



## **RASD – Sistema di votazione elettronica – Vers. 1.3**

### **Indice:**

#### **Descrizione del problema**

- Assignments
- Analisi e specifica dei requisiti:
  - Requisiti funzionali:
  - Requisiti non funzionali:
- Glossario

#### **Progettazione del Sistema**

- Diagramma dei casi d'uso
- Descrizione degli scenari
- Diagramma delle classi
- Diagrammi di sequenza
- Diagrammi delle attività
- Macchine di stato
- Diagramma dei componenti

#### **Implementazione del sistema**

- Diagramma delle classi
- Discussione dei Design Pattern utilizzati
- Gestione dei dati persistenti
- Descrizione della GUI
- Diagramma di deployment
- Specifica e verifica dei vincoli
- Descrizione del testing
- Nota per l'installazione e l'utilizzo

#### **Appendice**

#### **Note sulle versioni rilasciate**



## Descrizione del problema:

Il progetto riguarda la realizzazione di una piattaforma di voto sicura, in cui poter creare e gestire diversi tipi di votazione ed ovviamente, votare. Viene inoltre richiesto che la piattaforma possa integrarsi al meglio con le votazioni svolte in presenza.

- **Assignments:**

Ho riutilizzato alcune parti del lavoro commissionato in precedenza dalle Docenti. In particolare, seppure alcune modifiche, rimangono validi i diagrammi dei casi d'uso, i diagrammi di sequenza, le users stories e alcuni requisiti.

- **Analisi e specifica dei requisiti:**

La piattaforma realizzata sarà disponibile solamente ad *Amministratori* e a *Elettori* già inseriti nel sistema. Si presuppone quindi una collaborazione degli uffici della Pubblica Amministrazione nel registrare gli aventi diritto al voto e comunicare ad essi il loro codice fiscale e la loro password. Gli *amministratori* inoltre condividono le stesse credenziali, se hanno diritto di voto, per poter votare all'interno della piattaforma.

L'iscrizione da parte degli *elettori* avviene dunque in automatico grazie alla PA.

Come nelle votazioni in presenza, l'*Elettore* ha la possibilità di lasciare la scheda bianca.

Il voto mediante la piattaforma avviene dunque previa autenticazione dell'*Elettore*, il quale si troverà davanti ad una schermata che raccoglie tutte le votazioni attive, e selezionandone una, potrà avviarsi al voto.

Un *Amministratore* del sistema può creare, modificare o eliminare votazioni a sua discrezione. Può inoltre visualizzare i risultati una volta raggiunta la data di fine di una determinata votazione. Oltre a visualizzarne i risultati può immettere nel sistema i voti registrati in presenza e poi ottenere il risultato. Ha inoltre a disposizione due file di log, uno per l'utenza *elettori* e uno per l'utenza *amministratori*, per analizzare le azioni svolte nel sistema.

Per creare una votazione un *Amministratore* dovrà specificare:

- il nome/descrizione della votazione;
- la data di termine;
- la tipologia;
- eventuale quorum/maggioranza assoluta.



Le tipologie di voto disponibili sono le seguenti:

Tipologia	Descrizione
Voto Ordinale	L' <i>Elettore</i> troverà una lista di Candidati e dovrà ordinarli a seconda della sua preferenza. I voti vengono distribuiti in maniera decrescente: Con $n$ candidati si partirà da $n$ preferenze per il primo classificato, $n-1$ al secondo, ..., fino all'ultimo con solo 1 preferenza
Voto Categorico	L' <i>Elettore</i> troverà una lista di Candidati e dovrà selezionarne uno solo per esprimere la sua preferenza
Voto Categorico Preferenziale	L' <i>Elettore</i> troverà una lista di Gruppi/Partito e dovrà inserire una preferenza per uno di essi e avrà poi la possibilità di ordinare i candidati presentati da tale partito per esprimere le proprie preferenze; in caso contrario le preferenze saranno distribuite all'interno del partito
Referendum	L' <i>Elettore</i> dovrà rispondere al quesito referendario esprimendo consenso, dissenso o scheda bianca

Le tipologie di vittoria disponibili sono le seguenti:

Tipologia	Descrizione
Maggioranza	Il vincitore sarà il Candidato con il maggior numero di voti ottenuti
Maggioranza assoluta	Il vincitore sarà il Candidato con il maggior numero di voti ottenuti se e solo se ha ottenuto almeno il 50% + 1 voti sugli aventi diritto
Referendum con Quorum	Vincerà la scelta che ha avuto il maggior numero di voti se e solo se almeno il 50% + 1 degli aventi diritto di voto ha partecipato alla votazione
Referendum semplice	Vincerà la scelta che ha avuto il maggior numero di voti

Una volta che una votazione avrà raggiunto la sua scadenza e un *Amministratore* avrà calcolato i risultati, rimarrà disponibile nel sistema, verrà inoltre fornito all'*Amministratore* un file pdf dove consultare i risultati. La PA, attraverso i propri canali di comunicazione, comunicherà i risultati all'elettorato.



## Requisiti funzionali:

### ○ Requisiti utente:

1	Al suo avvio il sistema deve garantire la possibilità di scegliere il tipo di utente, così da poter impostare la votazione
2	Il sistema una volta avviato deve far scegliere all'utente <i>Amministratore</i> se inserire, modificare o ottenere i risultati di una votazione
3	Il sistema deve far scegliere la tipologia di votazione all'utente <i>Amministratore</i>
4	Il sistema deve far scegliere la condizione di vittoria all'utente <i>Amministratore</i>
5	Il sistema deve far impostare all'utente <i>Amministratore</i> la chiusura delle votazioni
6	Il sistema deve far esprimere il voto all'utente, indipendentemente dalla tipologia di votazione
7	Il sistema deve poter determinare un vincitore e fornire indicazioni sul risultato, indipendentemente dalla condizione di vittoria
8	Il sistema deve supportare le seguenti modalità di voto: voto ordinale, voto categorico, voto categorico preferenziale e referendum
9	Il sistema deve supportare le seguenti condizioni di vittoria: maggioranza, maggioranza assoluta, referendum senza quorum, referendum con quorum

### ○ User stories:

1	Come cittadino e <i>Amministratore</i> voglio poter scegliere in che ruolo entrare nel sistema, così potrò votare e/o impostare il sistema
6	Come votante voglio poter esprimere il mio voto mediante il sistema, così posso votare ovunque mi trovi
7	Come <i>Amministratore</i> voglio sapere i risultati delle votazioni elettroniche, così saprò chi ha vinto

### ○ Requisiti di sistema:

1.1	All'avvio del software, l'interfaccia dovrà permettere la scelta tra l'utente <i>Amministratore</i> e l'utente votante
1.2	In ognuno dei due casi sarà necessario autenticarsi nel sistema con le proprie credenziali
1.3	Una volta entrati, indipendentemente dal tipo di utente, l'applicazione sarà pronta all'utilizzo
2.1	L'utente <i>Amministratore</i> dovrà scegliere se inserire, modificare o ottenere i risultati di una votazione
3-4-5.1	Una volta entrati nella schermata di inserimento l' <i>Amministratore</i> potrà scegliere condizioni di vittoria, tipologia di votazione, scadenza e descrizione della votazione
3-4-5.2	Se la votazione scelta è diversa da Referendum l' <i>Amministratore</i> procederà all'inserimento dei Candidati
3-4-5.3	Prima di inserire la votazione il software richiederà conferma all' <i>Amministratore</i>



<b>6.1</b>	Una volta che l' <i>Elettore</i> si è autenticato potrà scegliere per quale votazione esprimere il voto (se ha diritto di voto)
<b>6.2.a</b>	Se ha scelto un referendum si troverà innanzi ad una schermata con la descrizione del referendum scelto e la possibilità di esprimere una preferenza tra [si] – [no] – [bianca]
<b>6.2.b</b>	Se ha scelto un altro tipo di votazione si troverà innanzi ad una schermata con la descrizione della votazione e la possibilità di esprimere la propria preferenza in base alla votazione scelta in precedenza
<b>6.3</b>	Prima di inserire la preferenza il software richiederà conferma all' <i>Elettore</i>
<b>7.1</b>	L'interfaccia mostrerà all' <i>Amministratore</i> le votazioni per le quali è possibile ottenere i risultati
<b>7.2.a</b>	Una volta scelta la votazione è possibile vedere i risultati direttamente
<b>7.2.b</b>	Una volta scelta la votazione è possibile aggiungere i voti delle votazioni svoltesi in presenza prima di visualizzare i risultati
<b>7.3</b>	Prima di ottenere i risultati il software richiederà conferma all' <i>Amministratore</i>

### Requisiti non funzionali:

#### ○ Requisiti di prodotto:

##### ▪ Requisiti di usabilità:

Una volta scelto chi votare il sistema richiede la conferma del voto

Il sistema deve essere utilizzabile su tutti i dispositivi

Il sistema deve essere conforme a principi di usabilità e accessibilità

##### ▪ Requisiti di sicurezza:

Il sistema deve mantenere l'anonimato dell'utente

Il software deve includere un sistema di auditing

##### ▪ Requisiti di affidabilità:

Il sistema deve permettere la verifica del voto prima di inviarlo

Il sistema deve riconoscere voto valido o scheda bianca

#### ○ Requisiti organizzativi:

##### ▪ Requisiti di sviluppo:

Il codice sarà Open Source

#### ○ Requisiti esterni:

##### ▪ Requisiti normativi:

Un utente votante può votare solamente una volta

Il sistema deve rispettare le linee guide AgID

Se, in caso di votazioni con maggioranza nessun candidato raggiunge la quota di voti richiesta, sarà necessario un ulteriore turno di votazioni tra i due candidati che hanno ricevuto più voti; il vincitore sarà chi avrà ricevuto più voti

Il sistema può contare i voti solo dopo la chiusura delle votazioni

##### ▪ Requisiti etici:

I dati personali devono essere protetti



Il sistema non deve influenzare il voto

Il sistema non deve fornire più informazioni del sistema di votazione in presenza

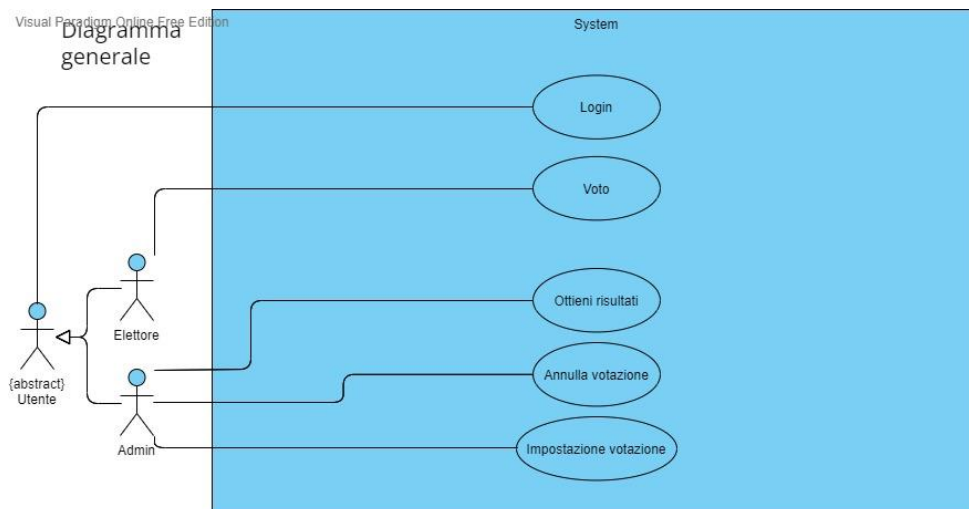
- **Glossario:**

<b>Maggioranza assoluta</b>	Tipologia di maggioranza in cui, per far sì che un candidato vinca serve il 50% + 1 degli aventi diritto al voto favorevoli;
<b>Referendum</b>	Istituto giuridico in virtù del quale il popolo viene chiamato alle urne per esprimersi su questioni istituzionali e politiche essenziali;
<b>Quorum</b>	Il numero minimo indispensabile per la validità giuridica della deliberazione o della votazione di un'assemblea, di un consiglio, in elezioni o referendum;
<b>Gruppo</b>	Collezione di persone che condividono le stesse idee, che si uniscono per promuovere e difendere un obiettivo comune, cercando di influenzare la decisione del governo;
<b>Partito</b>	Gruppo organizzato volontariamente di individui dedicati con un'ideologia politica simile;
<b>Seggio elettorale</b>	Indica sia il luogo in cui gli elettori si recano per votare, sia l'insieme delle persone preposte alla gestione delle operazioni di voto e di scrutinio;
<b>AgiD</b>	Agenzia per l'Italia digitale, che ha come obiettivo migliorare il modo in cui interagiamo con la Pubblica Amministrazione cercando di dargli uniformità e coerenza;
<b>Voto ordinale</b>	Modalità di voto in cui l'Elettore deve ordinare i candidati (o gruppi/partiti), presenti nella scheda in base alle proprie preferenze;
<b>Voto categorico</b>	Modalità di voto in cui l'Elettore inserisce una preferenza per un candidato (o gruppo/partito);
<b>Voto categorico con preferenze</b>	Modalità di voto in cui 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;
<b>Scheda bianca</b>	Esercizio della facoltà di astenersi dalla scelta di voto;
<b>Auditing</b>	Processo di valutazione pianificata, indipendente e sistematica di un sistema.
<b>Open Source</b>	Con open source (in italiano sorgente aperto), in informatica, si indica un sistema di sviluppo software decentralizzato basato sulla condivisione dei file sorgenti (source)
<b>PA</b>	Acronimo di Pubblica Amministrazione, ovvero l'insieme degli enti pubblici che concorrono all'esercizio e alle funzioni della gestione, direzione e coordinazione di uno Stato nelle materie di sua competenza.
<b>DP</b>	Acronimo di <u>Design Pattern</u>

