ANALISI E SPECIFICA DEI REQUISITI

GLOSSARIO

TERMINE	DEFINIZIONE
Utente	utente comune, con la sola capacità di voto.
Candidato	Utente che può essere votato in una elezione.
Auditing	operazione finalizzata a creare un file di log, con i movimenti e dettagli delle operazioni effettuate sulle votazioni
File di log	File che conterrà le operazioni svolte nel tempo dal sistema
Codice elezione	Codice numerico generato alla creazione di una sessione di voto. gli utenti in possesso di questo codice saranno in grado di accedere all'elezione corretta dall'applicazione. Può essere anche indicato come id votazione.
Database A	Database contenente informazioni per l'autenticazione
Triplo DES	Algoritmo crittografico usato per salvare le password degli utenti non in chiaro
DRAFT	Stato in cui si trova una votazione appena creata, in questo stato è possibile modificare le caratteristiche della votazione e aggiungere candidati, ma gli utenti non potranno votare. La votazione

REQUISITI FUNZIONALI

REQUISITI UTENTE

1. Il sistema deve dare la possibilità ad un utente di cerare una nuova sessione di voto
2. Le sessioni di voto create devono poter consentire diverse modalità di voto
3. Per ogni sessione di voto creata bisogna poter selezionare una modalità di calcolo del vincitore

4. Solo l'utente che ha creato la votazione dovrà avere i permessi per terminare la votazione
5. Una volta terminata la votazione il sistema deve calcolare il vincitore
6. La votazione può essere modificata, ad esempio si potranno cambiare la data di inizio, la data di fine e il nome della votazione
7. I voti devono essere mandati in maniera anonima
8. Deve essere possibile poter creare delle date di inizio e fine programmata, in cui l'elezione dovrà automaticamente iniziare e finire.

USER STORIES

1. come utente voglio poter visionare tutte le votazioni che ho creato

2. come utente voglio poter votare sia per le votazioni create da me stesso che per le votazioni create da altri utenti

3. come utente voglio essere in grado di avere un'interfaccia chiara e facilmente leggibile

4. come utente voglio poter terminare subito una votazione creata da me

REQUISITI DI SISTEMA

1. L'utente dovrà creare una nuova sessione di voto
   1.1 L'utente dovrà selezionare la modalità di voto
   1.2 L'utente dovrà selezionare la modalità di calcolo del vincitore
   1.3 L'utente dovrà inserire candidati o scrivere la domanda per un eventuale referendum
   1.4 L'utente dovrà impostare la data di inizio e di fine di una votazione

2. Gli utenti per poter votare dovranno essere registrati sulla piattaforma
   2.1 Una volta che un utente si è registrato, potrà votare inserendo l'id di una votazione creata da lui stesso o da un altro utente

3. L'utente dovrà essere in grado di autenticarsi tramite e-mail e password

   3.1 il codice fiscale dell'utente verrà salvato nel database A

   3.2 Dopo che un utente si è autenticato, il sistema mostrerà all'utente le opzioni disponibili per il voto

   3.3 La preferenza di voto dell'utente verrà salvata nel database A

3.4 Una volta espressa la preferenza di voto, il sistema cambierà interfaccia, facendo capire che la votazione ha avuto successo o è fallita

4. L'utente dovrà essere in grado di terminare le votazioni che lui stesso ha creato

4.1 Una volta terminata la votazione il sistema calcolerà il vincitore in base alla modalità selezionata

REQUISITI NON FUNZIONALI

REQUISITI DI PRODOTTO

Requisiti di usabilità:

1. L'interfaccia del sistema di voto dovrà essere implementata con caratteri large per migliorare la leggibilità alle persone anziane
2. Il sistema deve essere utilizzabile su Tablet con sistema operativo android Oreo o superiore
3. Il sistema deve sempre chiedere all'utente la conferma del voto
4. L'interfaccia dell'applicazione non deve essere ridimensionabile

Requisiti di sicurezza:

1. Il sistema non deve permettere ad un utente di votare più di una volta per la stessa votazione
2. Il sistema non deve permettere ad un amministratore generale di vedere gli identificativi degli utenti che hanno votato
3. Il sistema dovrà chiedere l'autenticazione prima di ogni voto espresso da un utente
4. Una volta che l'utente ha espresso il voto, il sistema deve nascondere la preferenza, rendendola anonima ai presenti
5. Il database A deve essere gestito dal ministero degli interni
6. Le password devono essere cifrate con algoritmo TRIPLO DES

Requisiti di affidabilità:

1. Il sistema deve riconoscere se il voto è valido

REQUISITI ORGANIZZATIVI

Requisiti di sviluppo:

1. Il codice del sistema deve essere Open Source
2. Una volta creati i database A il ministero degli interni dovrà gestirli

Requisiti Esterni:

