagrammi in Figura 16 e Figura 17 si svolge nei metodi della classe WinnerElection. I messaggi inviati e ricevuti indicano informazioni scambiate attraverso l'interfaccia SessionDAO.

3.6 Macchine di stato

3.6.1 Stati della Sessione di Voto

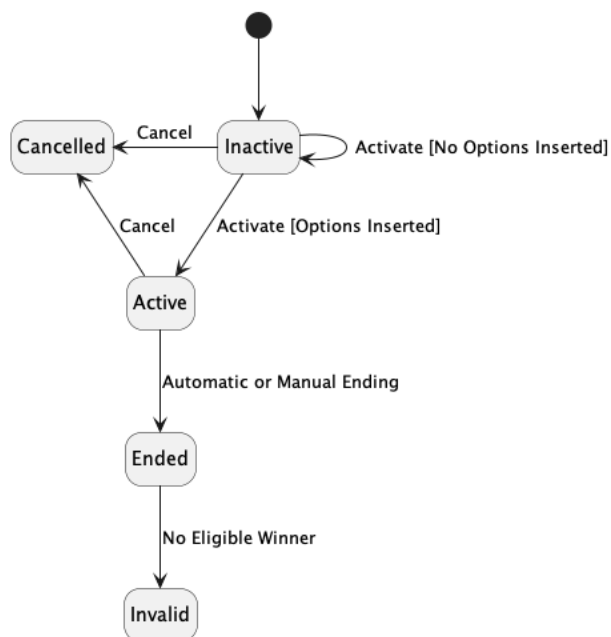


Figure 18 stati di una sessione

4 Implementazione del sistema

L'intero sistema di voto, quindi tutte e tre le sue componenti, sono state realizzate in Java, in particolare Java 18.

Il server è monolitico, quindi tutte le operazioni sono eseguite da moduli differenti, ma eseguiti dallo stesso processo. Per lo sviluppo del server è stato utilizzato il framework Spring, un framework consolidato e ampiamente utilizzato per la realizzazione di servizi accessibili tramite Internet.

In particolare, del framework, sono state utilizzate le seguenti componenti:

- Spring Boot: consente di fare il bootstrap di un http application server con tutte le dipendenze e componenti necessari già inclusi.
- Spring Security: consente di gestire gli aspetti di sicurezza di una applicazione Spring. Mette a disposizione diverse strategie per gestire in modo granulare l'autenticazione e autorizzazione degli utenti dell'applicazione. La strategia scelta è quella degli Access Token JWT.
- Spring Data JDBC: mette a disposizione strumenti integrati con il framework per accedere a database tramite i driver JDBC. Il driver specifico utilizzato è quello per MySQL, il database scelto come soluzione per la persistenza dei dati.

Il server così sviluppato è autoritativo su tutte le operazioni eseguibili nel contesto del sistema di voto, ed espone le sue funzionalità all'esterno tramite interfacce http REST.

Per lo sviluppo dei client e la relativa interfaccia grafica è stata utilizzata la libreria JavaFX che consente una semplice implementazione del modello Model, View, Controller (MVC).

Data la natura del server, il ruolo dei client è solo quello di tramite tra l'utilizzatore, quindi elettore o gestore, e il server; pertanto, non contiene alcuna logica di business, ma si limita ad eseguire il controllo degli input per migliorare l'esperienza utente e l'efficienza generale del sistema, andando ad evitare chiamate al server che risulterebbero in errori prevedibili.

Tutte le operazioni eseguite, sia sul server che sul client, sono tracciate attraverso le librerie di logging Slf4j e Log4j2. Tutte le informazioni sensibili utilizzate nelle varie operazioni vengono offuscate prima della costruzione dei log.

La gestione del progetto con relative dipendenze, quindi installazione, compilazione ed esecuzione, è gestita tramite il build tool Apache Maven. Lo storico delle modifiche ed evoluzione del progetto è gestito mediante una monorepo Git, che quindi include i tre progetti per tutte le componenti del sistema.

4.1 Diagrammi delle Classi di Programma

4.1.1 *Server*

Vedi allegato `diagramma_programma_server.png`

4.1.2 *Client Elector*

Vedi allegato `diagramma_programma_elettore.png`

4.1.3 *Client Manager*

Vedi allegato `diagramma_programma_gestore.png`

4.2 Discussione dei pattern utilizzati

4.2.1.1 MVC

Tutte le classi che finiscono con Controller fanno riferimento a questo pattern. I client utilizzano solo le view, astratta in parte attraverso JavaFX, e i controller per la corretta visualizzazione dei dati, il Model non è fornito ai client in quanto la logica è lasciata a lato server. A lato server troviamo, quasi in modo complementare ai client, dei controller, che forniscono punti di accesso per i client, e delle classi di modello che si occupano di gestire la logica vera e propria del sistema.

4.2.1.2 DAO

Il pattern DAO è stato utilizzato per isolare l'accesso al DB attraverso un insieme di interfacce (DAO) e le rispettive implementazioni (DAS), consentendo una semplice manutenibilità e interazione. È stato dedicato un package (db) sul server per il contenimento delle classi legate a questo compito, le classi DAO e DAS sono facilmente distinguibili in quanto citati nel nome della classe.

4.2.1.3 Observer

L'interfaccia costruita attraverso JavaFX utilizza pesantemente il pattern Observer, fornendo uno strato intermedio tra l'interfaccia e il programmatore che consiste nell'ascoltare i click dell'utente (e in generale tutte le sue azioni) e notificarli se coinvolgono dei nodi interessati consentendo al programmatore di intervenire quando questi si verificano, rendendo così l'interfaccia interattiva.

4.2.1.4 Strategy

L'utilizzo di questo pattern è presente in VoteValidator, questo fornisce al suo interno un'interfaccia Validator e quattro sue implementazioni, ognuna di queste specifica se un insieme di opzioni rispetta le proprietà legate al tipo di sessione di voto che il validatore vuole controllare. Questi validatori sono successivamente utilizzati per controllare che il formato di voto del client si conforme al tipo di sessione di voto scelta.

4.3 Autenticazione e Autorizzazione

Questi processi vengono gestiti inserendo due filtri nella “filter chain” di spring security: uno che si occupa dell’autorizzazione e uno che si occupa dell’autenticazione. La filter chain viene attraversata per ogni richiesta ricevuta.

Il filtro di autenticazione viene invocato soltanto se la chiamata effettuata è all’endpoint /login in POST; per tutte le altre chiamate viene saltato e sostituito dal filtro di autorizzazione. L’autenticazione consiste nel controllo delle credenziali presenti nel body della richiesta; se queste sono valide allora viene restituito al chiamante un token JWT, altrimenti è restituito il codice 401.

Il filtro di autorizzazione verifica se la richiesta ha nell’header Authorization un token JWT. Se questo è presente, verifica che sia valido: in caso di token valido la richiesta viene accettata e questa prosegue lungo la filter chain; in caso di token invalido la richiesta viene bloccata e viene restituito il codice 401.

In base al token ricevuto, contenente il ruolo dell’utente, verrà poi consentito o meno l’accesso all’endpoint richiesto.

Le password degli utenti vengono memorizzate nel database solo in seguito alla loro cifrazione mediante l’algoritmo BCrypt a dieci passaggi con salt.

4.4 Logging

La funzionalità di logging è built-in a Spring Boot, il quale si basa sull’interfaccia Slf4j. Questa consente di utilizzare la stessa API per scrivere i log indipendentemente dall’implementazione del logger utilizzata.

Gli eventi di Spring sono loggati in modo automatico, pertanto sono stati scritti log soltanto per le operazioni direttamente collegate al dominio del sistema di voto.

Per il logging nei due applicativi client è stata utilizzata sempre l’interfaccia Slf4j con l’implementazione della libreria Log4j2.

La configurazione del logger è effettuata attraverso il file *application.properties* tramite cui è possibile personalizzare il formato dei log e i livelli utilizzati per ciascuna classe o package.

4.5 Testing

L’attività di testing è stata svolta con il supporto del framework JUnit. Tramite questo sono stati realizzati test unitari seguendo il criterio di copertura delle decisioni. È stato così possibile verificare il corretto comportamento del sistema in tutti i casi di esecuzione principali, risultanti dai diversi esiti di ogni decisione.

Il testing si focalizza principalmente sulla procedura di determinazione del vincitore di una sessione di voto.

I casi di test trattati si trovano nella classe *VotingSessionTests*. Per effettuare il testing della funzionalità è stato necessario creare dei mock delle procedure di accesso al DB (*SessionDAO*) e del *SessionService*; questo per isolare le funzionalità da testare, dal resto della logica del programma. Lo scopo è evitare che le operazioni possano essere influenzate da fattori esterni, quali la disponibilità della connessione al database (oggetto di testing diverso da quello unitario).

4.6 Gestione Dati Persistenti

I dati persistenti sono memorizzati all'interno di un database MySQL. L'accesso a questi dati è ristretto al Server, che vi accede attraverso l'interfaccia JdbcTemplate, messa a disposizione da Spring Data Jdbc.

La connessione al database viene gestita in modo automatico da Spring. Nel file *application.yml* è specificata la configurazione per consentire a Spring di accedere al DB, quindi l'url e le credenziali.

L'utilizzo del JdbcTemplate prevede la scrittura manuale di tutte le query e gestione del mapping tra il contenuto del database e le classi Java. Questo approccio si sposa perfettamente con il pattern DAO e consente di creare interfacce di accesso al database finemente personalizzabili e quindi ottimizzabili, in modo da ridurre il carico sul server.

Così è possibile prelevare dal database solo le informazioni necessarie, che spesso si riducono a semplici id: una conseguenza dell'implementazione del modello REST.

