

**Documentazione
Progetto Ingegneria del
Software**

-

**Requirement Analysis
Specification Document
(RASD)**

-

**Sistema di Voto e
Scrutinio Elettronico**

Gruppo:

Federico Millefanti 922812

Lorenzo Borrelli 923420

Anno Accademico 2021-2022

Indice

Introduzione	2
1.1 Obiettivi	2
1.2 Destinatari	3
1.3 Scopo del Sistema	3
1.4 Glossario (definizioni, acronimi)	4
Descrizione Generale	5
2.1 Caratteristiche del Prodotto	5
2.2 Ambiente Operativo	5
2.3 Documentazione per l'utente	6
Specifica dei Requisiti	9
3.1 Requisiti Funzionali	9
3.1.1 Autenticazione	9
3.1.2 Inserimento Votazione	10
3.1.3 Modi per definire la vincita	10
3.1.4 Inserimento sessione	11
3.1.5 Votazione	11
3.2 Requisiti di Interfacce Esterne	11
3.2.1 Interfacce Utente e Software	11
3.3 Requisiti non Funzionali	14
3.3.1 Requisiti di Prestazione	15
3.3.2 Attributi di Qualità del Software	15
3.3.3 Requisiti di Sicurezza	15

4 Progettazione del Sistema	16
4.1 Diagrammi dei casi d'uso	16
4.2 Diagrammi delle Classi	19
4.2.1 Application	19
4.2.2 BaseClasses	21
4.2.3 dbManagement	25
4.2.4 DebugTool	26
4.2.5 Pages	27
4.3 Macchine di Stato	31
4.4 Diagramma delle Componenti	32
4.5 Diagramma di Deployment	33
4.6 Vincoli OCL	34
4.7 JUnit	36
4.8 Pattern Utilizzati	37
4.9 Gestione del Database e dei Dati Persistenti	38
5 Appendice e Note per l'Installazione	39

1. Introduzione

1.1 Obiettivi

L'obiettivo di questo progetto è quello di sviluppare un sistema di voto e scrutinio elettronico in modo tale che il committente sia soddisfatto ed il programma rispetti tutte le richieste fatte, eseguite in modo dettagliato, ordinato e completo.

In particolare all'interno di questo documento potete trovare disposti in modo organizzato tutti i capitoli e sottocapitoli che contengono la totalità del sistema progettato; dalla specifica dei requisiti all'implementazione del progetto, dall'utilizzo dei pattern ai contratti dei vincoli stipulati e molto altro ancora.

1.2 Destinatari

In questo documento chiamato RASD o SRS sono contenute tutte le varie indicazioni per i destinatari del programma ovvero gli Elettori e gli Admin del sistema.

Qualora qualche ente volesse utilizzare questo programma per un utilizzo pubblico è tenuto a verificare le linee guida del Ministero e controllare che le politiche di privacy siano dimensionate per l'utilizzo che se ne vuole fare.

Gli Elettori sono tutta quella fetta di popolazione che, in una determinata circostanza di referendum o elezione, è desiderosa di esprimere il proprio voto attraverso la piattaforma da noi ideata.

Gli Admin del sistema sono tutte quelle persone incaricate dall'ente di consultazione elettorale di gestire la votazione a 360 gradi, dall'inserimento alla modifica di tutti gli attori presenti nel software.

1.3 Scopo del Sistema

Lo scopo del sistema è quello di dare a disposizione una piattaforma che tenga conto, in modo imparziale, dei voti di un determinato referendum o elezione.

Il software garantisce una precisione nell'accesso per gli elettori ma anche una facilità e completezza nella creazione di nuove sessioni di voto che possono essere di natura nazionale, regionale o addirittura comunale.

La struttura del programma permette di avere più votazioni sulla stessa piattaforma, il voto segreto ed infine l'elettore può capire attraverso le notazioni presenti nel software se la votazione è ancora aperta oppure è finito il tempo massimo per esprimere la preferenza.

Per ottenere un corretto funzionamento del sistema, abbiamo pensato di facilitare l'utilizzo del programma, con schermate semplici con pochi tasti chiari ed evidenti, in modo tale da consentire una corretta fruibilità.

1.4 Glossario (definizioni, acronimi)

Referendum senza Quorum: si procede al conteggio dei voti indipendentemente se abbia partecipato o meno alla consultazione la maggioranza degli aventi diritto al voto.

Referendum con Quorum: si procede al conteggio dei voti espressi solo nel caso in cui abbia partecipato alla consultazione la maggioranza degli aventi diritto al voto.

Voto categorico: l'elettore inserisce una preferenza per un candidato (o gruppo/partito).

Voto categorico con preferenze: l'elettore inserisce una preferenza per un gruppo/partito e ha la possibilità di indicare una o più preferenze tra i candidati del gruppo/partito selezionato (niente voto disgiunto).

Voto ordinale: all'elettore è richiesto di ordinare i candidati (o gruppi/partiti) presenti nella scheda in base alle proprie preferenze.

Sessione di voto: momento in cui l'utente può esprimere la sua preferenza, il tempo è gestito dagli Admin del sistema.

VOTATO: indica una sessione di voto in cui l'utente ha già espresso il proprio voto.

SCADUTA: indica una sessione di voto non più disponibile, poiché il tempo necessario per esprimere il proprio voto è terminato.

Elettore: persona del popolo che ha diritto di voto

Admin o Admin del sistema: persona incaricata dall'ente che utilizza il programma per inizializzare una sessione di voto, una registrazione o semplicemente per controllare che tutto il sistema funzioni.

2. Descrizione Generale

2.1 Caratteristiche del Prodotto

Il prodotto si presenta completo sotto ogni punto di vista. E' adatto a tutti gli enti pubblici che hanno desiderio di trovare un mezzo rapido, veloce ed efficace per votazioni di ogni genere.

Le varie schermate presentano pochi bottoni/pulsanti ed anche quella riservata agli admin è semplice e chiara.

Gli utenti non ancora registrati potranno effettuare la propria registrazione direttamente sull'applicazione che salverà i dati in un apposito server.

Sarà possibile avviare una nuova sessione qualvolta necessario andando ad imporre un tempo massimo per gli elettori di esprimere il proprio voto.

Gli operatori incaricati di gestire la sezione dedicata agli "Admin" potranno usufruire di tutte le funzionalità necessarie per rendere possibile una votazione.

Sarà possibile aggiungere Partiti e Candidati attraverso un form compilativo oltre ad andare a selezionare la modalità di voto con nome e descrizione.

Nella schermata iniziale l'elettore (o anche l'admin) potrà visualizzare le varie sessioni aperte o scadute.

Nelle sessioni aperte è presente la dicitura "VOTATO" nel momento in cui la persona avrà confermato il proprio voto.

La dicitura "SCADUTA" indica la chiusura della sessione ovvero la fine del tempo massimo in cui si può esprimere la propria preferenza.

2.2 Ambiente Operativo

L'ambiente Operativo scelto dal team di progettazione del software (Borrelli-Millefanti) è Eclipse.

Il linguaggio di programmazione adottato è Java, mentre la parte grafica è stata implementata con JavaFX.

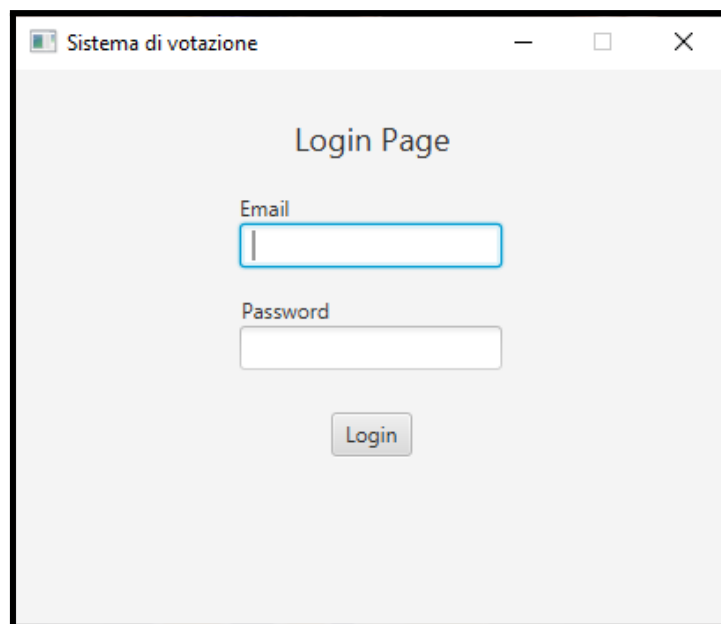
Per la memorizzazione dei dati persistenti e delle tabelle del database si è scelto di utilizzare MYSQL attraverso la piattaforma XAMPP.

Le librerie utilizzate di JavaFX sono: swt, base, controls, fxml, graphics, media, swing, web.

2.3 Documentazione per l'utente

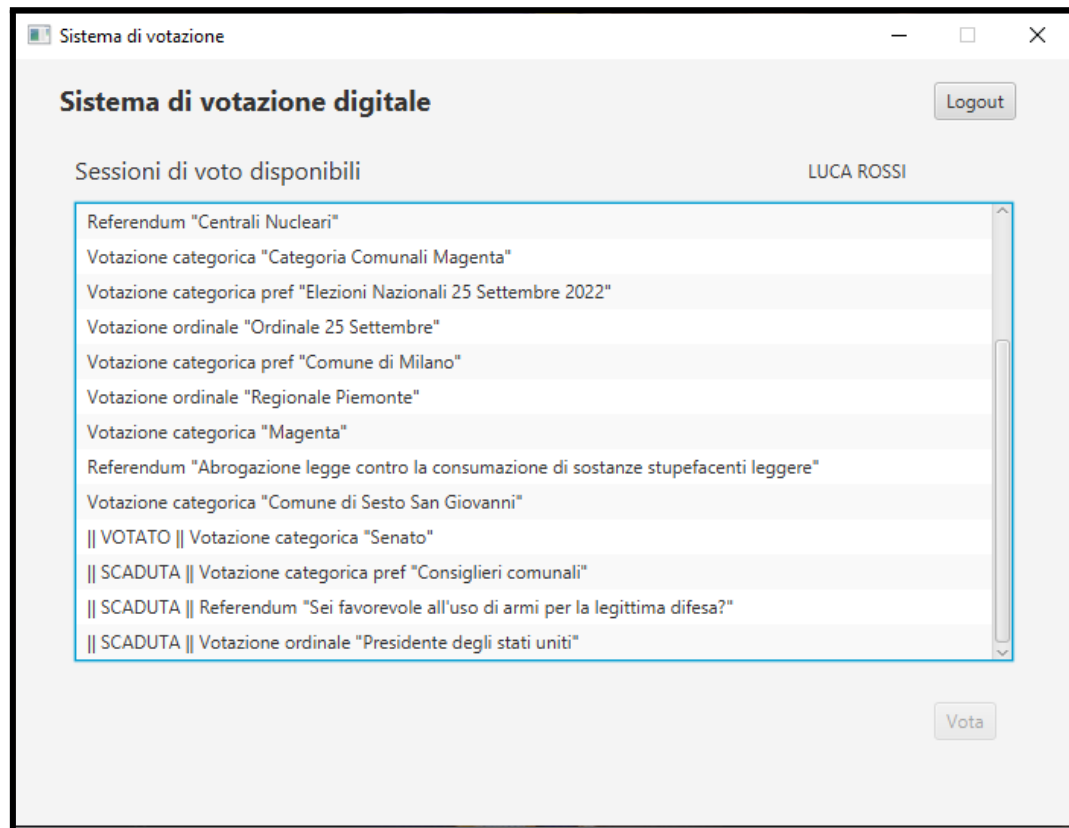
In questo sottocapitolo verranno descritte tutte le varie schermate visualizzate dell'elettore.

Dopo aver eseguito la Registrazione, fatta da un Admin del sistema, si può iniziare ad utilizzare il programma senza preoccuparsi di nulla avendo sempre l'accortezza di ricordarsi il proprio indirizzo email e la password utilizzate nella fase precedente. Andiamo quindi a vedere la seconda schermata, molto semplice, con due campi da compilare.