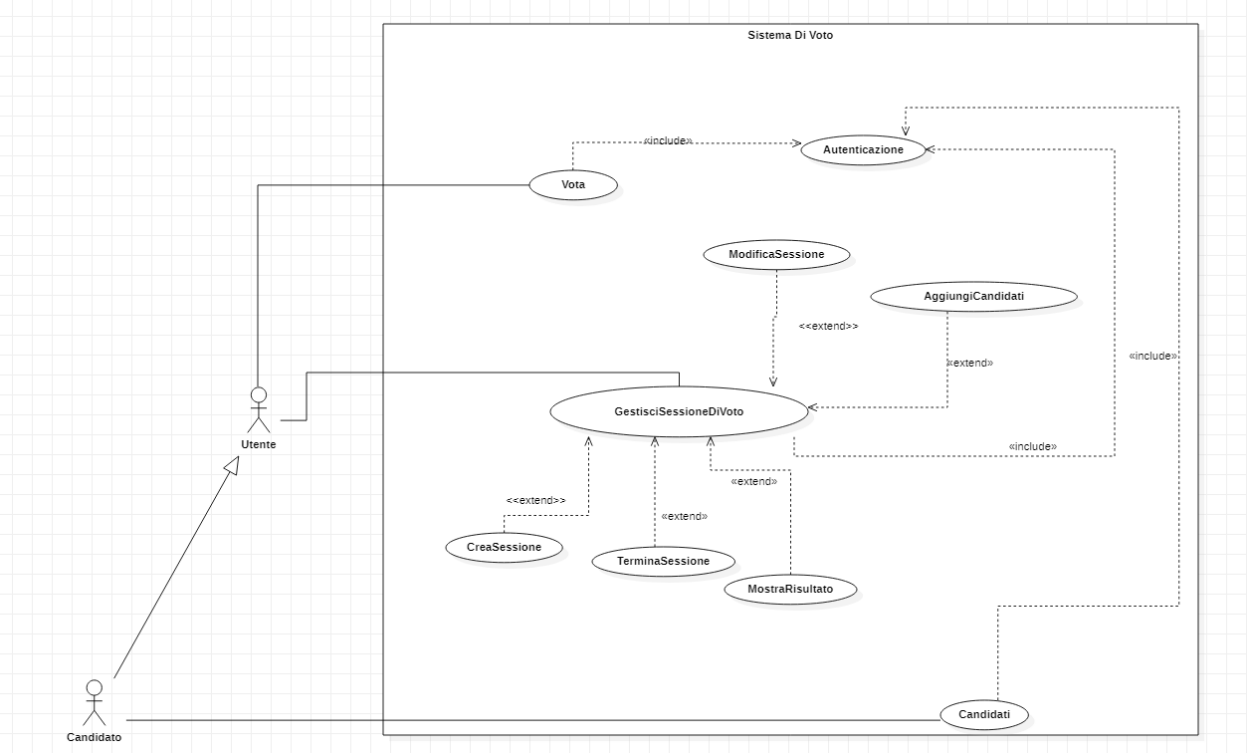
Requisiti normativi:

1. Il sistema deve permettere ad un utente di votare solo una volta
2. Il sistema deve rispettare tutte le normative riguardante il voto attualmente in vigore

Requisiti etici:

1. Bisogna sempre rispettare la privacy dell'utente
2. Il sistema non deve influenzare il voto

DIAGRAMMA DEI CASI D'USO



Nella versione originale dell'applicazione avevamo definito una gerarchia più "ampia" definendo gli utenti "Amministratore" e "Votante"

Procedendo con lo sviluppo ci siamo resi conto che questa gerarchia non aveva tanto senso; infatti, ciascun utente è amministratore delle proprie sezioni, inoltre ciascun utente può votare sia per le proprie votazioni che per le votazioni create da altri

DESCRIZIONE DEGLI SCENARI

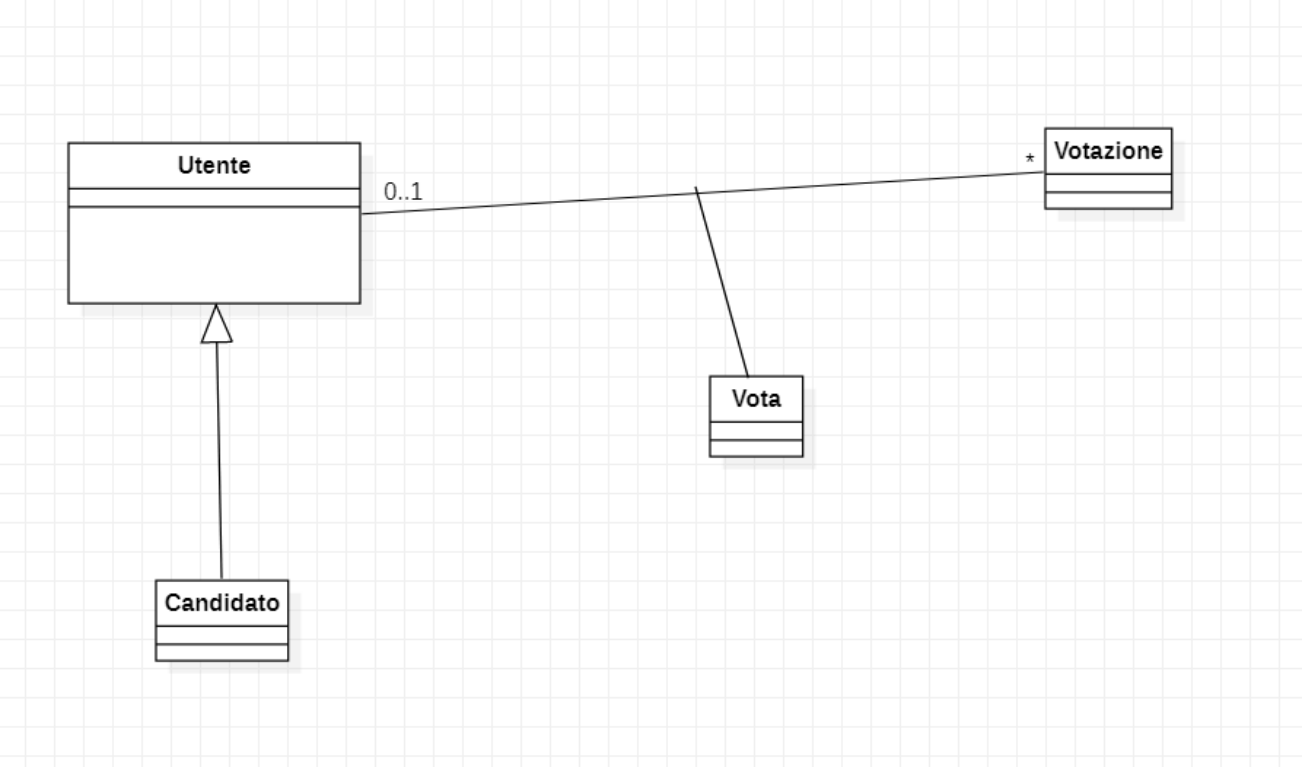
NOME	Vota
SCOPO	Permette ad un utente di votare
ATTORE/I	Utente
PRE-CONDIZIONI	L'utente deve essere registrato, e la votazione deve essere iniziata
TRIGGER	L'utente richiede di votare
DESCRIZIONE	
SEQUENZA EVENTI	1) Nella homepage l'utente si autentifica inserendo l'id della votazione per la quale vuole votare 2) Cliccando su "Vota" l'utente viene portato su una nuova interfaccia dove potrà esprimere la propria preferenza di voto 3) Una volta votato l'utente riceve un messaggio di conferma e non potrà votare nuovamente per la stessa sessione
ALTERNATIVA/E	1) L'utente inserisce un id non valido 2) L'utente non ha inserito delle credenziali valide
POST-CONDIZIONI	La preferenza di voto viene salvata