4.6.1 Struttura database

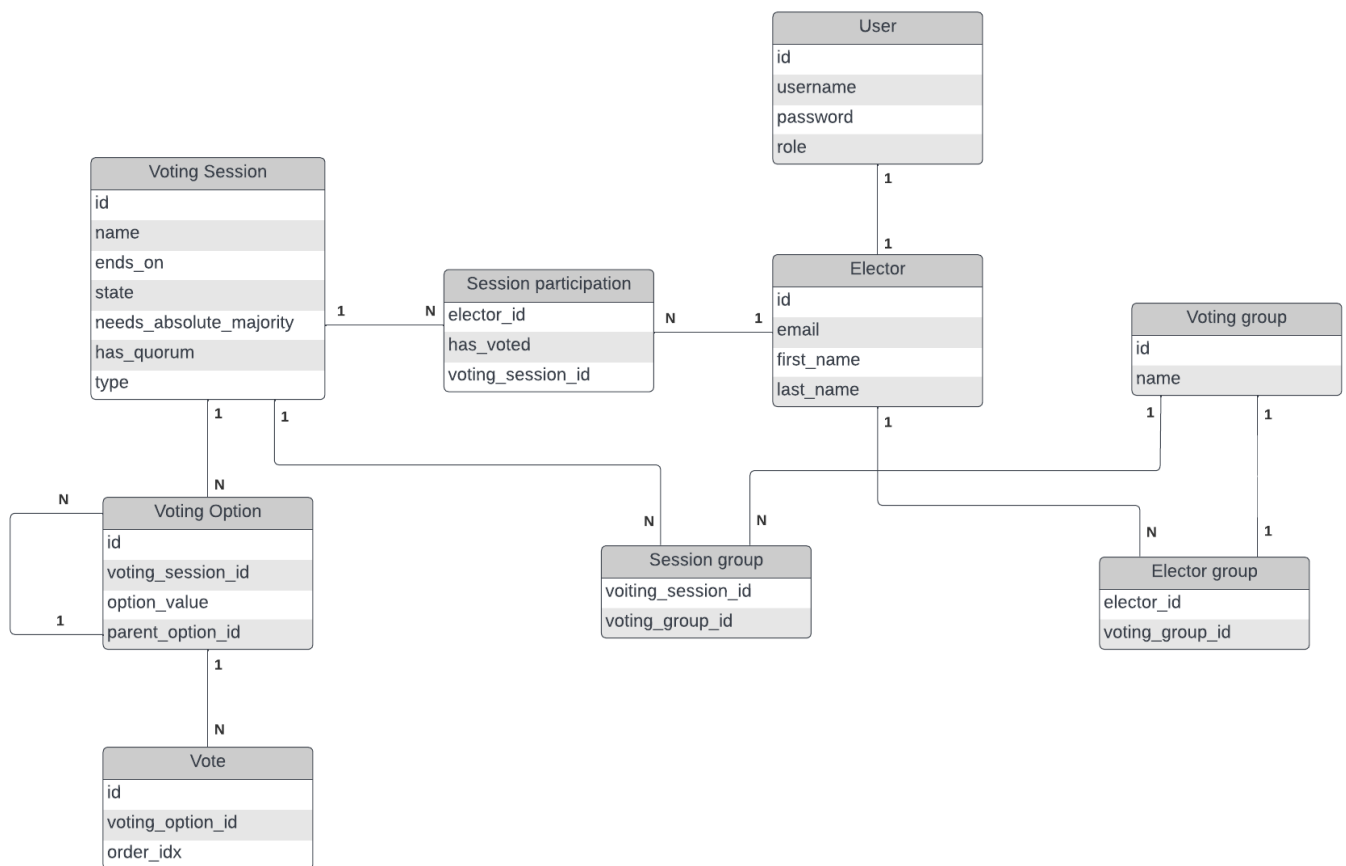


Figure 19 Modello ER

4.7 Descrizione dell'interfaccia grafica

Per lo sviluppo dell'interfaccia grafica si è optato di utilizzare JavaFX con il tool SceneBuilder per rendere più veloce e flessibile il layout delle componenti di ogni vista.

Il principio utilizzato per progettare l'interfaccia è stato “Out of mind out of sight” ovvero mantenere visibili solo e soltanto componenti necessarie all'utente e nulla di superfluo che possa confondere o distrarre utenti meno esperti, con pop-up di conferma nelle operazioni più critiche progettati per evitare incidenti quanto possibile.

Ogni campo in cui è possibile inserire testo libero è controllato dal server e viene eseguito l'escaping del contenuto attraverso Spring Jdbc per evitare attacchi di tipo SQL Injection. Alcuni di questi campi prevedono anche controlli lato client il cui unico scopo è migliorare la user experience, data la completa autorità del server.

4.7.1 Interfaccia grafica Client Manager

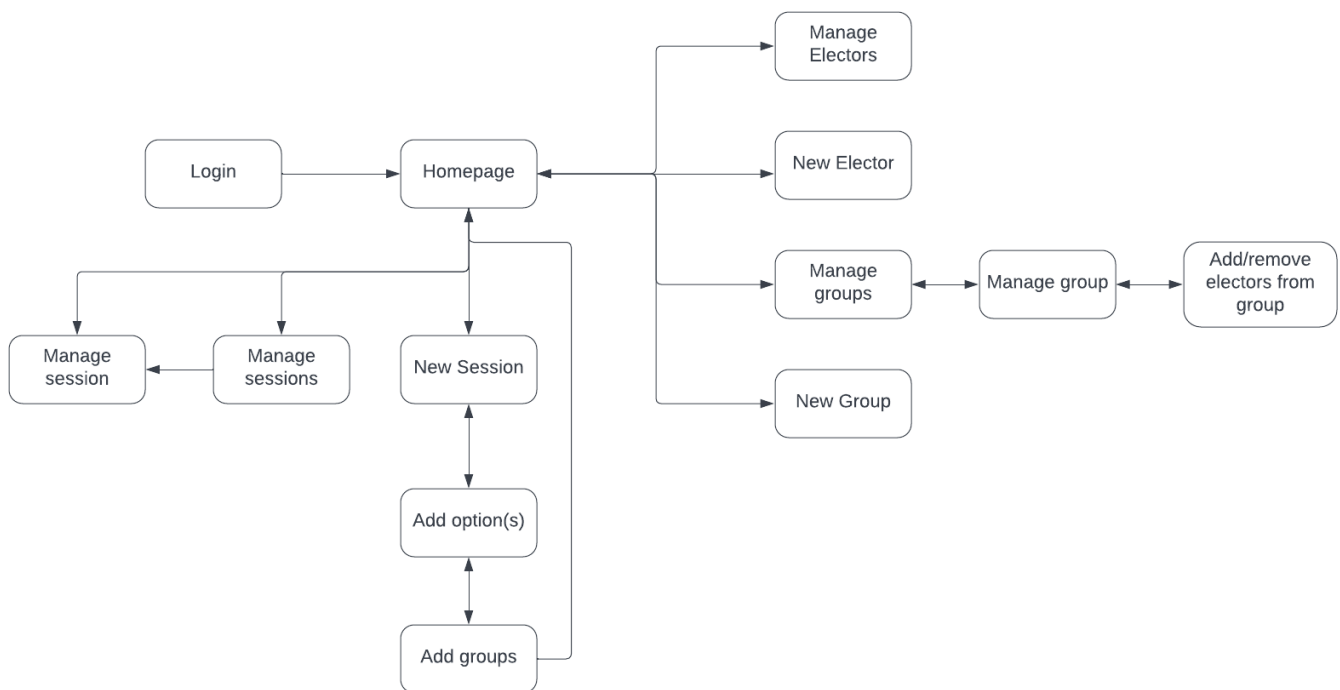


Figure 20 mappa di navigazione client Manager

4.7.1.1 Login

La schermata di login (Figura 21, condivisa anche dal client dell'elettore) è pensata per essere il più intuitiva possibile, rendendo possibile il solo utilizzo dei due campi per le credenziali, marcati con un prompt per guidare correttamente l'utente, e un bottone per tentare il login. In caso di credenziali errate un messaggio pop-up avvertirà l'utente e verrà invitato a riprovare.

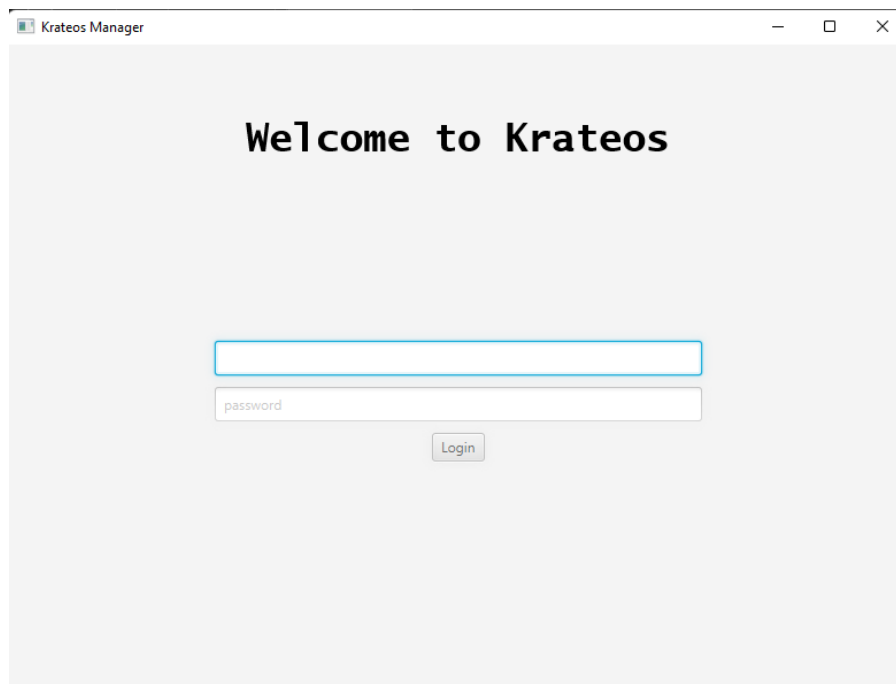


Figure 21 login client Manager

4.7.1.2 Homepage

La homepage (Figura 22) è la schermata di riferimento che contiene subito a vista le attuali sessioni attive, permettendo di analizzarle singolarmente selezionandole e cliccando sul bottone "open", e tre set di bottoni che dividono i campi nel quale l'amministratore può operare: elettori, gruppi e sessioni. I bottoni marcati dal "+" simbolizzano l'aggiunta, mentre quelli marcati da "manage" indicano la gestione delle entità già esistenti.