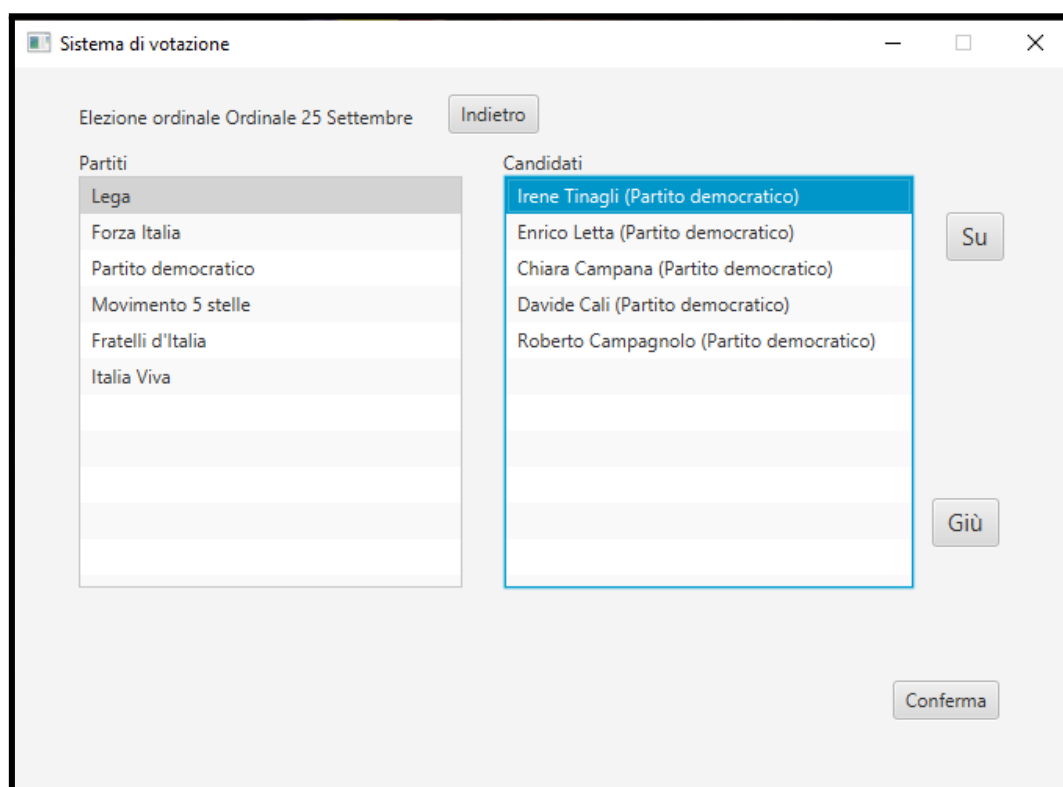
In seguito all'autenticazione, l'utente visualizzerà una schermata dove potrà vedere tutte le varie sessioni, sia quelle disponibili, sia quelle già

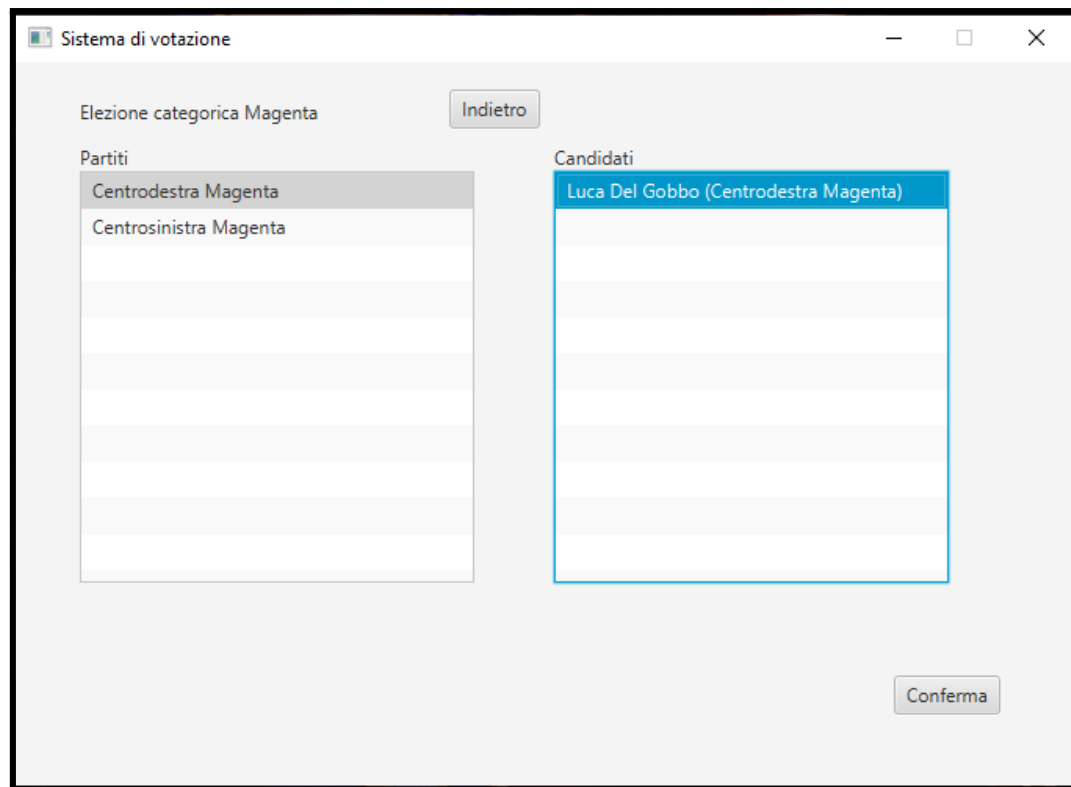
votate ma ancora attive (con la dicitura VOTATO), sia quelle non più disponibili (con la dicitura SCADUTA).



Per le prossime tre schermate verranno visualizzate le possibili modalità di voto con i rispettivi esempi. Il primo indica il Referendum, due bottoni, sì e no, con possibilità da parte degli incaricati di sistema di modifica e descrizione dell'oggetto. Il secondo esempio è quello della votazione ordinale nazionale che avverrà il 25 di Settembre con possibilità di scelta dei partiti e ordinamento dei candidati. L'ultimo è quello delle votazioni comunali di Magenta con possibilità di scegliere tra le liste proposte dal Comune. Le descrizioni, gli Oggetti, i Nomi sono stati presi dalle informazioni del sito del Governo o dai siti relativi ai vari Comuni. Chiaramente non è stata aggiunta la totalità delle scelte ed i nomi delle varie sessioni sono stati volutamente scelti semplici come esempio di utilizzo. Gli Admin del Sistema incaricati dall'Ente Nazionale, Regionale, Provinciale o Comunale saranno responsabili di ogni modifica

apportata al programma che lascia libero arbitrio sulle scelte di personalizzazione.





3. Specifica dei Requisiti

3.1 Requisiti Funzionali

Nei paragrafi successivi verranno riportati tutti i requisiti funzionali dell'intero sistema suddivisi in macro-categorie che servono semplicemente per una maggiore fruibilità. La totalità dei requisiti è stata discussa e fa riferimento ai requisiti di progettazione.

3.1.1 Autenticazione

Nel processo di autenticazione abbiamo voluto mantenere una schermata singola di login sia per l'utente elettore sia per l'admin del sistema in modo tale che anche gli admin di sistema possano votare a loro volta essendo anche elettori.

Dopo aver effettuato l'accesso con le proprie credenziali sarà possibile per gli elettori votare mentre invece per gli admin di sistema, dopo aver fatto accesso con l'apposito pulsante, sarà possibile gestire la piattaforma.

3.1.2 Inserimento Votazione

La possibilità di inserimento della votazione sarà possibile solo agli Admin di sistema. Le modalità di voto sono quattro:

- Voto Ordinale: all'elettore è richiesto di ordinare i candidati (o gruppi/partiti) presenti nella scheda in base alle proprie preferenze.
- Voto Categorico: l'elettore inserisce una preferenza per un candidato (o gruppo/partito).
- Voto Categorico con preferenza: L'elettore potrà inserire una preferenza per un gruppo/partito e avrà la possibilità di indicare una o più preferenze tra i candidati del gruppo/partito selezionato (niente voto disgiunto).
- Referendum: L'elettore potrà esprimere il suo parere in merito ad una domanda posta nella quale verrà chiesto se si è favorevoli o contrari ad un determinato quesito.

3.1.3 Modi per definire la vincita

La definizione del vincitore per l'elezione dipende dalle modalità di voto. Qui di seguito le varie regole:

- Maggioranza: il vincitore è il candidato che ha ottenuto il maggior numero di voti.
- Maggioranza Assoluta: il vincitore è il candidato che ha ottenuto la maggioranza assoluta dei voti, cioè il 50% + 1 dei voti espressi.
- Referendum Senza Quorum: si procede al conteggio dei voti indipendentemente se abbia partecipato o meno alla consultazione la maggioranza degli aventi diritto al voto.
- Referendum Con Quorum: si procede al conteggio dei voti espressi solo nel caso in cui abbia partecipato alla consultazione la maggioranza degli aventi diritto al voto.

3.1.4 Inserimento sessione

L'inserimento di una nuova sessione di voto è compito dell'admin del sistema. L'incaricato si troverà davanti alla schermata dove potrà visionare i partiti ed i candidati. Inoltre potrà aggiungere e togliere partiti come potrà aggiungere tramite Nome e Cognome i candidati. La scelta del partito per l'inserimento di un nuovo candidato potrà avvenire tra partiti precedentemente inseriti o su nuovi partiti appena aggiunti. Stesso discorso vale per i partiti che potranno essere inseriti o scelti da pregresse votazioni. In conclusione all'inserimento di una nuova votazione si dovrà decidere se la sessione sarà con voto categorico, ordinale, categorico con preferenza o referendum ed in caso scegliere se sarà necessario utilizzare il Quorum.

3.1.5 Votazione

La votazione consiste semplicemente nel scegliere la propria preferenza in merito ad una determinata sessione aperta. In questa fase l'elettore esprime la scelta in base alle modalità di voto proposte in modo anonimo (come spiegato nel paragrafo 2.3).

Lo scrutinio finale avverrà tramite un apposito pulsante presente nella sezione dedicata agli operatori di sistema. La visione dell'esito della votazione potrà essere visualizzata solamente dall'admin di sistema che sarà dovrà essere il responsabile della corretta divulgazione dell'esito.

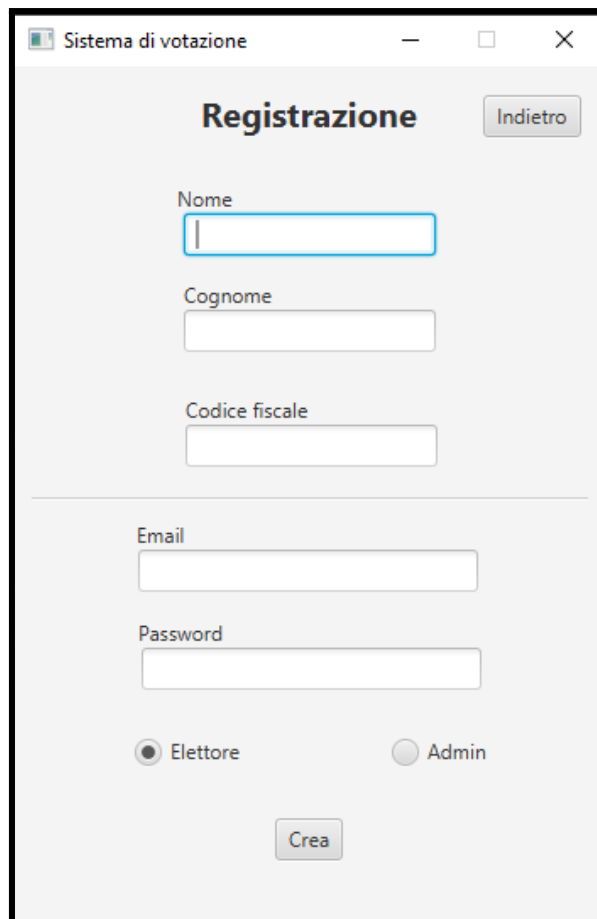
3.2 Requisiti di Interfacce Esterne

3.2.1 Interfacce Utente e Software

Le interfacce Utente sono state spiegate abbondantemente nel paragrafo 2.3 chiamato "documentazione per l'utente".