NOME	Autenticazione
SCOPO	Permette ad un utente di loggarsi
ATTORE/I	Utente
PRE-CONDIZIONI	L'utente deve essere registrato
TRIGGER	L'utente richiede di loggarsi oppure di votare
DESCRIZIONE	
SEQUENZA EVENTI	1) Nella homepage l'utente si autentifica inserendo l'email e la password 2) Cliccando su "Login" L'utente esegue l'accesso nella piattaforma
ALTERNATIVA/E	3) L'utente non ha inserito delle credenziali valide
POST-CONDIZIONI	L'utente viene portato su una nuova interfaccia

NOME	GestisciSessioneDiVoto
SCOPO	Permette ad un utente di gestire le proprie sessioni di voto
ATTORE/I	Utente
PRE-CONDIZIONI	L'utente deve essere registrato
TRIGGER	L'utente esegue il log-in
DESCRIZIONE	Una volta eseguito il log-in un utente può creare delle nuove votazioni, può modificare delle votazioni già create da lui oppure può eliminarle
SEQUENZA EVENTI	1) Una volta fatto il log-in l'utente può decidere di creare una nuova votazione/modificare una votazione/ eliminare una votazione
ALTERNATIVA/E	1) L'utente non ha mai creato nessuna votazione e quindi non potrà eliminarne o modificarne nessuna 1) L'utente sceglie di fare log-out e viene riportato sulla schermata di log-in
POST-CONDIZIONI	In base alla scelta fatta l'utente verrà portato su una diversa interfaccia

NOME	Crea Sessione
SCOPO	Permette ad un utente di creare una nuova sessione di voto
ATTORE/I	Utente
PRE-CONDIZIONI	L'utente deve essere registrato
TRIGGER	L'utente richiede di creare una nuova sessione di voto
DESCRIZIONE	
SEQUENZA EVENTI	1) L'utente dopo aver effettuato il log-in clicca su "Nuova elezione" 2) L'utente inserisce un nome per la votazione 3) L'utente inserisce una data di inizio ed una data di fine 4) L'utente sceglie la tipologia di votazione desiderata 5) L'utente clicca su "Crea"
ALTERNATIVA/E	2) L'utente non inserisce delle date valide 5) L'utente non compila tutti i campi obbligatori prima di cliccare su "crea"
POST-CONDIZIONI	La votazione viene creata con successo

