

Programmazione Java Avanzata

2024/25



Wordageddon

Gruppo 4

Cetraro Michele
Giachetta Corrado
Grimaldi Simone
Rodia Pasquale

0612707419
0612708054
0612707338
0612708053

Contents

1 General Description	2
2 Requirements Analisys	2
3 Use Case Model	4
3.1 Use Case Diagram	4
3.1.1 Diagrams Description	5
3.2 Use Cases Description	5
3.2.1 Registrazione Utente	5
3.2.2 Login Utente	5
3.2.3 Login Admin	6
3.2.4 Gioca	6
3.2.5 Visualizza Profilo	7
3.2.6 Logout	8
4 Design	9
4.1 Architectural Style	9
4.2 Package Diagram	9
4.3 Class Diagrams	10
4.3.1	10
4.3.2	11
4.3.3	12
4.3.4	13
4.4 Wireframe	14
4.5 Wireflow	16
5 Process View	17
5.1 Sequence Diagrams	17
5.1.1 SignUp	17
5.1.2 User Login	18
5.1.3 Admin Login	19
5.1.4 Adding Documents	20
5.1.5 Home - Profile visualization	21
5.1.6 Home - Start Match	22
5.1.7 Logout	23
6 Conclusion after implementation	24
6.1 Main Features	24
6.2 Game Rules	24
6.3 Limitations	25
6.4 Final Considerations	25

1 General Description

Wordageddon è un'applicazione interattiva sviluppata in Java che ha l'obiettivo di stimolare la memoria testuale dell'utente attraverso quiz generati su base statistica a partire da uno o più documenti. L'utente ha un tempo limitato per leggere i testi forniti, dopodiché deve rispondere a una serie di domande a risposta multipla che ne valutano la capacità di ricordare informazioni legate alla frequenza e presenza delle parole nei documenti.

Il sistema supporta più livelli di difficoltà, i quali influenzano parametri come la lunghezza dei testi, il numero di documenti visualizzati e il tempo disponibile per la lettura. Le domande, generate dinamicamente a partire dall'analisi testuale, comprendono confronti tra frequenze di parole, identificazione di termini assenti, e riconoscimento di parole all'interno di specifici documenti.

L'applicazione è multiutente: ciascun utente può registrarsi, effettuare il login, partecipare a sessioni di gioco e consultare i propri risultati. I punteggi sono salvati in un database relazionale e sono disponibili statistiche personali. È previsto inoltre un pannello amministrativo per il caricamento di nuovi documenti e liste di stopwords, al fine di personalizzare il materiale testuale e ottimizzare l'analisi semantica.

2 Requirements Analysis

- UI-1.1: Interfaccia di tipo GUI
- BF-1.1: Gestione utenti
- BF-1.2: Registrazione utente (Sign-up)
- BF-1.3: Login utente (Sign-in)
- UI-1.2: Visualizzazione pannello amministratore
- DF-1.1: Inserimento di un nuovo file testuale
- DF-1.2: Analisi semantica/statistica dei documenti (frequenze, stopwords)
- DF-2.1: Eliminazione di un file testuale esistente
- DF-3.1: Gestione dati utente (credenziali, profilo, punteggi)
- UI-1.3: Visualizzazione profilo utente (statistiche, cronologia partite)
- UI-1.4: Visualizzazione leaderboard utenti
- IF-1.1: Scelta della difficoltà della partita
- IF-2-1: Selezione della lingua del documento

- IF-3.1: Avvio di una nuova partita
- DF-4.1: Generazione dinamica delle domande
- IF-4.1: Scadenza tempo lettura e inizio automatico quiz
- UI-1.5: Visualizzazione dei testi con cronometro
- UI-1.6: Visualizzazione delle domande a risposta multipla
- UI-1.7: Visualizzazione punteggio e risultato partita
- DF-5.1: Salvataggio dati partita (risposte, punteggio, livello)
- IS-1.1: Persistenza dati su database relazionale
- IS-1.2: Persistenza documenti e stopwords