In questo specifico paragrafo volevamo invece presentarvi e spiegare le interfacce dell'admin del sistema con tutte le descrizioni e regole per il corretto utilizzo della piattaforma.

Entrando con l'apposito pulsante "ADMIN" nella pagina delle sessioni, che è identica sia per elettori che per gli impiegati di sistema, si entra in una finestra dedicata. In questa finestra è possibile rispondere a molteplici richieste. Prima fra tutte la registrazione dell'utente che avverrà solamente tramite l'admin che gestirà tutti gli elettori che votano per la prima volta, o che devono rifare l'account per accedere alla piattaforma. In questa pagina è necessario inserire tutti i dati richiesti: Nome, Cognome, Codice Fiscale, Email e Password. Ovviamente ci sono dei vincoli e delle restrizioni sulla creazione di un nuovo account, per esempio bisogna per forza riempire tutte le caselle (c'è un segnale se questo non viene fatto), oppure se il Codice Fiscale inserito non è corretto o se l'email è inesistente.



The screenshot shows a web application window titled "Sistema di votazione". Inside, there is a section titled "Registrazione" with a button labeled "Indietro" (Back) in the top right corner. Below the title, there are five input fields: "Nome" (Name), "Cognome" (Surname), "Codice fiscale" (Tax Code), "Email", and "Password". The "Nome" field is currently active, indicated by a blue border. Below these fields, there are two radio buttons: "Elettore" (Elector) and "Admin". The "Elettore" radio button is selected. At the bottom of the form, there is a button labeled "Crea" (Create).

Oltre alla pagina di Registrazione l'Admin di sistema potrà aggiungere partiti (nella parte sinistra della finestra) o aggiungere i candidati (nella parte destra della finestra). Dopo aver aggiunto i rispettivi membri votabili per le elezioni, l'impiegato di sistema potrà creare una nuova sessione di voto andando a scegliere tra le varie proposte: ordinale, categorica, categoria con preferenza, referendum e se è necessario o meno il quorum. Le sessioni di voto sono personalizzabili con nome o descrizione in modo tale da rendere chiaro cosa si sta andando a votare.

Sistema di votazione

Admin

Registrazione View Sessions Logout

Lista partiti

- Lega
- Forza Italia
- Partito democratico
- Movimento 5 stelle
- Fratelli d'Italia
- Italia Viva
- Centrodestra Magenta
- Centrosinistra Magenta

Aggiungi partito

AGGIUNGI

Lista candidati

- Emma Cioffi (Forza Italia)
- Paola Pesavento (Forza Italia)
- Cristina Spiller (Forza Italia)
- Giovanni Rizzo (Movimento 5 stelle)
- Enrico Civero (Movimento 5 stelle)
- Francesca Ferraro (Movimento 5 stelle)
- Davide Galvan (Movimento 5 stelle)
- Francesca Amatulli (Movimento 5 stelle)
- Giorgia Meloni (Fratelli d'Italia)

Aggiungi candidato (seleziona un partito)

AGGIUNGI

Crea votazione (Seleziona in alto i partiti da includere)

☒ Ordinale ☐ Categorica ☐ Categorica con preferenza ☐ Referendum

QUORUM

☐ SI ☒ NO

Nome o descrizione

Crea

Infine l'Admin può controllare le sessioni attive e può tranquillamente votare essendo anch'esso oltre che un impiegato di sistema anche un elettore tramite l'apposito pulsante in alto a destra "View Sessions".

3.3 Requisiti non Funzionali

I requisiti non funzionali presenti qui sotto saranno in grado di rispondere ad ogni vostro dubbio sull'utilizzo della piattaforma che sarà disponibile sia in loco, sui dispositivi presentati dall'ente che usufruisce del software, oppure online tramite computer fissi, pc portatili, smartphone o tablet.

- Tentativo di Registrazione fallito: durante il processo di registrazione di un nuovo utente elettore o admin non vengono inseriti tutti i campi necessari.
- Modalità di voto: potrà essere scelta una ed una sola modalità di voto e potrà essere scelta se farla con o senza quorum.
- Calcolo del vincitore: il calcolo del vincitore sarà stabilito dalla modalità di voto scelta.
- Sessioni aperte: quando si apre una nuova sessione non sarà possibile cancellarla o rimuoverla ma soltanto concluderla.
- Voto unico: è possibile esprimere solo un voto ed unico per elettore, quando si è confermato il voto non sarà più possibile tornare indietro.
- Anonimo: il voto sarà segreto e non sarà in nessun modo riconducibile all'elettore
- Conferma di voto: si può verificare che il voto è stato correttamente inserito dalla pagina sessioni con la dicitura "VOTATO".

3.3.1 Requisiti di Prestazione

Le prestazioni della piattaforma dipenderanno esclusivamente dall'ente che acquisterà il software che dovrà responsabilizzare gli Admin del sistema per un corretto uso di tutto il pacchetto. Qualsiasi dispositivo utilizzato per utilizzare la piattaforma dovrà correttamente essere connesso alla rete internet in modo tale da poter salvare tutti i dati, da quelli della registrazione a quelli delle sessioni ma soprattutto i voti degli elettori.

3.3.2 Attributi di Qualità del Software

Il software è completo sotto ogni punto di vista. La personalizzazione è il fattore principale su cui il team di sviluppo si è concentrato. Sarà possibile utilizzare qualsiasi dispositivo elettronico adatto per il voto, oppure utilizzare i seggi elettorali già presenti utilizzando in loco macchine o apparecchiature consone per facilitare l'utilizzo alle persone più anziane o persone portatrici di handicap. In generale la piattaforma può sostenere grandi carichi di utilizzo sia a livello dei dati sia a livello della contemporaneità di utilizzo nei vari seggi/online.

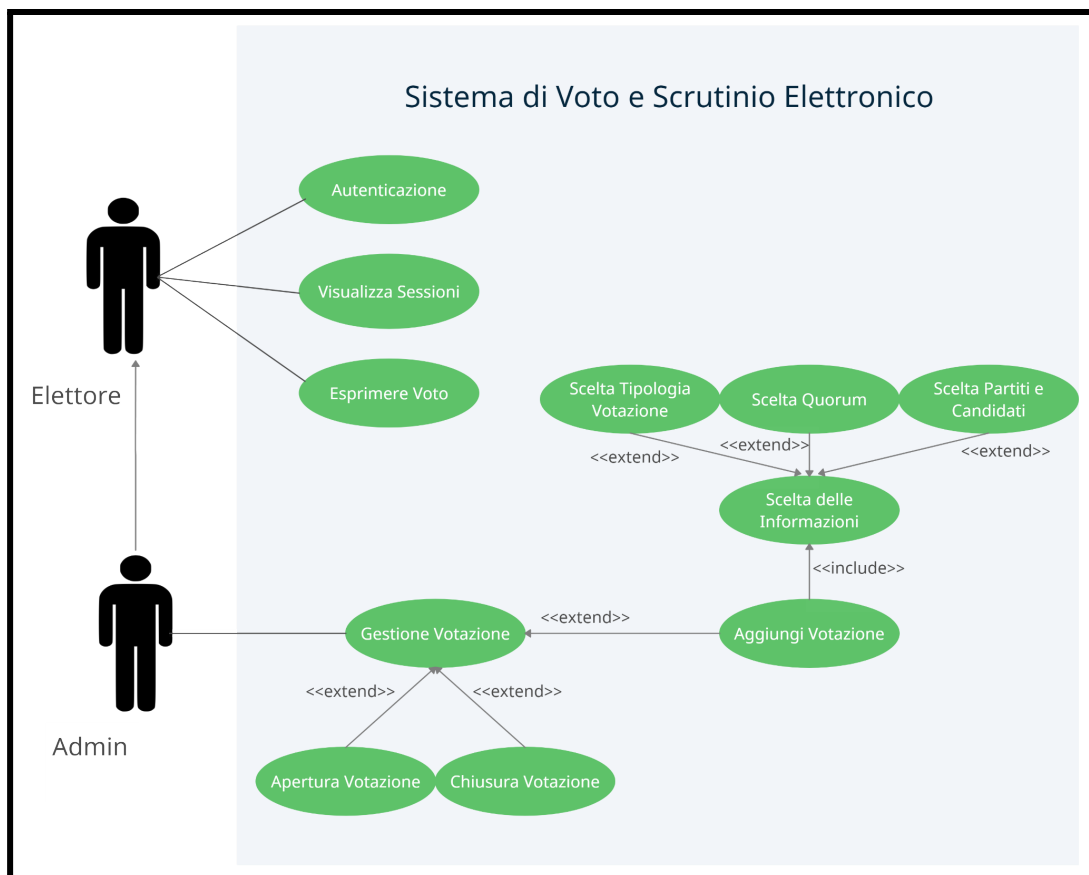
3.3.3 Requisiti di Sicurezza

L'accesso alla piattaforma avverrà previa registrazione da parte dell'utente elettore tramite un Admin di Sistema. La persona dovrà ricordarsi le credenziali in modo tale da facilitare la procedura. La password dell'account è criptata tramite l'algoritmo "SHA-256" ovvero un insieme di funzioni HASH crittografiche progettate dalla National Security Agency degli Stati Uniti D'America e pubblicate per la prima volta nel 2001. La password criptata sarà poi ulteriormente nascosta da una stringa alfanumerica per rendere ancora più sicuro il sistema. Inoltre l'account creato sarà univoco ovvero che ad una sola email e codice fiscale sarà possibile associare una ed una sola persona onde evitare casi di omonimia.

4 Progettazione del Sistema

4.1 Diagrammi dei casi d'uso

In UML, gli Use Case Diagram sono diagrammi dedicati alla descrizione delle funzioni o servizi offerti da un sistema, così come sono percepiti e utilizzati dagli attori che interagiscono col sistema stesso. Sono impiegati soprattutto nel contesto della Use Case View di un modello, e in tal caso si possono considerare come uno strumento di rappresentazione dei requisiti funzionali di un sistema. Tuttavia, è possibile ipotizzare l'uso degli UCD in altri contesti; durante la progettazione, per esempio, potrebbero essere usati per modellare i servizi offerti da un determinato modulo o sottosistema ad altri moduli o sottosistemi. In molti modelli di sviluppo software basati su UML, la Use Case View e gli Use Case Diagram che essa contiene rappresentano la vista più importante, attorno a cui si sviluppano tutte le altre attività del ciclo di vita del software.



