

Contenuti

- 1 Introduzione
 - 1.1 Scopo del sistema ProjectVote
 - 1.2 User stories
 - 1.3 Glossario
- 2 Requisiti
 - 2.1 specifica dei requisiti
 - 2.2 Requisiti funzionali
 - 2.3 Requisiti non funzionali
 - 2.4 Requisiti di sicurezza
 - 2.5 Requisiti di sistema
- 3 Diagramma casi d'uso
 - 3.1 Diagramma casi d'uso
 - 3.2 Descrizione degli scenari
- 4 Diagramma delle classi
- 5 Diagrammi di sequenza
 - 5.1 login
 - 5.2 visualizza vincitore
 - 5.3 vota
 - 5.3.1 seleziona elezione + voto categorico
 - 5.3.2 voto preferenziale
 - 5.3.3 voto referendum
 - 5.4 ferma elezione
 - 5.5 attiva elezione
 - 5.6 concludi elezione
 - 5.7 carica nuova elezione
- 6 Diagramma dei componenti
- 7 Diagramma macchina di stato della GUI
- 8 Design pattern Utilizzati
 - 8.1 MVC
 - 8.2 DAO
 - 8.3 Singleton
 - 8.4 Strategy
- 9 Gestione dati persistenti
 - 9.1 utenti
 - 9.2 elezioni
 - 9.3 candidati
 - 9.4 voti

- 10 Diagramma di deployment
- 11 Descrizione GUI
 - 11.1 login
 - 11.2 elettore autenticato
 - 11.3 vota per elezione categorica
 - 11.4 vota per elezione preferenziale
 - 11.5 vota per referendum
 - 11.6 scegli una elezione conclusa
 - 11.7 visualiza risultati elezione conclusa
 - 11.8 admin autenticato e gestione delle elezioni
- 12 Descrizione testing

13 Build del software

tutti i diagrammi sono visibili anche nella cartella diagrammi sia come file .mdj che come immagini

1 INTRODUZIONE

1.1 Scopo del sistema ProjectVote

Fino ad ora per votare bisognava recarsi in un seggio elettorale, aspettare una eventuale fila, votare e tornare a casa. ProjectVote ha lo scopo di semplificare questo processo e permetterà ai suoi utenti di votare da casa, permettendo a individui che a causa di problemi di salute, tempo, disponibilità o distanza da un seggio non possono votare e aumentare il numero di persone che votano.

1.2 User Stories

- Come elettore voglio poter votare anche se non sono vicino ad un seggio.
- Come amministratore voglio poter inizializzare velocemente una nuova elezione
- Come persona non avente diritto al voto voglio comunque poter vedere i risultati delle elezioni concluse.

1.3 Glossario

ProjectVote : è un sistema digitale che ha lo scopo principale di registrare voti e di visualizzare i risultati di elezioni concluse.

Elettore : persona che ha diritto al voto ed è fornito di credenziali per accedere a ProjectVote

Admin : amministratore del sistema ProjectVote fornito di credenziali per accedere come admin

2 REQUISITI

2.1 Specifica dei requisiti

ProjectVote è un sistema di voto digitale a distanza, che gestisce diverse elezioni in corso che possono essere di diversi tipi tra cui:

- Voto preferenziale: si chiede all'elettore di ordinare i candidati delle elezioni in base alle sue preferenze.
- Voto Categorico: si chiede all'elettore di selezionare uno dei candidati disponibili.
- Voto Categorico con preferenze: si chiede all'elettore di selezionare un partito/gruppo e un eventuale candidato all'interno del gruppo.
- Referendum: si chiede ai elettori se si è pro o contro una proposta.

I diversi tipi di elezioni hanno anche diversi criteri che devono essere soddisfatti per scegliere un vincitore tra cui:

- Maggioranza : vince chi ha la percentuale più alta di voti.
- Maggioranza assoluta : vince chi ha almeno il 50% + 1 dei voti.
- Referendum senza quorum : i voti vengono contati indipendentemente da quanti elettori hanno votato.
- Referendum con quorum : i voti vengono contati solo se almeno il 50% dei abilitati al voto ha votato.

2.2 Requisiti funzionali

- ProjectVote dev'averne un interfaccia grafica facile da utilizzare.
- ProjectVote permette ai utenti di autenticarsi e distingue da solo se si è autenticato un admin o un elettore.
- ProjectVote fornisce una spiegazione chiara per ogni modalità di voto.
- Nel caso in cui non si raggiungono tutti i requisiti per scegliere un vincitore ProjectVote permette ai suoi utenti di vedere quali requisiti non sono stati raggiunti.
- Un utente non registrato può comunque vedere gli esiti delle elezioni concluse.
- Un admin può creare nuove elezioni, fermare elezioni in corso, avviare elezioni fermate.
- Un admin deve poter specificare il nome di una nuova elezione, i suoi candidati, il tipo di elezione e la sua modalità di vittoria.
- Un elettore dovrà confermare la sua scelta prima di esprimere il suo voto.
- Un elettore può vedere tutte le elezioni in corso per cui può votare.
- Un elettore può scegliere una scheda bianca
- Il conteggio dei voti avviene solo su elezioni concluse.
- Elezioni concluse non possono essere modificate.
- E' possibile avere piu elezioni in corso.
- Usa un sistema di autenticazione esterno sicuro (come SPID).

2.3 Requisiti non funzionali

- ProjectVote dev'essere compatibile con un gran numero di dispositivi.
- Dopo operazioni importanti non reversibili il sistema deve chiedere conferma ai suoi utenti.
- Il conteggio dei voti dev'essere imparziale e non modificabile.
- Il disco su cui vengono salvati i dati persistenti deve avere un alto grado di affidabilità, ridondanza e devono essere eseguiti backup preferibilmente dopo ogni elezione conclusa.

2.4 Requisiti di sicurezza

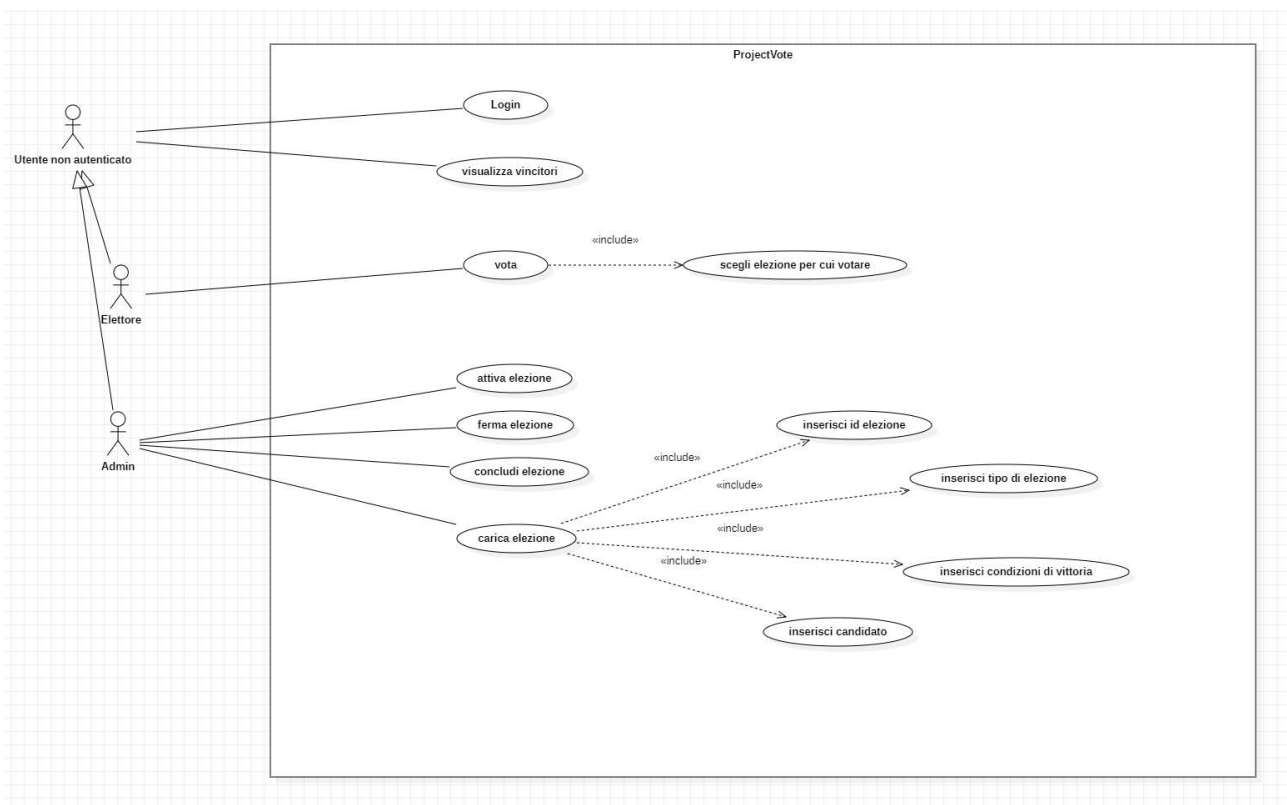
- ProjectVote utilizza il meno possibile input diretto dall'utente, utilizzando il più possibile menu e pulsanti , per un alto grado di sicurezza da attacchi di tipo SQL injection.
- Il sistema deve mantenere l'anonimato dell'utente.
- I voti espressi non possono essere modificati o cancellati.