DIAGRAMMA DELLE CLASSI DI PROGETTO



DIAGRAMMI DI SEQUENZA

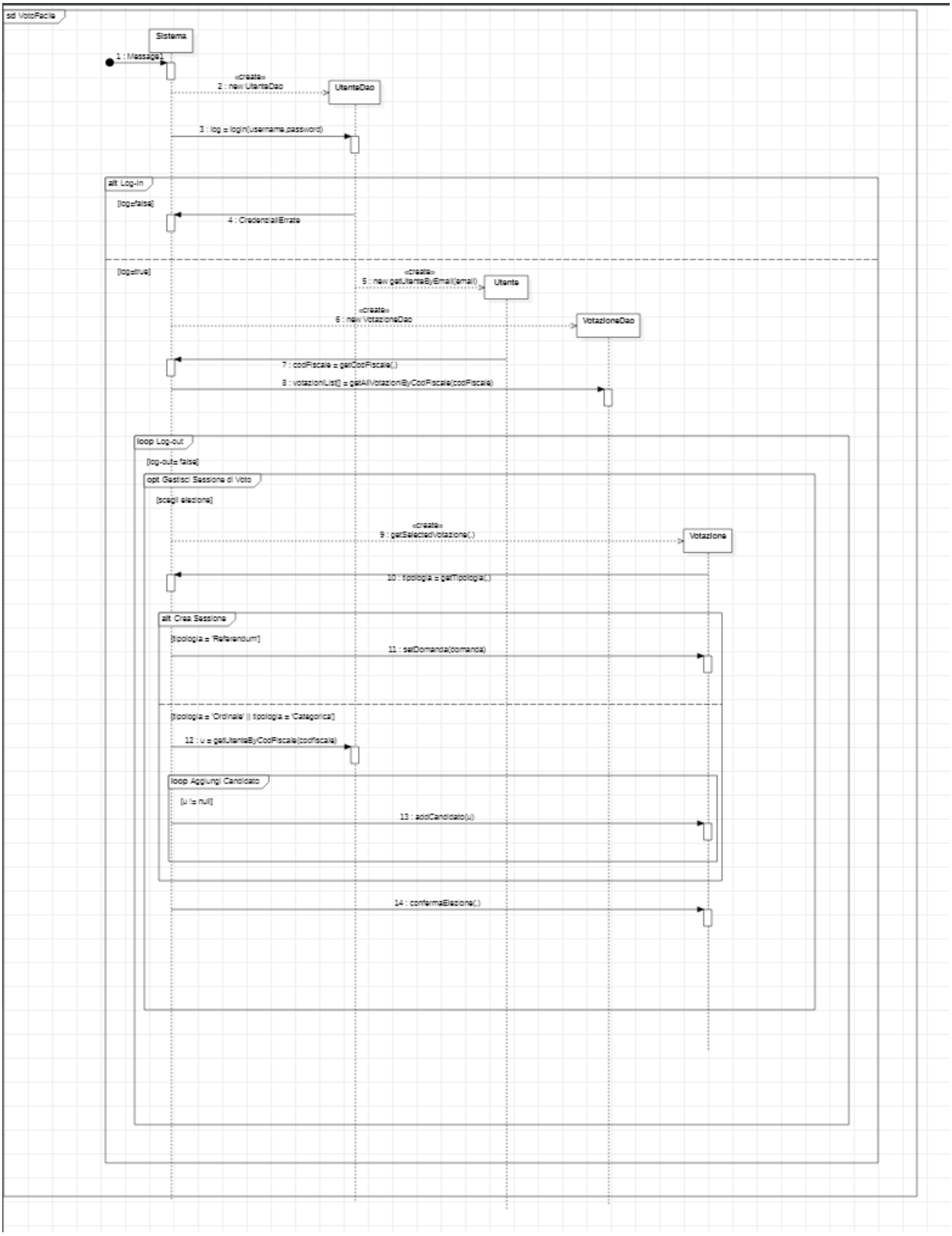
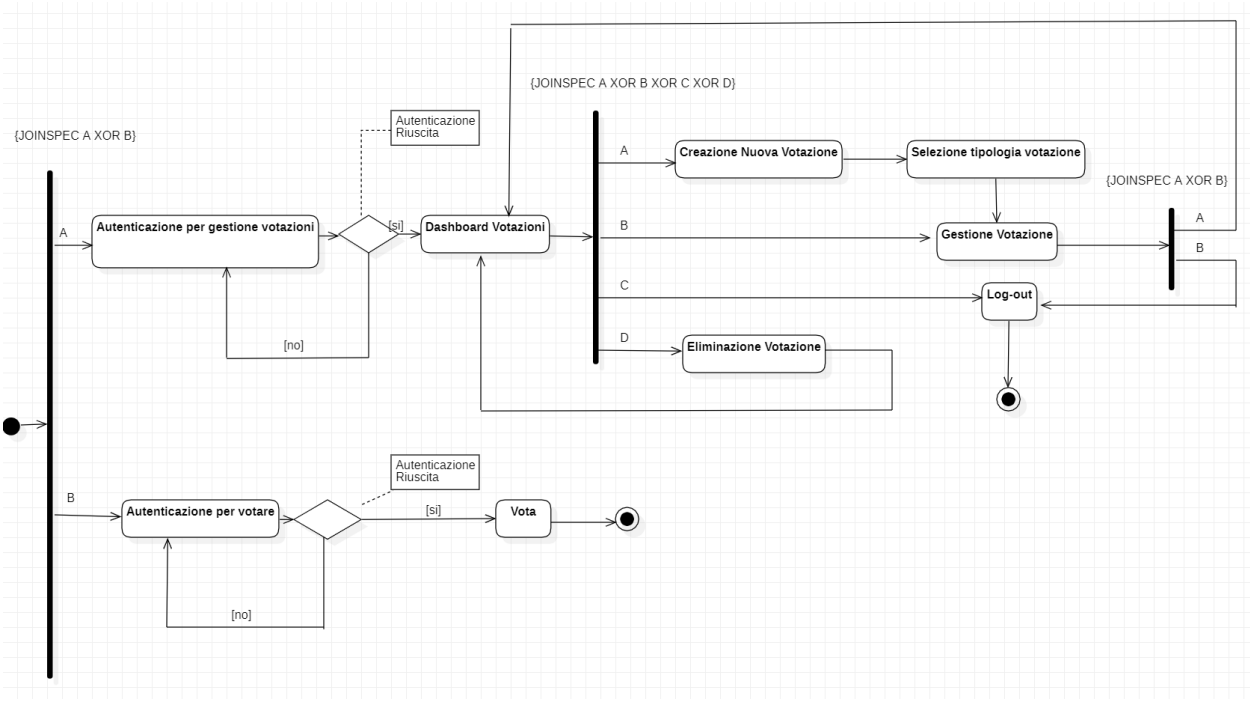
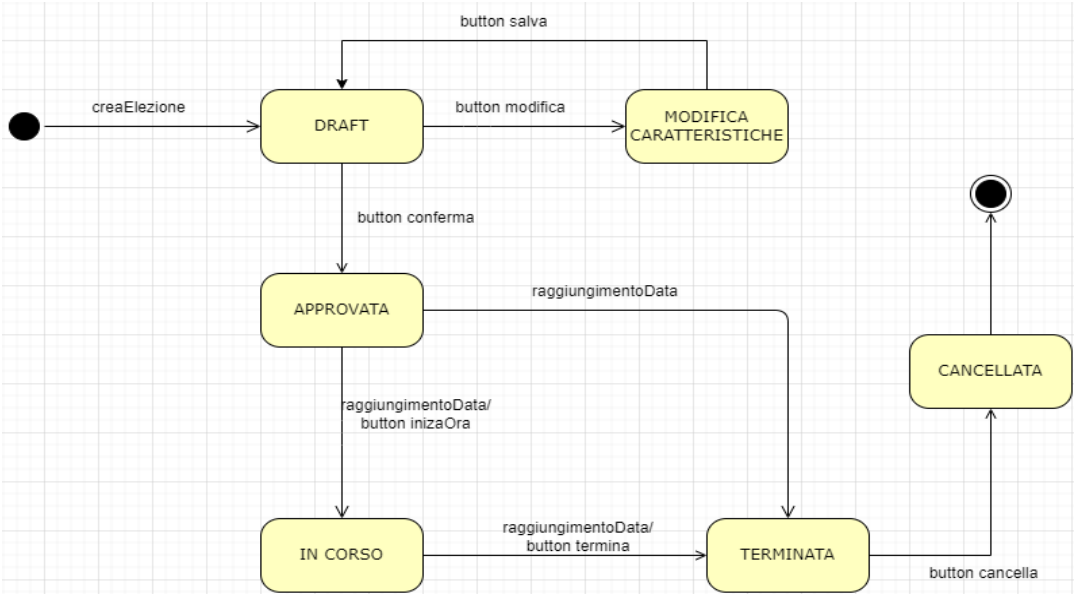


DIAGRAMMA DI ATTIVITA’