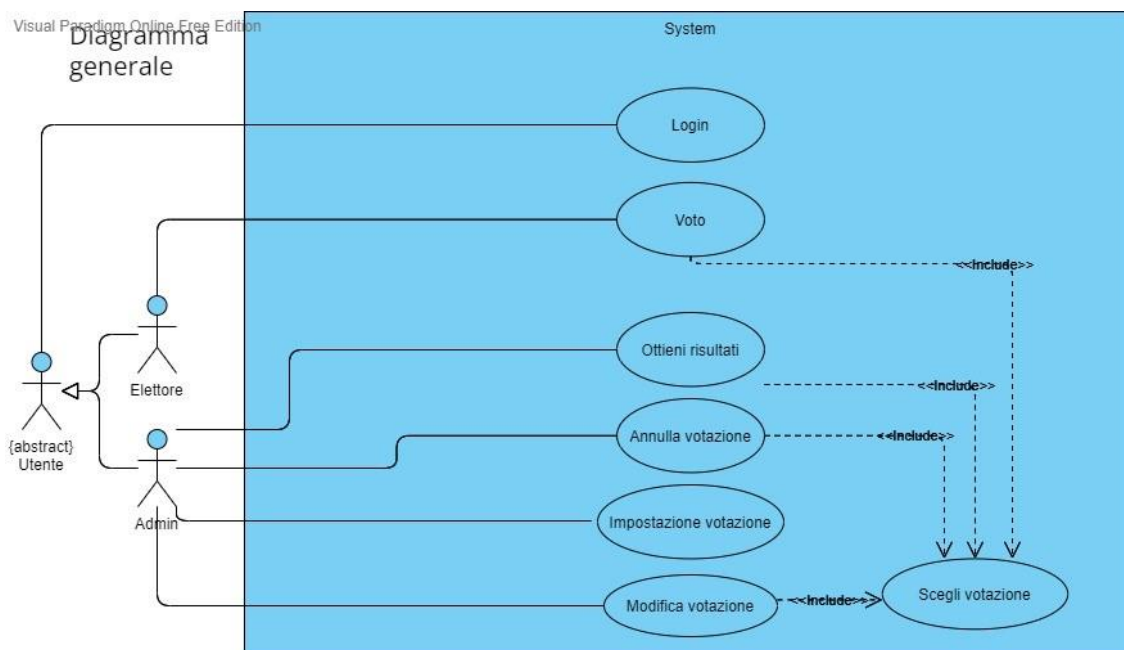
## Progettazione del Sistema:

- Diagramma dei casi d'uso:

Primo diagramma dei casi d'uso:

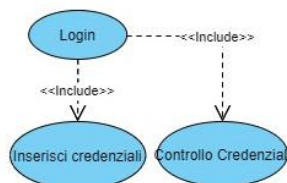


Secondo diagramma dei casi d'uso:



• Descrizione degli scenari:

Nome	Login
Scopo	Far accedere al sistema gli utenti
Attore/i	Persona
Pre-condizioni	L'attore Persona deve avere le credenziali per l'accesso
Trigger	L'attore Persona vuole autenticarsi
Descrizione	L'attore Persona dovrà inserire le credenziali di accesso, in base alle quali accederà al sistema come Admin o <i>Elettore</i>
Sequenza eventi	1 - Persona inserisce le credenziali 2 - Le credenziali sono corrette 3 - L'accesso viene effettuato
Alternativa/e	2'- Le credenziali non sono corrette 3'- L'accesso non viene effettuato
Post-condizioni	



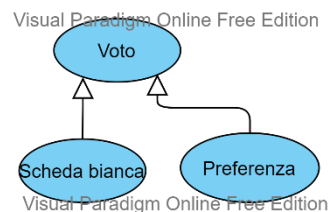
Nome	Impostazione votazione
Scopo	Far impostare le votazioni ad un Admin
Attore/i	Persona
Pre-condizioni	L'attore Persona deve aver effettuato il Login come Admin correttamente.
Trigger	L'attore Persona vuole impostare una votazione
Descrizione	L'attore Persona dovrà scegliere inizialmente la tipologia di votazione, la tipologia di vittoria e quando le votazioni saranno chiuse ufficialmente.
Sequenza eventi	1 - Persona sceglie tipologia di votazione 2 - Persona sceglie tipologia di vittoria 3 - Persona imposta la chiusura della votazione 4 - Persona conferma le scelte
Alternativa/e	4' - Persona non conferma le scelte 5' - Persona modifica le scelte 6' - Persona conferma le scelte
Post-condizioni	La scelta di data per la chiusura delle votazioni non può ricadere prima della data odierna.



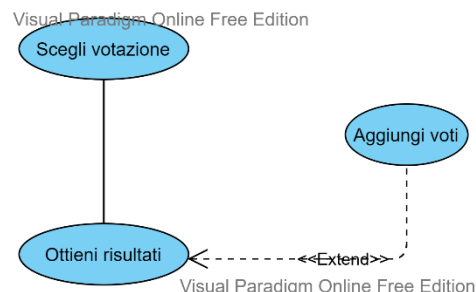




Nome	Voto
Scopo	Far votare l' <i>Elettore</i>
Attore/i	Persona
Pre-condizioni	L'attore Persona deve aver effettuato il Login come <i>Elettore</i> correttamente. Il sistema deve essere stato impostato da un Admin (caso d'uso 2). Il voto deve avvenire in una sessione di tempo valida. L'attore Persona può votare al più una volta. L'attore Persona deve essere maggiorenne
Trigger	L'attore Persona vuole esprimere un voto
Descrizione	L'attore Persona, indipendentemente dal tipo di votazione, procederà al voto scegliendo dalle opzioni presenti, il sistema richiederà conferma prima di procedere alla registrazione del voto
Sequenza eventi	1 - Persona esprime il suo voto 2 - Persona conferma la scelta
Alternativa/e	1' - Persona lascia scheda bianca 2' – Persona conferma la scelta
Post-condizioni	



Nome	Otteni risultati
Scopo	Determinare i risultati della votazione
Attore/i	Persona
Pre-condizioni	L'attore Persona deve essere un Admin che ha effettuato il Login
Trigger	L'attore Persona vuole sapere i risultati di una votazione
Descrizione	L'attore Persona avvierà la procedura di determina vincitore all'interno del sistema e i risultati verranno comunicati a video e in un file
Sequenza eventi	1 - Persona richiede di determinare il vincitore 2 – I risultati vengono comunicati
Alternativa/e	2' - Errore
Post-condizioni	Le votazioni devono essere concluse





- Diagramma delle classi:

Diagramma delle classi di progetto [Versione 1 – Assignment]:

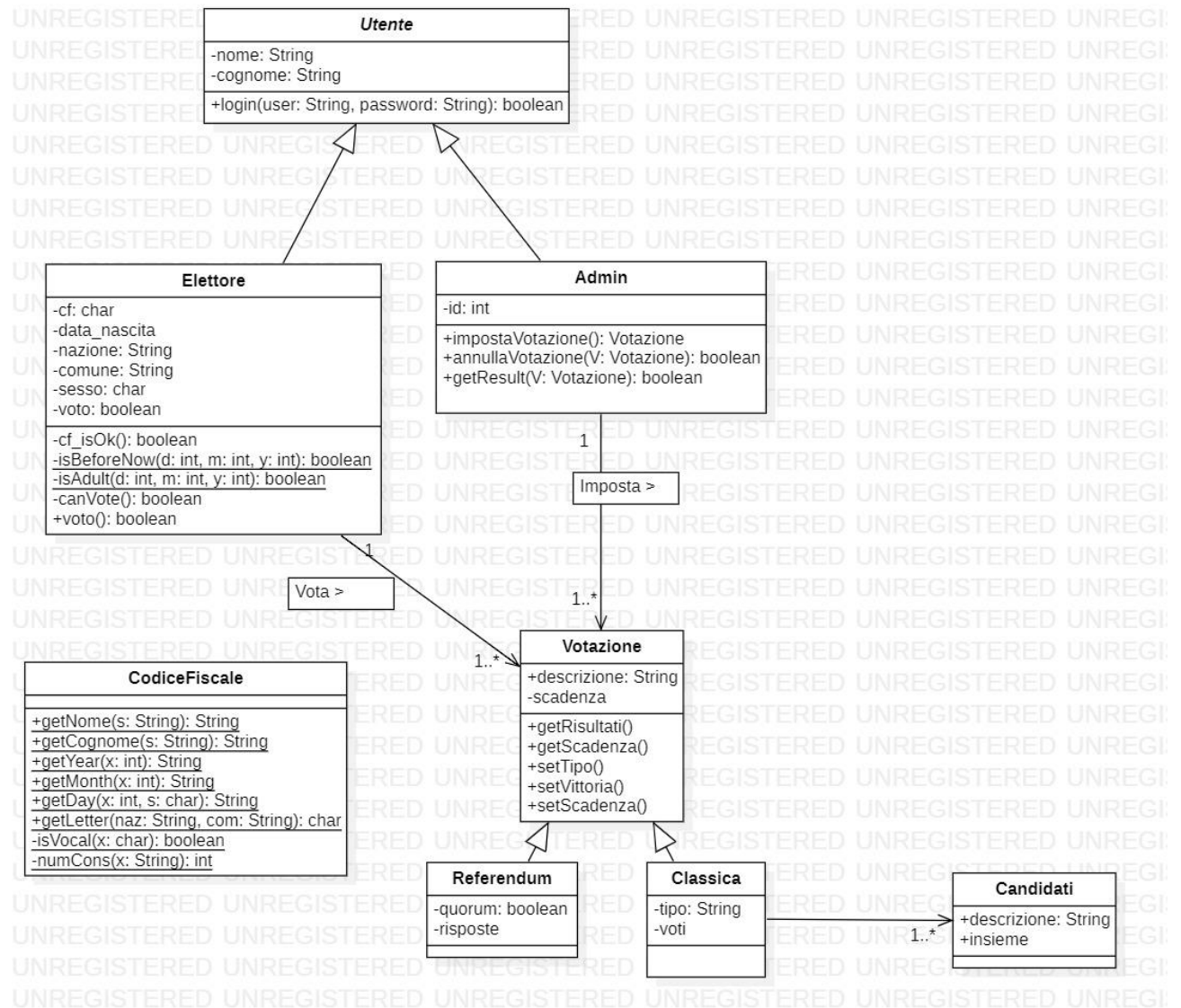
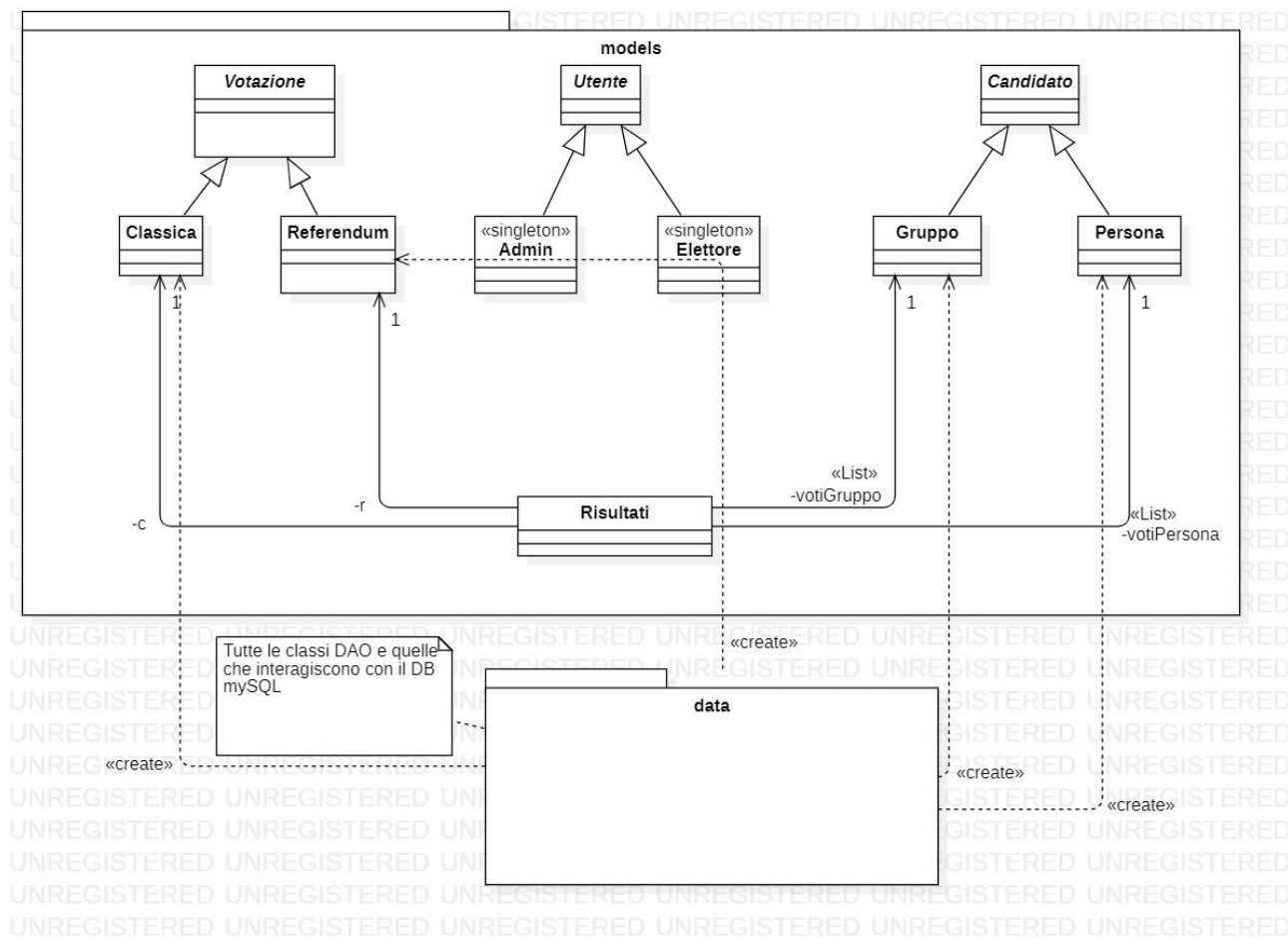