2.5 Requisiti di sistema

- Per poter avviare ProjectVote serve un dispositivo fornito da una JVM.
- Per poter autenticarsi, votare o eseguire altre operazioni serve una connessione internet stabile.

3 DIAGRAMMA CASI D'USO

3.1 Diagramma casi d'uso



3.2 Descrizione degli scenari

Nome	Login
Scopo	Autenticare l'utente
Attori	Utente non autenticato
Pre-condizioni	L'utente non autenticato possiede delle credenziali valide e le ha inserito nel Textfield username e password.
Trigger	L'utente clicca su Login
Descrizione sequenza eventi	1 il sistema verifica se le credenziali sono corrette. 2 il sistema verifica il tipo di utente 3 se l'utente è un elettore il sistema lo porta alla schermata di un elettore autenticato
Alternativa/e sequenza eventi	3a se l'utente è un admin il sistema lo porta alla schermata di un admin autenticato 3b se le credenziali non sono valide il sistema restituisce un messaggio d'errore
Post-condizioni	L'utente è autenticato come admin o elettore

Nome	Visualizza vincitore
Scopo	Vedere i risultati di elezioni concluse
Attori	Utente non autenticato
Pre-condizioni	
Trigger	L'utente clicca su visualizza vincitori
Descrizione sequenza eventi	1 l'utente viene portato a una schermata dove puo scegliere una delle elezioni concluse. 2 l'utente clicca su continua 3 l'utente viene portato nella schermata dove puo vedere i risultati della elezione.
Alternativa/e sequenza eventi	/
Post-condizioni	L'utente vede i risultati di una elezione

Nome	Vota
Scopo	L'elettore esprime il suo voto
Attori	Elettore
Pre-condizioni	L'elettore si è autenticato
Trigger	L'elettore clicca su continua dopo aver scelto l'elezione per cui votare
Descrizione sequenza eventi	1 il sistema controlla i parametri dell'elezione scelta dall'elettore 2 se l'elezione è di tipo categorico l'elettore viene portato nella schermata dove può esprimere un voto categorico. 3 l'elettore sceglie il candidato che preferisce 4 l'elettore clicca su conferma 5 l'elettore viene riportato alla schermata di login
Alternativa/e sequenza eventi	2a se l'elezione è di tipo referendum l'elettore viene portato nella schermata dove può esprimere un voto per referendum. 2b se l'elezione è di tipo preferenziale l'elettore viene portato nella schermata dove può esprimere una voto associato a una preferenza. 3b l'elettore associa un candidato a una posizione 4b l'elettore clicca su conferma 5b il candidato e la posizione confermata vengono eliminati dal elenco 6b il candidato ripete il passo 3b a 5b finché l'elenco non è vuoto 7b l'elettore viene riportato alla schermata di login
Post-condizioni	L'elettore ha espresso il suo voto

Nome	Disattiva elezione
Scopo	Disattivare una elezione attiva
Attori	Admin
Pre-condizioni	L'admin sceglie un elezione da fermare
Trigger	L'admin clicca su ferma elezione
Descrizione sequenza eventi	1 l'elezione selezionata viene fermata
Alternativa/e sequenza eventi	/
Post-condizioni	L'elezione attiva selezionata viene fermata, non sarà più possibile per utenti diversi da admin vedere l'elezione

Nome	Attiva elezione
Scopo	Attiva una elezione non attiva
Attori	Admin
Pre-condizioni	L'admin sceglie un elezione da attivare
Trigger	L'admin clicca su attiva elezione
Descrizione sequenza eventi	1 l'elezione selezionata viene attivata
Alternativa/e sequenza eventi	/
Post-condizioni	L'elezione selezionata viene attivata, sarà possibile per elettori di votare per questa elezione

Nome	Concludi elezione
Scopo	Conclude una elezione fermata
Attori	Admin
Pre-condizioni	L'admin sceglie una elezione fermata da concludere
Trigger	L'admin clicca su concludi elezione
Descrizione sequenza eventi	1 l'elezione selezionata viene conclusa
Alternativa/e sequenza eventi	/
Post-condizioni	L'elezione selezionata viene conclusa, sarà possibile per tutti vedere i risultati della elezione

Nome	Carica elezione
Scopo	Caricare una nuova elezione
Attori	Admin
Pre-condizioni	L'admin ha compilato i parametri della nuova elezione
Trigger	L'admin clicca su carica elezione
Descrizione sequenza eventi	1 l'elezione viene caricata se tutti i parametri sono compatibili
Alternativa/e sequenza eventi	1a se la combinazione tra tipo di elezione e modalità di vittoria è incompatibile viene visualizzato un messaggio d'errore che spiega il perché 2a l'elezione non viene caricata 3a l'admin corregge i parametri e torna al passo 1
Post-condizioni	Viene caricata una nuova elezione inizializzata come non attiva

4 DIAGRAMMA DELLE CLASSI

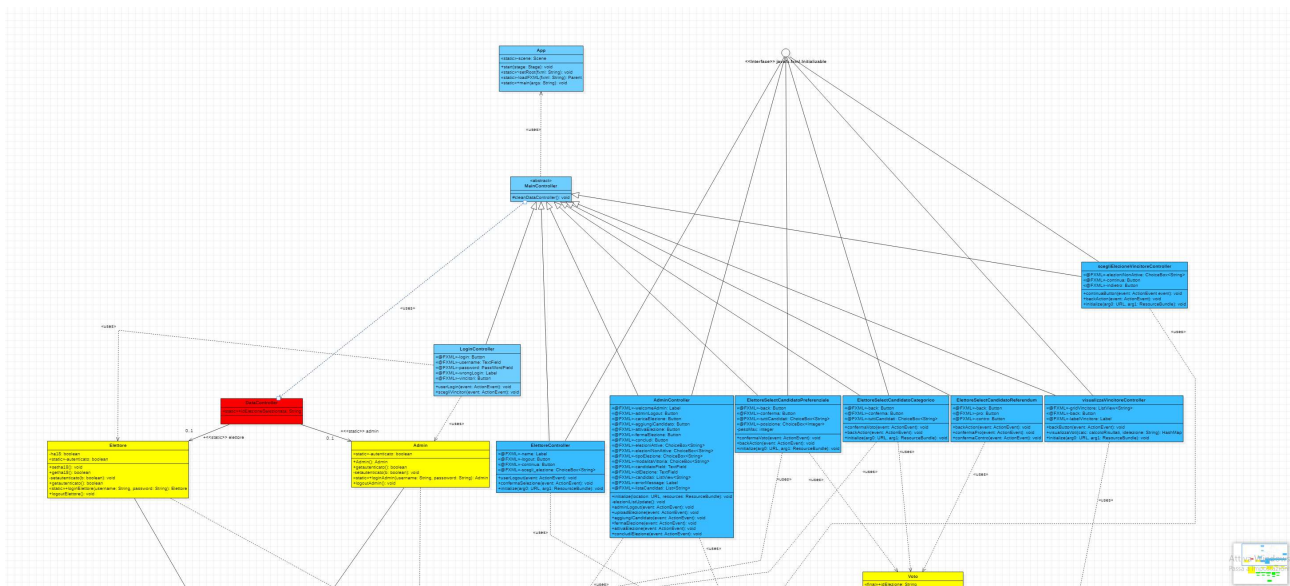
blu : pacchetto voto, contiene le classi controller

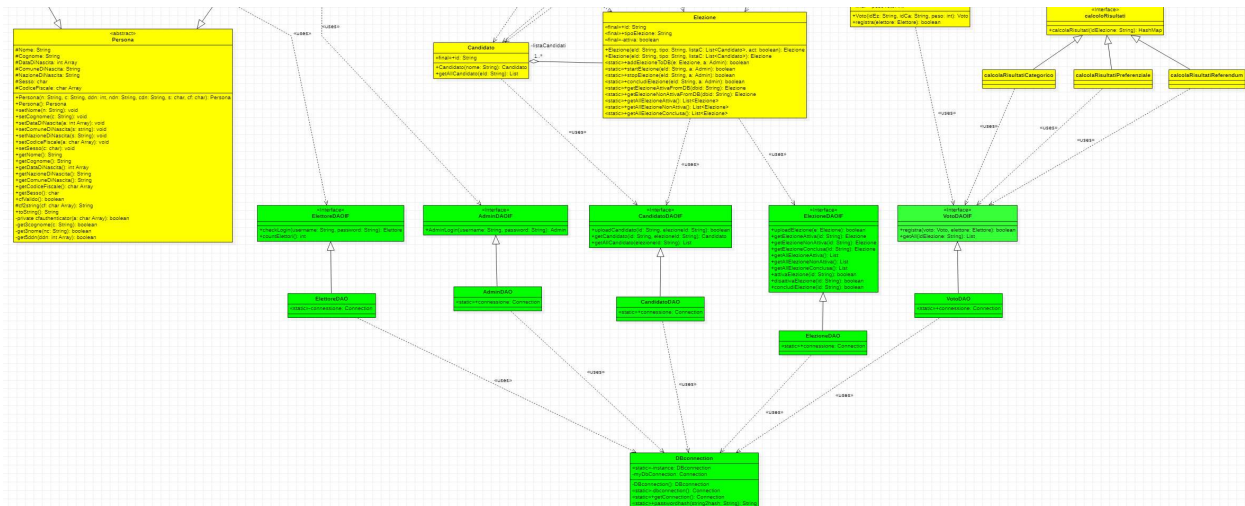
rosso : pacchetto data_manager, contiene le classi che servono ai controller per passare dati tra di loro

giallo : pacchetto model, contiene le classi model

verde : pacchetto db_access, contiene le classi e interfacce DAO e gestisce la connessione al DB