Descrizione degli Scenari:

Nome	Autenticazione
Scopo	Permette di poter identificare l'utente che si approccia alla piattaforma fornendo due modalità di autenticazione, a seconda del ruolo, una per l'elettore e una per l'impiegato di sistema
Attori	Elettore
Pre-Condizioni	Il nome, cognome, data di nascita non devono essere nulli, la data di nascita deve essere conforme, la password inserita deve essere corretta
Trigger	Accedere al sistema e inserire le credenziali
Descrizione Sequenza Eventi	Un utente accede al sistema e inserisce le credenziali, il sistema le verifica e, se sono corrette, l'utente potrà accedere al sistema nel ruolo a lui designato
Alternativa	L'utente inserisce credenziali non corrette, il sistema provvederà a notificarlo con un messaggio
Post-Condizioni	L'elettore registrato accede alla piattaforma e può usufruire dei servizi ad esso destinati

Nome	Esprimere voto
Scopo	Permette di esprimere il proprio voto
Attori	Elettore
Pre-Condizioni	L'elettore deve essere autenticato correttamente, deve essere maggiorenne, la votazione selezionata deve essere accessibile all'elettore
Trigger	Selezionare la votazione desiderata e selezionare esprime voto

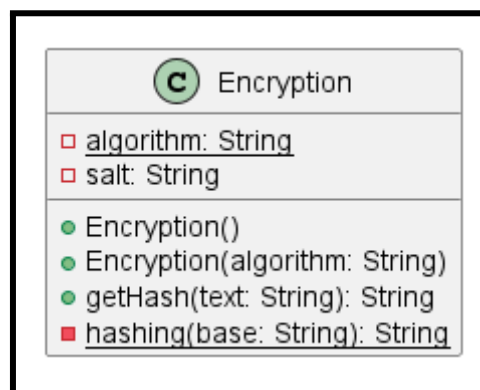
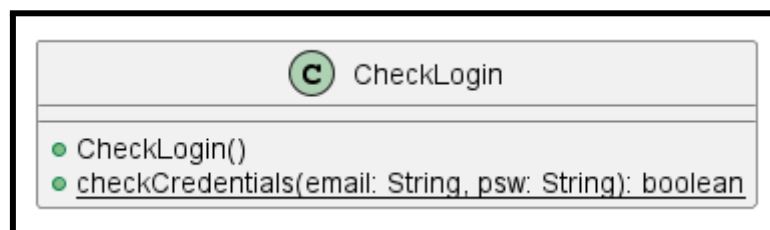
Descrizione Sequenza Eventi	L'elettore accede al sistema, seleziona una votazione per la quale esprimere il proprio voto. Il sistema verifica la possibilità di votare dell'elettore. La verifica va a buon fine e l'elettore può esprimere il proprio voto.
Alternativa	La verifica della possibilità di votare non va a buon fine, l'elettore viene notificato e reindirizzato sulla pagina del sistema a lui dedicata.
Post-Condizioni	Il voto espresso viene inserito nel sistema, il voto deve essere unico, segreto e non riconducibile all'elettore

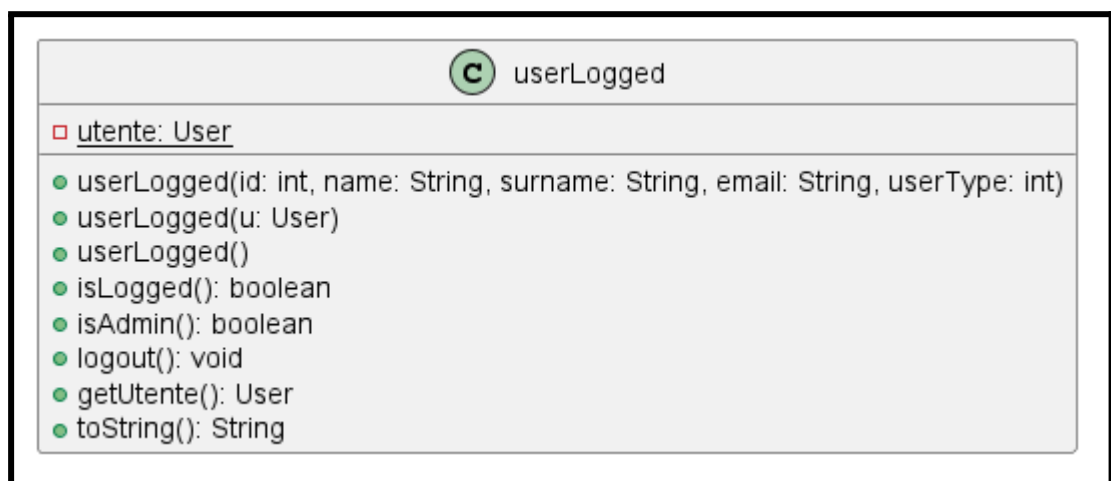
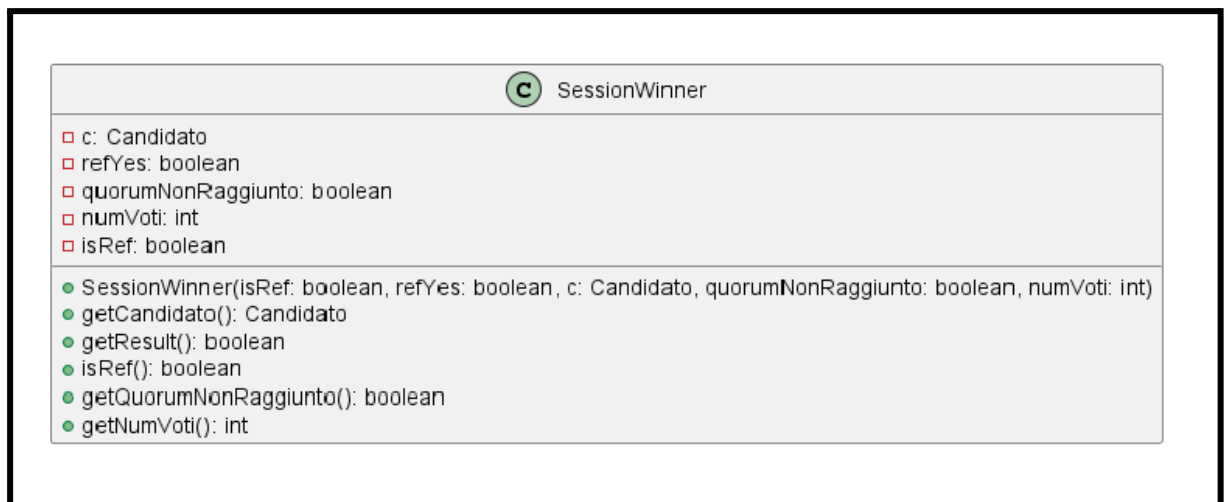
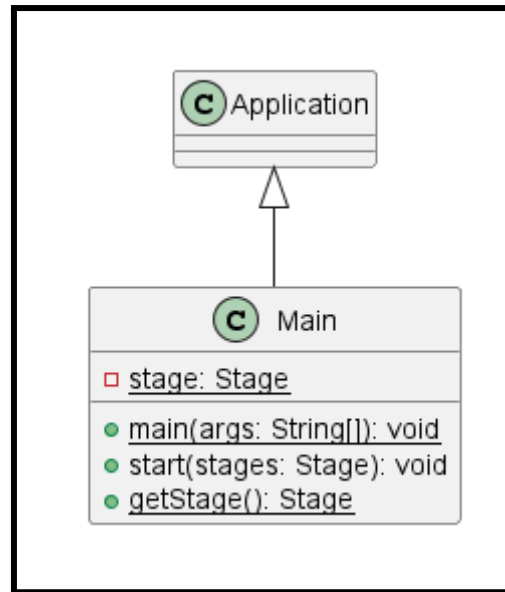
Nome	Gestione Votazione
Scopo	Permette di Gestire la Sessione di Voto
Attori	Admin
Pre-Condizioni	L'Admin di sistema deve essere correttamente autenticato
Trigger	Selezionare tutte le informazioni necessarie per iniziare una votazione, a partire dai partiti e i candidati dei partiti scelti.
Descrizione Sequenza Eventi	L'admin accede al sistema e dopo aver selezionato tutti i partiti rispetto alla votazione che si vuole intraprendere, sceglie la modalità di voto.
Alternativa	L'Admin può visualizzare la pagina della registrazione oppure quella delle sessioni attive
Post-Condizioni	Dopo aver avviato la votazione, apparirà nella finestra dove sono presenti le altre sessioni, sia quelle scadute che quelle già votate. Questa determinata finestra potrà essere visualizzata sia dagli Elettori, sia dagli Admin che devono ancora prendere parte al voto.

4.2 Diagrammi delle Classi

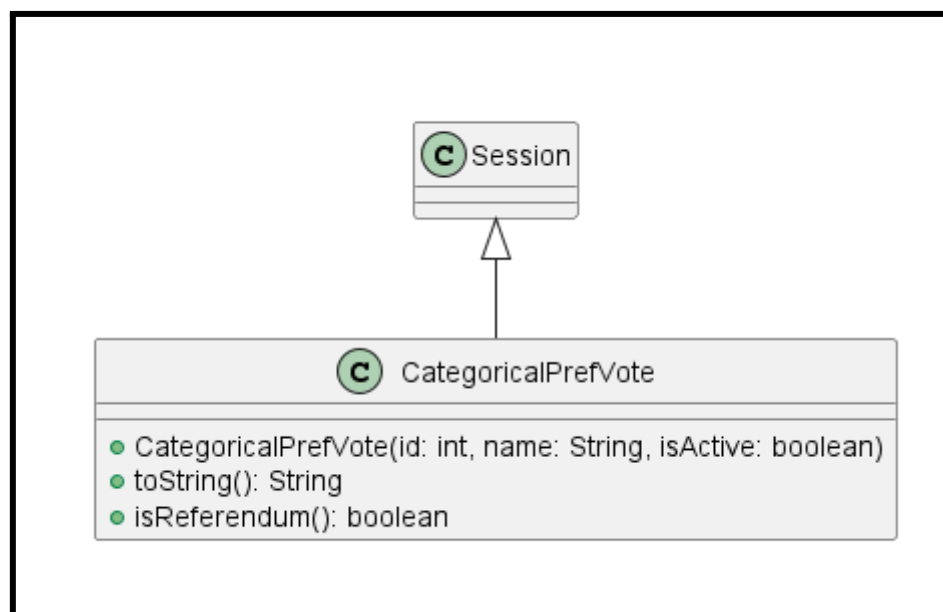
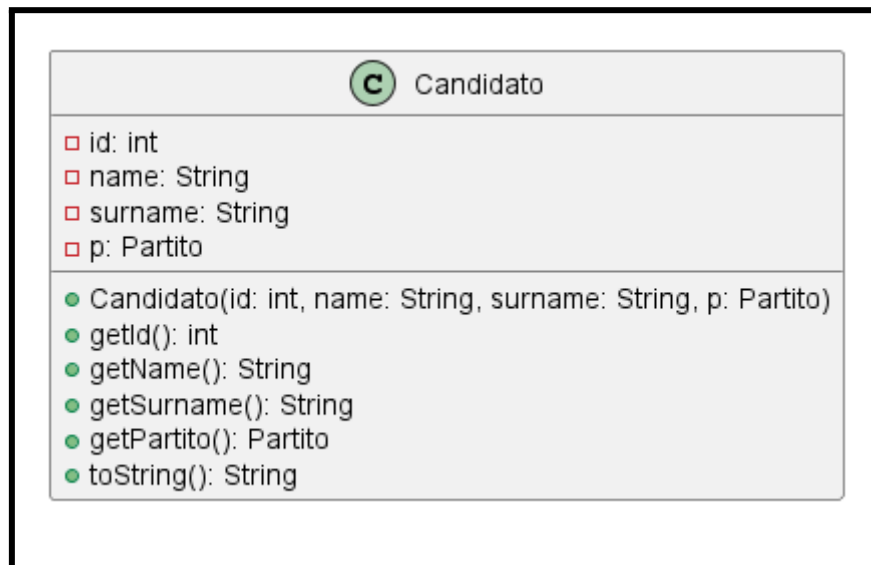
I diagrammi delle classi (class diagram) sono uno dei tipi di diagrammi che possono comparire in un modello UML. In termini generali, consentono di descrivere tipi di entità, con le loro caratteristiche e le eventuali relazioni fra questi tipi. Gli strumenti concettuali utilizzati sono il concetto di classe del paradigma object-oriented e altri correlati (per esempio la generalizzazione, che è una relazione concettuale assimilabile al meccanismo object-oriented dell'ereditarietà).