Diagramma delle classi di progetto [Versione 2 – definitiva]:

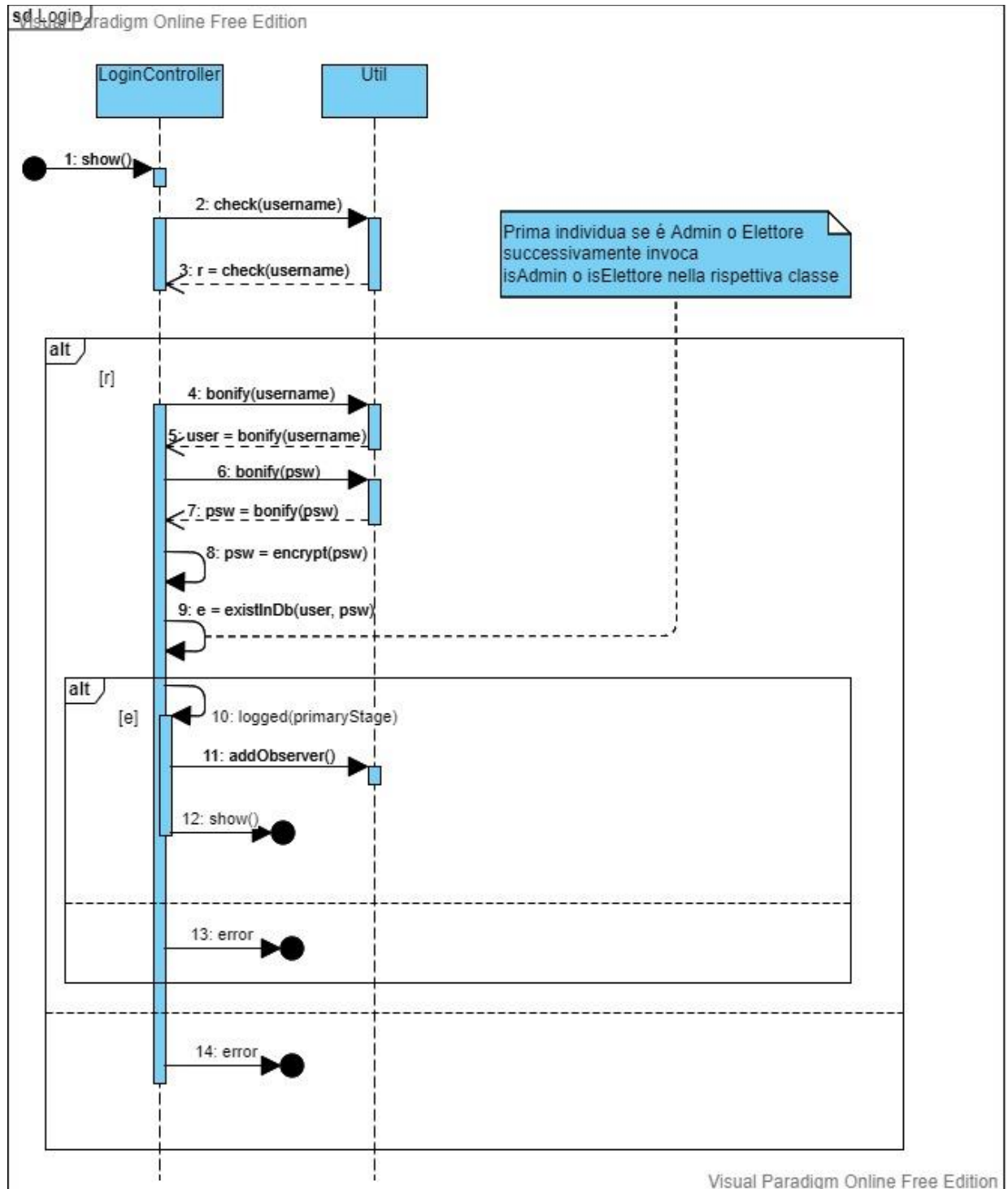


- **Diagrammi di sequenza:**

Prima di entrare nel vivo dei diagrammi tenevo a specificare che sono stati posti i metodi più ad alto livello, in modo tale da rendere i diagrammi più leggibili.

Nel caso si rendesse necessaria una visione più specifica [qui](#) si possono trovare dei diagrammi più “zoomati”, realizzati con [VisualParadigm](#).

Diagramma che mostra i passi per autenticarsi all’interno del sistema:



[*encrypt* è una abbreviazione per la chiamata di metodi della classe interna responsabile del criptaggio]



Diagramma che mostra i passi necessari per accertare la presenza di un *utente* all'interno della piattaforma:

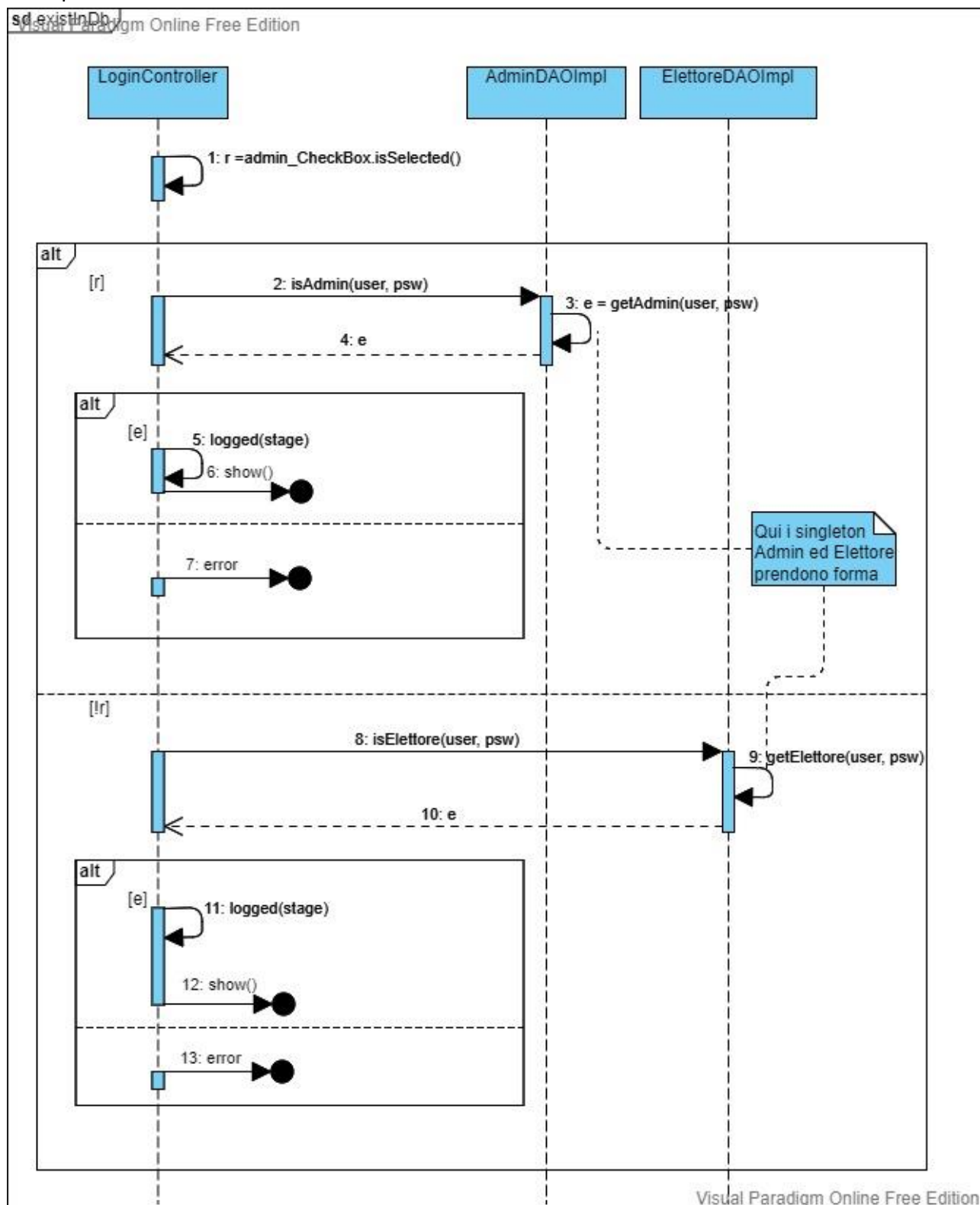




Diagramma che mostra i passi per creare una nuova votazione di tipo ordinale essendo  
*Amministratore*:

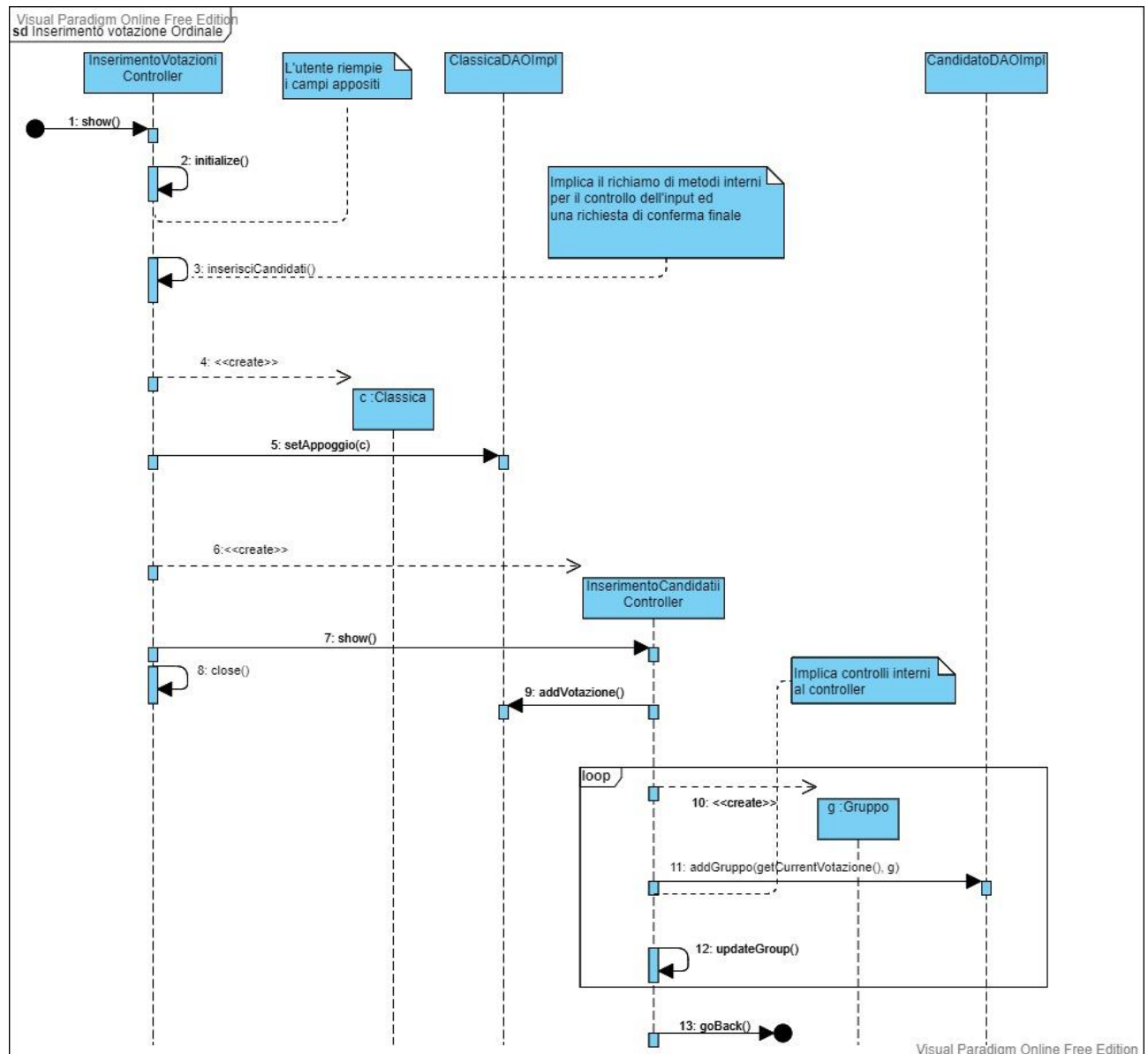




Diagramma che mostra i passi necessari ad un *Elettore* per votare ad un referendum:

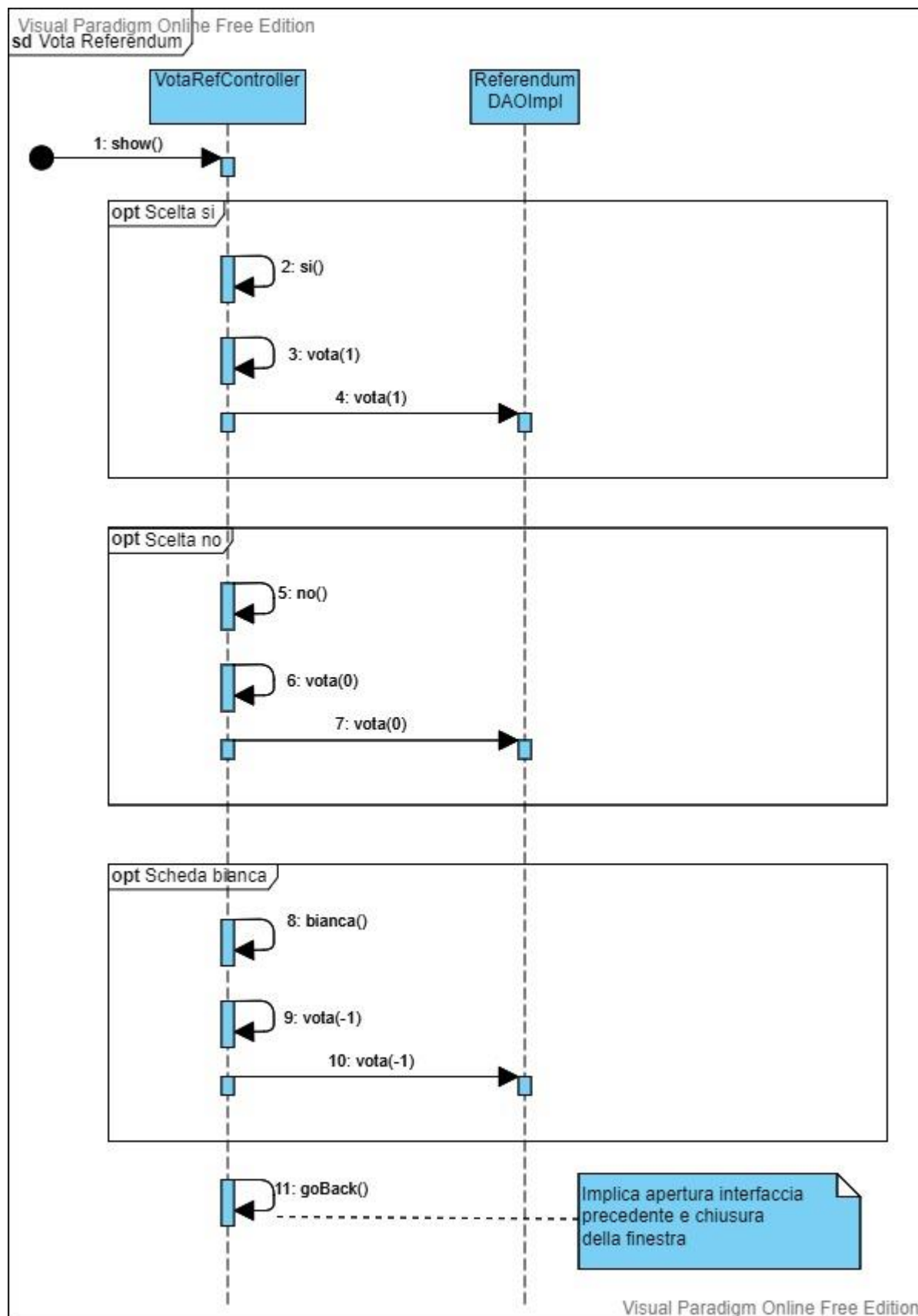
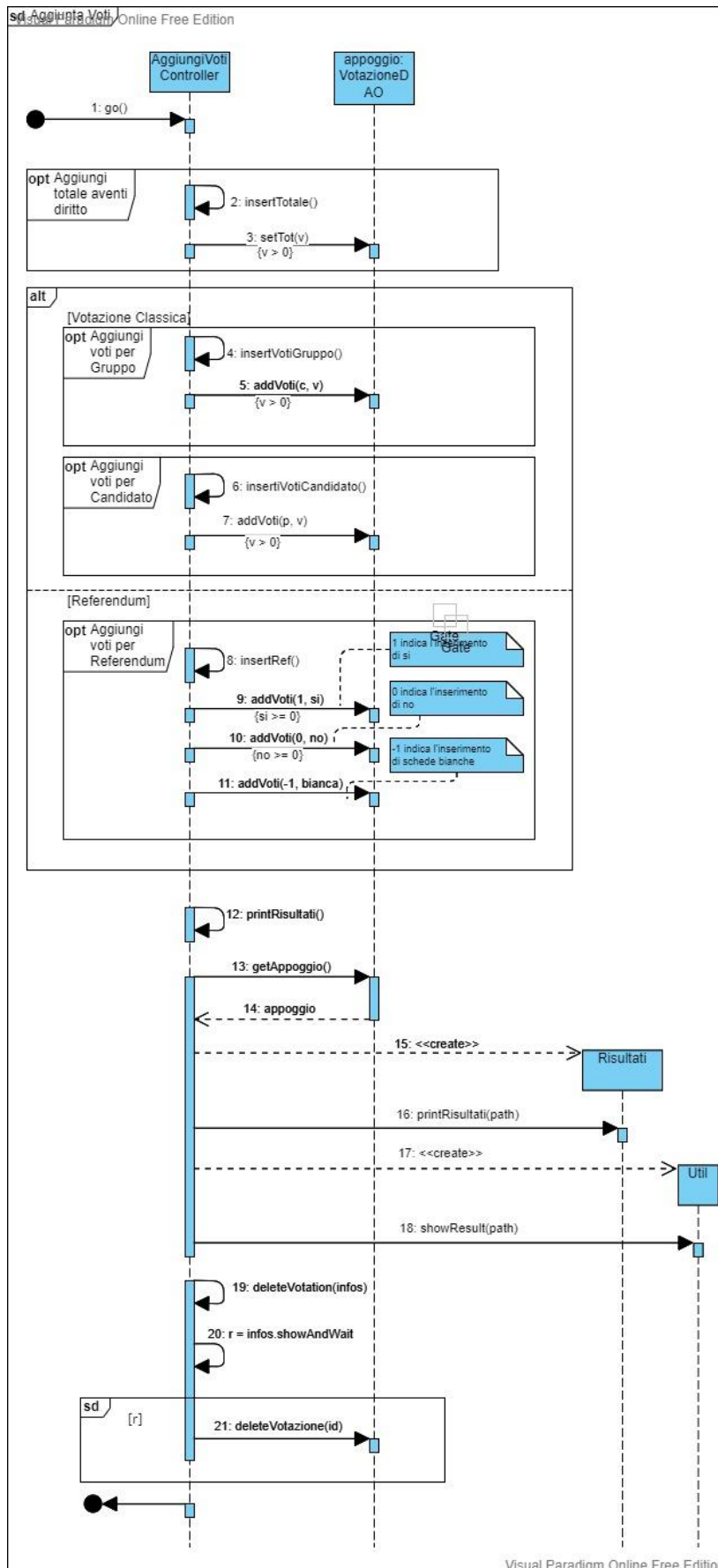






Diagramma che mostra i passi necessari ad un *Amministratore* per aggiungere e visualizzare i risultati di una votazione:







- Diagrammi delle attività:

Diagramma che mostra l'ordine di esecuzione delle operazioni legate al *login* all'interno della piattaforma:

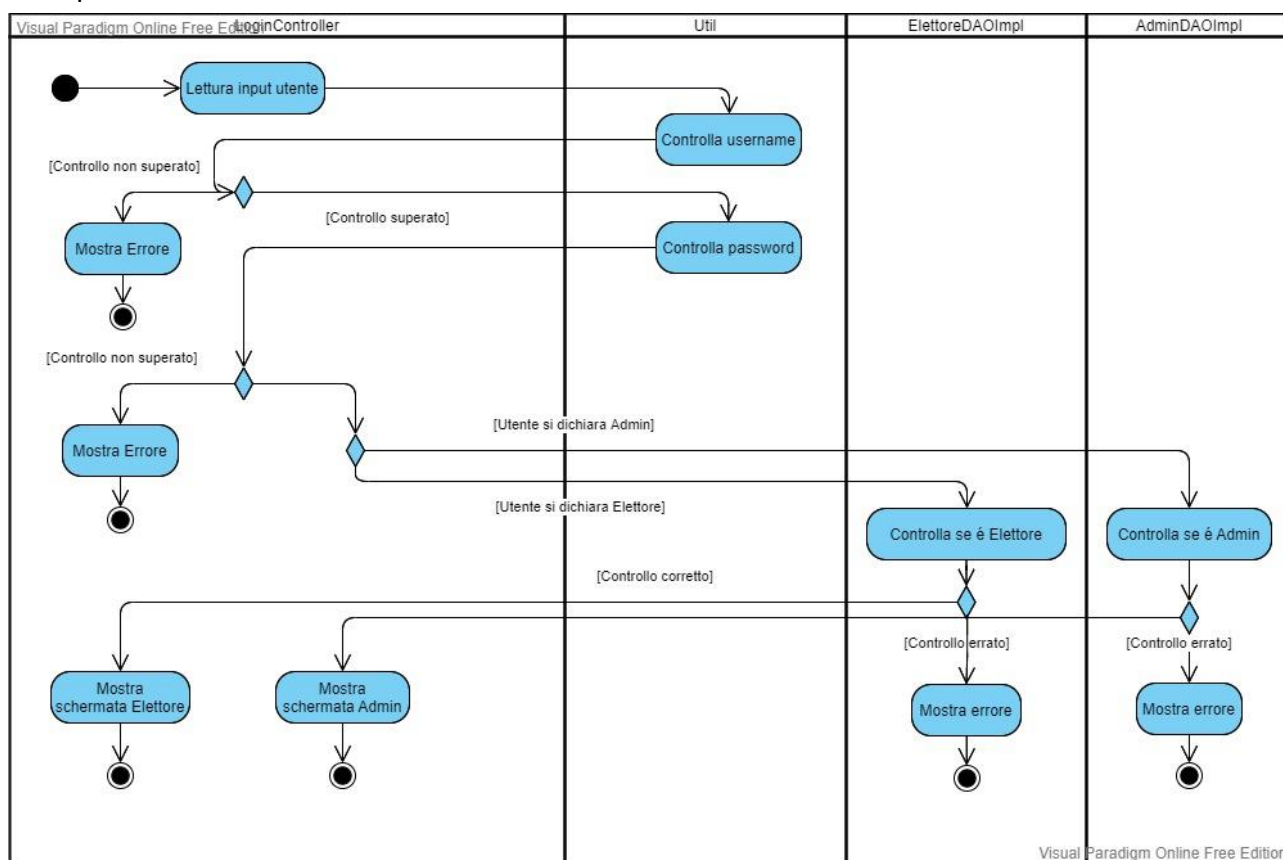




Diagramma che mostra l'ordine di esecuzione delle azioni/attività necessarie per ottenere i risultati di una votazione, con la possibilità di inserire i voti relativi alla sessione svoltasi in presenza:

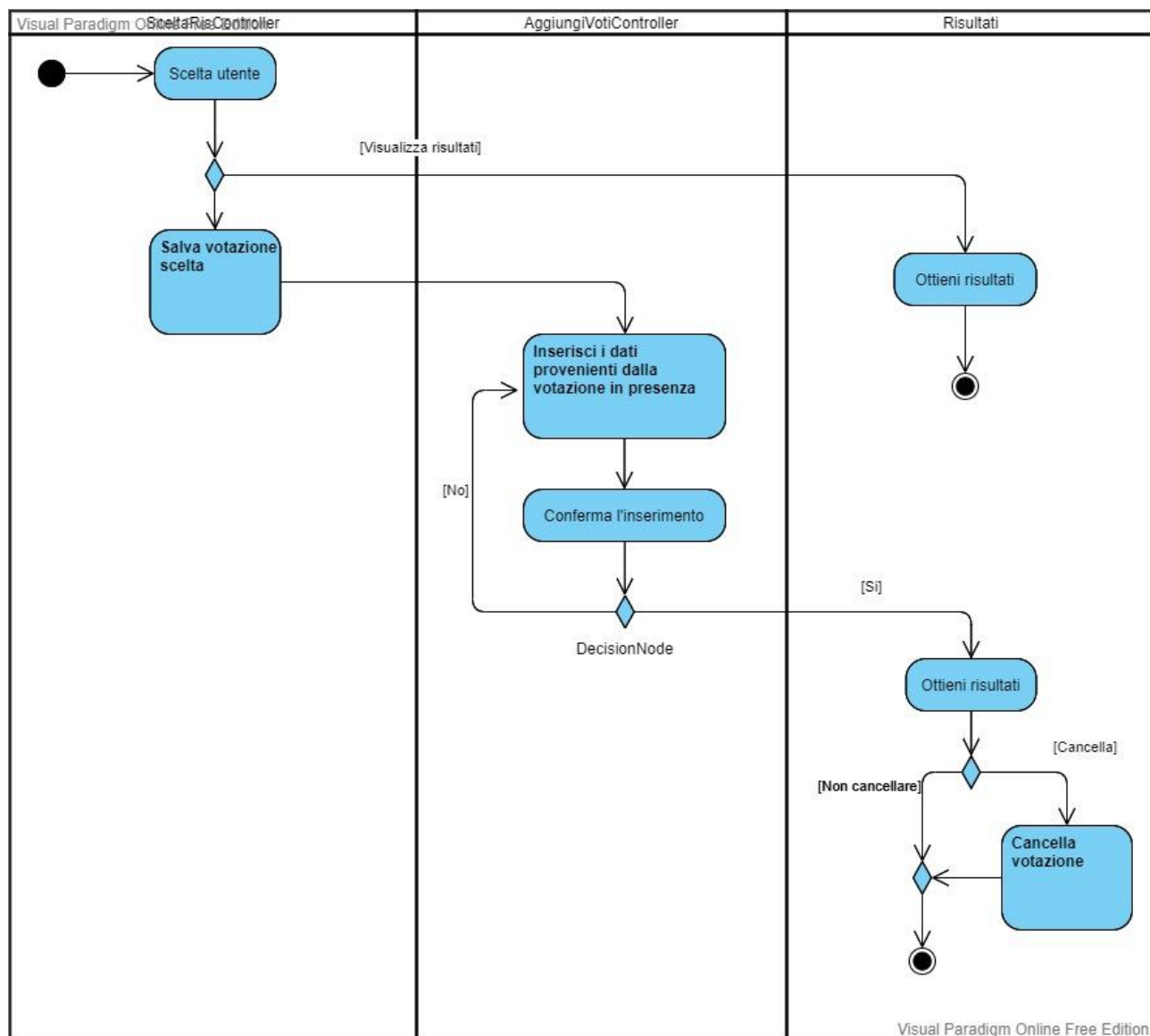




Diagramma che mostra l'ordine di esecuzione delle azioni che permettono ad un *Elettore* di esprimere la propria preferenza ad una votazione:

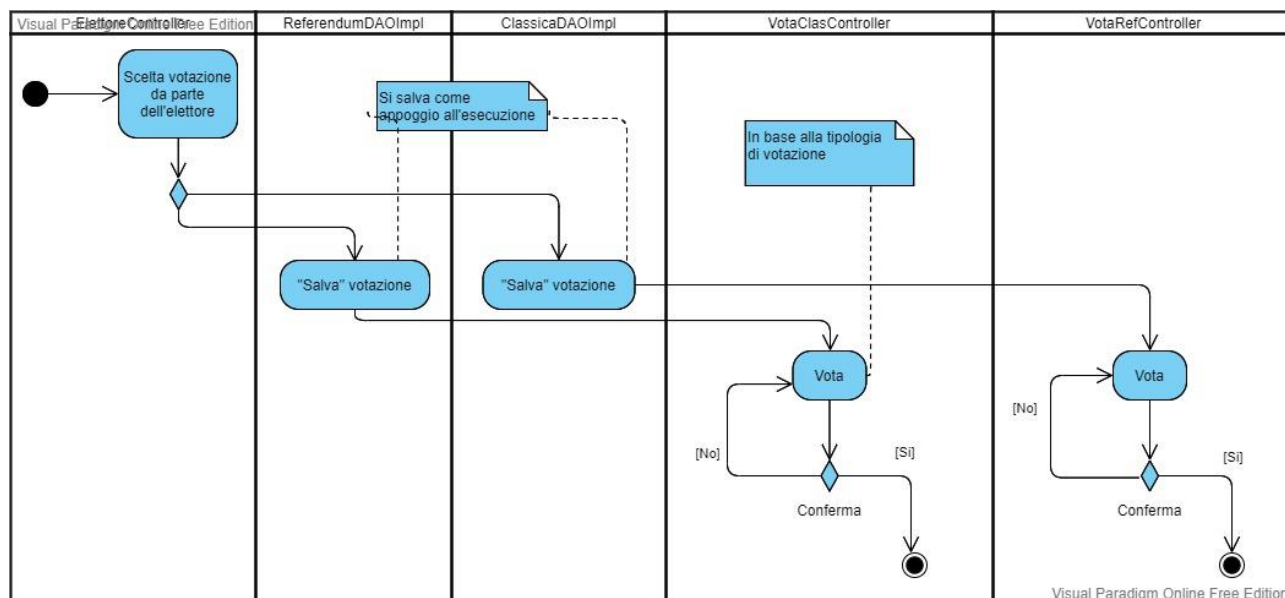
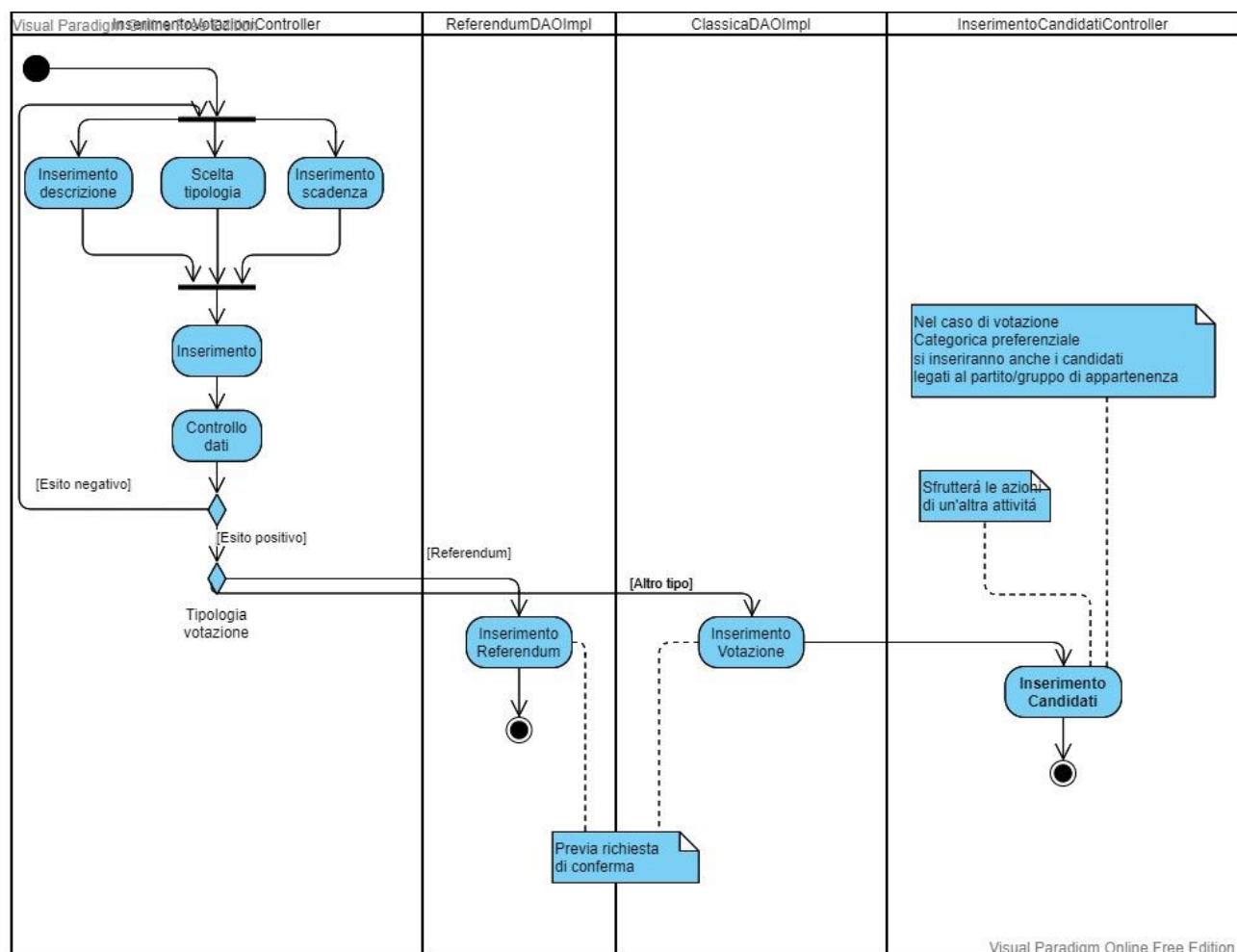


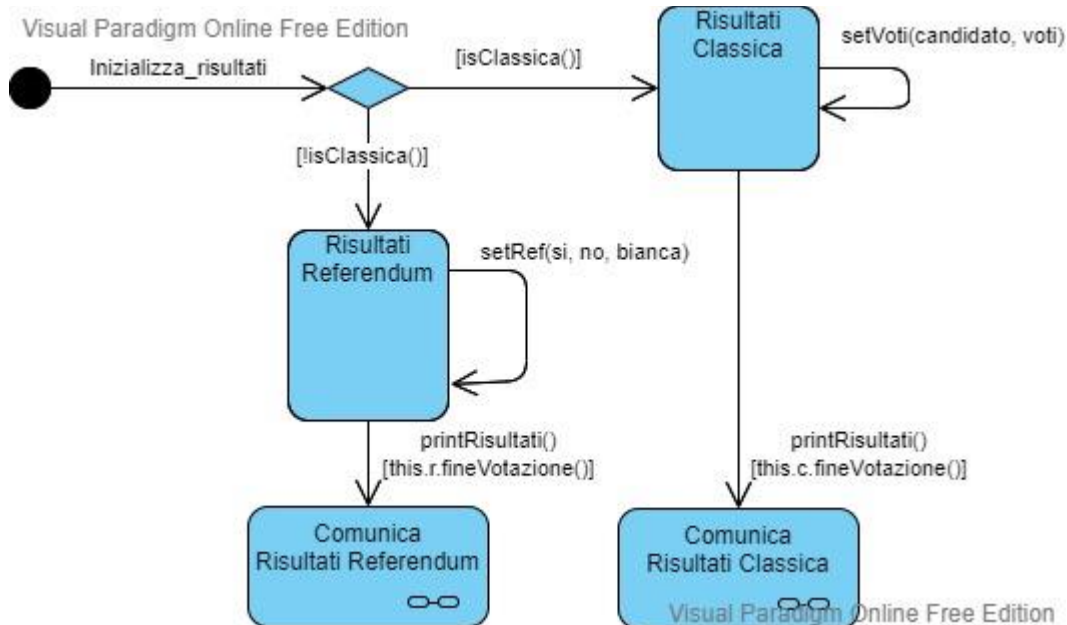
Diagramma che mostra l'ordine di esecuzione delle azioni/attività che permettono ad un *Amministratore* l'inserimento di una votazione:



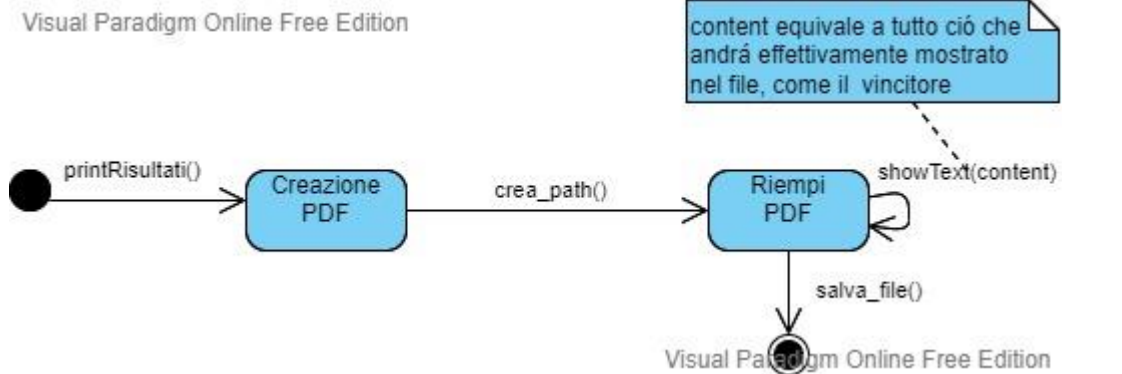
- **Macchine di stato:**

Non facendo troppo affidamento sullo stato delle classi, ma riponendo il gravoso compito al database la realizzazione di troppe macchine di stato risulta un compito futile, pertanto ho deciso di mostrarne solamente una, che fa riferimento alla classe che più di tutte fa uso del suo stato interno, Risultati.java.