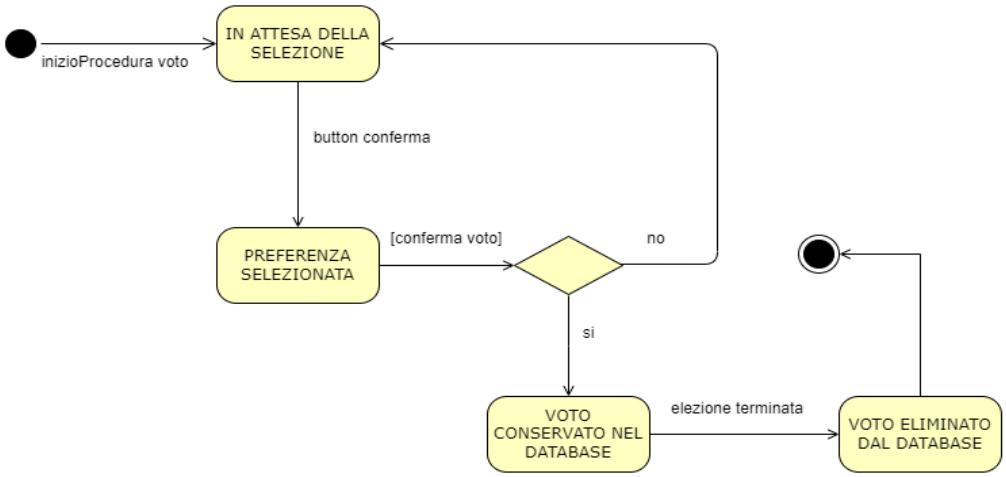


MACCHINE DI STATO



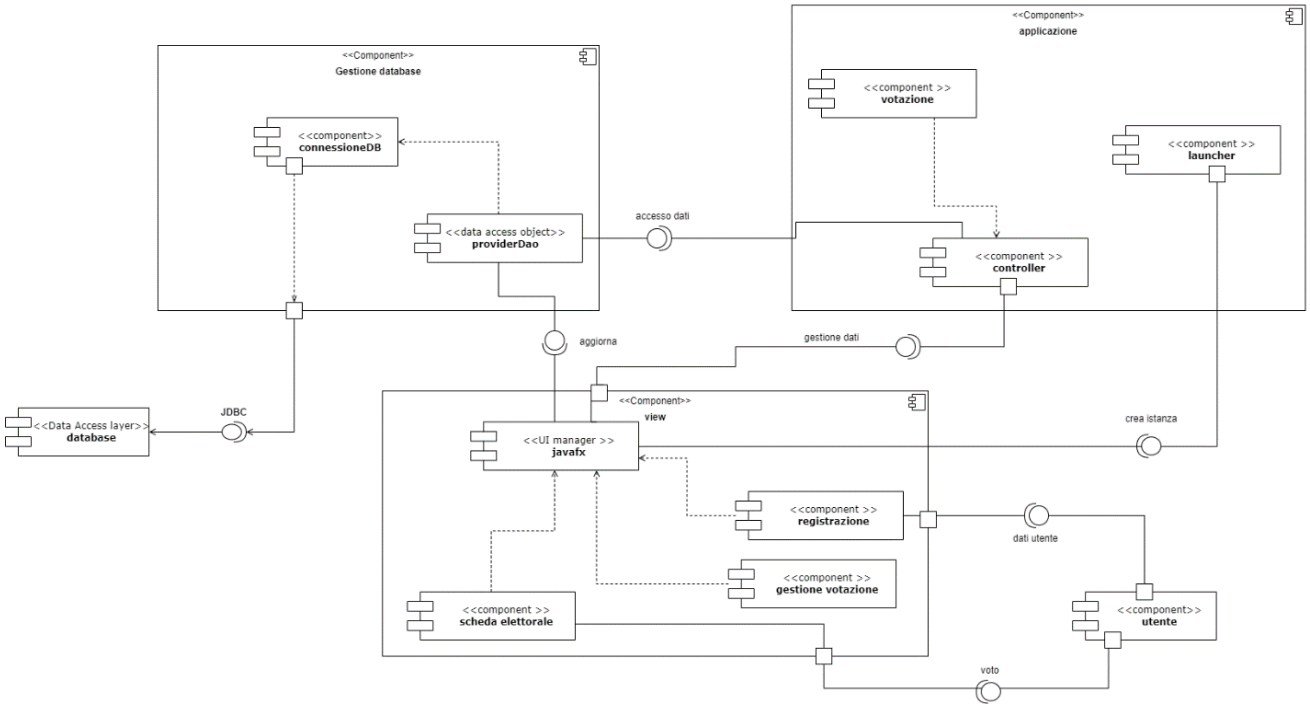
la macchina a stati finiti rappresenta lo stato di una votazione nel suo ciclo di vita, appena creata entra in stato di draft ed è possibile alterare opzioni come la data di inizio e fine, nome e aggiungere candidati, questi eventi sono racchiusi nello stato di modifica caratteristiche.

Una volta che la votazione viene approvata non può essere modificata e al raggiungimento della data di inizio, per gli utenti sarà possibile votare. Una volta terminata apparirà nella schermata delle votazioni create dall'utente e sarà possibile cancellarla



Il diagramma rappresenta lo stato del voto dell'utente, salvato come record nel database, deve essere previsto prima della conferma definitiva una schermata che chiede all'utente se è sicuro della scelta selezionata, una volta che la preferenza è confermata allora il voto verrà salvato nel database, per non sovraccaricare il database il voto deve essere cancellato una volta che l'utente elimina la votazione.