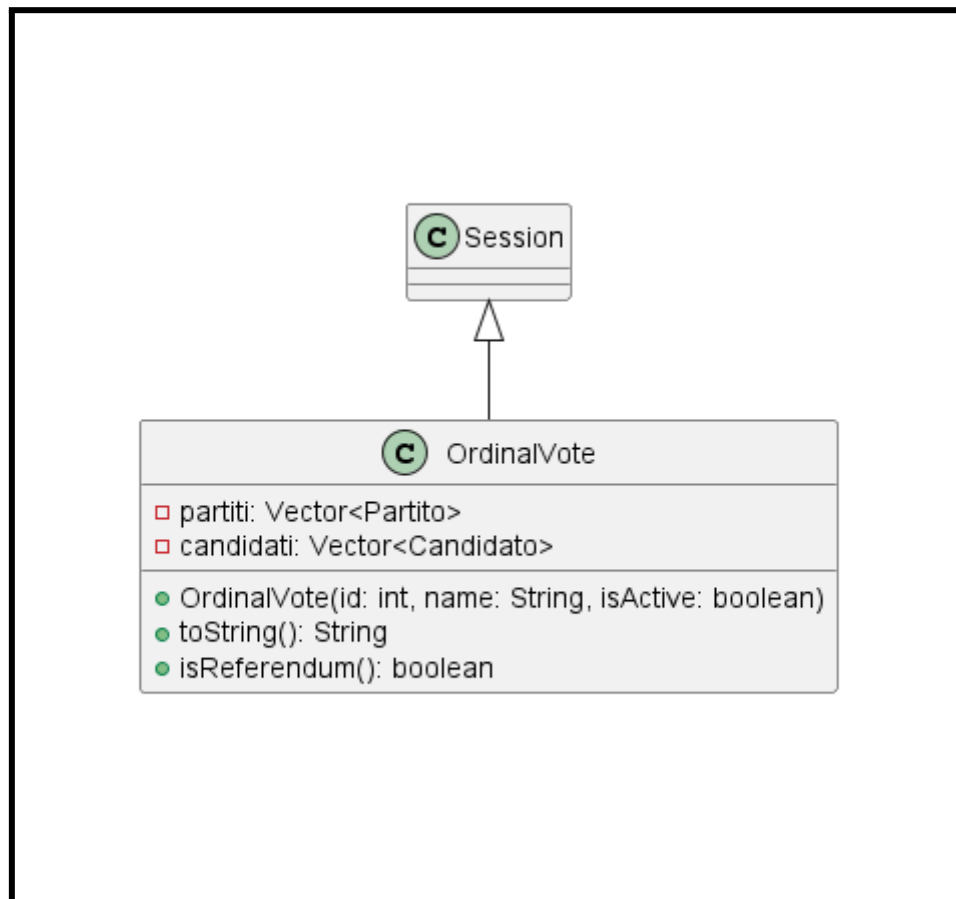
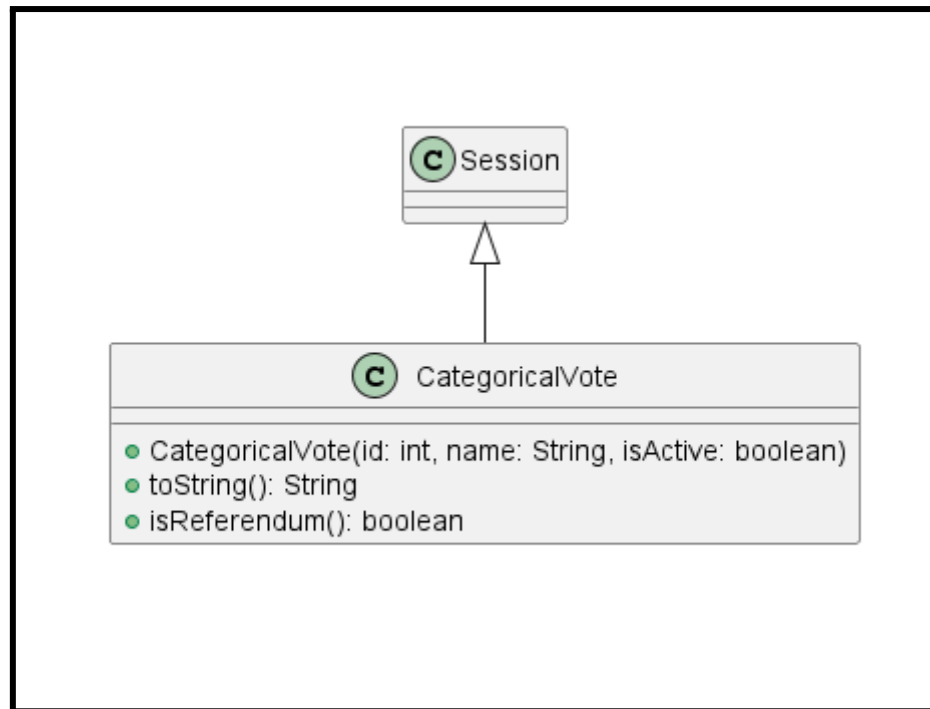
4.2.1 Application

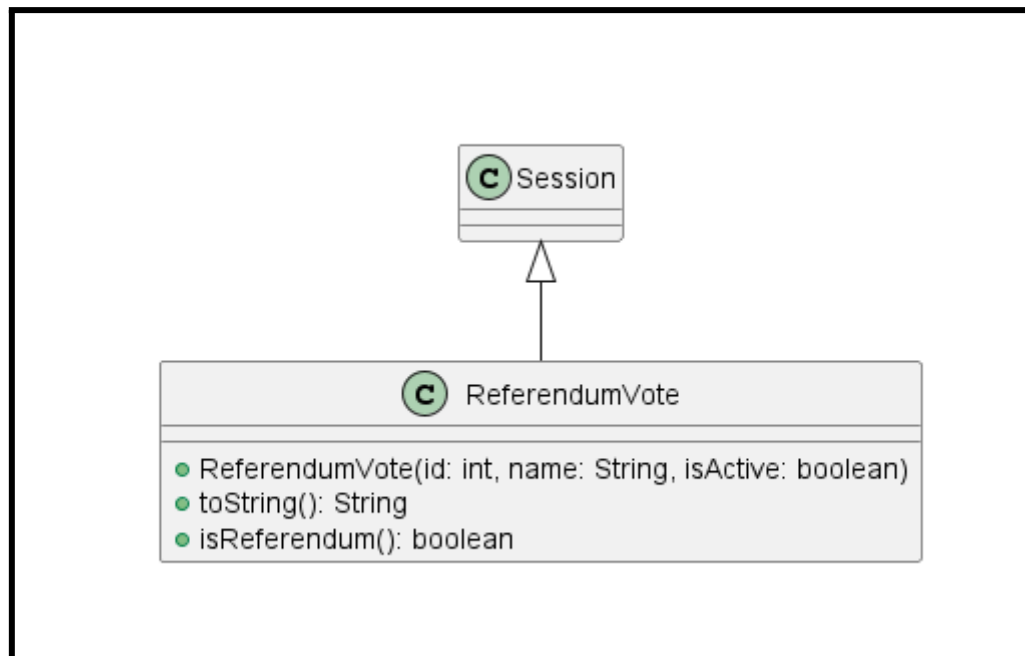
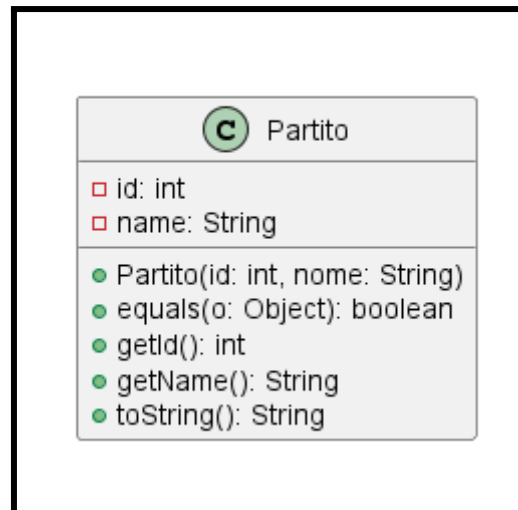


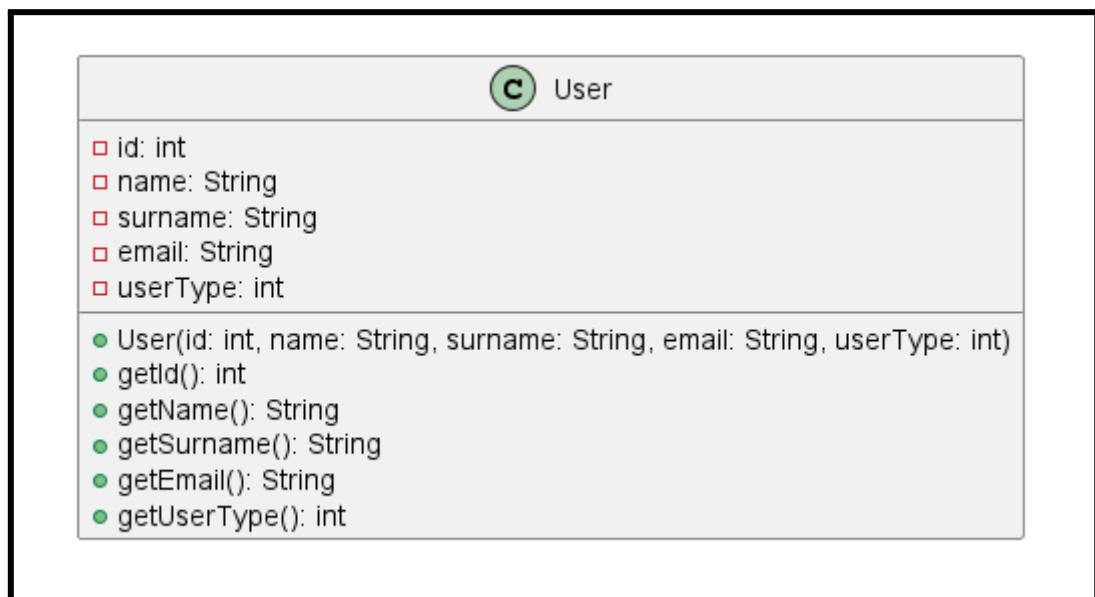
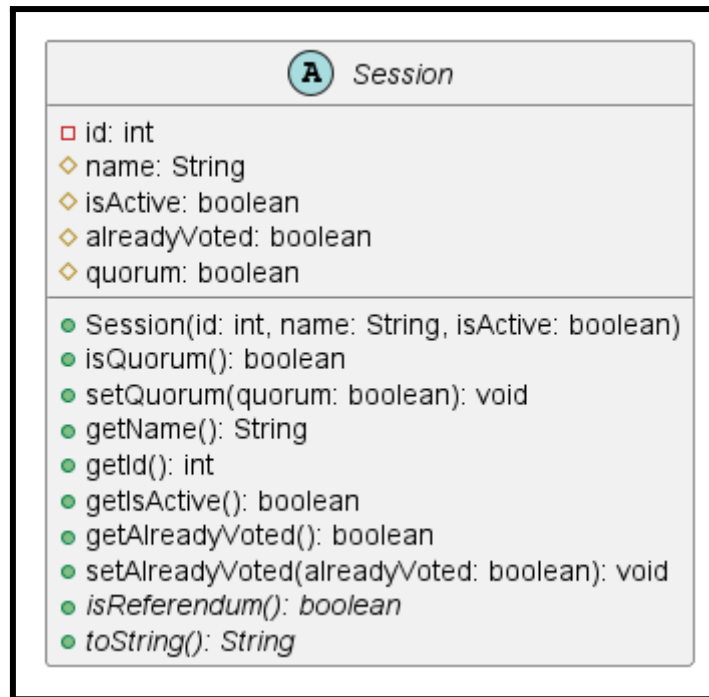