**Risultati:**



**Metodo Risultati.printRisultati():**







## Implementazione del sistema:

### • Diagramma delle classi:

Diagramma delle classi di Programma, senza le classi legate a Controller e View del MVC applicato all'interno del progetto:

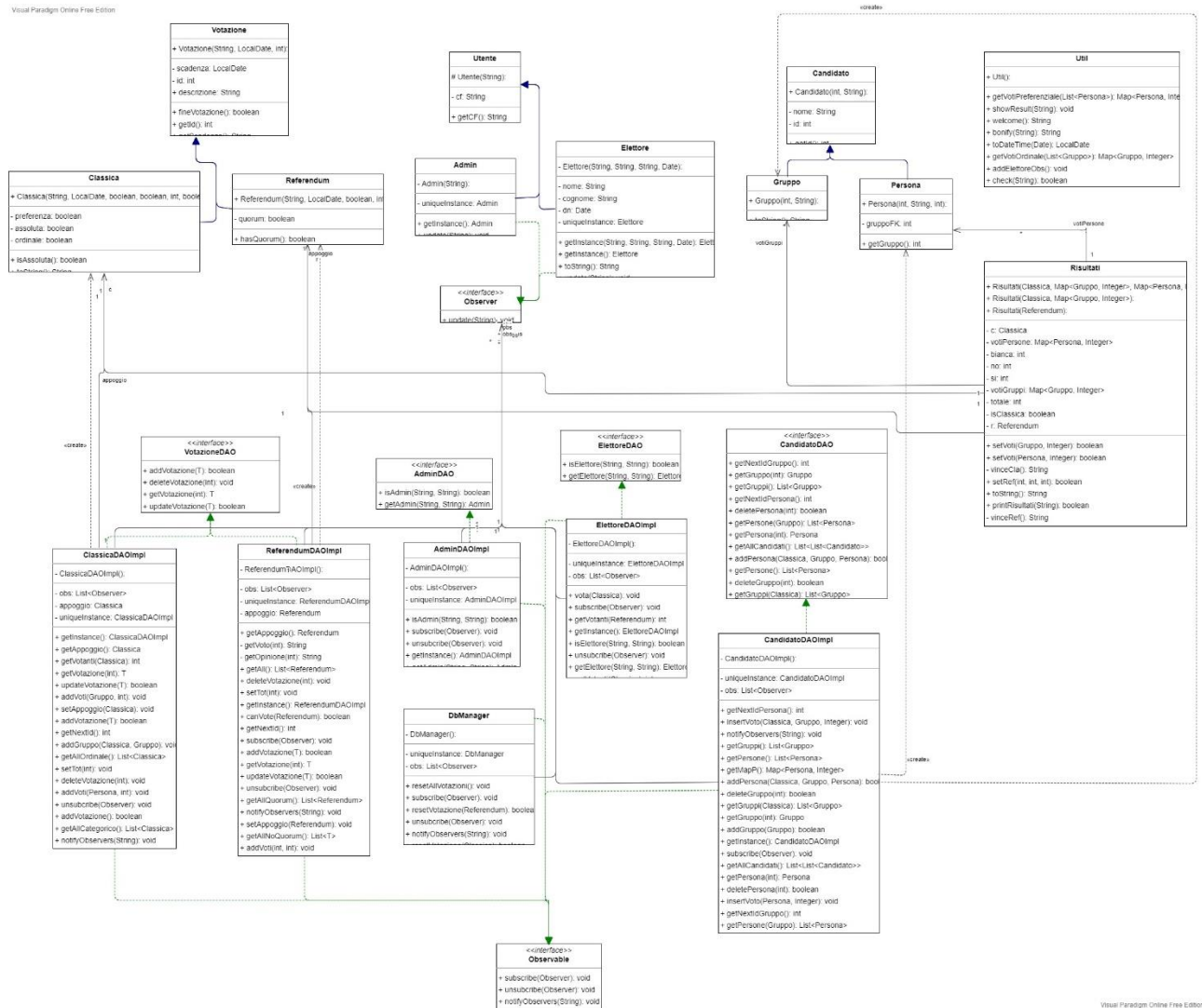


Diagramma delle classi di programma completo e file .drawio per modificarlo a piacimento.



- **Discussione dei Design Pattern utilizzati:**

- **MVC - Model View Controller:**

Ho utilizzato questo pattern per gestire al meglio ogni interazione dell'utente con l'interfaccia grafica.

Seguire questo pattern implica avere un *modello* (Model), una *vista* (vista) ed un *controllore* (controller) per ogni finestra della GUI.

Pertanto ad ogni file .fxml (views) viene associato, tramite stringa di codice fx:controller, un controllore .java (controllers) e sarà proprio quest'ultimo ad utilizzare le classi .java (models) per portare a termine il compito dettato dall'utente.

Questo DP permette di suddividere i compiti in un tritico di tipi di file riconosciuti dal loro compito, rendendo più fluido il controllo degli input utente e la manutenzione/implementazione dei task del software.

All'interno della file system del progetto si possono notare le directory (packages) atte ad implementare questo pattern.

Come già immesso in precedenza nel quinto assignment, ho esteso l'uso del pattern a tutte le finestre del software.

- **DAO – Data Abstract Object:**

Ho utilizzato questo pattern per "impacchettare" in interfacce e classi dedicate l'interazione con il db MySQL.

Ciò permette di suddividere l'accesso ai dati persistenti in classi dedicate a quest'ultimo compito, permettendo facile manutenzione e comprensione.

È stata creata una directory (package) dedicata a questo tipo di classi, l'unica eccezione riguarda la classe DBManager che ricalca una classe Utility, per tasks non appartenenti ad un singolo tipo di dati persistenti.

Le classi modello (*Elettore*, Admin, *Votazione*, *Classica*, *Referendum*, ...) sono state tutte "dotate" di DAO.

Come già immesso in precedenza nel quinto assignment, ho esteso l'uso del pattern alla maggior parte delle classi modello (*Elettore*, Admin, *Votazione*, *Classica*, *Referendum*, ...).

- **Singleton:**

Data la grossa mole di classi che non necessitano una creazione multipla (o addirittura di creazione) è stato naturale utilizzare questo design pattern sia per le classi *Elettore* e Admin, semplificando esponenzialmente il loro richiamo in tutte le parti del software (specialmente per la stesura dei log di sistema), sia per tutte le classi che seguono il design pattern DAO esposto in precedenza.

In queste ultime classi è stato utile, alle volte, porre una variabile di appoggio al loro interno, per semplificare il passaggio di dati tra Stage.





- **Observer/Observable:**

Questo pattern è stato implementato mappando gli oggetti Observer nelle classi *Elettore* ed *Admin* e gli oggetti Observable nelle classi che seguono il DP DAO. È stato sfruttato per scrivere i log utente/*Amministratore* in maniera comoda e fluida.

- **Strategy/Composite/Observer:**

Per chiarezza teorica è bene specificare che, questo trittico di DP, vanno a formare il Model View Controller.

Strategy viene riconosciuto nel Controller, il quale lo implementa, questo poichè la View di un oggetto è configurato con una strategia, il controllore dà questa strategia. Infatti, la vista si occupa solamente degli aspetti visuali e delega il controllore per ogni decisione sui comportamenti dell'interfaccia. L'utilizzo di questo pattern tiene inoltre la View disaccoppiata dal Model, per la responsabilità data al controller.

Composite viene riconosciuto nella View, che è un insieme di finestre, pannelli, bottoni, testi, ecc innestati.

Observer viene riconosciuto nel Model che tiene gli oggetti interessati aggiornati quando lo stato cambia.

Questo piccolo paragrafo è stato tratto dal libro *Head First Design Patterns* di *Eric Freeman*, tengo a specificare che è stato fatto solo per completezza, inoltre nel caso di questo progetto non è stata seguita alla lettera questa composizione, ma è possibile riconoscerla solo a tratti.

- **Simple Factory:**

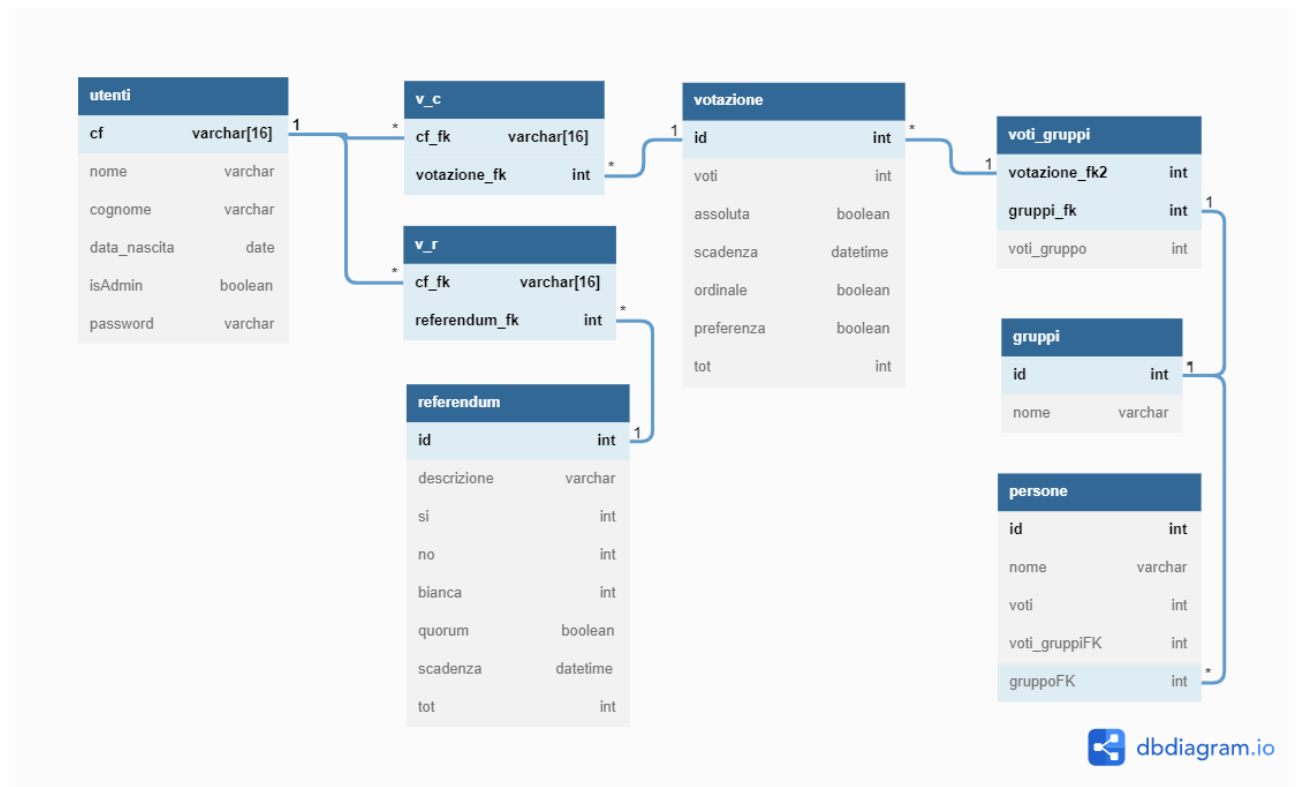
Questo pattern è stato riconosciuto all'interno della struttura del progetto dopo la sua implementazione.

Di fatto la classe DAO *VotazioneDAO* segue i principi di questo DP, questo in quanto può produrre una lista di *Votazione* in tutte le sue sottotipazioni.





- Gestione dei dati persistenti:



**Piccolo brief sull'organizzazione del database MySQL:**

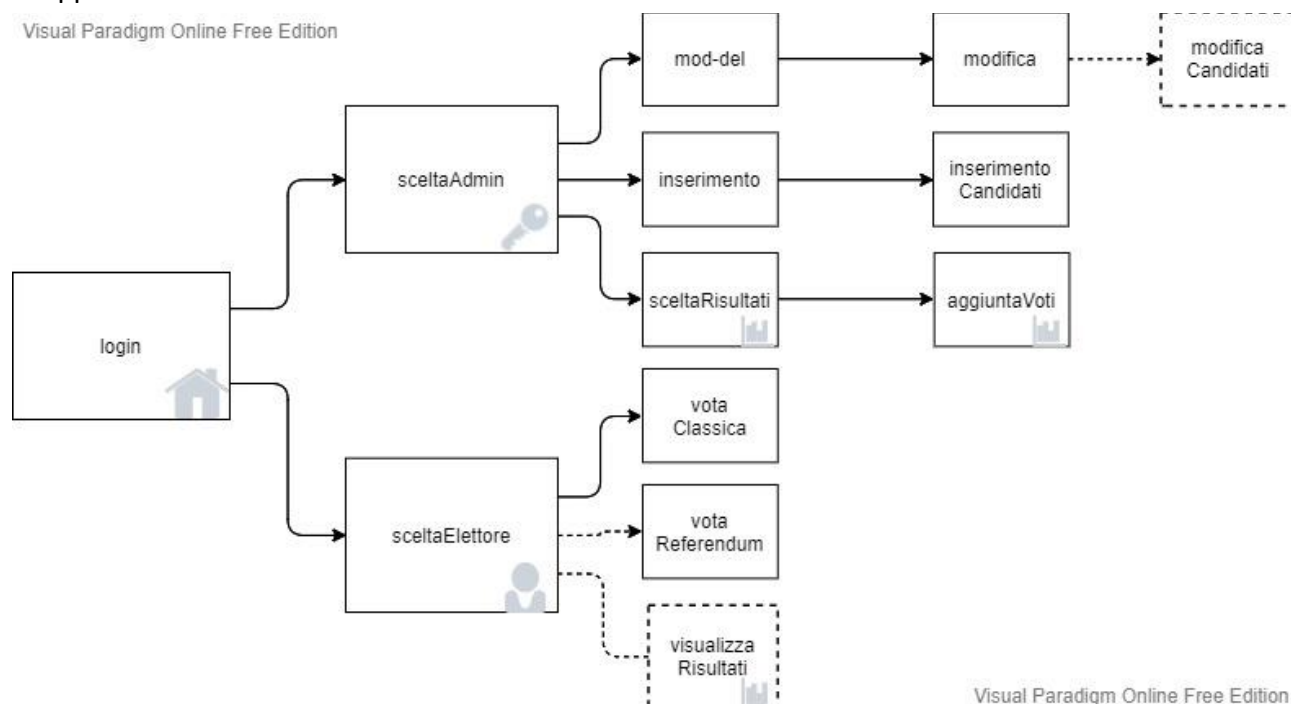
- **Utenti**: tabella utilizzata per salvare *elettori* e *admin* con le relative informazioni (password criptate);
- **v\_c**: tabella di collegamento tra gli *elettori* e le *votazioni*, usata per stabilire l'avvenuta votazione;
- **v\_r**: tabella di collegamento tra gli *elettori* e i *referendum*, usata per stabilire l'avvenuta votazione;
- **votazione**: tabella utilizzata per salvare tutte le informazioni riguardanti una votazione *classica*;
- **voti\_gruppi**: tabella di collegamento tra le votazioni *classiche* e i relativi candidati;
- **gruppi**: tabella utilizzata per salvare le informazioni riguardanti i gruppi/partiti (in caso di votazioni diverse dalla categorica preferenziale viene usata anche per i candidati);
- **persone**: tabella utilizzata per salvare le informazioni riguardanti i candidati collegati ai gruppi/partiti politici;
- **referendum**: tabella utilizzata per salvare tutte le informazioni riguardanti i *referendum*;

Dopo l'ottenimento dei risultati è possibile cancellare i dati per la votazione appena calcolata ed esportata in PDF.