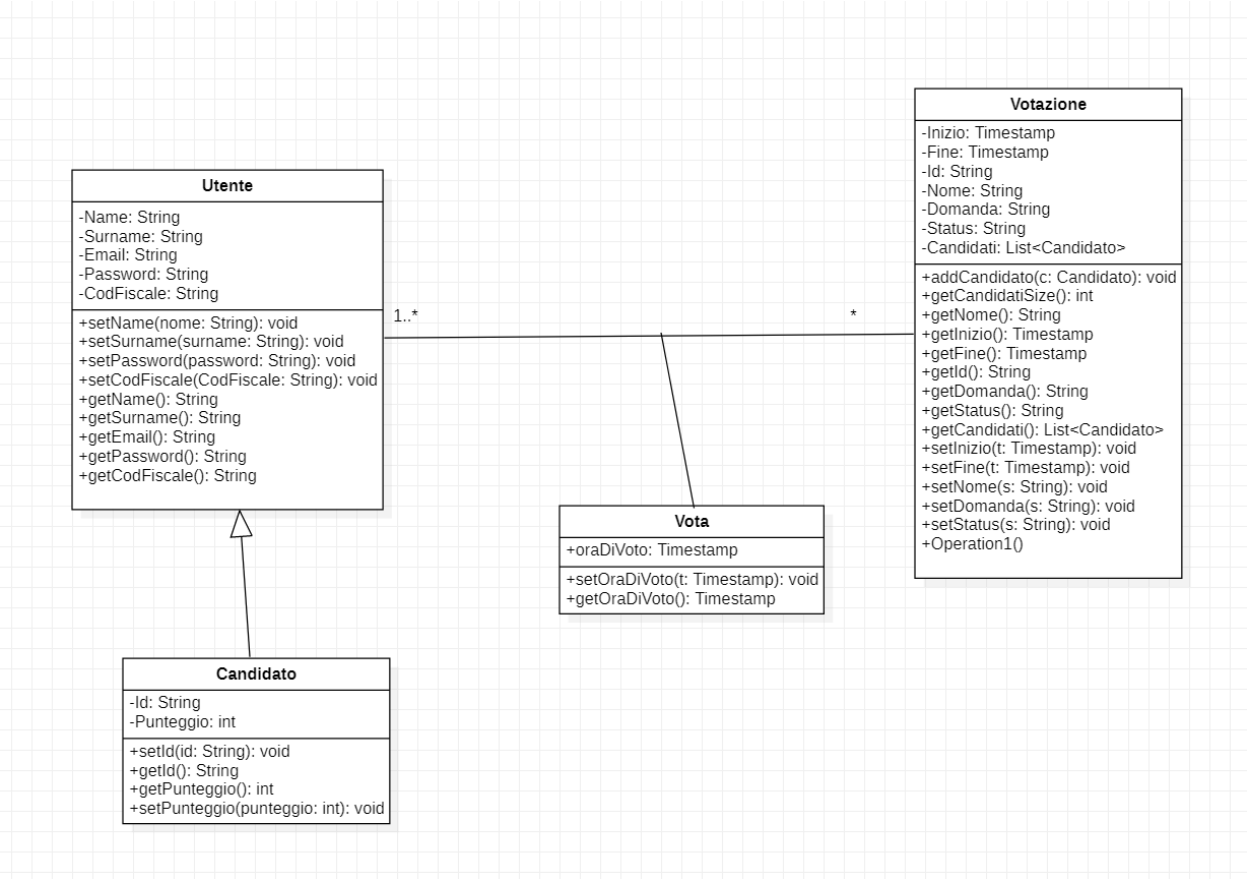
DIAGRAMMA DEI COMPONENTI



(Per la visualizzazione è presente nella cartella del progetto il file **diagrammaComponenti.png**)

Nel diagramma delle componenti si può vedere la distinzione che è stata fatta tra la **view,** e le parti che compongono l'applicazione, il model e il controller, si può osservare come quest'ultimo sia l'intermediario tra view, applicazione e database, e di come gestisca l'accesso e la retribuzione dei dati inseriti dall'utente. La view è composta da classi di javafx, che gestiscono l'interfaccia utene. Il database è stato implementato seguendo i pattern singleton e dao.

DIAGRAMMA DELLE CLASSI DI PROGRAMMA



DISCUSSIONE DEI DESIGN PATTERN UTILIZZATI

Design pattern utilizzati: MVC, Singleton, DAO, Observer.

Per quanto riguarda il Singleton la classe che lo implementa è **SingletonDbConnection**, viene usato per garantire l'unicità della connessione al database, è sto implementato per porre un limite al numero di connessioni che si possono creare, abbiamo notato che con questo particolare DBSM si verificavano malfunzionamenti quando si tenevano aperte più connessioni.

Quando viene istanziata una volta la connessione, viene richiamata la classe quando risulta necessario effettuare un'operazione che coinvolga il database.

Il pattern DAO è utilizzato per le interfacce **UtenteDao** e **VotazioneDao**, e consente di tenere in classi separate le operazioni che vengono effettuate sul database. Le classi contengono delle richieste in forma di query che possono essere richiamate come metodi dalle classi controller quando richiesto. Se si dovesse intervenire solo sulla parte legata al database, l'intervento è focalizzato su queste classi e ciò garantisce una migliore portabilità e manutenibilità del sistema.

il pattern MVC viene istanziato con le classi **Utente,Candidato,Votazione**(model), i **file FXML**(view), e le classi denominate con **controller**(ex: loginController).

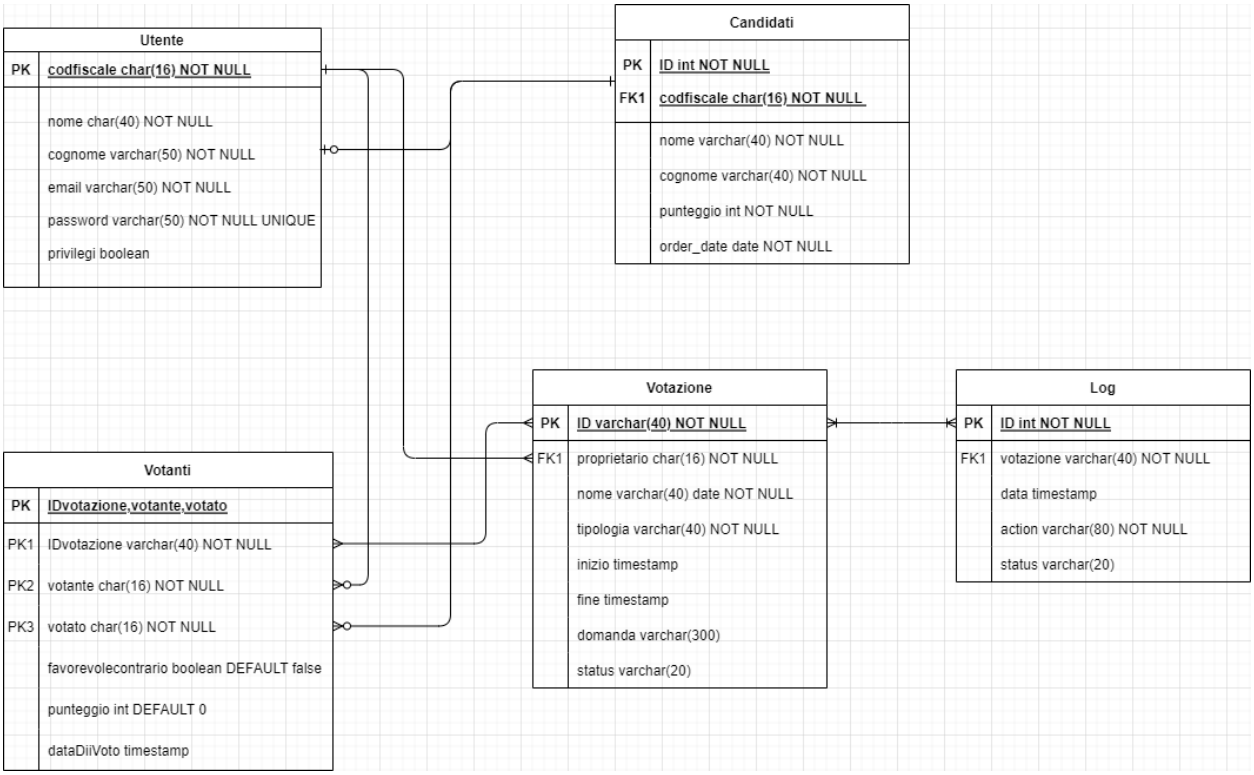
Il pattern è utilizzato per distinguere meglio i moduli funzionali del sistema.