4.2.2 BaseClasses

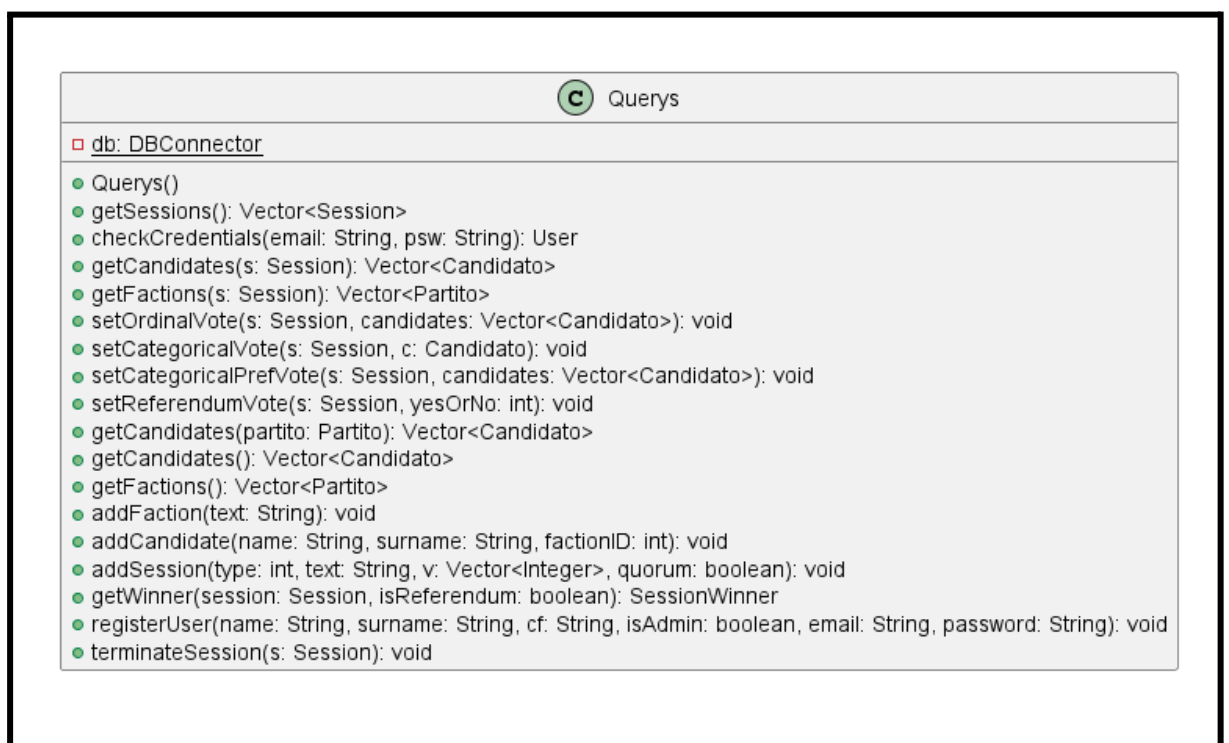
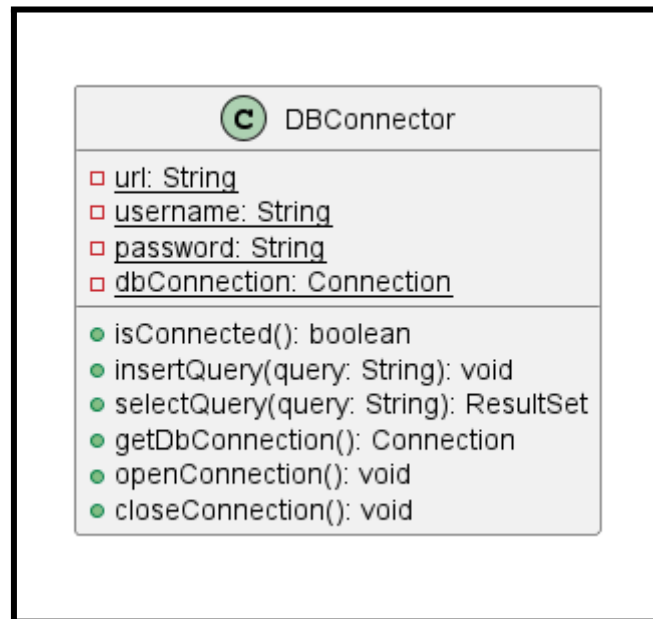




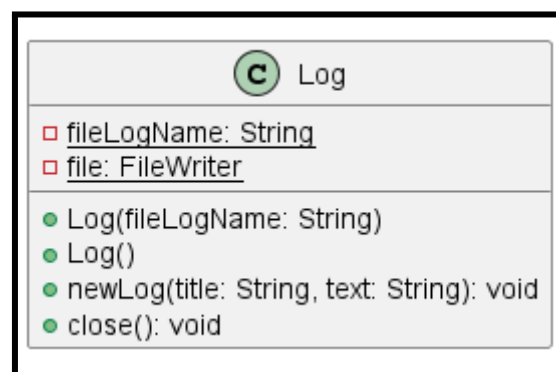
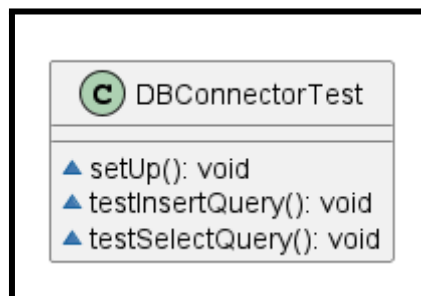
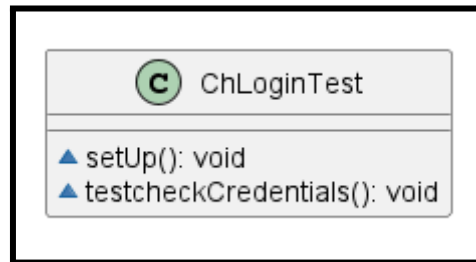




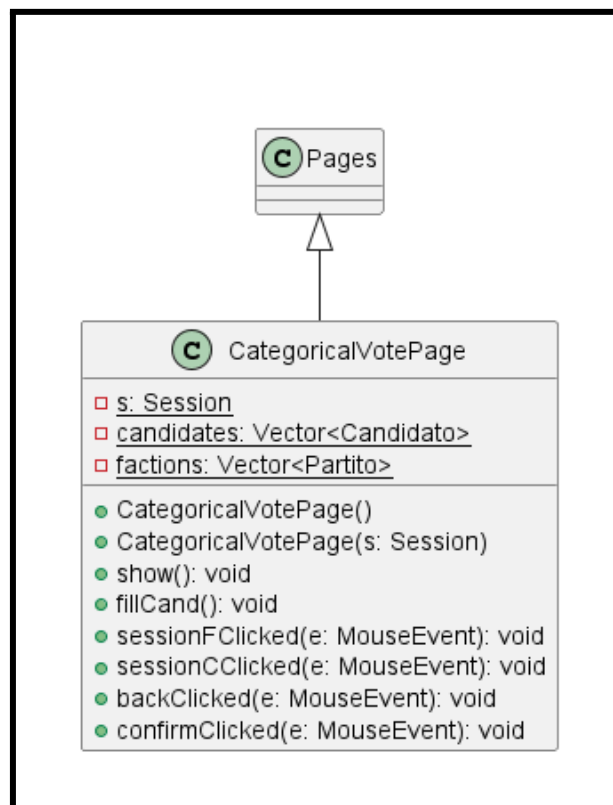
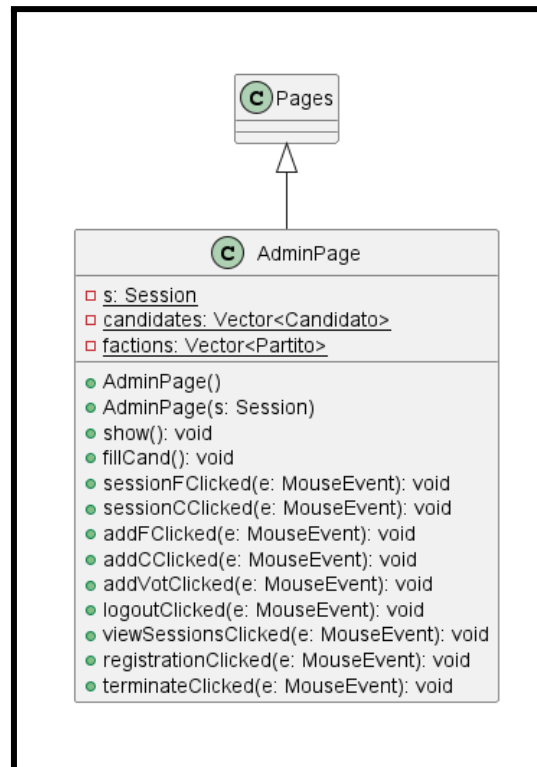
4.2.3 dbManagement

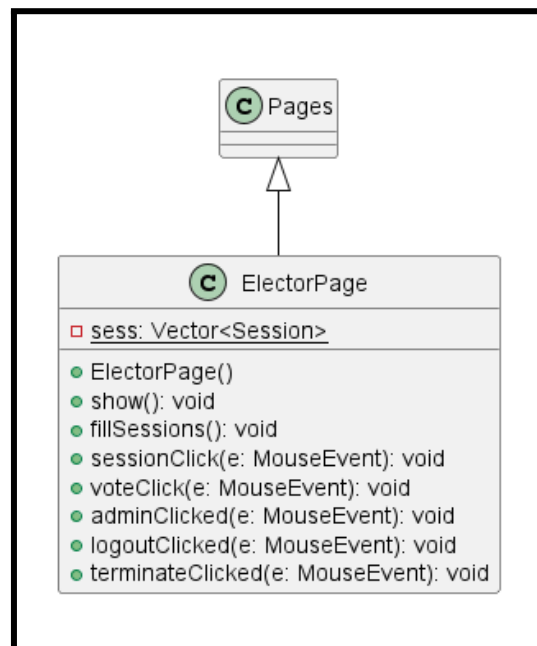
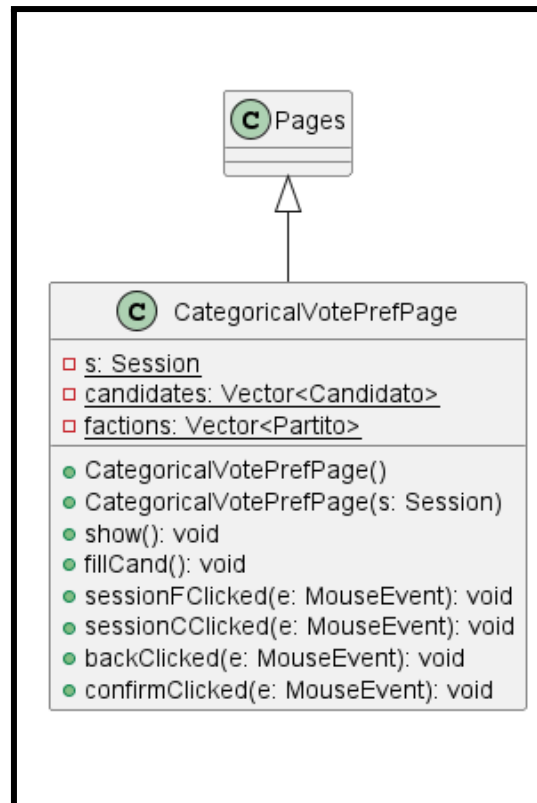


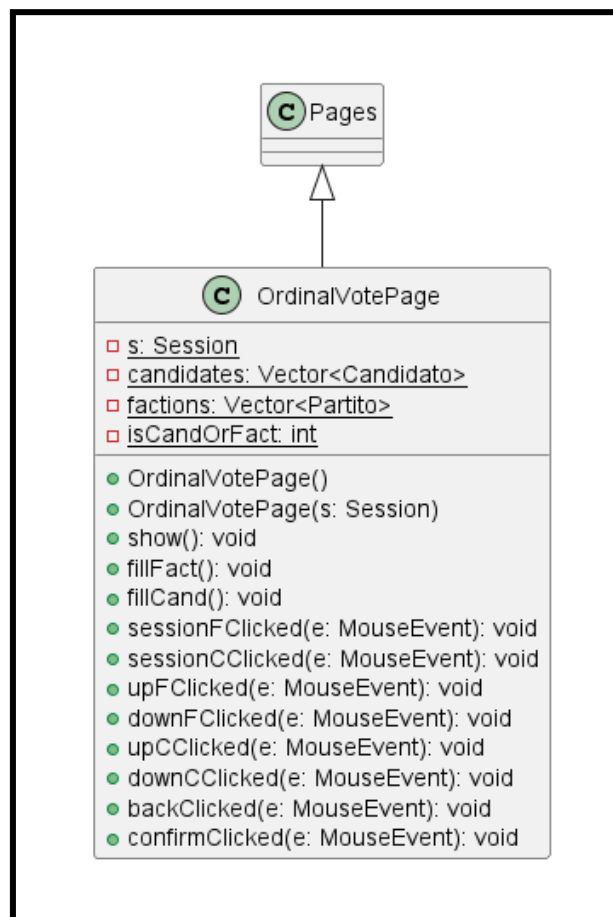
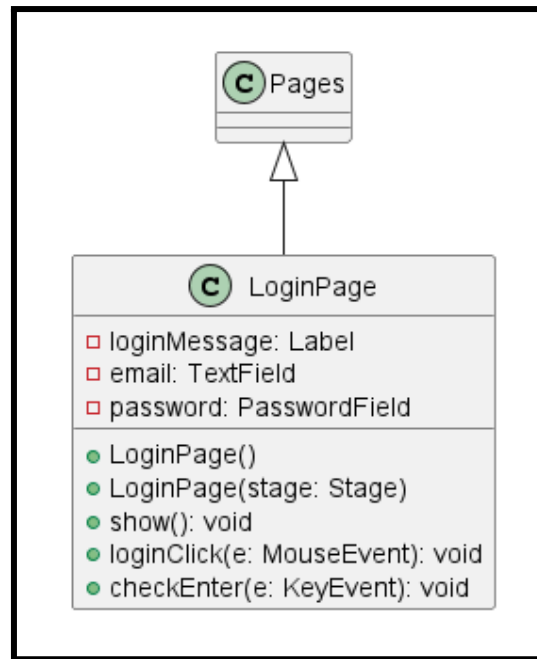
4.2.4 DebugTool