Requirements	Business Value	Technical Risk	Requirement Priority
UI-1.1	Must Have	Medium	Priority 1
BF-1.1	Must Have	Low	Priority 1
BF-1.2	Must Have	Low	Priority 1
BF-1.3	Must Have	Low	Priority 1
UI-1.2	Must Have	Low	Priority 1
DF-1.1	Must Have	Low	Priority 1
DF-1.2	Must Have	Medium	Priority 1
DF-2.1	Nice to Have	Medium	Priority 3
DF-3.1	Must Have	Medium	Priority 1
UI-1.3	Should Have	Medium	Priority 2
UI-1.4	Nice to Have	High	Priority 3
IF-1.1	Must Have	Low	Priority 1
IF-2.1	Nice to Have	High	Priority 3
IF-3.1	Must Have	Low	Priority 1
DF-4.1	Must Have	Medium	Priority 1
IF-3.1	Must Have	Low	Priority 1
UI-1.5	Must Have	Medium	Priority 1
UI-1.6	Must Have	Low	Priority 1
UI-1.7	Should Have	Medium	Priority 2
DF-5.1	Should Have	Medium	Priority 2
IS-1.1	Must Have	High	Priority 1
IS-1.2	Must Have	High	Priority 1

3 Use Case Model

3.1 Use Case Diagram

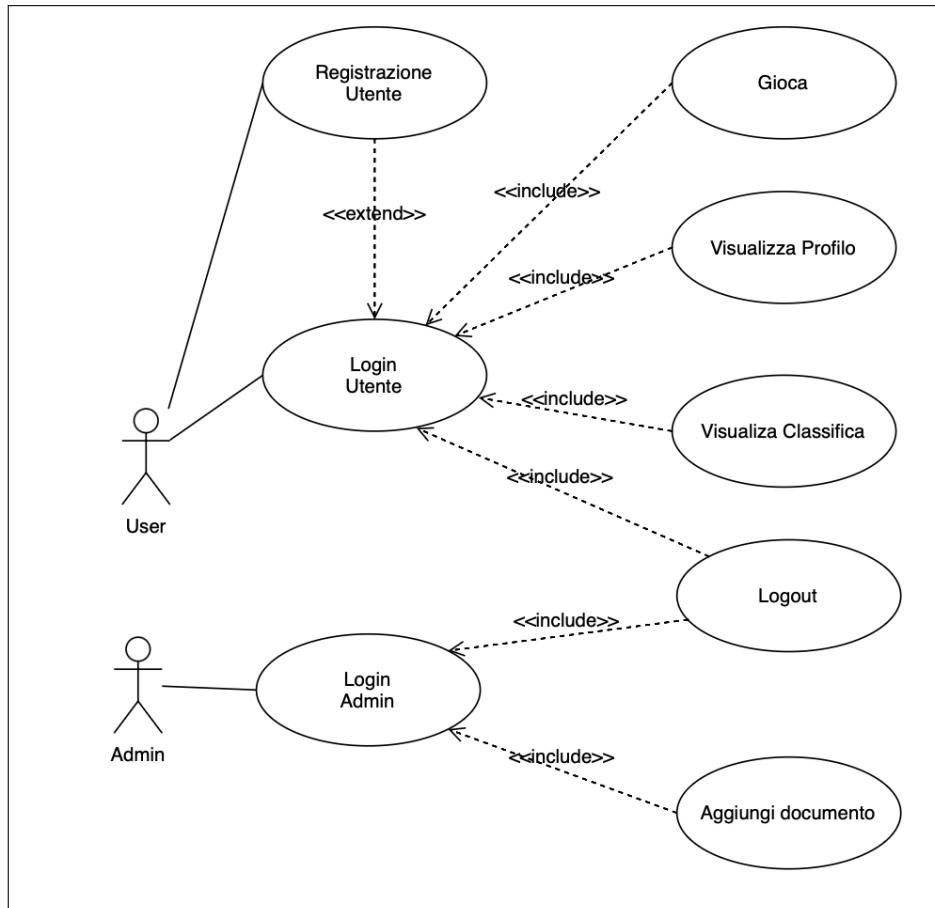


Figure 1: Diagramma dei casi d'uso

3.1.1 Diagrams Description

Il diagramma mostrato in Figura 1 mappa le interazioni dell'utente con il sistema descrivendone i main flows. Non sono presenti flussi alternativi, poiché tutte le possibili azioni dei partecipanti ai casi d'uso sono chiaramente descritti nella Figura 1.