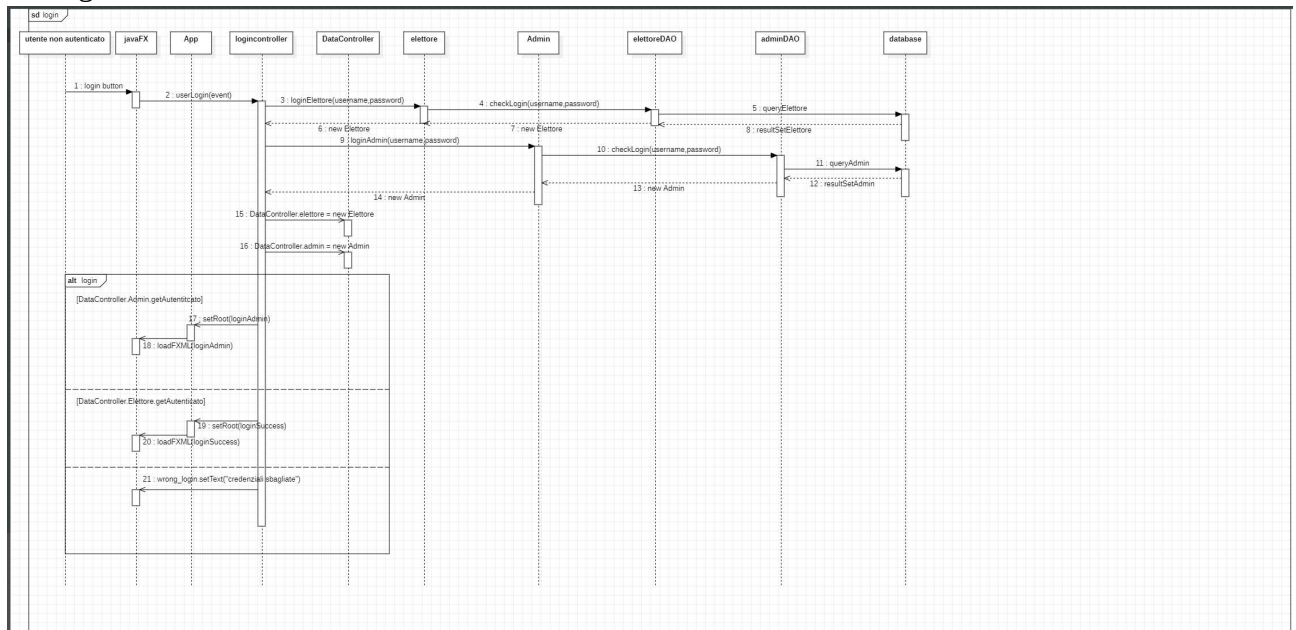
Si consiglia di guardare il diagramma delle classi direttamente dal file `class.mdj` in `diagrammi/class` o dalle immagini nella stessa directory.