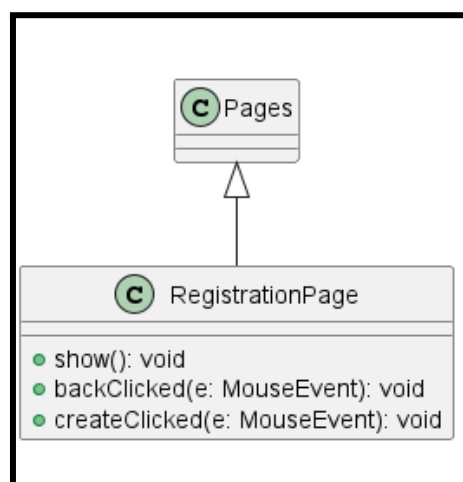
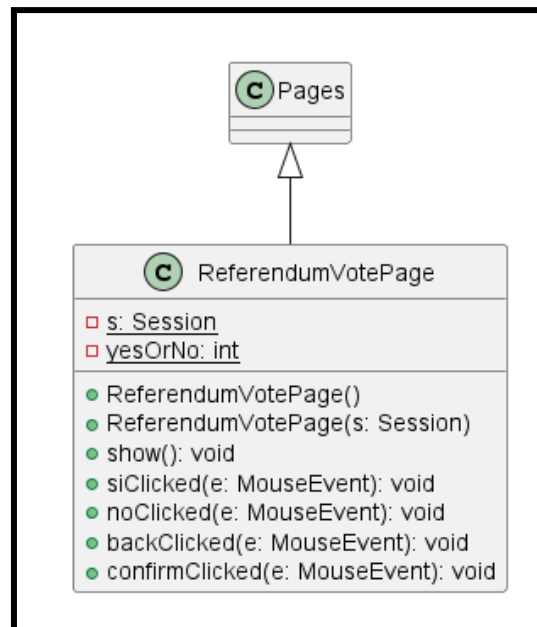
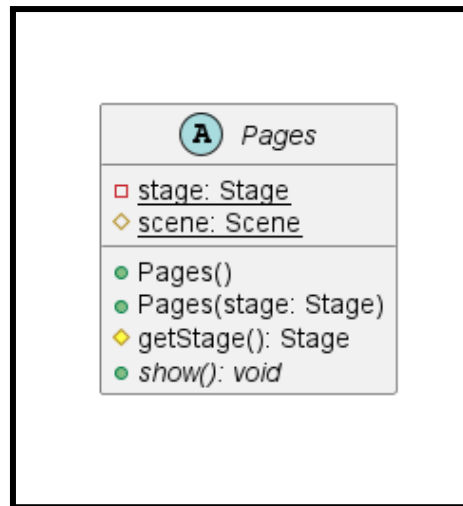


4.2.5 Pages





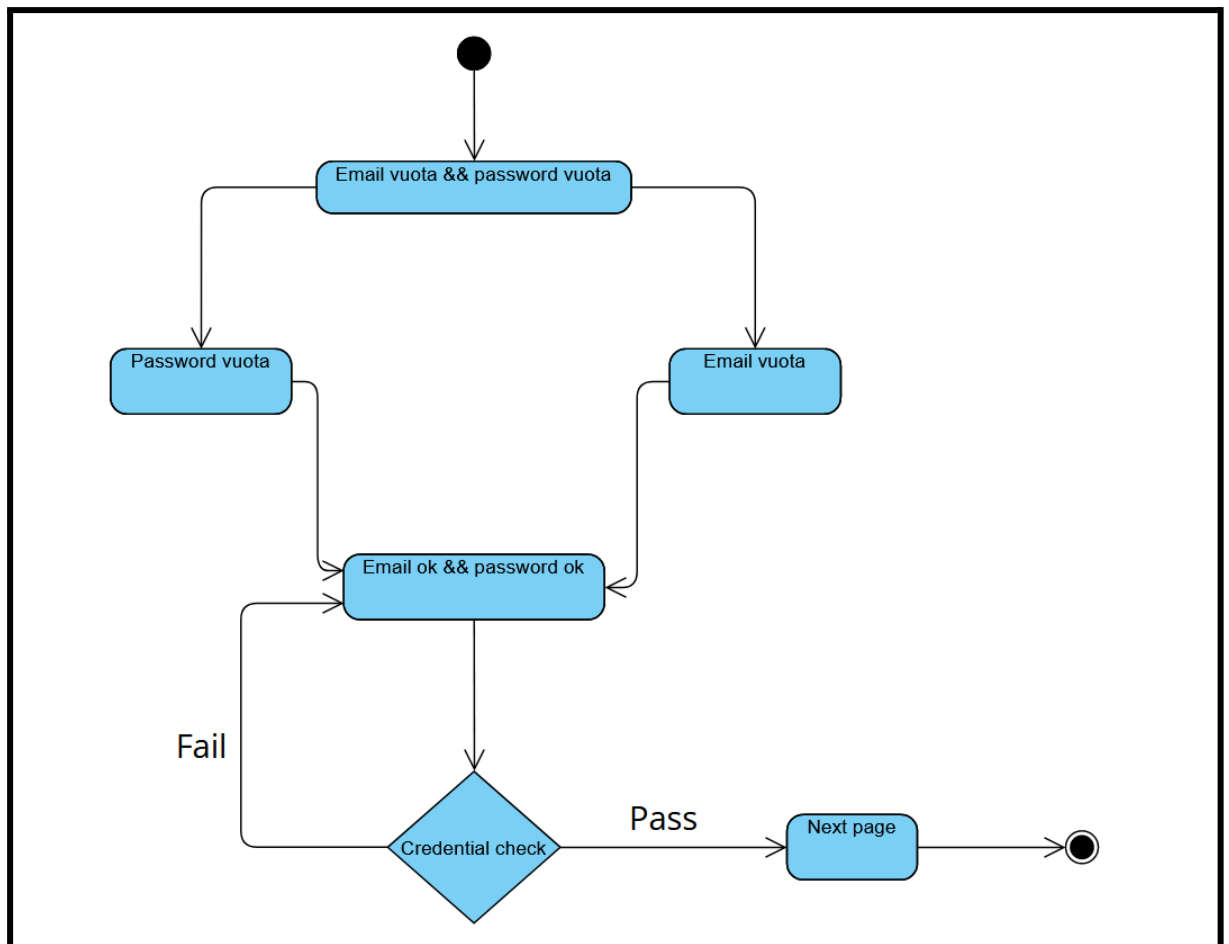




4.3 Macchine di Stato

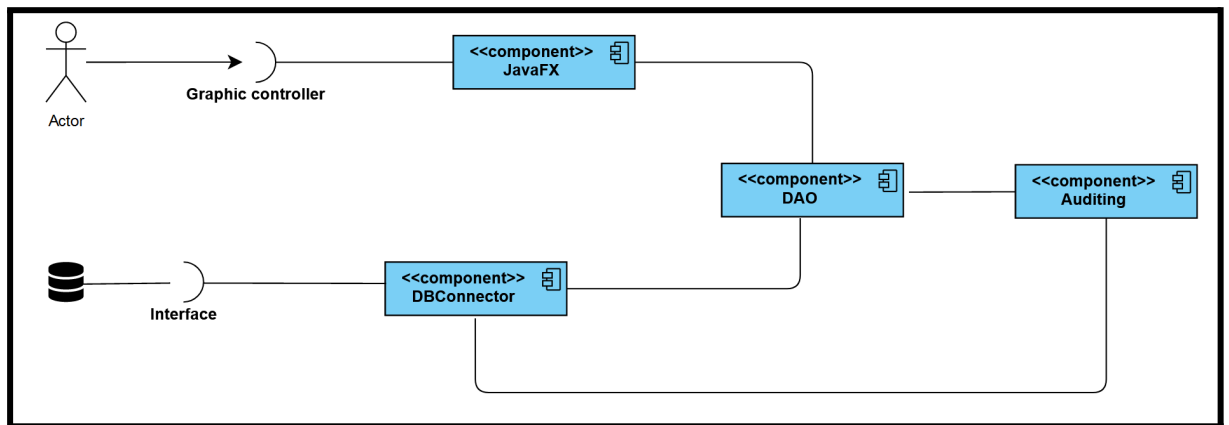
Una macchina a stati finiti (FSM, dall'inglese finite state machine) è un modello matematico di calcolo: è un tipo di automa che permette di descrivere con precisione e in maniera formale il comportamento di molti sistemi. Grazie alla sua semplicità e chiarezza questo modello è molto diffuso nell'ingegneria e nelle scienze, soprattutto nel campo dell'informatica e della ricerca operativa.

Un automa a stati finiti può essere utilizzato sia per modellare un sistema esistente che per modellare un nuovo sistema formale in grado di risolvere alcuni problemi esistenti. A quest'ultima categoria appartengono i cosiddetti riconoscitori di linguaggi e i traduttori. L'usuale rappresentazione grafica di un automa a stati finiti è il grafo orientato.



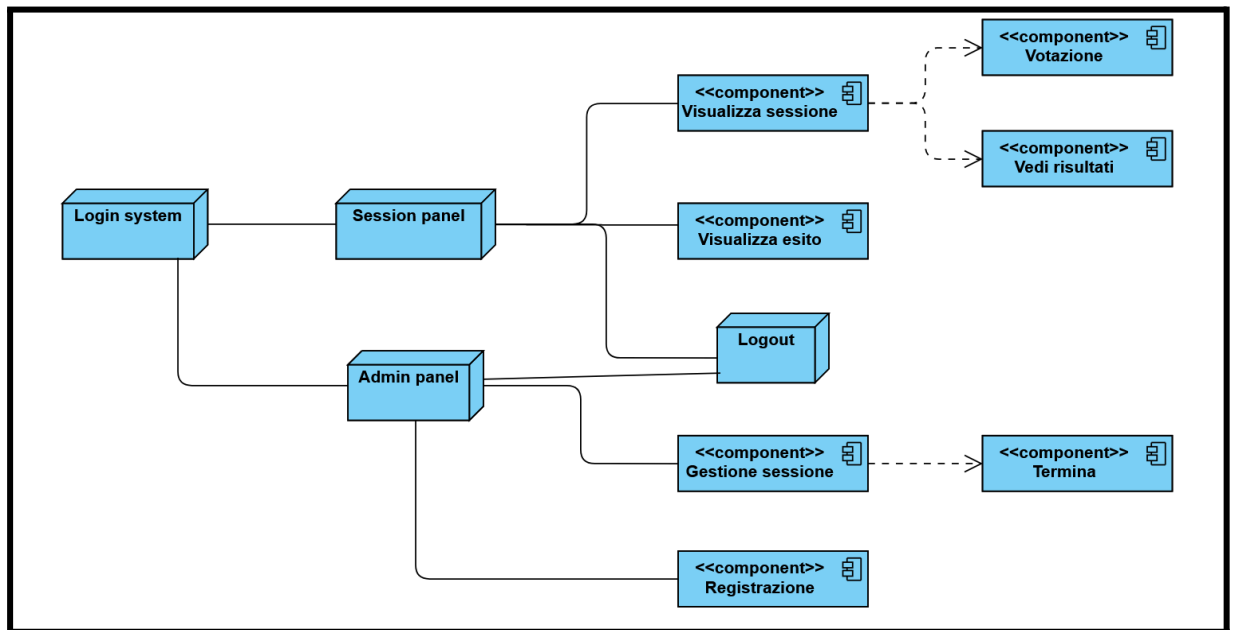
4.4 Diagramma delle Componenti

Nella terminologia del linguaggio di modellazione UML, un diagramma dei componenti (in inglese Component diagram) è un diagramma che ha lo scopo di rappresentare la struttura interna del sistema software modellato in termini dei suoi componenti principali e delle relazioni fra di essi. Per componente si intende una unità software dotata di una precisa identità, nonché responsabilità e interfacce ben definite.