3.2 Use Cases Description

3.2.1 Registrazione Utente

Nome: Registrazione utente

Attori partecipanti: User

Precondizioni: User visualizza correttamente l'interfaccia grafica e decide di registrarsi

Postcondizioni: User si registra correttamente e viene reindirizzato all'interno dell'applicazione

Flusso di eventi:

1. User clicca il tasto "Sign-Up" sull'interfaccia grafica iniziale
2. User compila i campi "Username" e "Password"
3. User clicca il tasto "Sign-Up"

Flussi alternativi:

3a User inserisce uno username già esistente

3a.1 User ricomincia dal punto 2 di Registrazione Utente

3b User clicca il tasto "Back" sull'interfaccia grafica

3b.1 User viene reindirizzato al punto 1 di "Login Utente"

3.2.2 Login Utente

Nome: Login Utente

Attori partecipanti: User

Precondizioni: User visualizza correttamente l'interfaccia grafica iniziale e decide di voler effettuare il login

Postcondizioni: User effettua correttamente il login e viene reindirizzato all'interno dell'applicazione

Flusso di eventi:

1. User compila i campi "Username" e "Password"
2. User clicca il tasto "Log-in"

Flussi alternativi:

- 3a User inserisce username o password errati.
 - 3a.1 User ricomincia dal punto 1 di Login Utente

3.2.3 Login Admin

Nome: Login Admin

Attori partecipanti: Admin

Precondizioni: Admin visualizza correttamente l'interfaccia grafica iniziale e decide di effettuare il Login

Postcondizioni: Admin effettua correttamente il login e viene reindirizzato nel pannello amministrativo

Flusso di eventi:

1. Admin compila i campi "Username" e "Password"
2. Admin clicca il tasto "Log-in"

Flussi alternativi:

- 3a Admin inserisce username o password errati
 - 3a.1 Admin ricomincia dal punto 1 di Login Admin

Note: Le credenziali Admin permettono di accedere solo alla sezione dedicata alla modifica dei documenti; un Admin non ha la possibilità di giocare con il proprio account.

3.2.4 Gioca

Nome: Gioca

Attori partecipanti: User

Precondizioni: User ha effettuato il login e visualizza correttamente l'interfaccia grafica

Postcondizioni: L'user completa una sessione di gioco

Flusso di eventi:

1. User seleziona la difficoltà
2. User avvia la sessione di gioco
3. User visualizza uno o più testi ed un timer
4. Il timer scade
5. User viene reindirizzato alla sezione dedicata alle domande
6. User clicca il tasto "Finish" per completare il quiz

7. User visualizza il punteggio ottenuto
8. User è reindirizzato alla schermata principale

Flussi alternativi:

- 1a User seleziona l'opzione **Logout**, e viene reindirizzato al punto 1 di Logout
- 2a User visualizza un messaggio di errore relativo alla mancanza di testi
 - 2a.1 User viene reindirizzato alla schermata home
- 3a User ha la possibilità di accedere alla sezione dedicata alle domande prima che scada il timer

Flussi alternativi che possono essere eseguiti in qualsiasi momento del caso d'uso Gioca:

- User può lasciare in qualunque momento la partita.
- User viene reindirizzato al punto 8 di Gioca

3.2.5 Visualizza Profilo

Nome: Visualizza Profilo

Attori partecipanti: User

Precondizioni: User ha effettuato il login e visualizza correttamente l'interfaccia grafica

Postcondizioni: User visualizza le informazioni e le statistiche relative al suo profilo

Flusso di eventi:

1. User accede alla sezione "Profilo"
2. User visualizza le statistiche e i dati relativi allo storico delle sue partite
3. User può visualizzare la classifica globale dove sono presenti i punteggi migliori di tutti gli utenti
4. User può visualizzare la sua posizione nella classifica globale