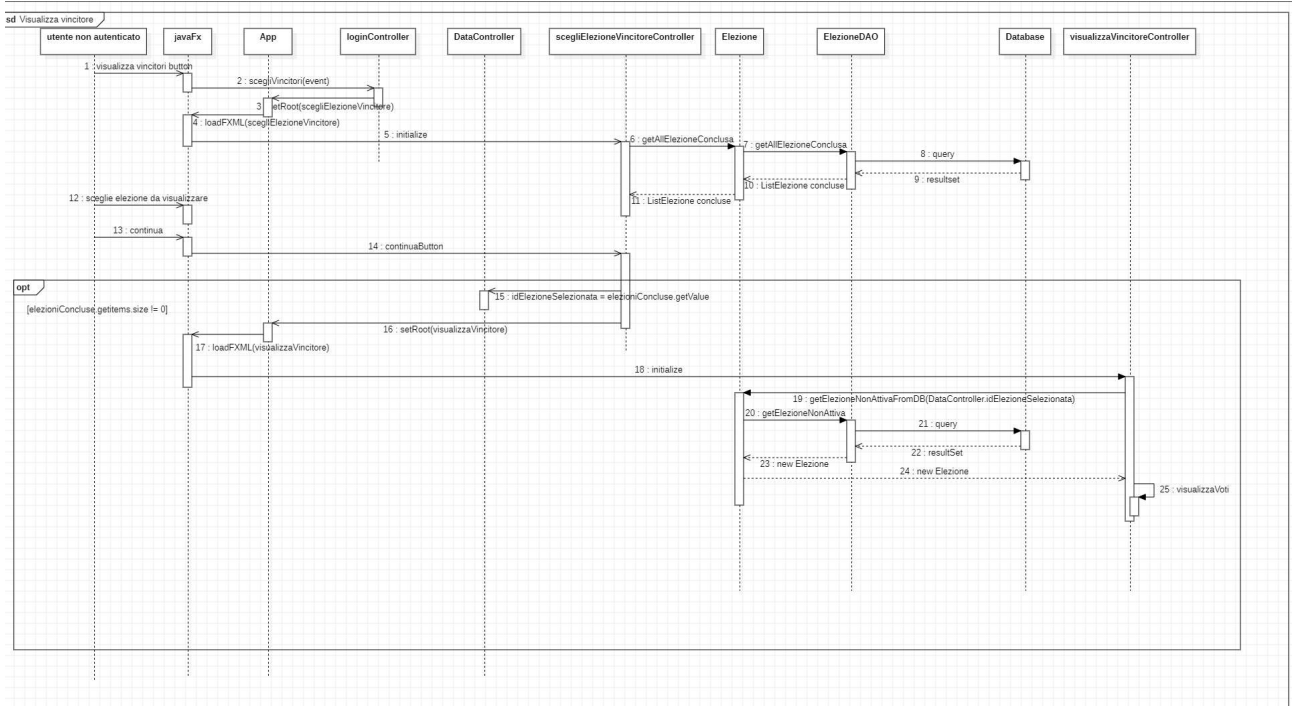


5 Diagrammi di sequenza

5.1 login



5.2 visualizza vincitore



5.3.1 seleziona elezione per cui votare e voto categorico



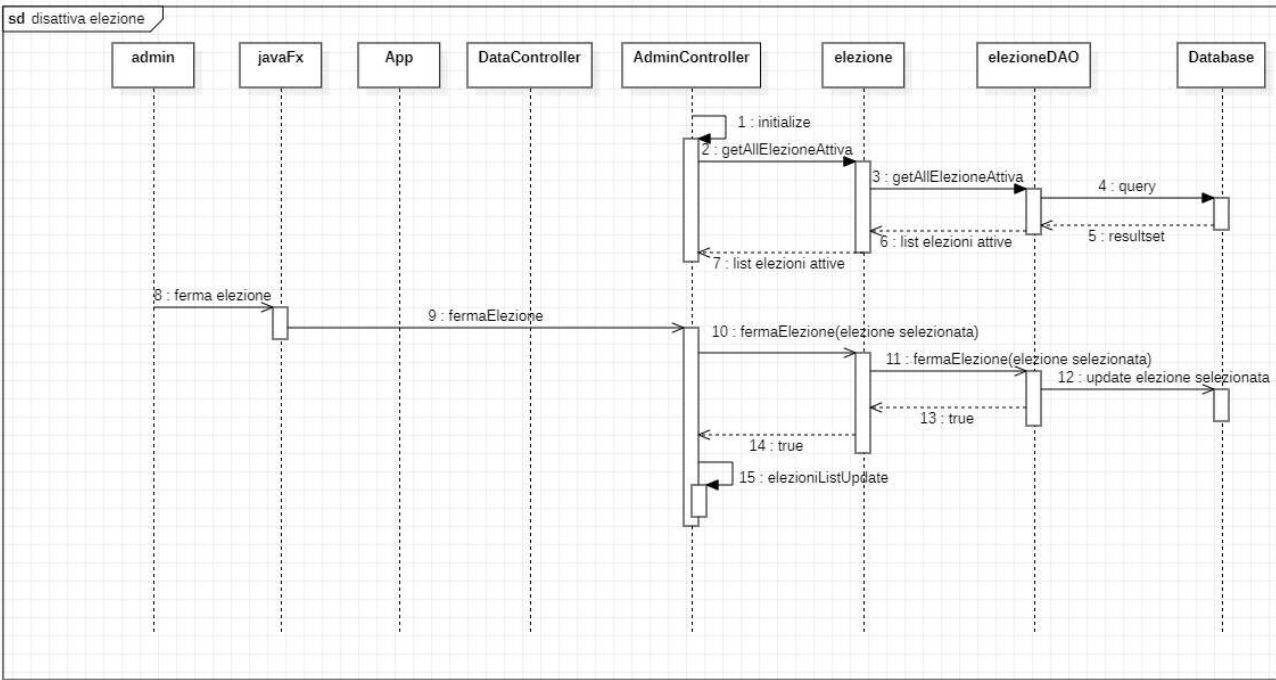
5.3.2 voto preferenziale



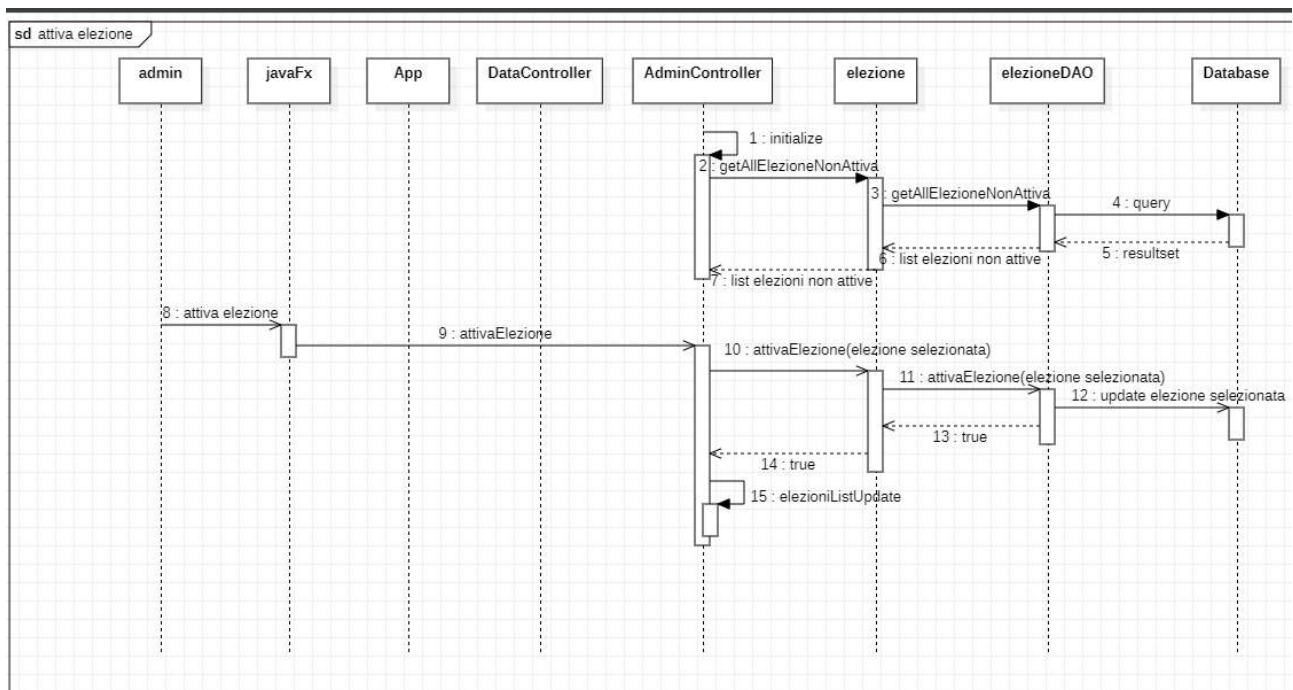
5.3.3 voto referendum



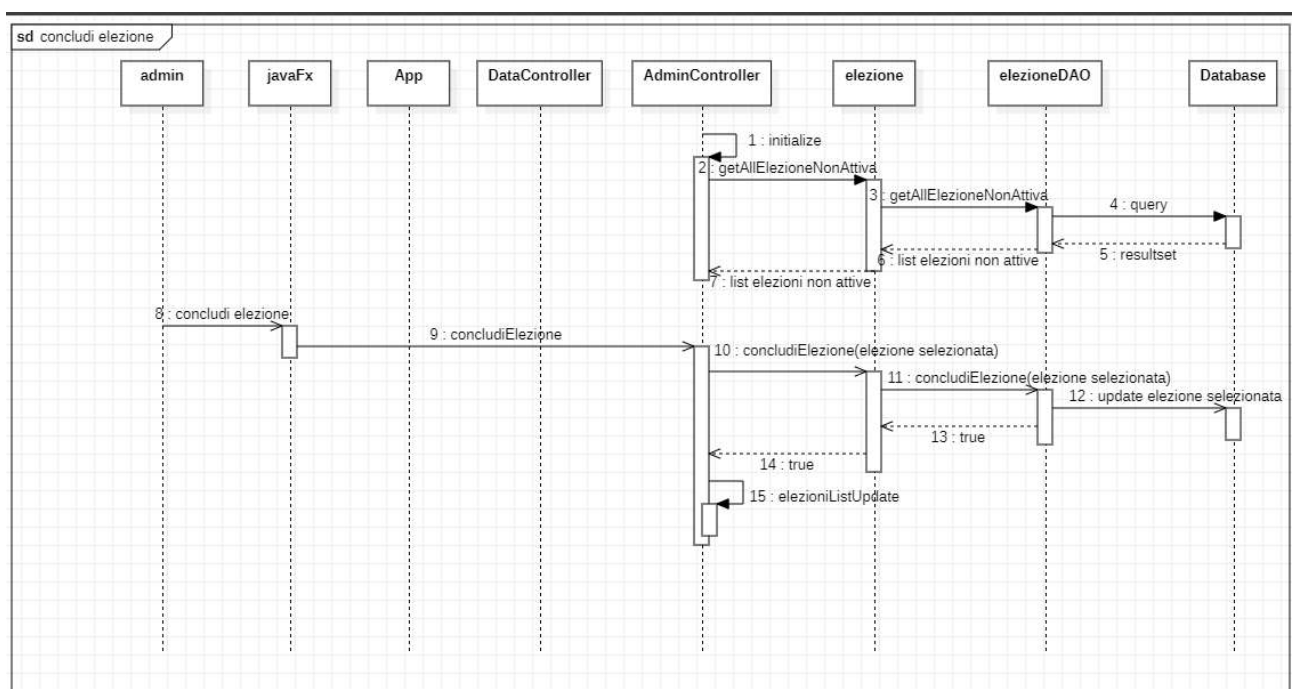
5.4 ferma elezione



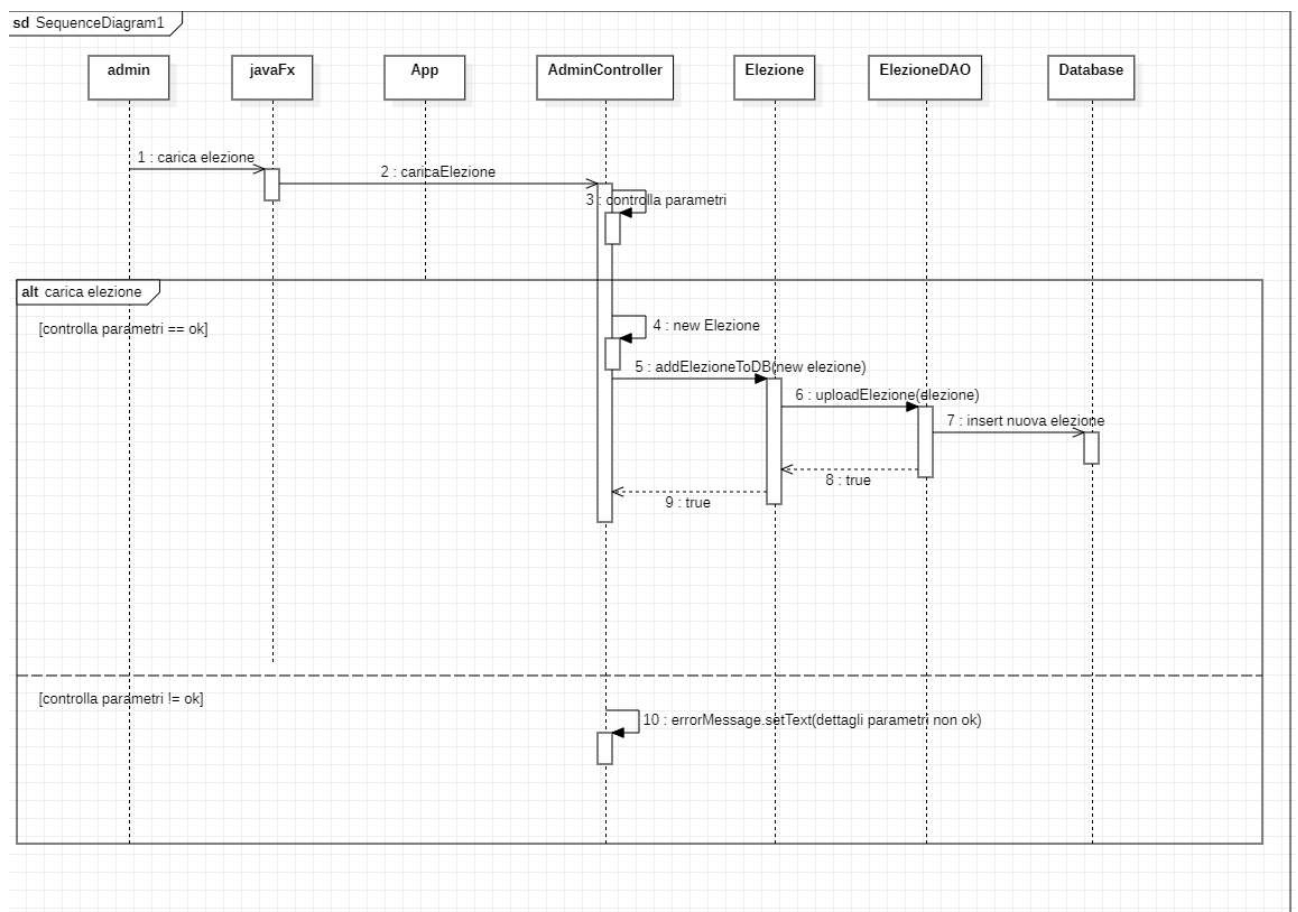
5.5 attiva elezione



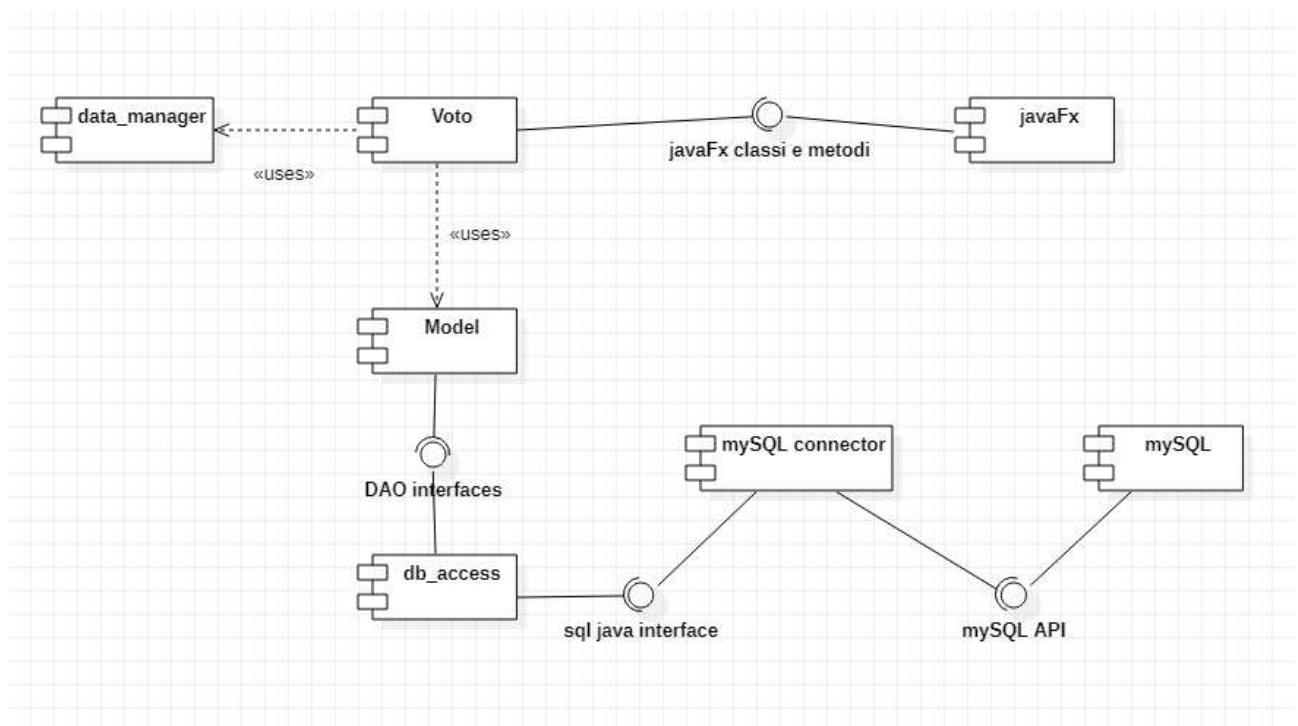
5.6 concludi elezione



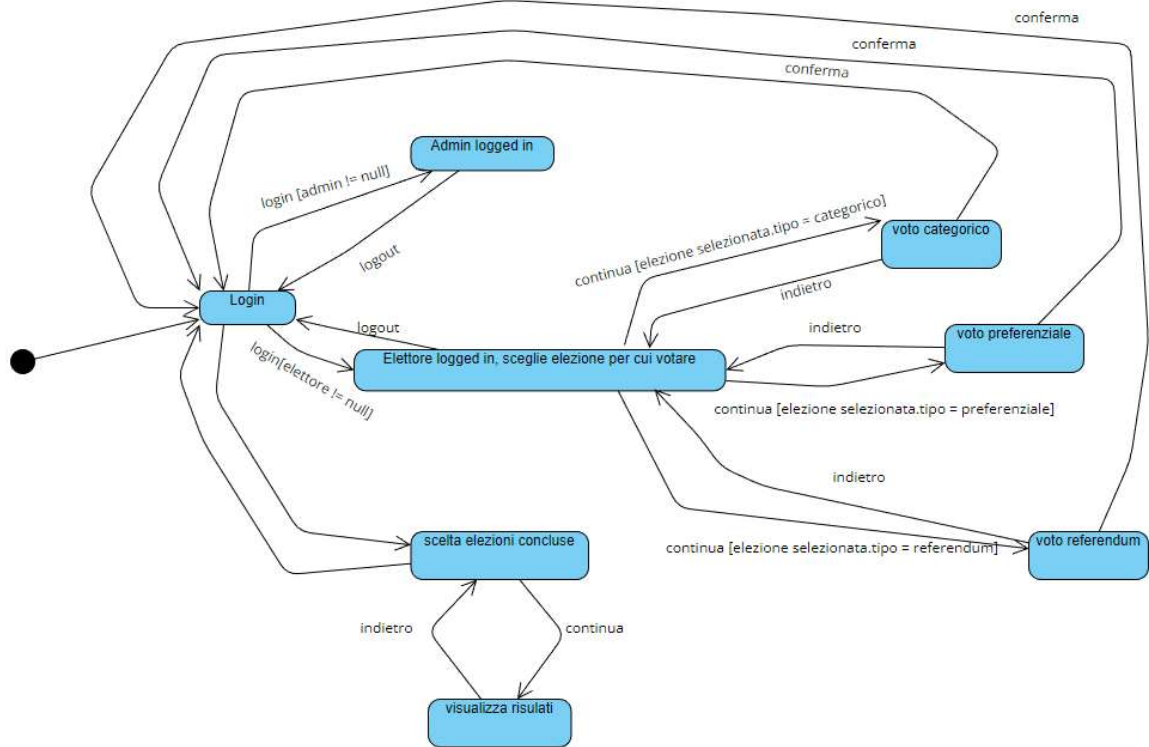
5.7 carica elezione nuova



6 Diagramma dei componenti



7 Macchina di stato della GUI



8 Design pattern utilizzati

8.1 MVC

Model-View-Controller (MVC) è un pattern utilizzato per dividere grafica dalla parte funzionale del sistema, contiene 3 tipi di classi e oggetti: model , view , controller.

Model: contiene le classi le cui istanze devono essere manipolate e visualizzate (pacchetto model)

Controller: contiene gli oggetti che gestiscono l'input dell'utente e manipolano gli oggetti model e view in base all'input dell'utente. (pacchetto voto)

View: contiene gli oggetti utilizzati per visualizzare l'interfaccia grafica (risorse fxm1)

8.2 DAO

Data Access Object Pattern è un pattern utilizzato per separare le operazioni di basso livello che accedono a API dalle classi client che usano tali operazioni. Questo permette una alta riusabilità e facile modificabilità.

La struttura del pattern è divisa in 3 parti:

classi client: le classi a cui servono le informazioni dal database. (pacchetto model)

Interfacce DAO: le interfacce che definiscono le operazioni che le classi DAO devono implementare e sono usate dalle classi client. (interfacce *DAOIF.java nel pacchetto db_access)

Classi DAO: le classi DAO che implementano l'interfaccia e gestiscono i dati dalla risorsa fornita (in questo caso un database)(classi *DAO.java nel pacchetto db_access).

8.3 Singleton

Singleton è un pattern utilizzato quando si vuole una e una sola istanza di una classe.

In projectVote è usata per avere una sola istanza della classe che contiene la connessione al database (db_access/Dbconnection.java)

8.4 Strategy

Strategy è un pattern utilizzato quando si vuole gestire un insieme di algoritmi diversi ma correlati. Il pattern ha lo scopo di aumentare la facilità di modifica e l'aggiunta di nuovi algoritmi correlati.

E' fatto da un'interfaccia con il metodo, le classi che implementano questo metodo incapsulano i diversi algoritmi.

In projectVote questo pattern è stato usato per incapsulare gli algoritmi che calcolano le percentuali di voti per ogni elezione di tipo diverso. L'interfaccia comune è calcoloRisultati implementata dalle classi calcoloRisultatiCategorico, calcoloRisultatiPreferenziale e calcoloRisultatiReferendum.

9 Gestione dei dati persistenti

Per la gestione dei dati persistenti viene usato un database MySQL con cui interagisce projectVote attraverso JDBC. Le tabelle usate da projectVote sono candidati, elezioni, voti e utenti.