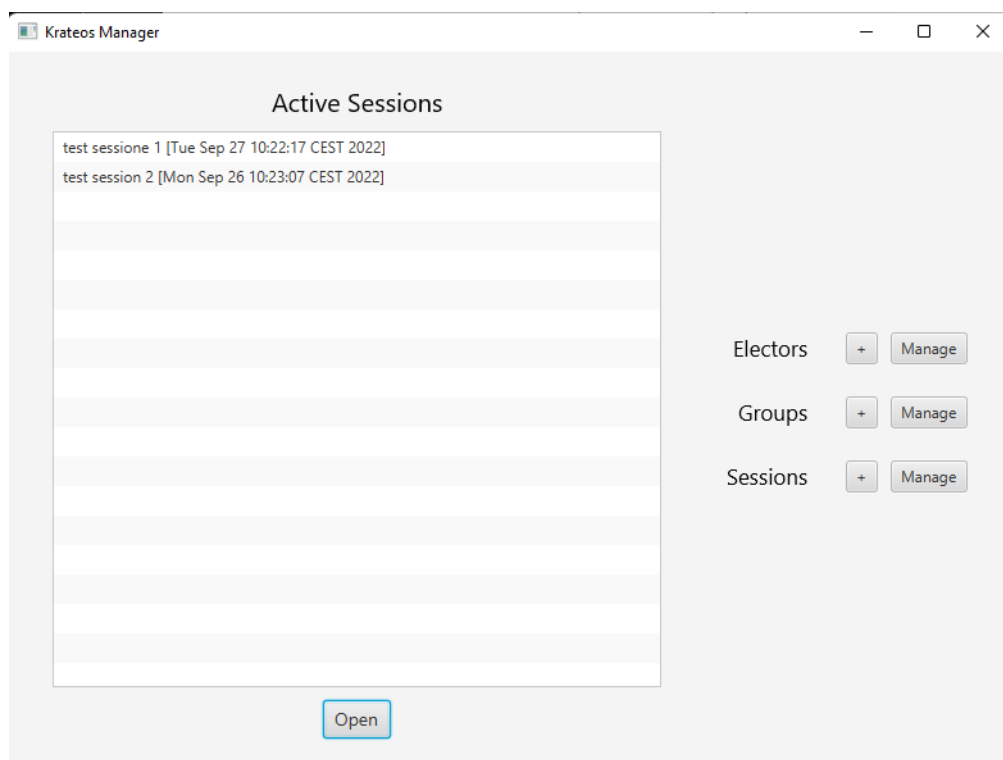
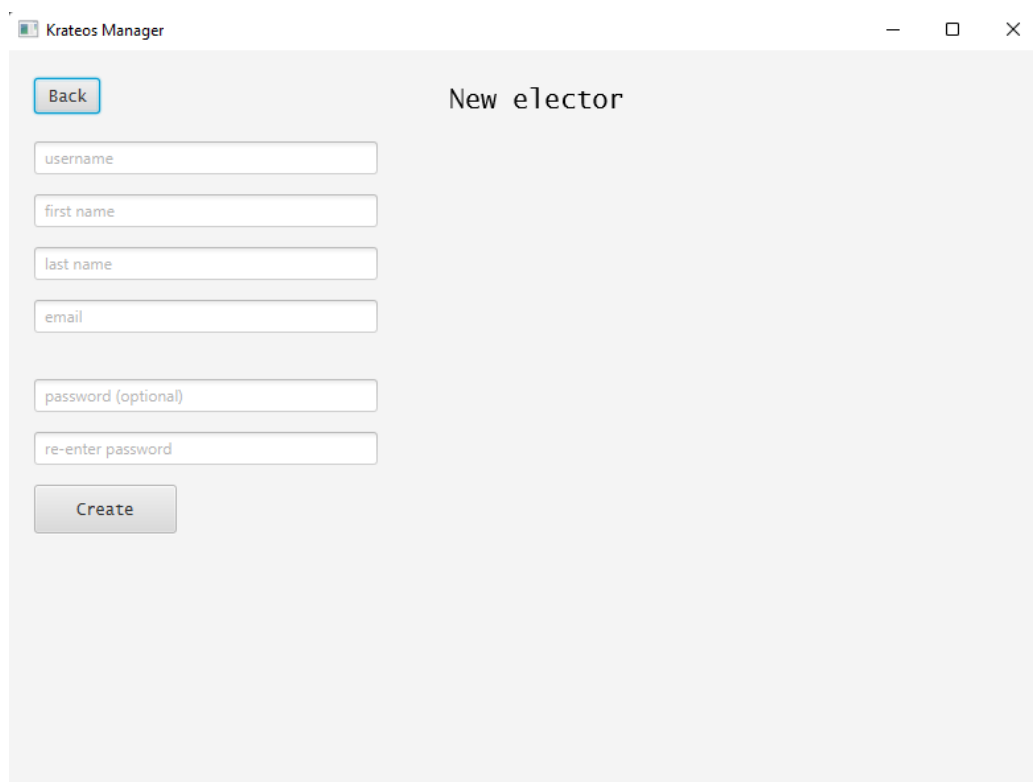


Figure 22 Homepage

4.7.1.3 Elettori

4.7.1.3.1 Aggiunta

L'aggiunta di un elettore (Figura 23) è presentata da un insieme di campi obbligatori che richiedono tutte le informazioni necessarie riguardanti l'elettore che vogliamo aggiungere.

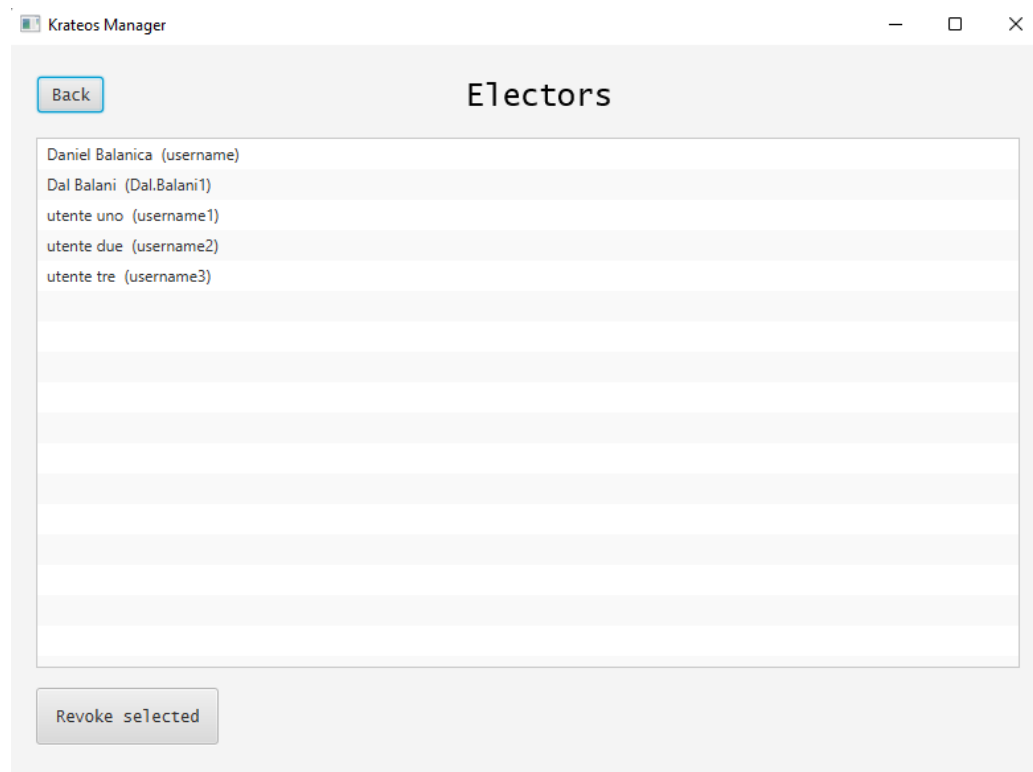


The screenshot shows a web application window titled "Krateos Manager". Inside, there is a form titled "New elector". The form contains several input fields: "username", "first name", "last name", "email", "password (optional)", and "re-enter password". A "Back" button is located at the top left of the form area, and a "Create" button is at the bottom left. The form is set against a light gray background.

Figure 23 Aggiunta di un elettore

4.7.1.3.2 Gestione

La gestione degli utenti (Figura 24) consiste semplicemente nella possibilità di revocare le credenziali di ognuno.



The screenshot shows a web application window titled "Krateos Manager". Inside, there is a page titled "Electors". At the top left, there is a "Back" button. Below it is a table with the following rows:

Daniel Balanica (username)
Dal Balani (Dal.Balani1)
utente uno (username1)
utente due (username2)
utente tre (username3)

At the bottom left of the table, there is a "Revoke selected" button.

Figure 24 Gestione elettori

4.7.1.4 Gruppi

4.7.1.4.1 Aggiunta

La seguente è una delle interfacce più complesse (Figura 25), in quanto composta da due liste, quella a sinistra comprende tutti gli elettori che non fanno parte del gruppo che vogliamo creare, quella a destra comprende tutti gli elettori aggiunti al gruppo che vogliamo creare.

È obbligatorio aggiungere un nome identificativo al nuovo gruppo.

Krateos Manager

Back

New group

gruppo esempio

Available electors

utente uno (username1)

utente tre (username3)

Add

Added electors

Daniel Balanica (username)

Dal Balani (Dal.Balani1)

utente due (username2)

Remove

Create

Figure 25 Creazione di un gruppo con elettori

4.7.1.4.2 Gestione

Per la gestione dei gruppi abbiamo adottato tre interfacce, la prima per elencare tutte i gruppi (Figura 26), permettendo all'amministratore di scegliere quale gestire. La seconda elenca gli elettori di uno specifico gruppo (Figura 27) e permette di modificare il gruppo selezionato spostandosi sulla terza interfaccia, utilizzata per la modifica del gruppo, che permette di aggiungere o togliere elettori (Figura 28).