Flussi alternativi che possono essere eseguiti in qualsiasi momento del caso d'uso Gioca:

- User può scegliere in qualsiasi momento di effettuare il **Logout**
- User viene reindirizzato al punto 1 di Logout

3.2.6 Logout

Nome: Logout

Attori partecipanti: User

Precondizioni: User si trova in una qualsiasi posizione all'interno dell'applicazione"

Postcondizioni: User è disconnesso

Flusso di eventi:

1. User clicca il tasto Logout

Note: Si può accedere a questo caso d'uso a partire da qualsiasi caso d'uso in cui l'utente ha effettuato correttamente il Login

Nome: Aggiungi Documento

Attori partecipanti: Admin

Precondizioni: Admin ha effettuato il login e visualizza correttamente la schermata dedicata all'amministratore

Postcondizioni: I documenti vengono modificati correttamente

Flusso di eventi:

1. Admin seleziona l'opzione "Aggiungi Documento"
2. Admin aggiunge un documento in formato .txt
3. Admin fornisce una eventuale lista di StopWords
4. Admin elimina uno dei documenti presenti sull'applicazione
5. Admin conferma le operazioni eseguite

Flussi alternativi:

2a Il file non è in formato .txt

2a.1 Admin ricomincia dal punto 1 di Modifiche ai Documenti di gioco

Azioni sempre disponibili :

1. 1.1 Admin può cliccare il tasto **Undo** in qualunque momento, eliminando le modifiche effettuate fino a quel momento nella sessione corrente
 - 1.2 L'Admin ricomincia dal punto 1 di Modifiche ai Documenti di gioco
2. 2.1 Admin può in qualsiasi momento premere il tasto Logout
 - 2.2 Admin viene reindirizzato al punto 1 di Logout

4 Design

4.1 Architectural Style

Lo stile architetturale adottato è Model-View-Controller poiché particolarmente adatto per applicazioni con interfacce grafiche (GUI). Nello specifico il Model si costituisce della sezione di logica gestionale (gestione delle analisi e dei quiz), la View si compone dei file FXML che forniscono l'interfaccia grafica, con tutte le sue sotto-interfacce, e il Controller funge da intermediario regolando le interazioni dell'utente con l'interfaccia e la logica di background.

4.2 Package Diagram

Per rispettare il pattern Model-View-Controller (MVC), la struttura del progetto è stata organizzata in modo da separare chiaramente le responsabilità di ciascun componente (Model, View, Controller) :

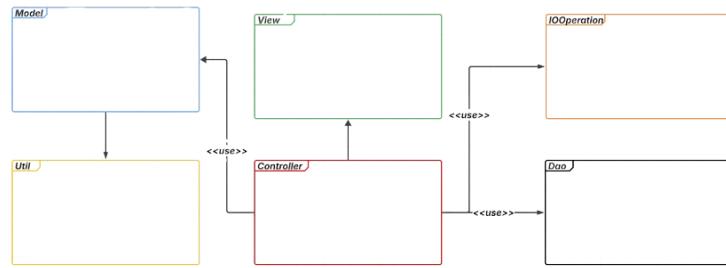


Figure 2: Diagramma dei package

- Package Model : è rappresentato dalle classi che gestiscono la logica del quiz e l'analisi dei dati.
- Package View : contiene tutte le viste dell'applicazione in .fxml e il relativo foglio di syile in .css.
- Package Controller : contiene l'insieme dei controller che gestiscono l'interazione tra la View e il Model.
- Package DAO : è responsabile per l'interazione con il database, ovvero l'estrazione e l'inserimento dei dati. Si occupa di astrarre l'accesso al database per separarlo dalla logica di business.
- Package Utility : contiene le classi utili al funzionamento dell'applicazione
- Package IOOperation : gestisce le operazioni di input/output (lettura e scrittura su file), permettendo di separare la logica di lettura/scrittura dai controller e dal modello.

4.3 Class Diagrams

In questa sezione sono riportati i diagrammi delle classi che descrivono la struttura e le relazioni tra gli oggetti nel progetto Java.