9.1 utenti

[la tabella utenti serve a simulare un sistema di autenticazione esterno come SPID per questo [motivo il sistema projectVote non ha una funzione di registrazione di nuovi utenti.

username: chiave primaria, ciò che viene usato per identificare l'utente

password: una password scelta dall'utente e passata attraverso una funzione hash "SHA-256"

nome: nome dell'utente

cognome: cognome dell'utente

data di nascita: data di nascita dell'utente

nazione di nascita: nazione di nascita dell'utente

comune di nascita: comune di nascita dell'utente

isAdmin: usato per verificare se l'utente è un admin 1 = sì , 0 = no

codicefiscale: cf dell'utente

sex: sesso dell'utente

un esempio di una tabella users:

	username	password	nome	cognome	dataDiNascita	nazioneNascita	comuneNascita	isAdmin	codiceFiscale	sex
▶	alexAdmin	2VCBIsoUPWnfipvzYkt7yyuKyB7SEKD1JkVe8Rn...	aleksandar	manasiev	1999-06-10	IT	milano	1	MNSLSN99E10F205V	M
	alexander	2fDR7zm3Zl2yy/1grYYSoGupkQGJblvAcs5BWYh...	aleksandar	manasiev	1999-06-10	IT	lodi	0	MNSLSN99E10E205V	M
	testAdmin1	oUDAwE2i3vK4MDY7o2KqTX0lxCyYFRiIfVW4...	admin	admin	1999-06-15	IT	crema	1	DMNDMN99E15D142Q	M
	testUser1	G08OmFGXGZjnMgeFRMlrNsPQHO33yqMyNZ1v...	mario	kart	1999-06-11	IT	bergamo	0	KRTMRA99E11A794G	M
	testUser2	YDA64iuZIGG847KPM+7BvnWKITyGyTwHbb6fV...	luigi	kart	1999-06-12	IT	roma	0	KRTLGU99E12H501W	M
	testUser3	/WGgOvT3fYcPwh4F5+gGeAlcktgIz7O1wnnuB...	lalala	nonloso	1999-06-13	GER	NULL	0	NNLLLL99E13Z112T	M
	testUser4	pOYk1obgPTj2fAq9hcFEJrCxFOs6B0nu0/k9vA...	lululu	nonloso2	1999-06-14	FR	NULL	0	NNLLLL99E14Z110R	M
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

9.2 elezioni

idElezioni: il nome dell'elezione.

Attiva: usato per verificare se una elezione è in corso o no 1 = sì, 0 = no

tipoElezioni: il tipo di elezione

conclusa: se una elezione è conclusa o no 1 = sì, 0 = no

un esempio di una tabella elezioni :

	idElezioni	attiva	tipoElezioni	conclusa
▶	moto migliore	1	categorico	0
	ranking moto	0	preferenziale	1
	referendum Ducati	0	referendum con quorum	1
	referendum senape	0	referendum con quorum	0
	salsa migliore	0	categorico	1
	testElezioniCategorico	1	categorico maggioranza non assoluta	0
	testPreferenziale	1	preferenziale maggioranza non assoluta	0
*	NULL	NULL	NULL	NULL

9.3 candidati

idCandidato: usato per il nome del candidato

idElezione: usato per controllare a quale elezione appartiene

FOREIGN KEY(idElezione) REFERENCES('sistemavoto.elezioni.idElezione)

un esempio di una tabella candidato

	idCandidati	idElezione
►	Ducati	moto migliore
	Honda	moto migliore
	Kawasaki	moto migliore
	Suzuki	moto migliore
	Triumph	moto migliore
	Yamaha	moto migliore
	Ducati	ranking moto
	Honda	ranking moto
	Kawasaki	ranking moto
	Suzuki	ranking moto
	Triumph	ranking moto
	Yamaha	ranking moto
	contro	referendum ...
	pro	referendum ...

9.4 voti

idElezione: id dell'elezione a quale appartiene

idCandidato: id dell'candidato per cui si ha votato

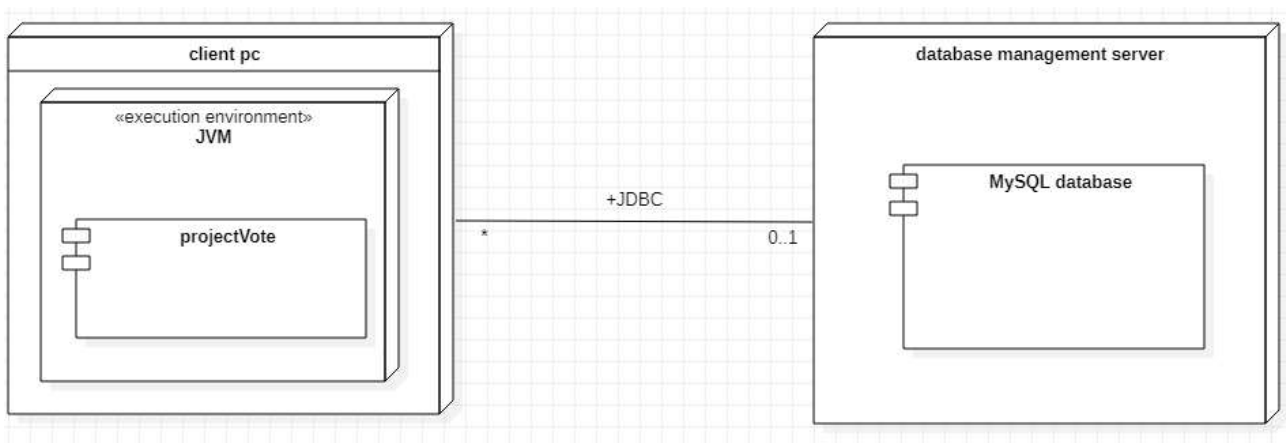
peso: indica quanto preferisce un candidato su un altro, più è alto il peso, più lo preferisce

idElettore: il codice fiscale dell'elettore che ha votato passato attraverso una funzione hash "SHA-256" per preservare l'anonimato.

un esempio di una tabella voti

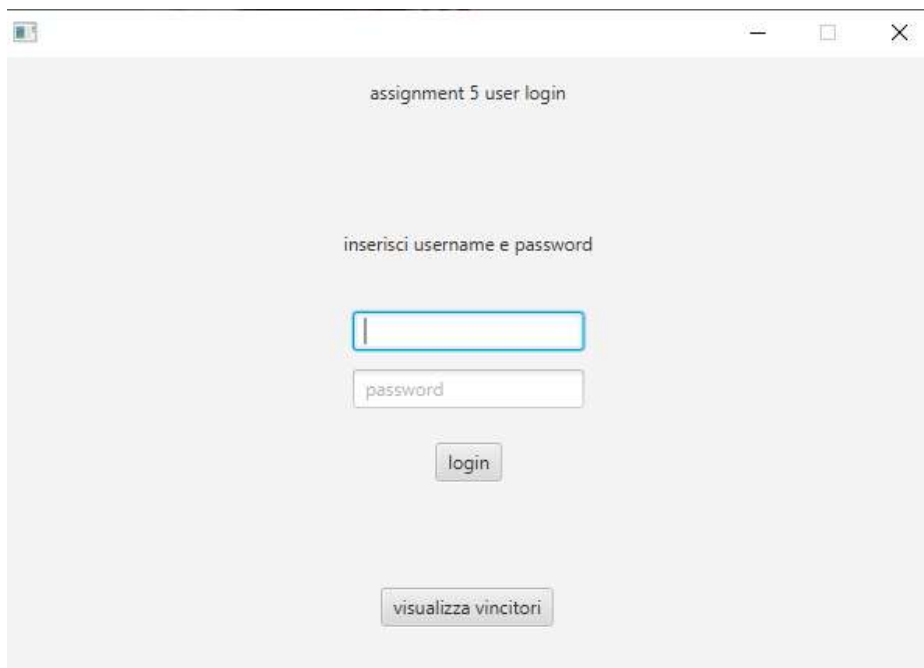
	idElezione	idCandidato	peso	idElettore
►	referendum Ducati	contro	1	bB4GWH5/z7Ose0Lmsw2cNIoyUmfiz01Oe4fVC...
	referendum Ducati	contro	1	dc8DkRy39H+b513toYWtw4fi/kure6qXwisEcVx...
	moto migliore	Ducati	1	bB4GWH5/z7Ose0Lmsw2cNIoyUmfiz01Oe4fVC...
	moto migliore	Ducati	1	Ircf0iNMkhCcnk8CPLtGa/oaC9pGgnyBtOifiziDtPo
	ranking moto	Ducati	1	dc8DkRy39H+b513toYWtw4fi/kure6qXwisEcVx...
	ranking moto	Ducati	1	Ircf0iNMkhCcnk8CPLtGa/oaC9pGgnyBtOifiziDtPo
	ranking moto	Ducati	6	bB4GWH5/z7Ose0Lmsw2cNIoyUmfiz01Oe4fVC...
	ranking moto	Honda	2	dc8DkRy39H+b513toYWtw4fi/kure6qXwisEcVx...
	ranking moto	Honda	2	Ircf0iNMkhCcnk8CPLtGa/oaC9pGgnyBtOifiziDtPo
	ranking moto	Honda	5	bB4GWH5/z7Ose0Lmsw2cNIoyUmfiz01Oe4fVC...
	ranking moto	Kawasaki	3	dc8DkRy39H+b513toYWtw4fi/kure6qXwisEcVx...
	ranking moto	Kawasaki	3	Ircf0iNMkhCcnk8CPLtGa/oaC9pGgnyBtOifiziDtPo
	ranking moto	Kawasaki	4	bB4GWH5/z7Ose0Lmsw2cNIoyUmfiz01Oe4fVC...