Il le classi controller si occupano della comunicazione tra view e model, aggiornando i dati quando vengono inseriti dall'utente o vengono richiesti al database. La view rappresenta l'interfaccia grafica, ovvero ciò con cui interagirà l'utente finale. le classi model rappresentano tutti gli oggetti con cui sono organizzati i dati dell'applicazione.

Per il pattern **Observer** è stato implementato nell'utilizzo delle **"observable List" (nelle classi CandidatiController, VotazioniController)** di javafx, per la gestione delle tabelle che mostrano l'utente le votazioni, i candidati e i risultati

Inoltre, quando si è nell'interfaccia di modifica della votazione è possibile vedere in ogni momento il nome dell'utente e della votazione su cui si sta lavorando.

GESTIONE DEI DATI PERSISTENTI



Il DBMS utilizzato è postgreSQL. sono presenti 5 tabelle, Utente, candidati, votanti, votazione, log.

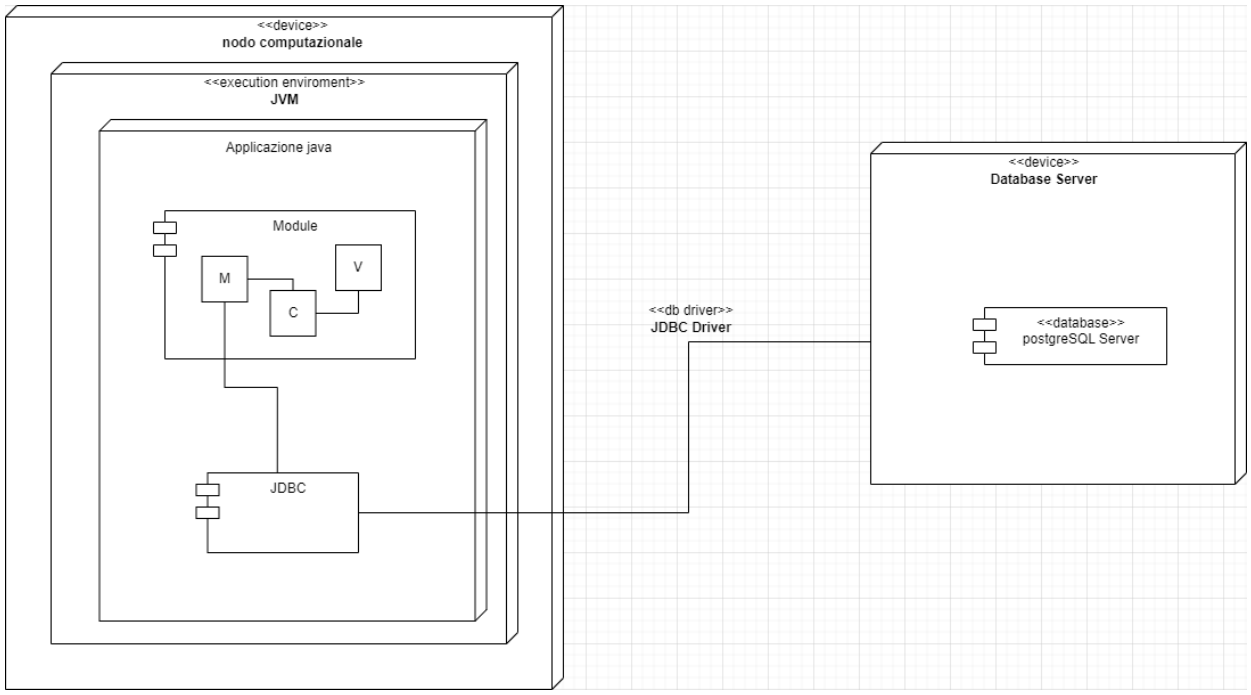
la tabella utente raccoglie i dati delle persone che hanno effettuato la registrazione al sistema, un utente può apparire nella tabella candidato qualora dovesse essere inserito da un utente come candidato di una votazione. La tabella Utente ha inoltre altre 2 relazioni(1 a molti) con la tabella Votazione, che rappresenta appunto le elezioni che l'utente ha creato, e con la tabella votanti, che racchiude i voti per una determinata elezione, ha infatti un collegamento di chiave esterna con la tabella candidati, in questa tabella si memorizza anche il punteggio di ognuno di essi.

La tabella di log contiene un resoconto delle operazioni effettuate sulle votazioni.