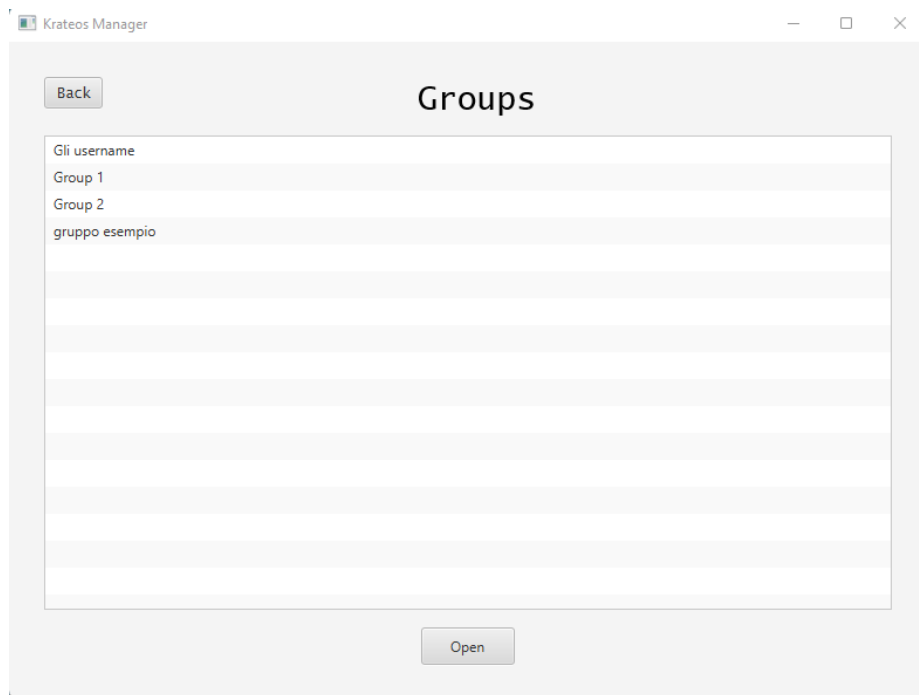


Figure 26 Visualizzazione di tutti i gruppi

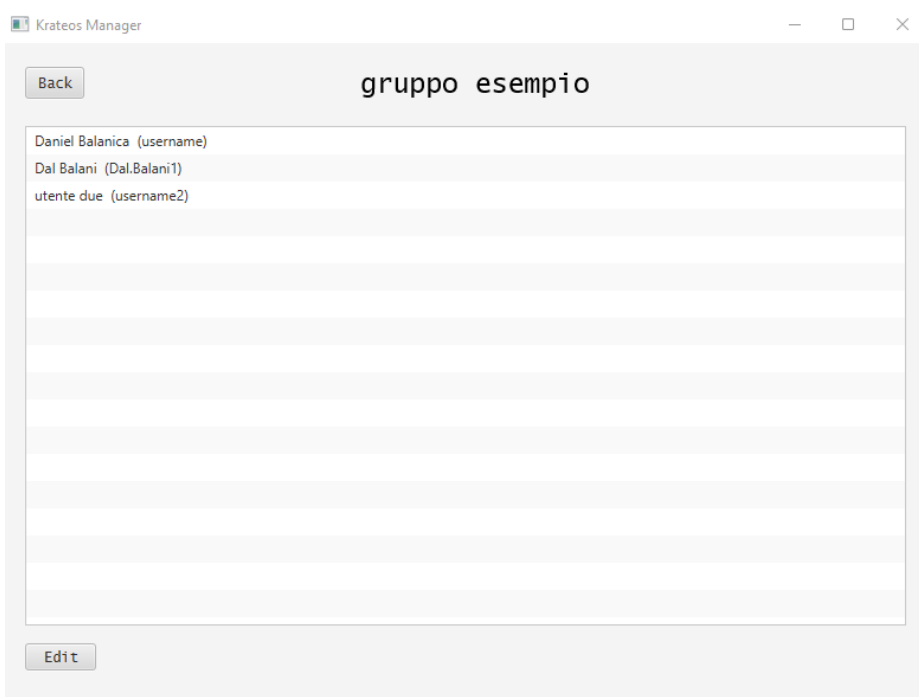


Figure 27 Elenco degli elettori di un singolo gruppo

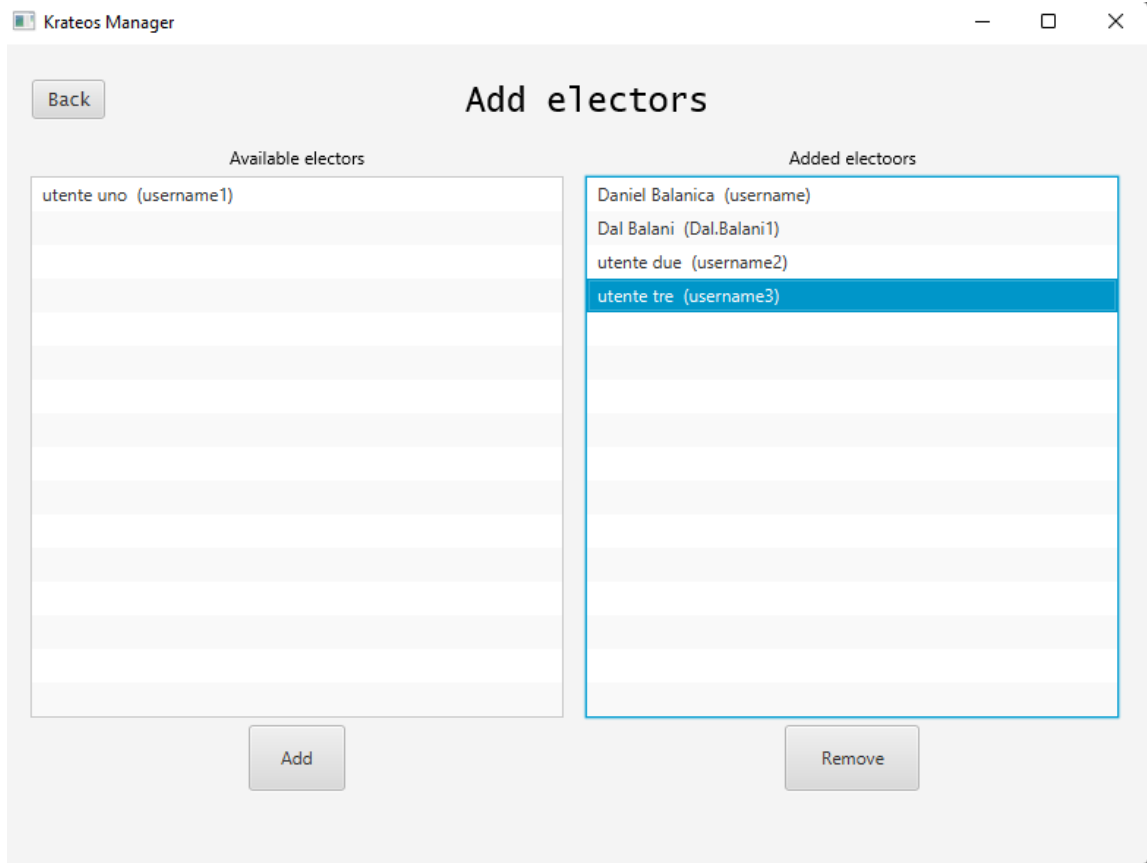


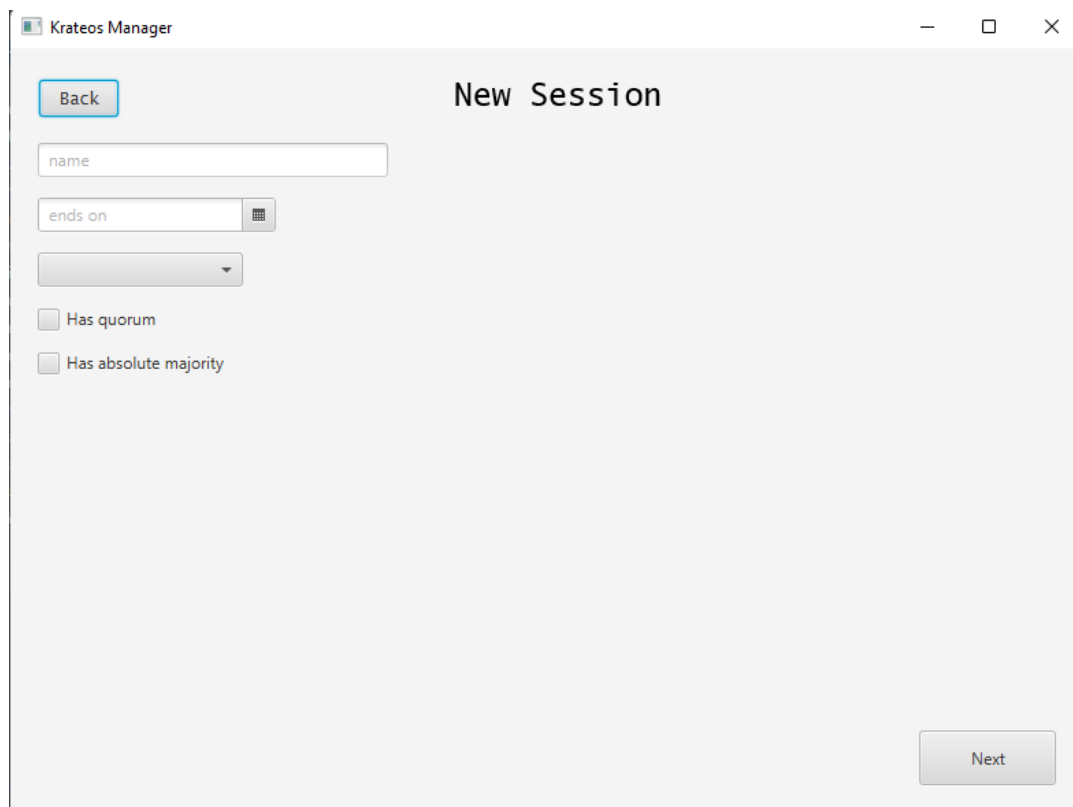
Figure 28 Modifica degli elettori di un gruppo

4.7.1.5 Sessioni

4.7.1.5.1 Aggiunta

La creazione di una sessione è suddivisa in 3 passaggi, di conseguenza diviso in 3 interfacce.

La prima (Figura 29) richiede l'inserimento di un nome, una data di scadenza, il tipo di sessione ed eventualmente selezionare il quorum e maggioranza assoluta. Se si seleziona una data nel passato viene comunicato un messaggio di errore. Ad eccezione della selezione di quorum e maggioranza assoluta tutti i campi sono obbligatori. Il secondo passo per la creazione di una sessione consiste nell'aggiunta delle opzioni di voto, l'interfaccia utilizzata in questo passaggio cambia in base alla tipologia di sessione, la più complicata delle quali è l'aggiunta di opzioni e elementi delle sessioni categoriche con preferenza (Figura 30), la quale richiama l'aggiunta di elettori ad un gruppo (Figura 28): una volta inserito il prompt, basta decidere se questo si vuole aggiungere come opzione o, selezionando a priori l'opzione padre, come elemento. Non è consentito lasciare la lista di opzioni vuote, dare un valore vuoto all'opzione o inserire opzioni duplicate. L'ultimo passaggio consiste nell'aggiunta dei gruppi che avranno diritto al voto nella sessione, una volta inseriti la sessione è creata in uno stato inattivo.



The screenshot shows a web application window titled "Krateos Manager" with a "New Session" form. The form includes a "Back" button at the top left. Below it are three input fields: "name", "ends on" (with a calendar icon), and a dropdown menu. At the bottom left, there are two checkboxes: "Has quorum" and "Has absolute majority". A "Next" button is located at the bottom right of the form area.