10 Diagramma di deployment

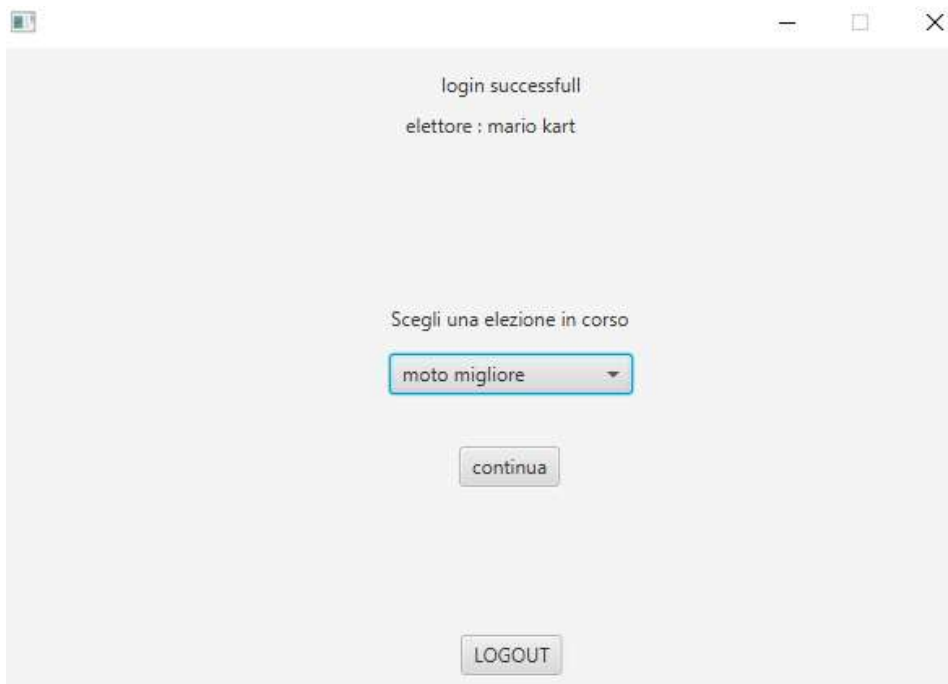


11 Descrizione GUI

11.1 schermata login



11.2 schermata elettore autenticato e scelta elezione per cui votare



login successfull
elettore : mario kart

Scegli una elezione in corso

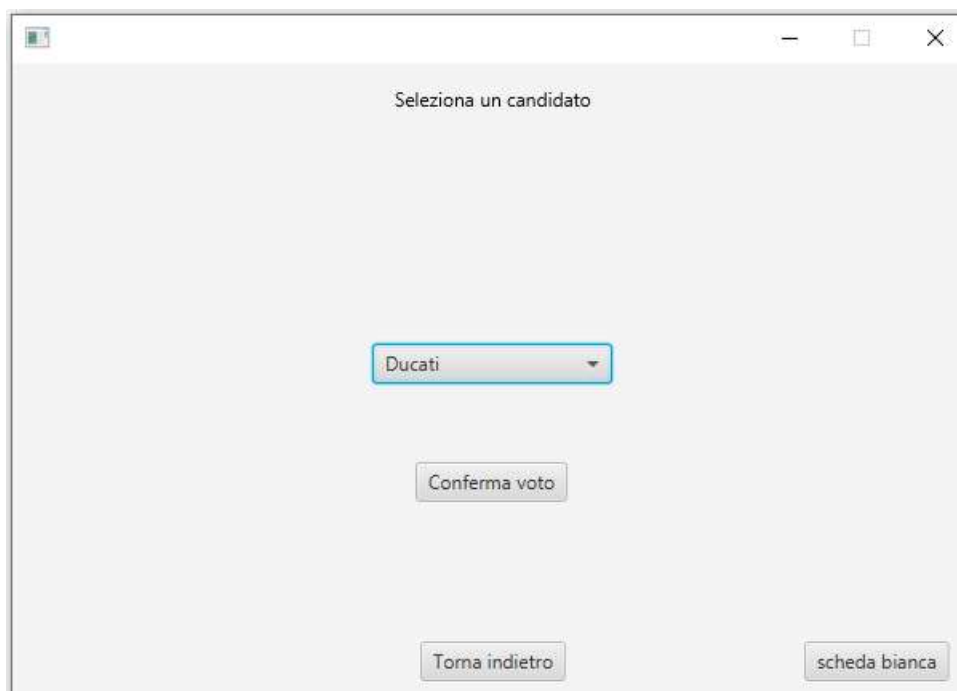
moto migliore ▼

continua

LOGOUT

This screenshot shows a web application window with a light gray background. At the top, it displays 'login successfull' and 'elettore : mario kart'. Below this, the instruction 'Scegli una elezione in corso' is followed by a dropdown menu currently showing 'moto migliore'. A 'continua' button is positioned below the dropdown, and a 'LOGOUT' button is at the bottom center.

11.3 vota per una elezione categorica



Seleziona un candidato

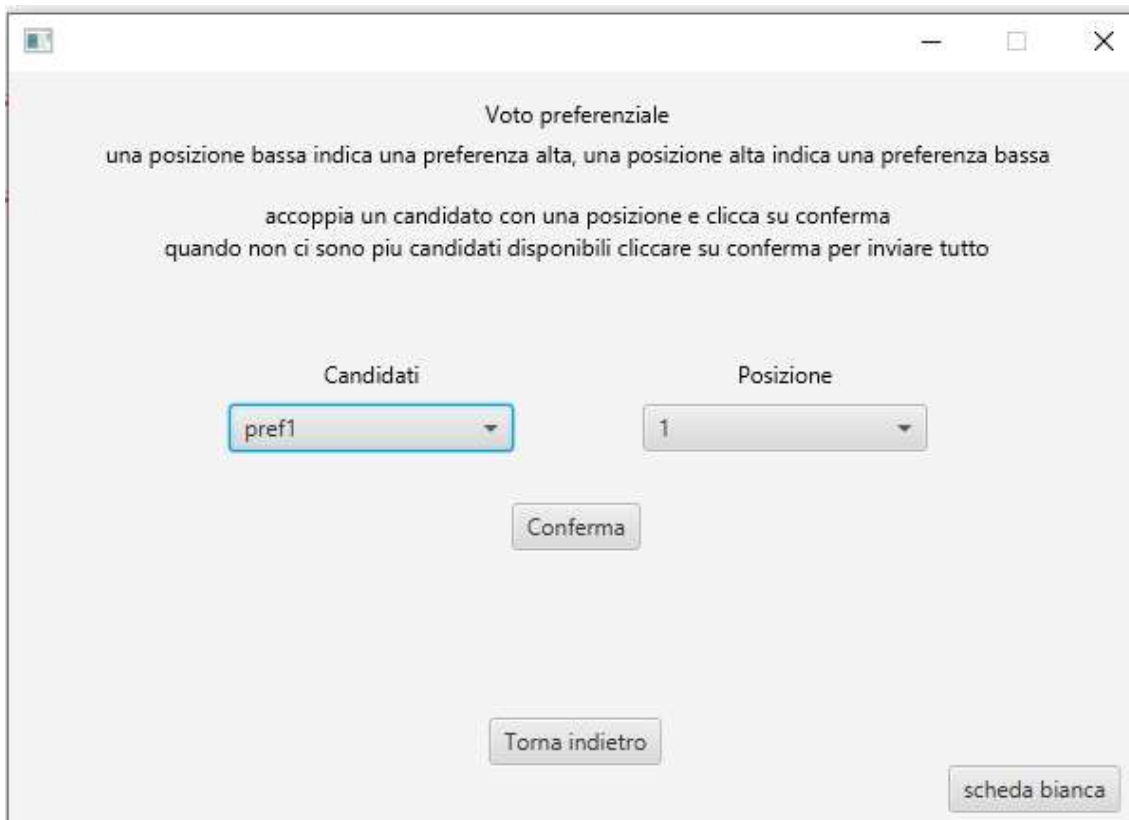
Ducati ▼

Conferma voto

Torna indietro scheda bianca

This screenshot shows a web application window for selecting a candidate. It features the title 'Seleziona un candidato' at the top. Below it is a dropdown menu with 'Ducati' selected. A 'Conferma voto' button is centered below the dropdown. At the bottom, there are two buttons: 'Torna indietro' on the left and 'scheda bianca' on the right.

11.4 vota per una elezione preferenziale



Voto preferenziale

una posizione bassa indica una preferenza alta, una posizione alta indica una preferenza bassa

accoppia un candidato con una posizione e clicca su conferma
quando non ci sono piu candidati disponibili cliccare su conferma per inviare tutto