4.3.1

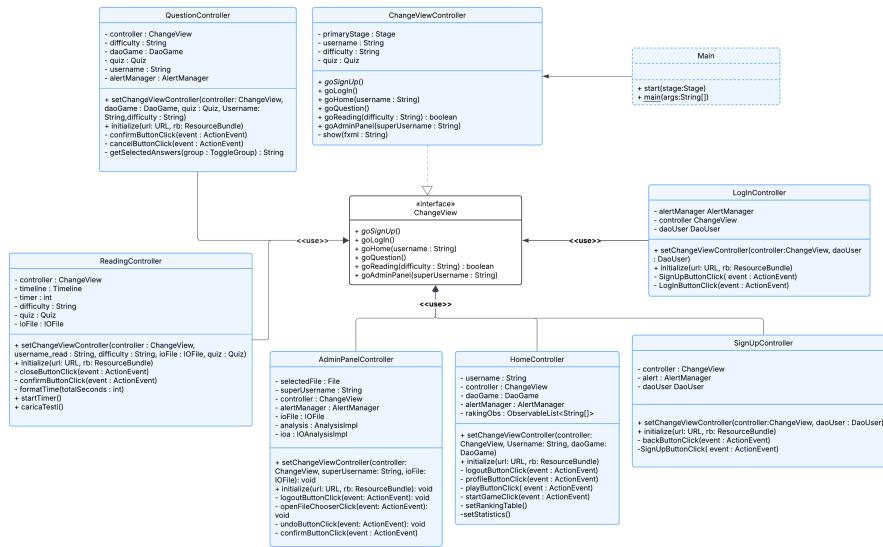


Figure 3: Diagramma delle classi - Controller

In figura sono rappresentati tutti i controller relativi alle view(.fxml) dell'applicazione, oltre al controller principale. Per seguire i principi di buona progettazione e le buone pratiche di sviluppo software, tutti i controller relativi alle viste non si occupano della gestione diretta degli oggetti o delle logiche aziendali bensì, questi oggetti gli sono passati (iniettati) tramite il metodo 'setChangeViewController' dal controller principale. Quanto definito rappresenta un buon approccio per ridurre il forte accoppiamento e migliorare la coesione delle classi, facilitando la manutenibilità, la testabilità e la flessibilità del software.