Figure 29 Aggiunta caratteristiche di sessione

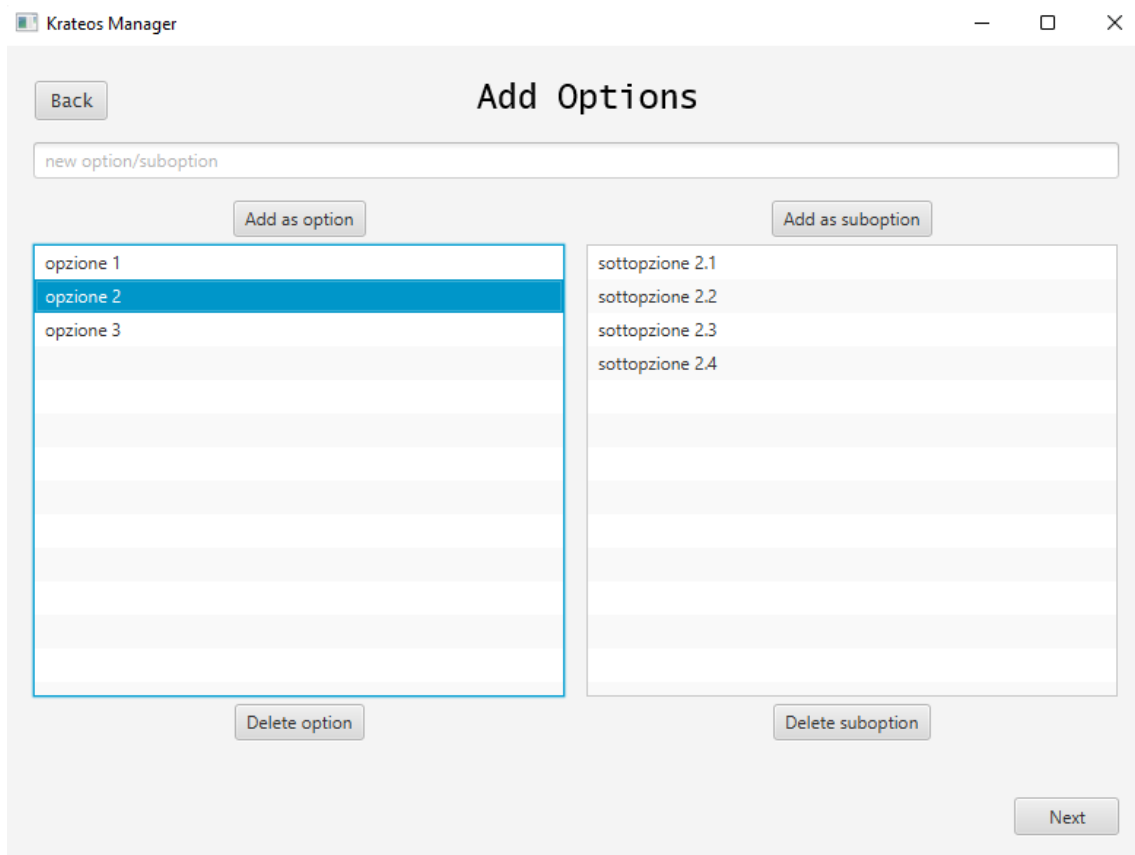


Figure 30 Aggiunta opzioni - sessione categorica con preferenza

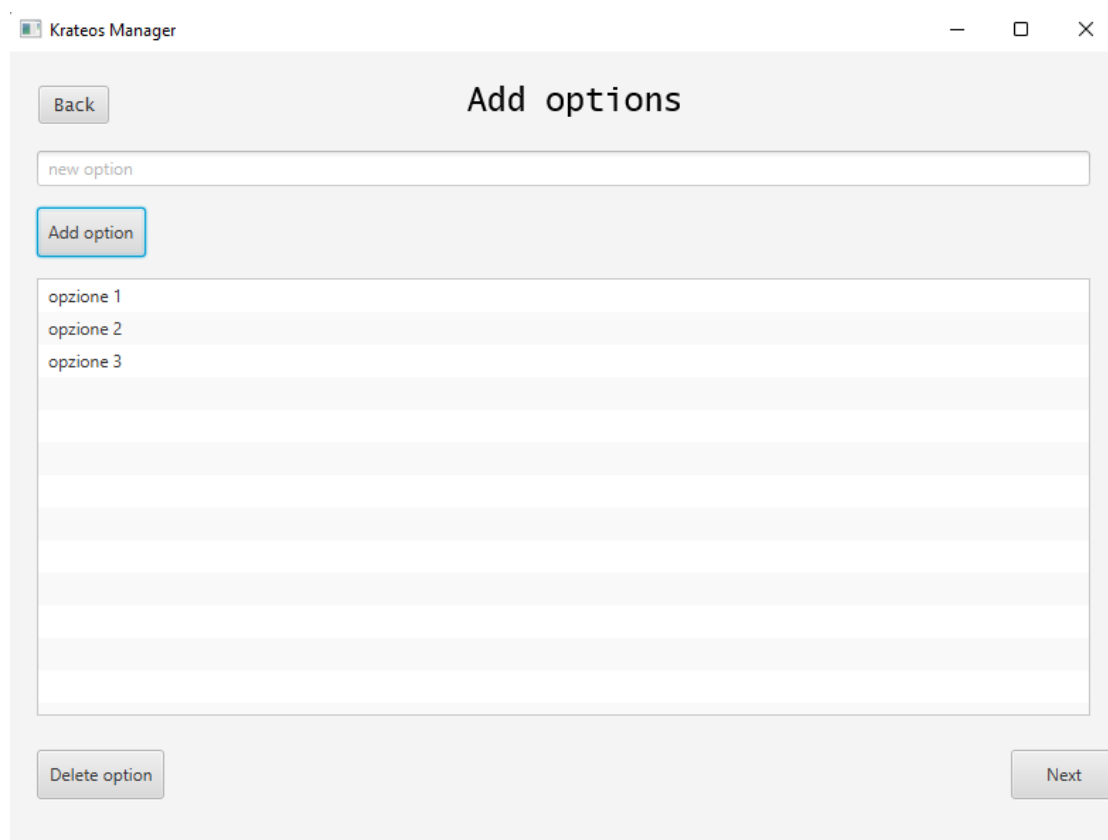


Figure 31 Aggiunta opzioni - sessione categorica/ordinale

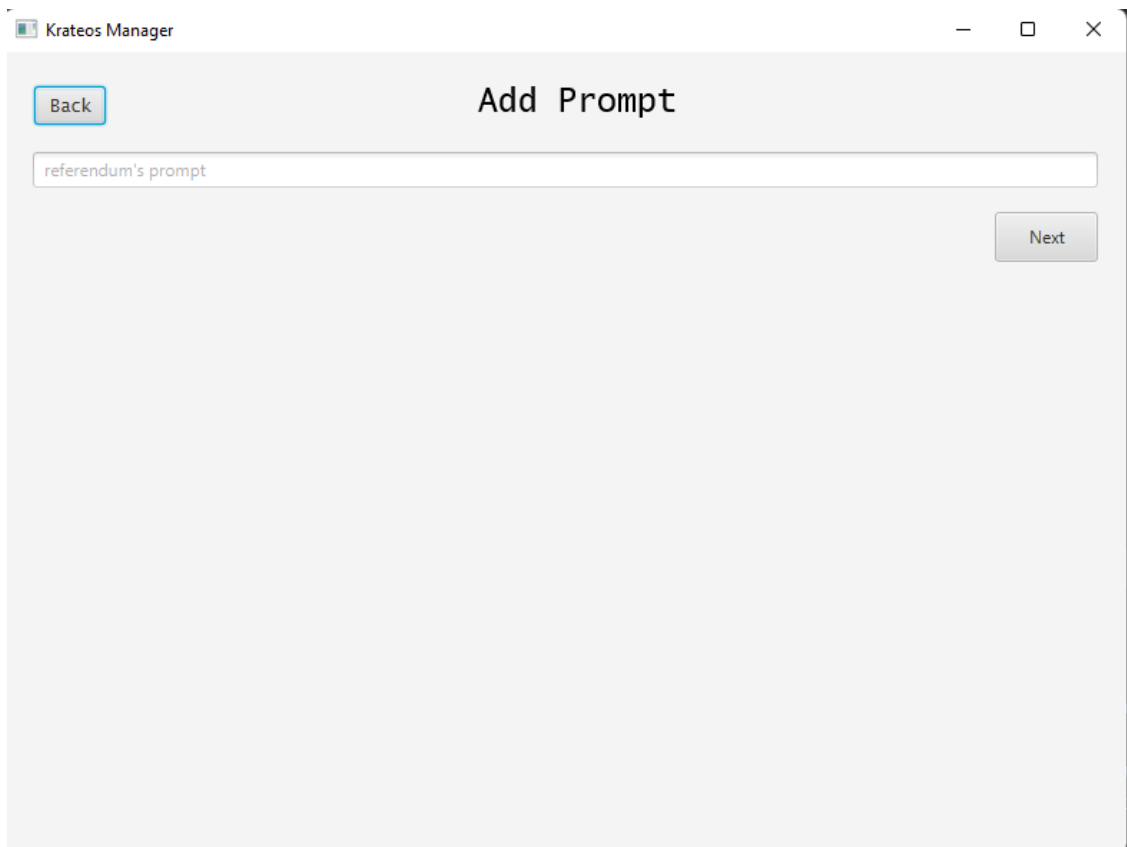


Figure 32 Aggiunta prompt - referendum

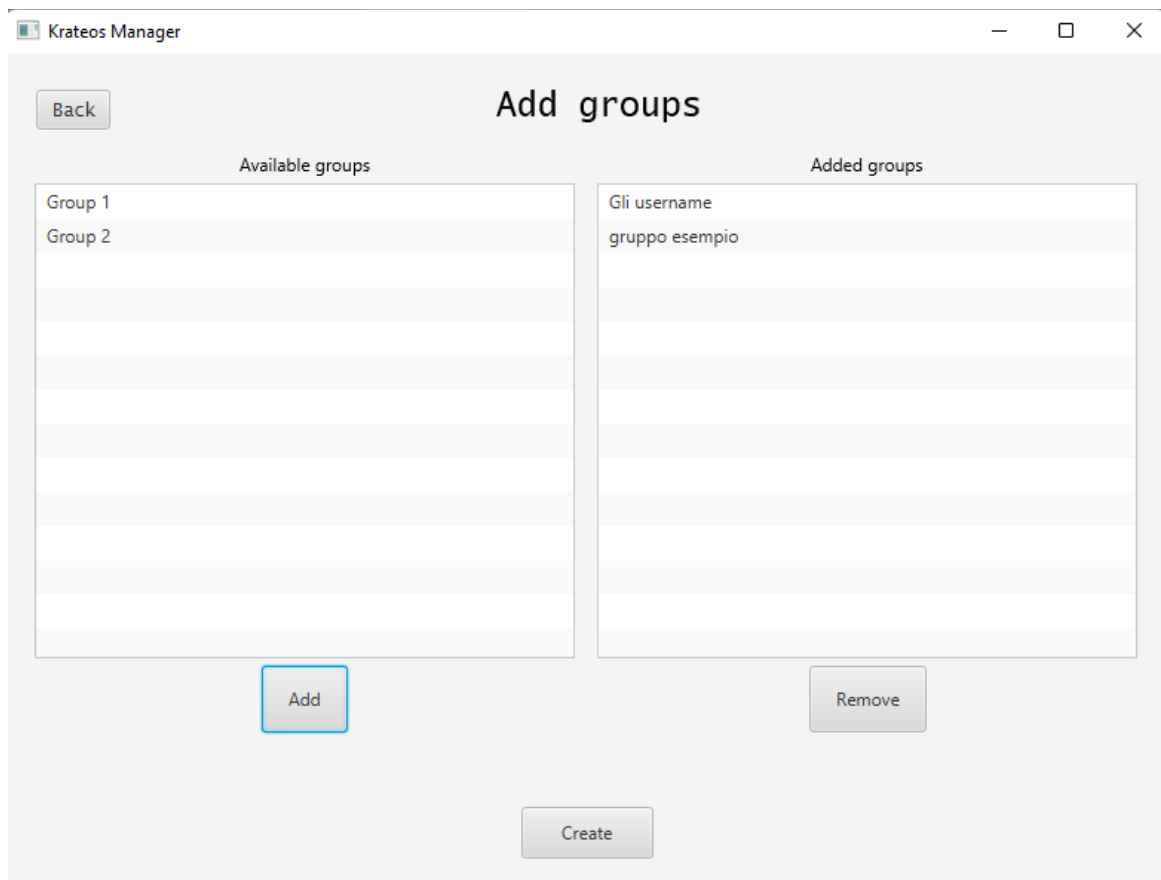


Figure 33 Selezione gruppi aventi diritto di voto

4.7.1.5.2 Gestione

La gestione delle sessioni è suddivisa in due interfacce, di cui una dinamica. La prima consiste nell'elencare tutte le sessioni create (Figura 34), indipendentemente dallo stato, da questa vista è possibile selezionare una sessione da aprire, per passare nella seconda vista che fornisce un numero di informazioni riguardanti la sessione selezionata (Figura 35). In quest'ultima interfaccia è possibile attivare, terminare, annullare ed eventualmente eliminare una sessione di voto. Lo stato della sessione è mostrato in alto, insieme al nome e alla data e ora di termine. Quando viene aperta una sessione terminata (ended) il resoconto dei voti (Figura 36) e il vincitore sono mostrati in questa interfaccia attraverso un grafico o una coppia di liste per le sessioni categoriche con preferenza, in caso di quorum o maggioranza assoluta non raggiunta un messaggio avverte l'amministratore di tale evento (Figura 37).