Candidati

Posizione

pref1

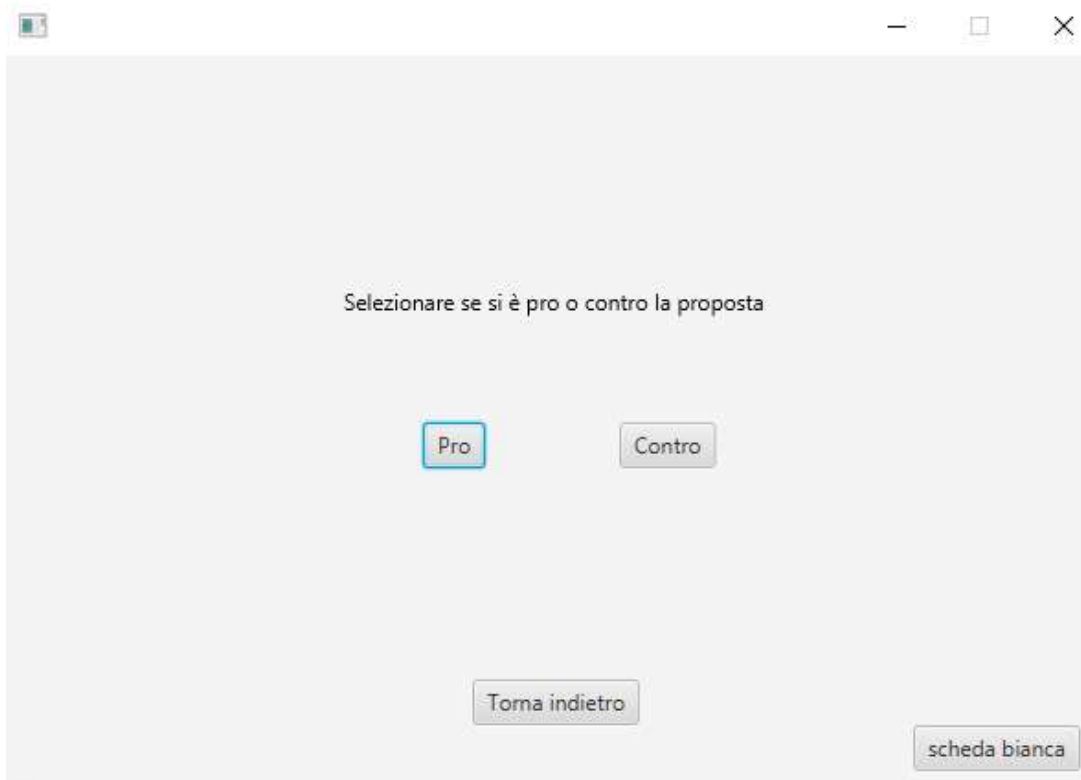
1

Conferma

Torna indietro

scheda bianca

11.5 vota per un referendum



Selezionare se si è pro o contro la proposta

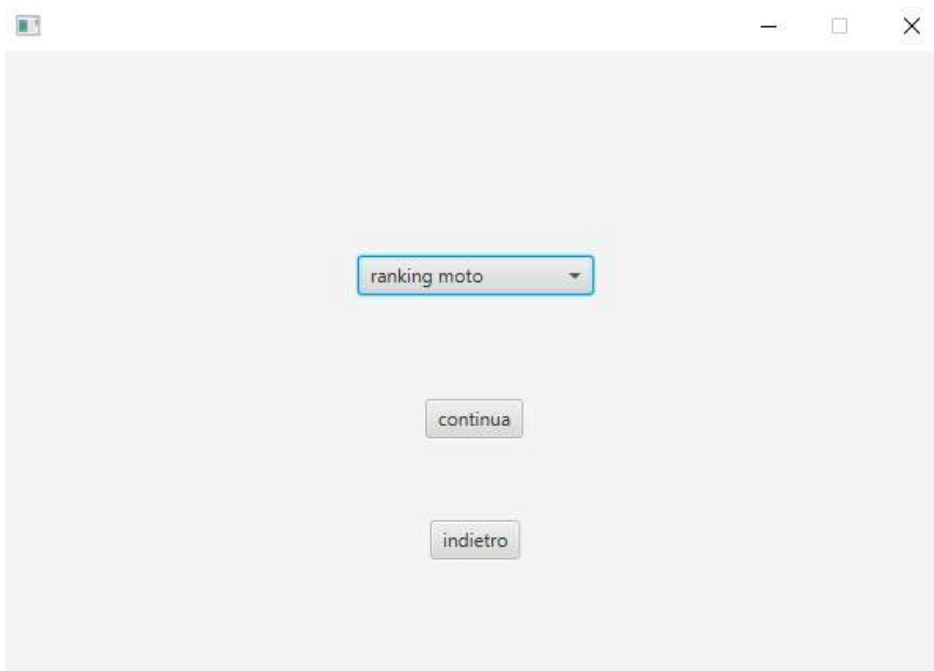
Pro

Contro

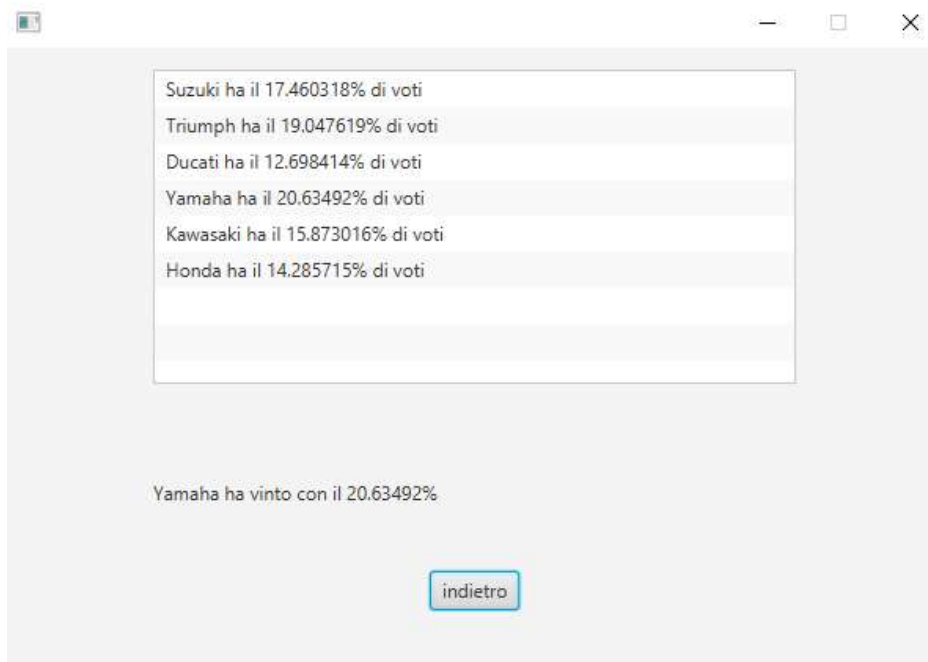
Torna indietro

scheda bianca

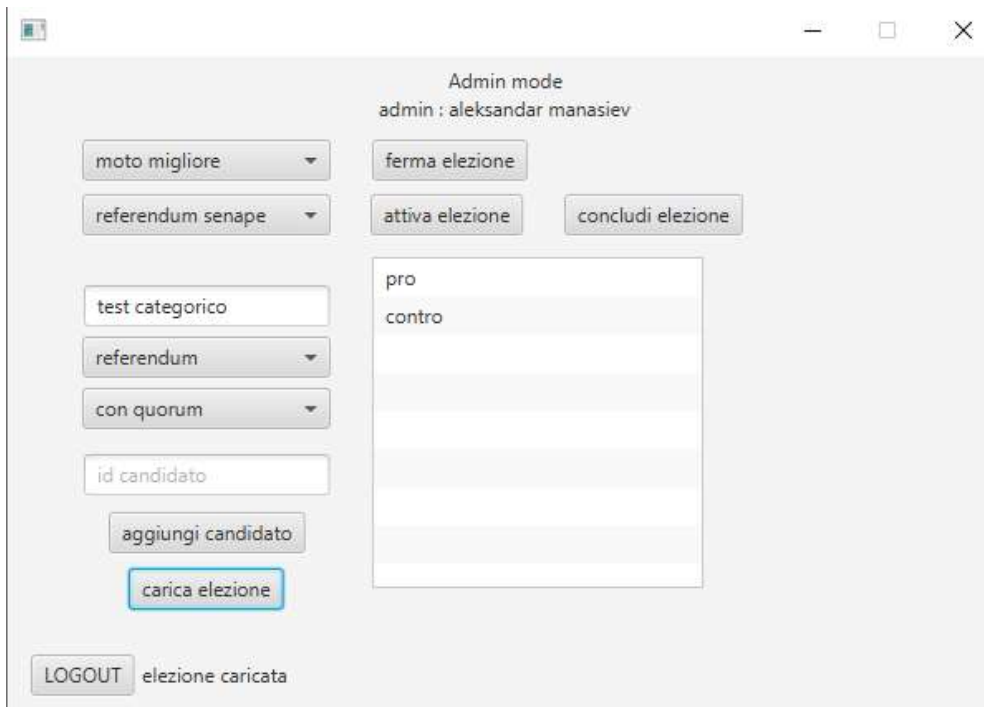
11.6 scegli l'elezione conclusa di cui vuoi vedere i vincitori



11.7 il risultato della elezione conclusa



11.8 admin autenticato e gestione delle elezioni



The screenshot shows a web application interface titled "Admin mode" with the user "admin : aleksandar manasiev". The interface is divided into two main sections. On the left, there are several controls: a dropdown menu with "moto migliore" selected, another dropdown with "referendum senape", a text input field containing "test categorico", a dropdown with "referendum", another dropdown with "con quorum", a text input field for "id candidato", a button labeled "aggiungi candidato", and a button labeled "carica elezione" which is highlighted with a blue border. On the right, there are three buttons: "ferma elezione", "attiva elezione", and "concludi elezione". Below these buttons is a table with two columns, "pro" and "contro", and several empty rows for data entry. At the bottom left, there is a "LOGOUT" button and the text "elezione caricata".

12 Descrizione testing

Il testing è stato effettuato tramite Junit e la copertura delle istruzioni tramite il plugin EcLlemma in eclipse. Sono state testati i pacchetti Model e db_access seguendo il criterio della copertura delle decisioni a cause delle condizioni semplici e non poca presenza di condizioni multiple.

Ogni test case ha lo scopo di testare sia la classe model che la classe DAO.

Le istruzioni non coperta sono principalmente istruzioni setter e getter molto semplici o eccezioni.

La classe calcoloRisultatiPreferenziale.java è stata verificata manualmente a causa di piccole differenze nel calcolo di numeri Float.

13 Build del software

ricostruire il db dal dump sul proprio db MySQL

modificare le variabili dbUser, dbPass, connectionString in modo da poter accedere al proprio DB.

Importare la cartella voto come progetto maven in eclipse

Eseguire il build del progetto o usare Project → Build automatically

Run App.java