- Descrizione della GUI:

Mapa dell'interfaccia:

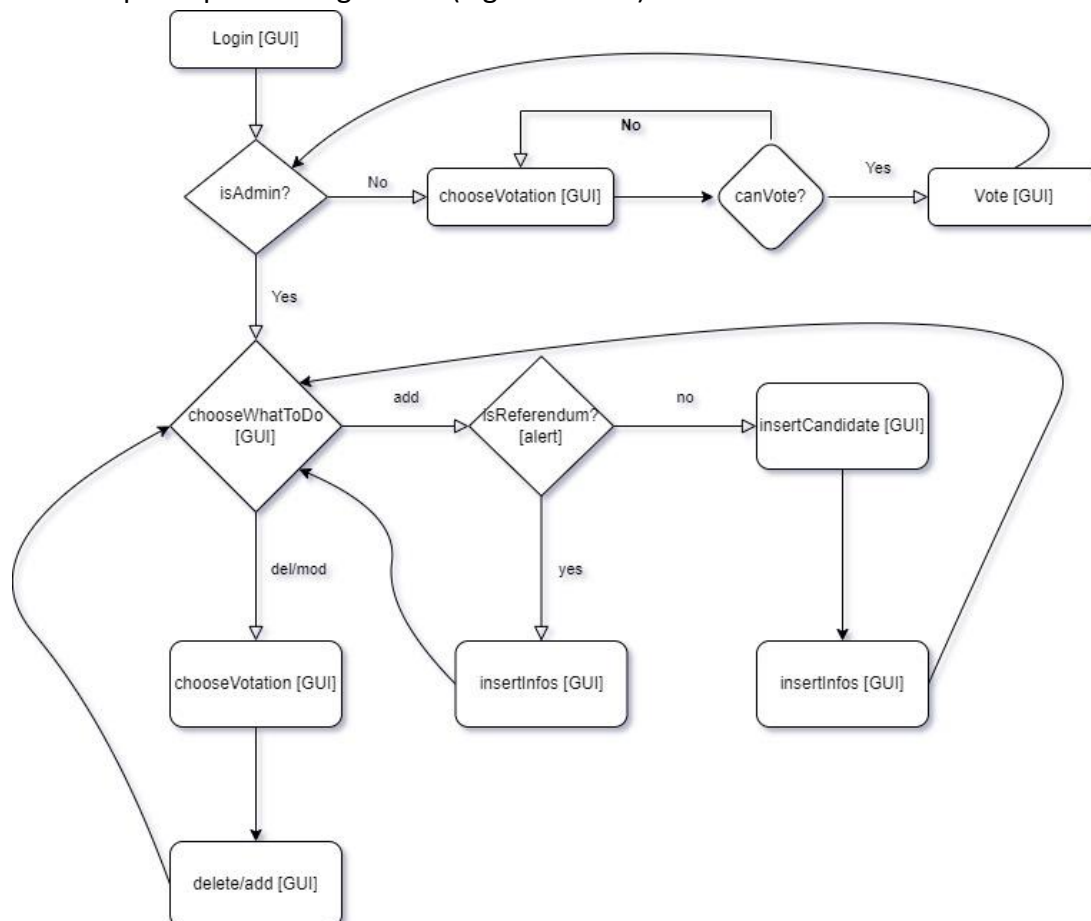
Visual Paradigm Online Free Edition



Visual Paradigm Online Free Edition

*[le figure tratteggiate indicano una possibile espansione della piattaforma]*

Schema per capire i collegamenti (a grandi linee) tra finestre:





### Login:

The screenshot shows a classic Mac OS-style window titled "Login". The title bar is purple and contains standard window controls (minimize, maximize, close). Below the title bar is a menu bar with "File", "Edit", and "Help" menus. The main content area is light gray and contains the following elements:

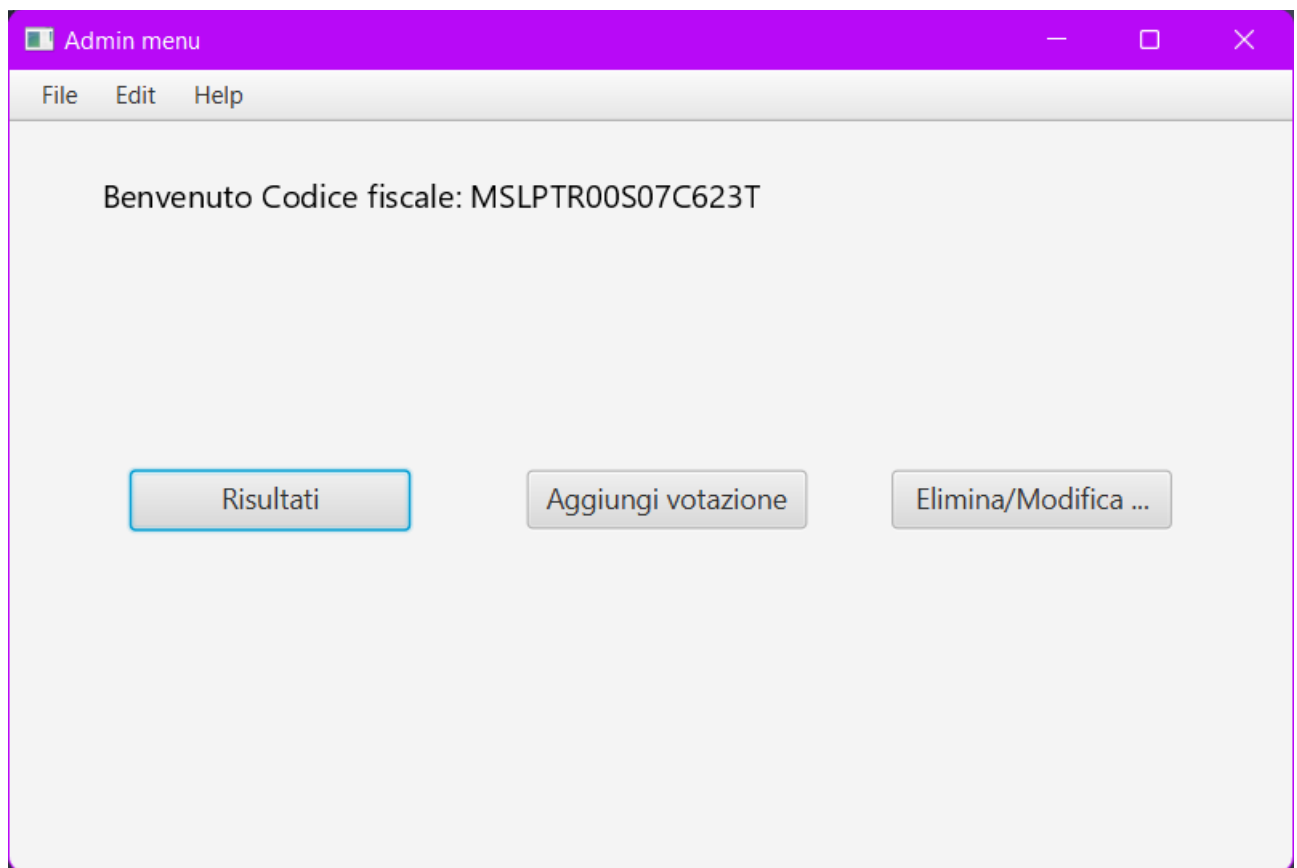
- A label "Codice Fiscale:" followed by a text input field containing the value "mslptr00s7c623t".
- A label "Password:" followed by a password input field with 10 dots.
- A checkbox labeled "Admin" which is checked.
- A blue "Login" button.

### controlli in input:

- username deve corrispondere al codice fiscale, viene effettuato un controllo sulla lunghezza (16 caratteri), successivamente la stringa viene “bonificata” contro possibili attacchi di *SQL Injection* e successivamente verrà controllata nel db;
- password viene “bonificata” da possibili attacchi di *SQL Injection*, successivamente criptata e poi confrontata nel db;
- se la checkbox Admin viene flaggata il software controllerà se l’utente esiste nel db ed è abilitato come *Amministratore*.



### Scelta admin:





### Inserimento votazione:

Admin menu

File Edit Help

☐ Votazione ☐ Quorum ☒ Assoluta

Mettere la spunta a 'Votazione' per inserire un referendum

Amministrative Velletri

Ordinale 23/07/2022

Procedi a inserire candid... Inserisci Referendum

Indietro

### controlli in input:

- se la checkbox Referendum è flaggata le opzioni relative ad una votazione Classica vengono disabilitate e viceversa;
- la data di scadenza non può essere prima della giornata odierna, ne vuota;
- la descrizione deve essere superiore ai 5 caratteri e viene “bonificata”.



### Scelta di modifica o cancellazione:

Admin menu

File Edit Help

Buon pomeriggio amministratore:  
Codice fiscale: MSLPTR00S07C623T

☐ Elimina Una sola selezione alla volta é ammessa

Votazione per Amministrative Chiavenna  
Votazione categorica non preferenziale -/- 2022-07-28

Votazione per Amministrative Roma  
Votazione categorica preferenziale -/- 2022-07-28

Votazione per Amministrative Milano  
Votazione ordinale -/- 2022-07-28

Referendum per Eutanasia Legale  
Con quorum -/- Scadenza: 2022-07-21

Referendum per Cannabis Legale  
Senza quorum -/- Scadenza: 2022-07-22

Indietro Aggio...

### controlli in input:

- è necessaria la scelta di una votazione nella lista;
- se la checkbox è flaggata il sistema richiederà conferma prima di eliminare.



### Modifica:

Modifica Classica

File Edit Help

☐ Quorum ☒ Assoluta

Votazione per Amministrative Milano  
Votazione ordinale -/- 2022-07-28

Amministrative Milano

Categorico senza preferenze ▼ 28/07/2022

Modifica

Indietro

### controlli di input:

- la data di scadenza non può essere prima della giornata odierna, né vuota;
- la descrizione deve essere superiore ai 5 caratteri e viene "bonificata".



### Inserimento candidati (gruppi/partiti/persone):

Nome:

Selezionare appartenenza a Gruppo/Partito: ☐ Gruppo/Partito

- Unione Politica M5S
- Unione Politica PD

### controlli in input:

- il nome non può essere nullo e viene inoltre “bonificato”;
- la checkbox indica il collegamento al Gruppo/Partito selezionato.





### Scelta della votazione per il voto:

Buon pomeriggio elettore  
Nome: Pietro -/- Cognome: Masolini -/- Data di nascita: 2000-11-07  
Codice fiscale: MSLPTR00S07C623T

Una sola selezione alla volta è ammessa

- Votazione per Amministrative Chiavenna  
Votazione categorica non preferenziale -/- 2022-07-28
- Votazione per Amministrative Roma  
Votazione categorica preferenziale -/- 2022-07-28
- Votazione per Amministrative Milano  
Votazione ordinale -/- 2022-07-28
- Referendum per Eutanasia Legale  
Con quorum -/- Scadenza: 2022-07-21
- Referendum per Cannabis Legale  
Senza quorum -/- Scadenza: 2022-07-22

Vota

### controlli in input:

- è necessaria la scelta di una votazione nella lista.



### Voto Referendum:

A screenshot of a web application window titled "Eutanasia Legale". The window has a purple title bar with standard minimize, maximize, and close buttons. Below the title bar is a menu bar with "File", "Edit", and "Help" options. The main content area is light gray and contains the text "Referendum per Eutanasia Legale" and "Con quorum -/- Scadenza: 2022-07-21". At the bottom, there are three buttons: "Si" (highlighted with a blue border), "Bianca", and "No".

Eutanasia Legale

File Edit Help

Referendum per Eutanasia Legale  
Con quorum -/- Scadenza: 2022-07-21

Si Bianca No



### Voto Classica:

### controlli in input:

- in caso di votazione categorica e categorica preferenziale è necessario selezionare un candidato dalla lista di sinistra prima di votare;
- in caso di votazione ordinale o categorica preferenziale è necessario ordinare i candidati, rispettivamente tabella sinistra e tabella destra, prima di votare.



### Scelta votazione per aggiungere/ottenere risultati:

Scelta votazione per risultati

File Edit Help

Seleziona una votazione per poter aggiungere i voti in presenza

- Votazione per Amministrative Chiavenna
- Votazione categorica non preferenziale -/- 2022-07-14

Indietro Aggiungi Ottieni

### controlli in input:

- è necessaria la scelta di una votazione nella lista.