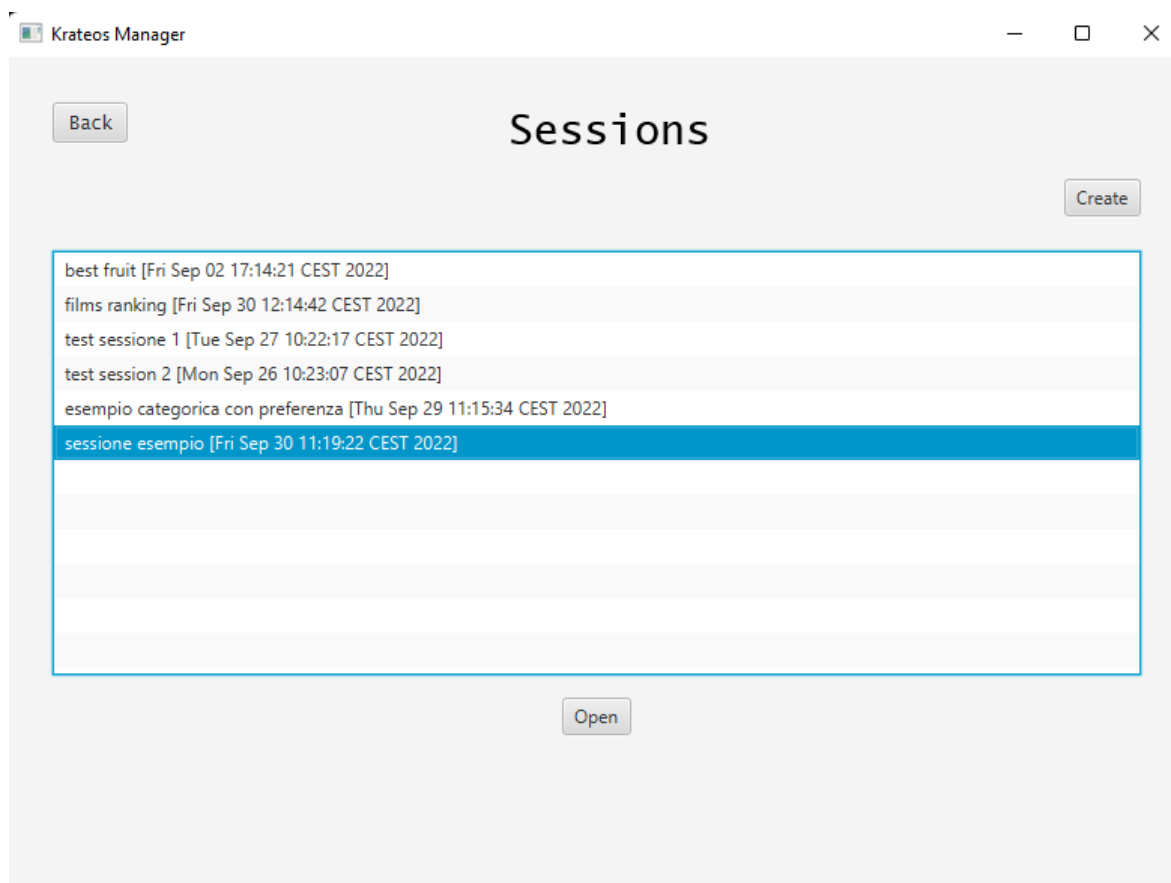


Figure 34 Visualizzazione di tutte le sessioni



Figure 35 Gestione sessione

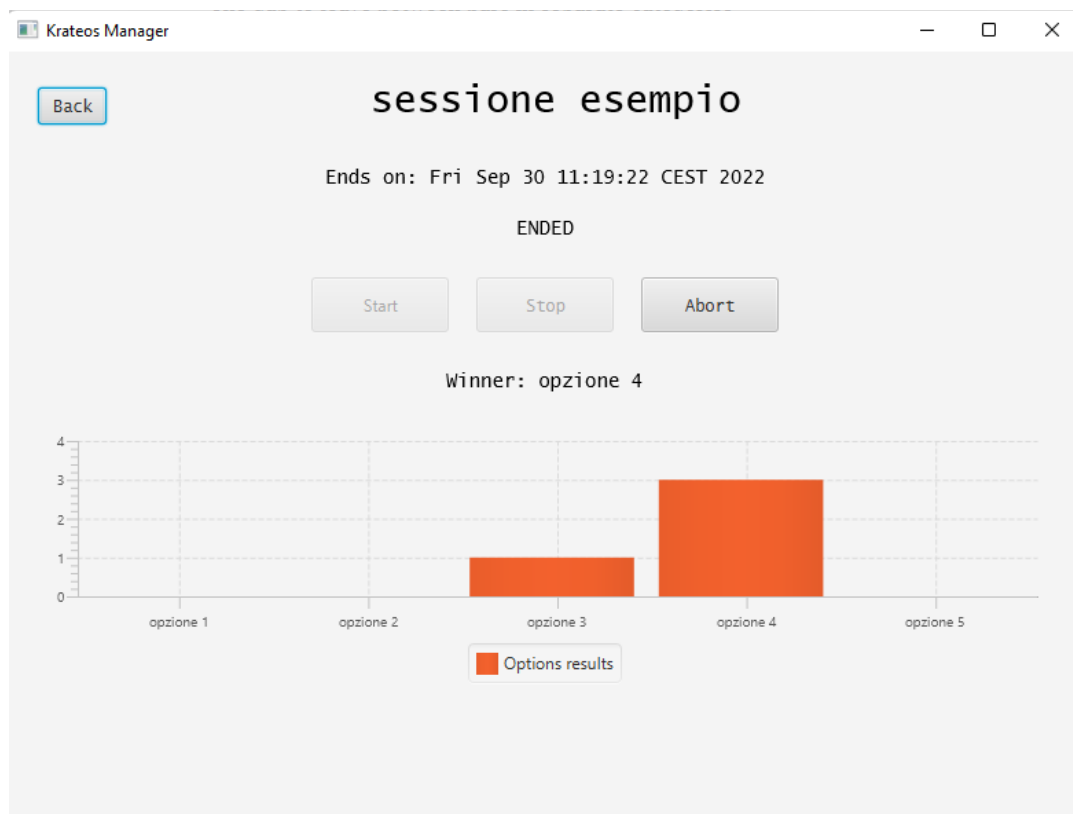


Figure 36 Gestione sessione - sessione terminata con calcolo dei voti e del vincitore

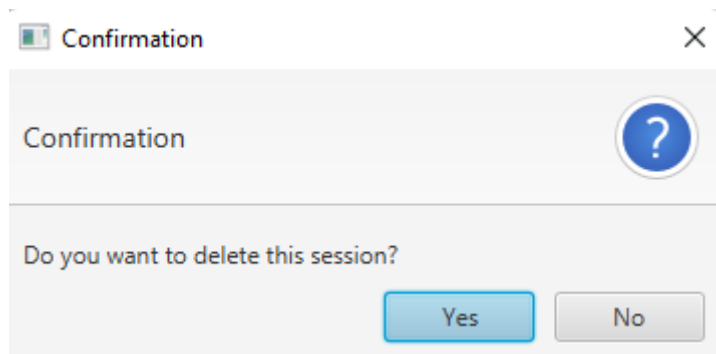


Figure 37 messaggio pop-up di conferma

4.7.2 Interfaccia grafica Client Elector

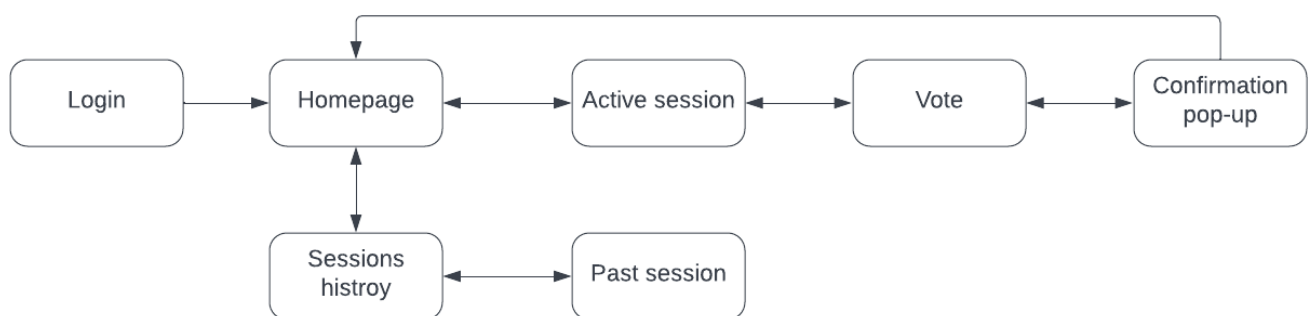


Figure 38 Mappa di navigazione del client Elector

3 Mappa di navigazione del client Elector

4.7.2.1 Login

L'interfaccia di login (Figura 39) dell'elettore condivide tutte le sue caratteristiche con l'interfaccia di login del manager (Vedi 5.7.1.1).

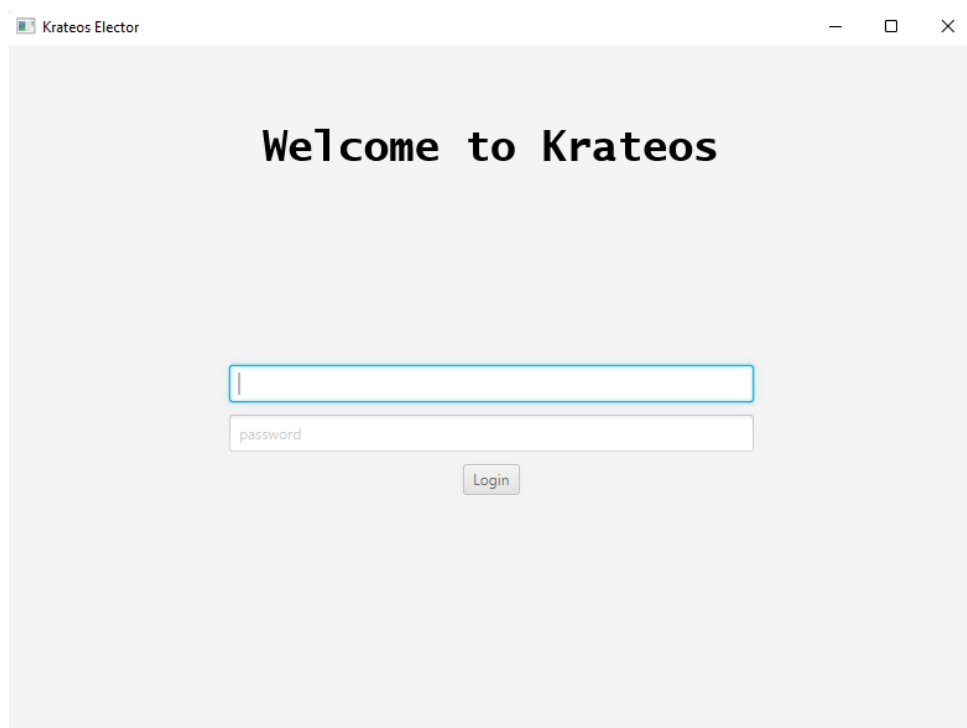


Figure 39 Login client elector

4.7.2.2 Homepage

L'homepage (Figura 40) è il punto di riferimento principale dell'elettore, qui potrà visualizzare tutte le sessioni di voto attive alla quale ha diritto di voto e potrà di conseguenza scegliere a quale votare oppure può decidere di visualizzare passare allo storico delle sessioni.