4.5 Diagramma di Deployment

Il Deployment Diagram ("diagramma di dispiegamento") è un diagramma di tipo statico previsto dal linguaggio di modellazione object-oriented UML per descrivere un sistema in termini di risorse hardware, dette nodi, e di relazioni fra di esse. Spesso si utilizza un diagramma che mostra come le componenti software siano distribuite rispetto alle risorse hardware disponibili sul sistema; questo diagramma è costruito unendo il Component Diagram e il Deployment Diagram.



4.6 Vincoli OCL

OCL è un linguaggio formale di semplice uso e non ambiguo che serve per descrivere testualmente i vincoli. Non è un linguaggio di programmazione, per cui non causa effetti collaterali, ed è basato su espressioni. Java Modelling Language (JML) è un linguaggio di specifica che permette di definire astrazioni procedurali su un modello di programmazione per contratto, effettuando dei controlli sui parametri d'ingresso di un metodo e sul suo valore di ritorno. Ed è stato usato in questo progetto per mappare i vincoli OCL definiti. Di seguito riportiamo alcuni esempi.

```

1 package application;
2
3 import java.security.MessageDigest;
4
5 public class Encryption {
6     private /*@ not_null */ static String algorithm;
7     private final String salt = "fu34FInj5Dn";
8
9
10
11     public Encryption(){
12         this.algorithm = "SHA-256";
13     }
14     public Encryption(String algorithm){
15         this.algorithm = algorithm;
16     }
17
18
19
20     public String getHash(String text){
21         return hashing(salt + text);
22     }
23
24
25
26     /*@ requires base != null && !base.equals("")
27      * requires algorithm != null && !algorithm.equals("")
28      * ensures !old(base).equals(/result)
29      */
30     @*/
31     private static String hashing(final String base) {
32         try{
33             final MessageDigest digest = MessageDigest.getInstance(algorithm);
34             final byte[] hash = digest.digest(base.getBytes("UTF-8"));
35             final StringBuilder hexString = new StringBuilder();
36             for (int i = 0; i < hash.length; i++) {
37                 final String hex = Integer.toHexString(0xff & hash[i]);
38                 if(hex.length() == 1)
39                     hexString.append('0');
40                 hexString.append(hex);
41             }
42             return hexString.toString();
43         } catch(Exception ex){
44             throw new RuntimeException(ex);
45         }
46     }
47 }
48

```

```

/*@
 * requires u != null
 */
public userLogged(User u) {
    utente = u;
}

```

```

/*@ requires file != null
 * requires title != null
 *
 *
 */
public void newLog(String title, String text){
    // scrittura log su file

    try {
        BufferedWriter bw = new BufferedWriter(file);
        bw.write(title + " - " + text + "\n");
        bw.flush();
        bw.close();
    }
    catch(IOException e) {
        e.printStackTrace();
    }
}

```

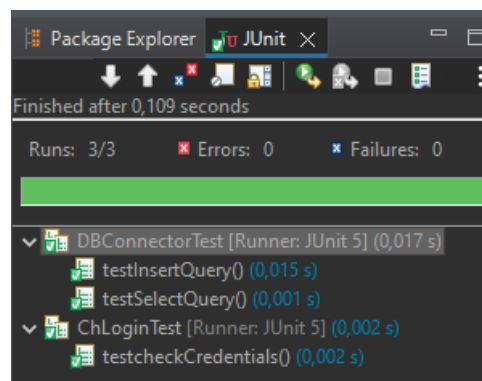
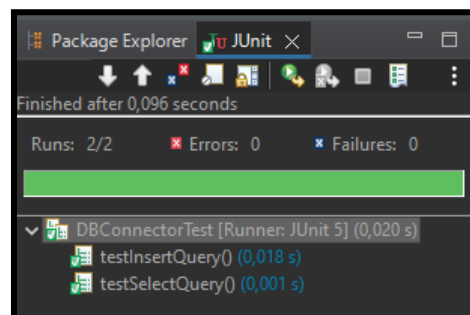
4.7 JUnit

Un test di unità è una semplice prova fatta per verificare che un requisito sia soddisfatto dal codice che abbiamo scritto.

Sul nostro progetto è stata testata la classe `CheckLogin` nella classe `ChLoginTest` presente nel package `DebugTool`.

Inoltre sono presentati anche gli screenshot del tool JUnit di Eclipse che testano il codice Sorgente.

```
1 package debugTool;
2
3 import static org.junit.jupiter.api.Assertions.*;
4
5
6
7
8
9
10
11 class ChLoginTest {
12
13     @BeforeEach
14     void setUp() throws Exception {
15     }
16
17     @Test
18     void testcheckCredentials() {
19         assertThrows(NullPointerException.class, () -> CheckLogin.checkCredentials(null, "abc"));
20         assertThrows(NullPointerException.class, () -> CheckLogin.checkCredentials("abc", null));
21         assertThrows(NullPointerException.class, () -> CheckLogin.checkCredentials(null, null));
22
23         assertFalse(CheckLogin.checkCredentials("", ""));
24
25         assertFalse(CheckLogin.checkCredentials("x", "x"));
26     }
27
28 }
29
```



4.8 Pattern Utilizzati

- Singleton: Assicura che la classe abbia una sola istanza e fornisce un metodo di accesso globale a tale istanza
- Observer: Definisce una dipendenza 1:N tra oggetti in modo che se uno degli oggetti cambia stato gli altri siano aggiornati automaticamente
- DAO: Data Access Object Pattern pattern viene utilizzato per separare i dati di basso livello che accedono all'API o alle operazioni dai servizi di alto livello. I vantaggi nell'utilizzare questo pattern sono molteplici, tra cui: il vantaggio di utilizzare oggetti di accesso ai dati è la separazione relativamente semplice e rigorosa tra due parti importanti di un'applicazione che possono ma non dovrebbero sapere nulla l'una dell'altra e che ci si può aspettare che si evolvano frequentemente e in modo indipendente. Se si dovesse avere la necessità di cambiare il meccanismo di persistenza sottostante, bisogna solo cambiare il livello DAO e non tutti i punti della logica di dominio da cui viene utilizzato il livello DAO.
- MVC: Model-View-Controller (MVC) è un pattern utilizzato in programmazione per dividere il codice in blocchi dalle funzionalità ben distinte. In particolare: - Model: contiene i metodi di accesso ai dati. - View: si occupa di visualizzare i dati all'utente e gestisce l'interazione fra quest'ultimo e l'infrastruttura sottostante. - Controller: riceve i comandi dell'utente attraverso il View e reagisce eseguendo delle operazioni che possono interessare il Model e che portano generalmente ad un cambiamento di stato del View.

Singleton è stato utilizzato nella classe User Logged.

Observer è stato utilizzato nelle classi Pages.

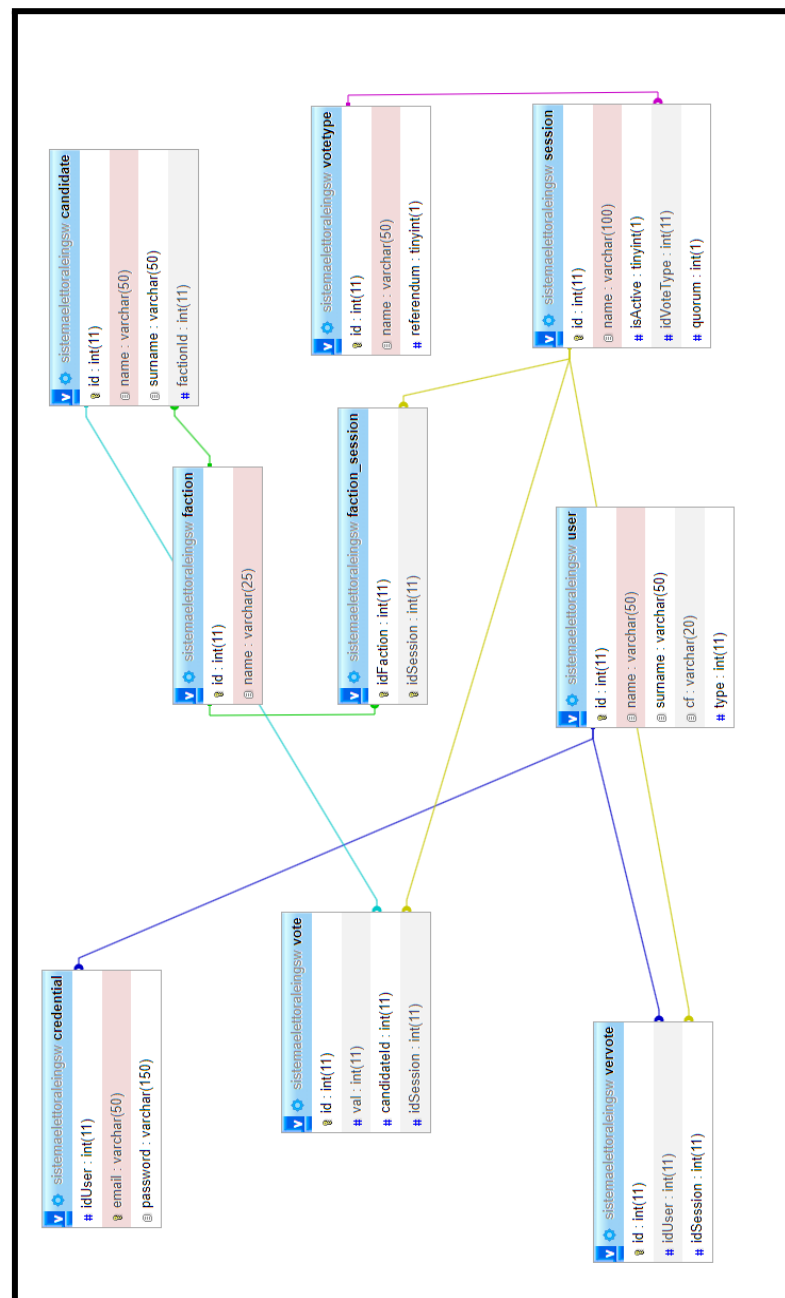
DAO è stato utilizzato nella classe Queries.

MVC è stato utilizzato nella classe Sessione - Utente.

Federico Millefanti
Lorenzo Borrelli

922812
923420

38



5 Appendice e Note per l'Installazione

Con il pacchetto zippato vengono forniti tutti i codici eseguibili, il dump del database e la documentazione.

I due software utilizzati per il corretto utilizzo dell'applicazione sono Eclipse e XAMPP.

Per poter utilizzare il programma è necessario scaricare l'ultima versione di eclipse disponibile ed installare tutti i vari pacchetti utilizzati. Oltre all'utilizzo del linguaggio di programmazione JAVA è stato indispensabile l'utilizzo di JavaFX con i seguenti pacchetti: wt, base, controls, fxml, graphics, media, swing, web.

Dopo aver correttamente predisposto l'ambiente di lavoro è indispensabile scaricare il dump del database ed inserirlo nella piattaforma di XAMPP, che usufruisce del sistema di scrittura di MySQL. E' possibile utilizzare il programma da qualsiasi ente e per qualsiasi scopo purché segua il regolamento sopra descritto.

I creatori di questo progetto vi invitano a controllare la pagina GitHub per nuove modifiche o aggiornamenti.

Lorenzo Borrelli
Federico Millefanti