### **Aggiunta risultati:**

Aggiungi voti a Classica

File Edit Help

Partito/Gruppo:

Unione Politica Fratelli d'Italia

1000

Inserisci Ottieni Candidati

Candidati:

Voti

Inserisci

Si: Voti

No: Voti

Bianca: Voti

Inserisci

Totale aventi diritto: Possibili elettori

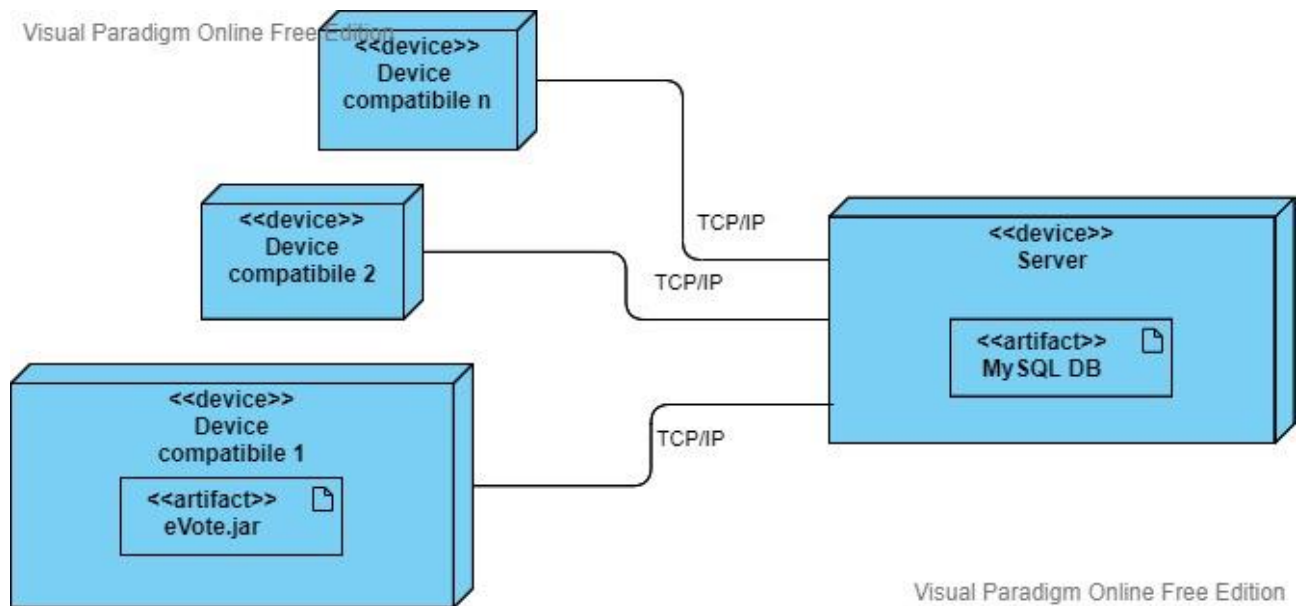
Inserisci

Indietro Avanti

### **controlli in input:**

- i voti aggiunti devono essere superiori a 0;
- in base al tipo di votazione scelta verranno disabilitate le aggiunte per l'altro tipo.

- **Diagramma di deployment:**



- **Specifica e verifica dei vincoli:**

In questa relazione vengono riportati i vincoli più significativi tra tutti quelli possibili.  
Le parti sottolineate implicano una mappatura in JML

context Utente inv:

Util.check(self.cf)

```
public class Utente{ //JML
```

```
    //@ invariant Util.check(cf)
```

```
    ...
```

```
}
```

```
public class Util{ //JML
```

```
    //@ requires (cf != null) && (cf.length() == 16);
```

```
    public static /*@ pure @*/boolean check(String cf) {
```

```
        ...
```

```
    }
```

```
}
```

context Elettore inv:

```
self.nome != null && self.nome.length() > 0 && self.cognome != null &&
self.cognome.length() > 0 && self.dn != null
```



```
public class Elettore extends Utente implements Observer{ //JML
    //@ invariant nome != null && nome.length() > 0 &&
    //@cognome != null && cognome.length() > 0 && dn != null ;
    ...
}
```

context **Votazione** inv:

```
self.descrizione != null && self.descrizione.length() > 0 && self.scadenza != null &&
self.id > -1
```

```
public abstract class Votazione{ //JML
    //@ invariant descrizione != null && descrizione.length() > 0 &&
    //@ scadenza != null && id > -1;
    ...
}
```

context **Candidato** inv:

```
self.id > -1 && self.nome != null && self.nome.length() > 0
```

```
public abstract class Candidato{ //JML
    //@ invariant id > -1 && nome != null && nome.length() > 0;
    ...
}
```

- **Descrizione del testing:**

Il testing è stato effettuato su tre classi (Referendum, Classica e Util).

Ho testato tutti i metodi pubblici utili al funzionamento della piattaforma, utilizzando il **criterio di copertura delle decisioni**.

Le classi di test si trovano nelle rispettive directory (*test/models* e *test/util*) ed è stato usato Junit4. Ecco le classi di test:

- [Classe di testReferendum](#)
- [Classe di testClassica](#)
- [Classe di testUtil](#)

Tutte le decisioni presenti nei metodi sono state coperte.



Per quanto riguarda il test di una delle Interfacce Utente è stata scelta quella relativa all'inserimento di una nuova votazione da parte di un *Amministratore*:

1. Consistenza:

il font ricalca le abitudini degli utenti, risultando normale ai loro occhi;  
i colori dell'interfaccia sono grigio, bianco e nero che donano una visione molto classica e semplice;  
gli elementi grafici inseriti aiutano a capire il funzionamento del software;

2. Spelling:

sulla schermata non si notano refusi, neanche nei messaggi di errore;

3. Leggibilità:

la schermata risulta leggibile e gli elementi sono ben delineati, inoltre le interazioni inutili vengono disabilitate a seconda del tipo di votazione da inserire;

4. Comportamento elementi interattivi:

ogni elemento ha una sua funzionalità e si aggiorna in maniera automatica;

5. Validazione delle funzionalità:

dopo una serie di test manuali è stato possibile confermare il corretto inserimento di ogni tipo di input utente consentito, *sono stati invece bloccati gli inserimenti privi di descrizione o con data di scadenza antecedente alla giornata odierna* (relativa a quando si va ad inserire);  
sfruttando molti elementi statici, che riducono l'interazione ad un semplice click dell'utente è stato possibile disabilitare combinazioni pericolose di input;

6. Adattabilità:

la schermata occupa una piccola parte dello schermo, rimanendo però leggibile e adattandosi ad ogni device con uno schermo pari o superiore ad otto pollici.

[Il controllo degli input utenti è poi presente ovunque l'utente è libero di immettere stringhe di testo]

• **Nota per l'installazione e l'utilizzo:**

Il sistema è stato realizzato tramite progetto Maven e precisamente:

- compilato con **openjdk-18.0.1**;
- interfaccia con **javaFX-11.0.1**;
- connettore al db MySQL **mysql.connector 8.0.29**;
- integrazione PDF con **itextpdf-5.5.13.3** ed **org.apache.pdfbox-2.0.26**;
- testing J-Unit con **junit-4.13.2**.

È stato usato anche il seguente *virtual machine argument*:

--module-path





```
yourPath\javafx-sdk-18.0.1\lib  
--add-modules=javafx.controls,javafx.fxml
```

Per quanto riguarda il DataBase la scelta è ricaduta, per abitudine personale, su MySQL.  
Le credenziali sono user -> *root*, psw -> *root*.  
Ho utilizzato molto il tool *MySQLWorkbench* nella versione *8.0*.

All'interno delle directory fornite (precisamente in *src/main/resources/db\_dump*) si trova un dump dello Schema, da utilizzare come sample.

Lista di utenti abilitati: *[la password decriptata per tutti gli utenti è **Modificala**]*

[Utilizza la funzione anonima di criptazione per creare una password da inserire nel DB]

[Il primo utente nella lista ha già espresso i voti nel sample del DB]

## Appendice:

Sono state utilizzate le seguenti risorse aggiuntive:

- [StarUML](#) in fase di progettazione;
- [VisualParadigm Online](#) per alcuni diagrammi;
- [DBDiagrams](#) per il diagramma del database;
- [Eclipse IDE](#) nelle fasi iniziali di implementazione;
- [IntelliJ IDEA](#) nelle fasi intermedie e finali di implementazione;
- [Github](#) per permettermi di lavorare da diversi dispositivi più facilmente.

Di seguito riporto il Code Coverage di una esecuzione in cui si effettua il login come Admin e si eseguono tutte le operazioni tranne l'aggiunta di voti:

Current scope: all classes			
Overall Coverage Summary			
Package	Class, %	Method, %	Line, %
all classes	76,5% (26/34)	52% (131/252)	37,2% (725/1948)
Coverage Breakdown			
Package	Class, %	Method, %	Line, %
<empty package name>	100% (1/1)	100% (3/3)	81,8% (9/11)
controllers	66,7% (8/12)	55,8% (43/77)	38,6% (289/749)
data	85,7% (6/7)	50,5% (47/93)	35,9% (289/804)
models	100% (10/10)	63,3% (31/49)	40,6% (112/276)
test.models	0% (0/2)	0% (0/9)	0% (0/28)
test.util	0% (0/1)	0% (0/10)	0% (0/27)
util	100% (1/1)	63,6% (7/11)	49,1% (26/53)

generated on 2022-07-18 10:08

Qui invece riporto il Code Coverage di un'esecuzione in cui si effettua il login come *Elettore*:



Current scope: all classes

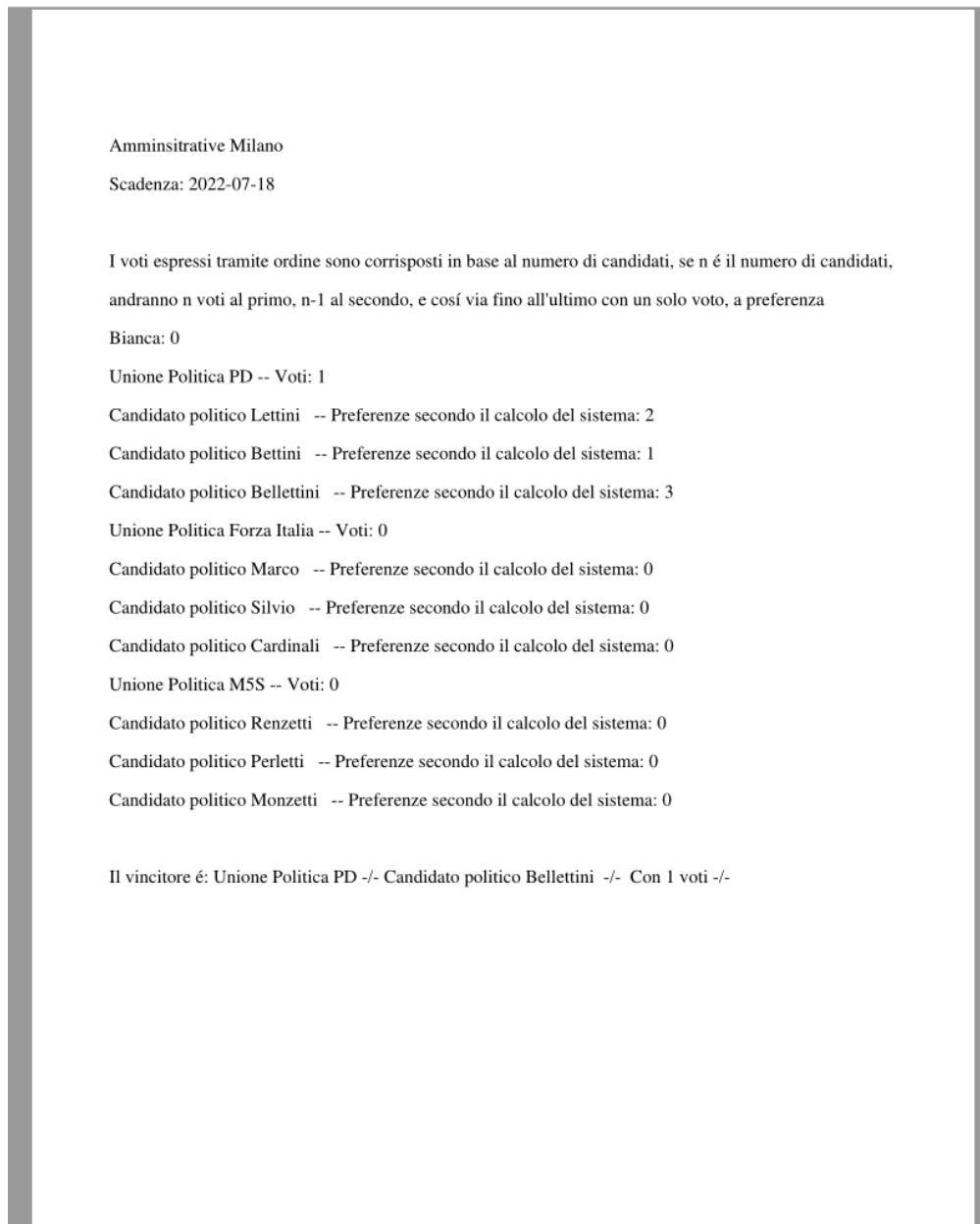
Overall Coverage Summary			
Package	Class, %	Method, %	Line, %
all classes	61,8% (21/34)	38,1% (96/252)	24,7% (482/1948)

Coverage Breakdown			
Package	Class, %	Method, %	Line, %
<empty package name>	100% (1/1)	100% (3/3)	81,8% (9/11)
controllers	41,7% (5/12)	31,2% (24/77)	26,3% (197/749)
data	71,4% (5/7)	37,6% (35/93)	22,5% (181/804)
models	90% (9/10)	55,1% (27/49)	25% (69/276)
test.models	0% (0/2)	0% (0/9)	0% (0/28)
test.util	0% (0/1)	0% (0/10)	0% (0/27)
util	100% (1/1)	63,6% (7/11)	49,1% (26/53)

generated on 2022-07-18 10:16

Riporto un esempio di file PDF generato per i risultati:



## Note sulle versioni rilasciate:

- **Versione 1.0:**

Il documento è stato creato includendo una breve descrizione iniziale del sistema, una prima definizione dei requisiti ed infine una prima definizione del glossario.



- **Versione 1.1:**

Al documento è stato aggiunto il diagramma dei casi d'uso, con relativo approfondimento, il diagramma delle classi e alcuni diagrammi di sequenza.

- **Versione 1.2:**

Il documento è stato riadattato alle modifiche progettuali e vengono recuperati solo alcuni punti cardine posti in precedenza.

- **Versione 1.3:**

Al documento sono stati aggiunte le note per l'installazione e l'appendice.