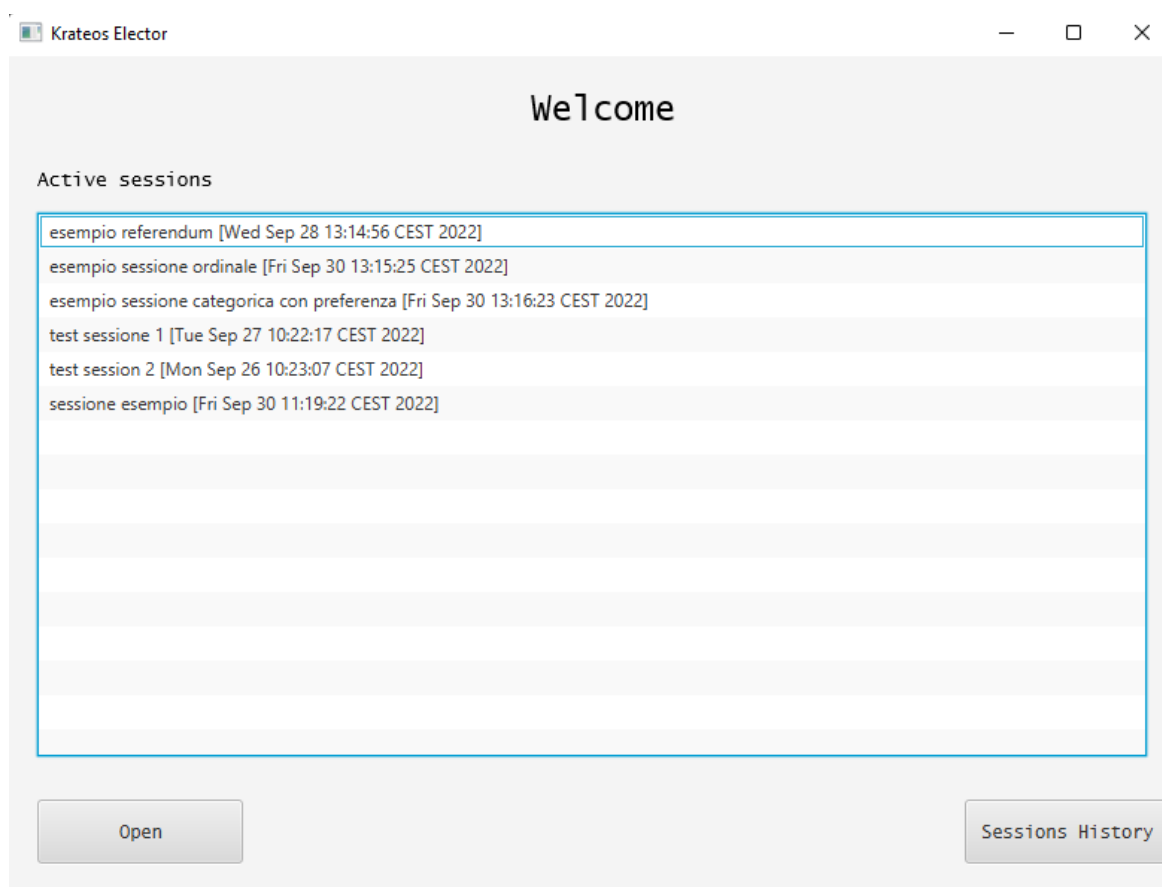


Figure 40 Homepage elector

4.7.2.3 Sessioni

Quando si procede a cliccare il bottone “Open” su una sessione nella vista Homepage si prosegue nella vista in Figura 41 che ci consente di visualizzare la data e ora di termine di questa sessione e ci permette di passare alla schermata di voto.



Figure 41 Informazioni sessione attiva

4.7.2.3.1 Voto

Le schermate di voto differiscono visivamente dipendentemente dal tipo di sessione alla quale vogliamo votare. Il voto categorico con preferenza (Figura 42) segue lo schema visto in precedenza della doppia lista: una volta selezionata un'opzione la lista di destra si aggiorna per visualizzare i suoi elementi; similmente il voto categorico (Figura 43) richiede semplicemente di selezionare l'opzione desiderata. Il voto ordinale (Figura 44) richiede l'utilizzo di due bottoni che permettono di spostare l'opzione selezionata di una posizione in alto o in basso per costruire l'ordine desiderato. Infine, il referendum (Figura 45) richiede la semplice selezione di una check-box per esprimere o meno il consenso sul tema trattato, non è consentita la scheda vuota.

Ogni volta che si tenta di esprimere un voto un pop-up di conferma (Figura 46) appare per una seconda conferma da parte dell'elettore per evitare incidenti durante il processo di votazione.

The screenshot shows a web application window titled "Krateos Elector". The main heading is "esempio sessione categorica con preferenza". There are two columns of options. The left column contains "opzione 1", "opzione 2", and "opzione 3". The right column contains "sottopzione 1.1", "sottopzione 1.2", and "sottopzione 1.3". The "sottopzione 1.2" is highlighted in blue. At the bottom, there is a "Vote" button. A "Back" button is located in the top left corner.

opzione 1	sottopzione 1.1
opzione 2	sottopzione 1.2
opzione 3	sottopzione 1.3

Figure 42 Voto categorico con preferenza

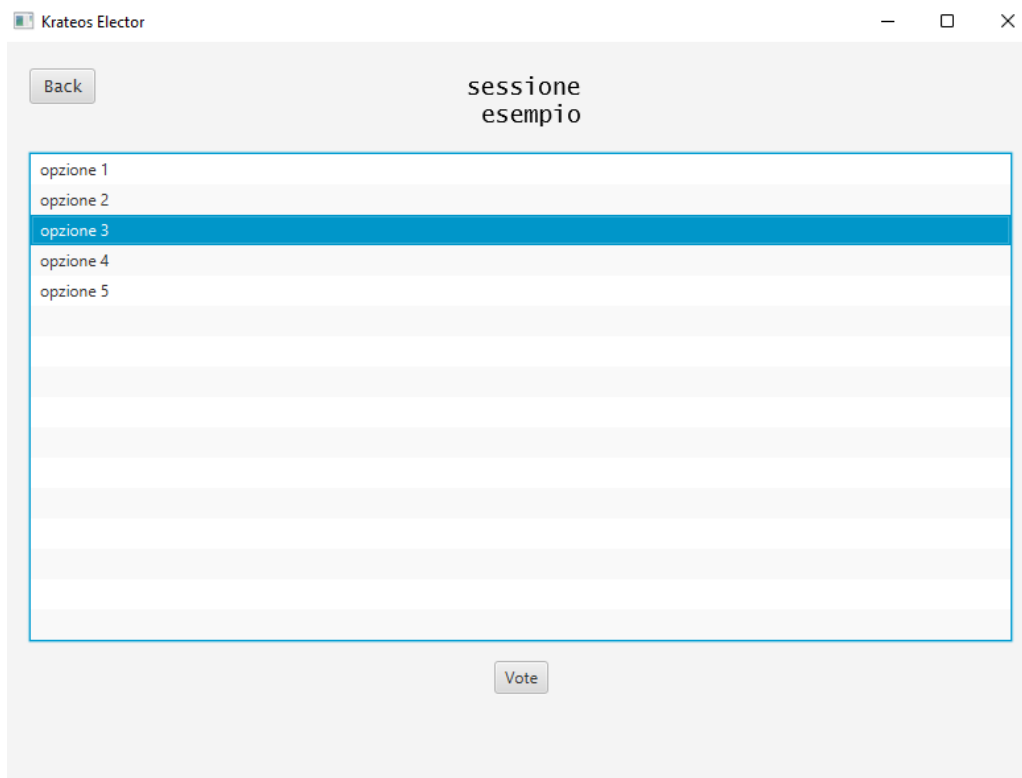


Figure 43 Voto categorico

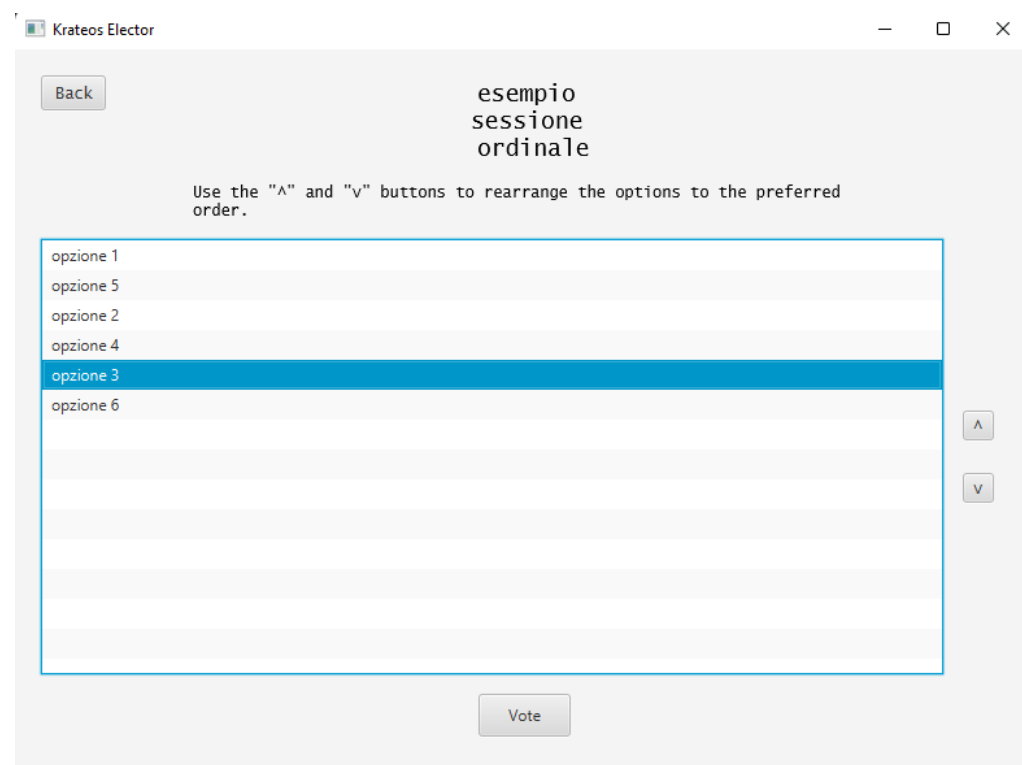


Figure 44 Voto ordinale



Figure 45 Voto referendum

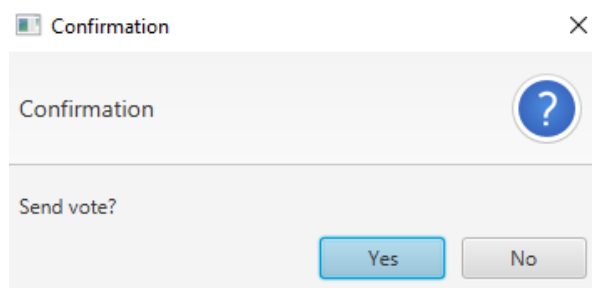


Figure 46 Conferma registrazione voto

4.7.2.4 Storico sessioni

Lo storico delle sessioni (Figura 47) consiste in una lista delle sessioni terminate, annullate o invalidate. È possibile selezionare una sessione per visualizzare l'esito di tale sessione con eventualmente il vincitore (Figura 48).