DESCRIZIONE DELL'INTERFACCIA GRAFICA

È possibile visualizzare l'interfaccia grafica anche tramite il power-point o pdf.
(presentazioneVotoFacile.pdf, VotoFacile.pptx)

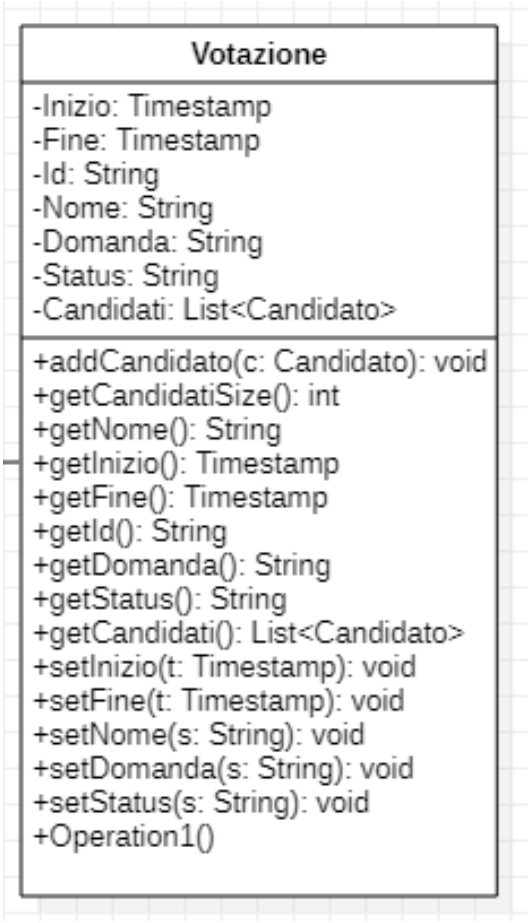
DIAGRAMMA DI DEPLOYMENT



L'applicazione utilizza come architettura fisica esterna il server contente i dati persistenti dell'applicazione, in questo caso è stato utilizzato posgreSQL come dbsm

SPECIFICA DEI VINCOLI

Vincoli OCL per la classe votazione:



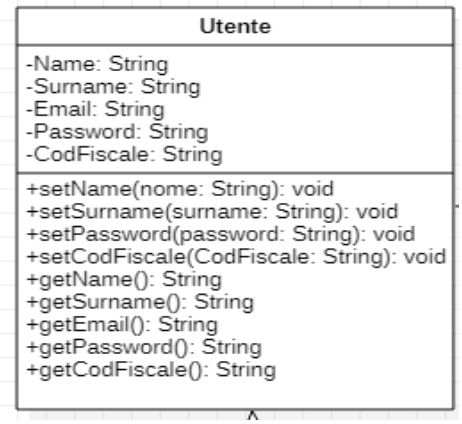
la data d'inizio non deve superare la data di fine:

{ context Votazione
inv: self.Inizio <= self.Fine }

il tipo della votazione deve essere compreso tra "referendum, cardinale e ordinale"

{ context Votazione
inv: self.Inizio <= self.Fine }

Vincoli OCL per la classe Utente:



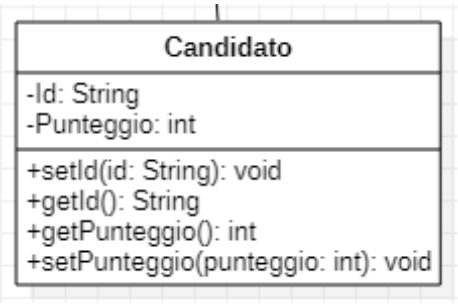
vengono espressi i seguenti vincoli per il controllo del codice fiscale:

1. le prime 3 lettere contengono le consonanti del cognome, prese nel loro ordine.
2. le 3 lettere successive contengono le consonanti del nome
3. i successivi 2 numeri devono corrispondere alle ultime due cifre dell'anno di nascita.
4. una lettera che indica il mese di nascita
5. due numeri indicano il giorno di nascita
6. una lettera e tre numeri indicano il Comune di nascita, per chi è nato all'estero la sigla inizia per Z ed è seguita dal numero che identifica lo Stato di nascita.
7. un carattere alfabetico di controllo generato attraverso uno speciale algoritmo e che chiude ogni codice fiscale valido

la password deve essere lunga almeno 8 caratteri

{context Utente
inv: self.password.length >= 8 }

Vincoli OCL per la classe candidato:



il punteggio deve essere un valore non nullo e maggiore di 0

{context Candidato
inv: self.Punteggio >= 8 and self.Punteggio <> null }

c nella classe SingletonDbConnection deve esistere in un'unica istanza
{context SingletonDbConnection
inv: self.allInstances()->isUnique(c) }

DESCRIZIONE DEL TESTING

i test effettuati sono implementati nelle classi EncriptionTest, UtenteDaoImplTest e VotazionedaoTest.

Per implementare correttamente le politiche di sicurezza il sistema salva l'hash delle password degli utenti **EncriptionTest** è una suite scritta per testare il corretto funzionamento della funzione encrypt(), che si occupa di calcolare tale hash, e che non avesse particolari comportamenti nel caso di input non validi.

UtenteDaoImpl si occupa della gestione dei dati degli utenti, per questo la test suite si occupa di controllare la gestione completa di un utente: creazione(addUtente()), cancellazione (deleteUtente()) e ottenimento dei sui dati(getUtente()), inoltre vengono provati vari tentativi di accesso testando il metodo login() su vari input.

Il testing della classe **VotazioneDaoImpl** si concentra sui metodi: **getRisultatoReferendum()**, e **getRisultatoVotazione(),** per assicurare che i vari metodi di calcolo della votazione siano corretti, il test viene effettuato su record che crea il test del metodo **addVotazione()** e i voti creati dal test del metodo **vota().**

Note per l'installazione e l'utilizzo

È stato utilizzato java come linguaggio di programmazione, con l'implementazione dello strumento **Maven** per la gestione delle librerie e delle dipendenze.

Lo sviluppo è avvenuto tramite l'utilizzo dell'IDE **intellij idea** su windows 10.

Il server è ospitato sul servizio cloud Heroku, è presente nella cartella il dump del database.

per utilizzare l'applicazione è sufficiente avviare il file **VotoFacile.jar** presente nella cartella **VotoFacileApp** (o sul file VotoFacile.bat se il jar dovesse funzionare) oppure aprire il progetto su intelliJ e lanciare in esecuzione la classe Main.

Per accedere all'applicazione è sufficiente registrarsi nell'apposita pagina, oppure utilizzare le credenziali: **"email@gmail.com" e "password"** per accedere direttamente.