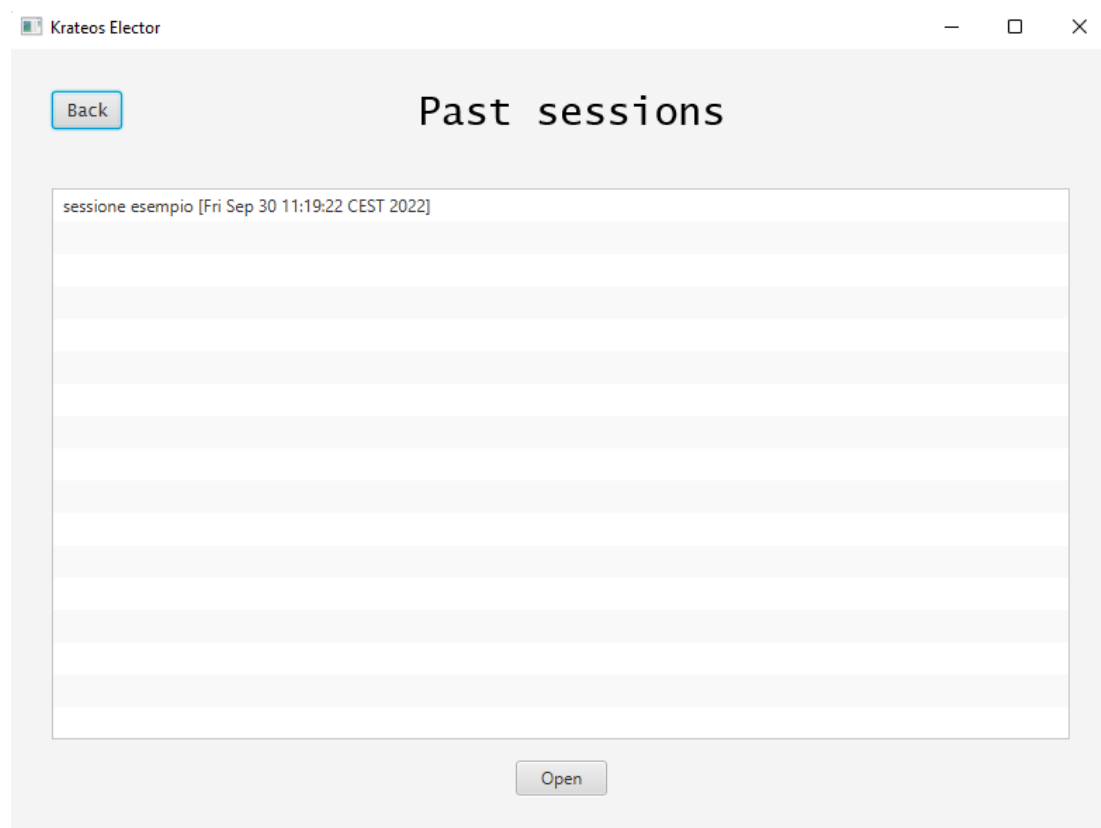


Figure 47 Storico sessioni



Figure 48 Informazioni sessione passata

4.8 Diagramma di deployment

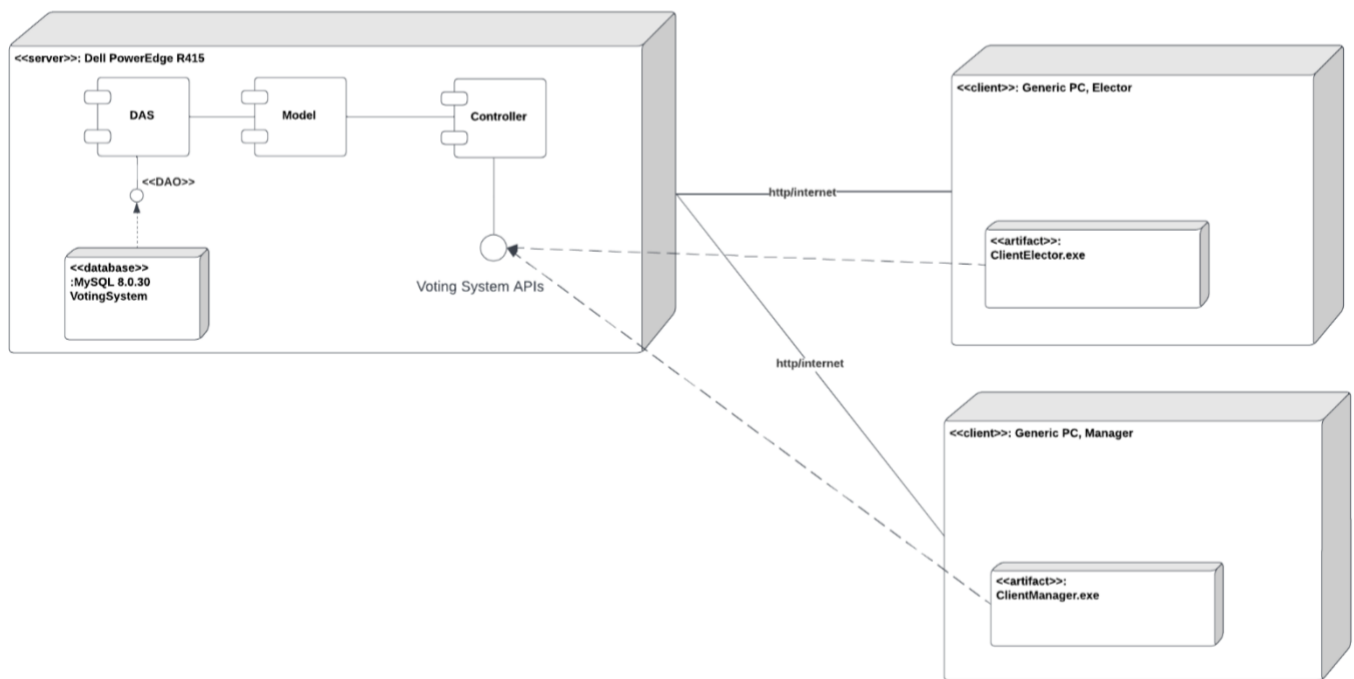


Figure 49 Diagramma di deployment

4.9 Specifica e verifica dei vincoli

Vengono riportati i vincoli più significativi tra quelli possibili:

Voting Session:

Per una corretta rappresentazione di una sessione di voto è necessario che il suo id, corrispondente a quello presente nel database sia maggiore o uguale a 0, che il suo nome non sia vuoto, che il suo tipo sia stato definito (categorica, categorica con preferenza, ordinale o referendum), che il suo stato sia definito e che una data di fine sia stata decisa.

Questa invariante viene rispettata grazie all'utilizzo di annotazioni come `@NotNull` e `@NotBlank` che ci assicurano che determinati campi non siano mai nulli o vuoti nel caso di stringhe. Per l'id si utilizza l'annotazione `@Min()` che impone un limite inferiore ad una variabile numerale.

```
context VotingSession inv:
  id >= 0 and name <> null and (not (name -> isBlank)) and type <> null
  and state <> null and endsOn <> null
```

OCL: Invariante di VotingSession

```
/*@ invariant
id >= 0 && name != null && !name.isBlank() && type != null
&& state != null && endsOn != null
```

4 JML: invariante per VotingSession

Aggiunta di un voto

Il metodo per l'aggiunta del voto necessita di particolari attenzioni in quanto un voto può essere accettato solo nel caso che la sessione di voto esista e sia attiva, ricevuto rispetti la sintassi corretta per la tipologia della sessione presa in considerazione. Tutti questi controlli vengono effettuati dal metodo `SessionServiceImpl::expressVote()` se questo rileva che qualcuno di questi vincoli è rispettato l'operazione di voto non viene completata e il client riceve un messaggio di errore.

```
context SessionDAS::addVotes(sessionId :long, votes :List<Vote>)
pre: sessionId >= 0 and votes <> null and votes.size() > 0
    and electorService.getElectorByUsername(electorUserName) <> null
    and getSessionStatus(sessionId) = Session.Status.ACTIVE
    and validate(getVotingOptions(votes, sessionId)) = true
```

5 OCL precondizioni per l'aggiunta di un voto

```
/*@ requires
sessionId >= 0 && votes != null && votes.size() > 0
&& electorService.getElectorByUsername(electorUserName) != null
&& getSessionStatus(sessionId) == Session.Status.ACTIVE
&& validate(getVotingOptions(votes, sessionId)) == true;
@*/
```

6 JML precondizioni per l'aggiunta di un voto

Marcare l'elettore dopo il voto:

Questo metodo non consente all'elettore di votare una seconda volta per la stessa sessione di voto, marcandolo su un'apposita tabella sul db, ma senza associarlo al voto stesso per preservare la segretezza di questo. Bisogna assicurarsi che l'id della sessione e dell'elettore presi in considerazione siano positivi, in quanto rappresentano gli id degli stessi sul db, è inoltre critico assicurarsi che la sessione alla quale l'elettore sta votando esista e sia attiva, similmente che l'elettore esista e soprattutto che il bit che segna se questo ha già votato in precedenza alla stessa sessione di voto sia vuoto. Tutti questi controlli vengono effettuati dal metodo `SessionServiceImpl::expressVote()` se questo rileva che qualcuno di questi vincoli è rispettato l'operazione di voto non viene completata e il client riceve un messaggio di errore.

```
context SessionDAS::setHasVoted(sessionId :long, electorId :long)
pre:  sessionId >= 0 and electorId >= 0
      and getSessionStatus(sessionId) = Session.Status.ACTIVE
      and electorService.getElectorByUsername(electorUserName) <> null
      and not (getParticipationStatus(sessionId, electorId) -> isEmpty)
```

7 OCL precondizioni per segnare che l'elettore ha espresso registrato un voto per la sessione specificata

```
/*@ requires
  sessionId >= 0 && electorId >= 0
  && getSessionStatus(sessionId).equals(Session.State.ACTIVE)
  && electorService.getElectorByUsername(electorUsername) != null
  && !(getParticipationStatus(sessionId, electorId).isEmpty());
@*/
```

8 JML precondizioni per segnare che l'elettore ha espresso registrato un voto per la sessione specificata

5 Appendice A: Guida all'installazione

Requisiti:

- Java 18 SDK
- Maven
- MySQL

È possibile configurare la porta utilizzata dal server e le credenziali di accesso al database mysql nel file `properties.yaml`, nelle risorse del progetto del server.

È reso disponibile il dump del database al cui interno si trovano già le seguenti utenze:

- username : password
- username1 : password
- username2 : password
- username3 : password
- username4 : password
- username5 : password
- username6 : password

Le credenziali di accesso con ruolo di gestore sono:

- admin : password

6 Appendice B: Sviluppi Futuri

- Timer di voto: aggiunta di un timer entro il quale l'elettore deve confermare il voto, nel caso dovesse scadere, nessun voto viene registrato e il sistema torna alla schermata principale.
- Supporto csv elettori: aggiunta supporto di csv per l'inserimento di elettori
- Autogenerazione username e password elettore: autogenerazione dell'username dato nome e cognome e generazione casuale della password
- Autogenerazione sessione di ballottaggio: nel caso di pareggio, maggioranza assoluta non raggiunta o quorum non raggiunto in una sessione di voto categorica, referendum o ordinale, questa viene dichiarata invalida e una nuova sessione di voto viene creata con le medesime caratteristiche. Nel caso di pareggio t elementi in una sessione categorica con preferenza, questa viene portata nello stato di terminata, mostrando l'opzione vincitrice ma creando automaticamente una sessione di ballottaggio categorica tra gli elementi dell'opzione vincitrice.
- Testing: aggiunta di test esaustivi nelle parti del sistema non ancora coperte.
- Modifica opzioni: possibilità di modificare le opzioni delle sessioni inattive
- Filtri e Barre di ricerca: aggiunta di filtri e barre di ricerca nello storico sessioni, gestione delle sessioni, gruppi e elettori.