4.3.2

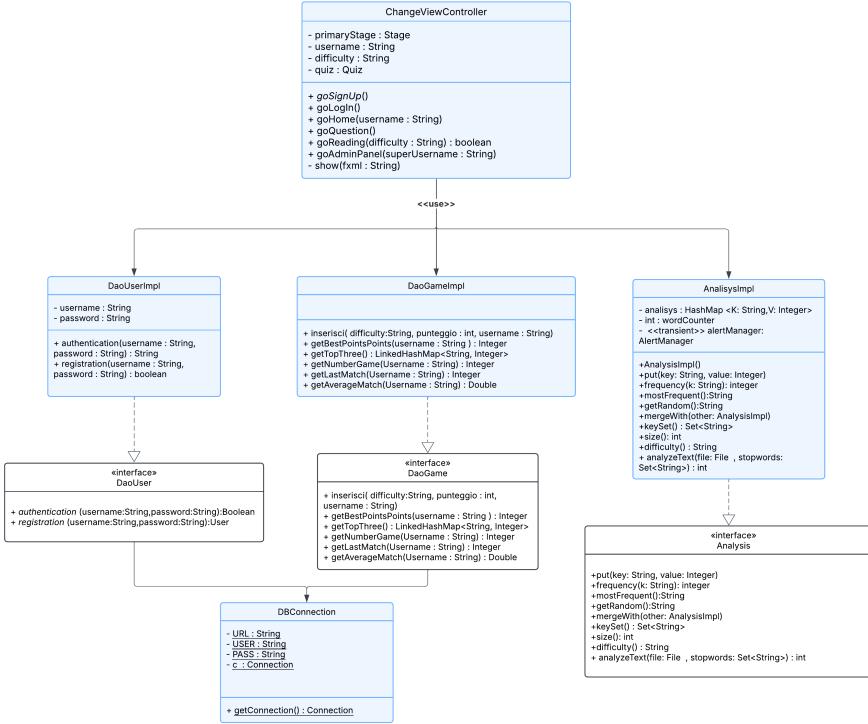


Figure 4: Diagramma delle classi - DAO e Analysis

L'accesso ai dati persistenti è gestito attraverso il pattern DAO, che consente di mantenere separata la logica applicativa dall'interazione con il database. La classe `DBConnection` si occupa di mantenere aperta e riutilizzare la stessa connessione per più operazioni consecutive, migliorando così le prestazioni e ottimizzando l'uso delle risorse. La classe `AnalysisImpl`, che implementa l'interfaccia `Analysis`, incapsula le informazioni relative alla frequenza delle parole presenti nei testi, rappresentandole in modo strutturato. Le componenti che necessitano di queste funzionalità non fanno riferimento direttamente alle classi concrete, ma si appoggiano sulle interfacce corrispondenti, garantendo così un elevato grado di modularità.

4.3.3

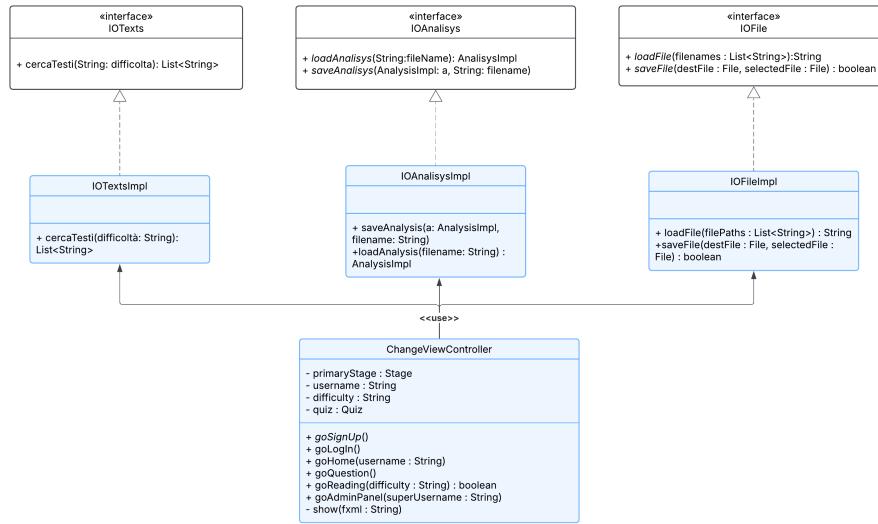


Figure 5: Diagramma delle classi - IOOperation

In figura sono rappresentate le classi e le relative interfacce utilizzate per tutte le operazioni di Input/Output (I/O) su e da file. Ogni controller, che necessita di interagire con i file, fa uso di queste interfacce, senza dover gestire direttamente le implementazioni concrete.

4.3.4

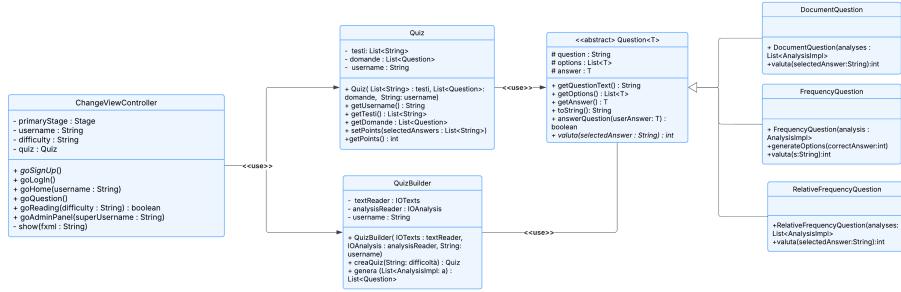


Figure 6: Diagramma delle classi - Quiz Generation

In figura sono rappresentate le classi utilizzate per la generazione dei quiz. Il ChangeViewController dipende da Quiz per accedere alle domande, ma non gestisce direttamente la logica di creazione delle domande, in quanto questa logica è delegata alla classe QuizBuilder, che a sua volta dipende da componenti come IOTexts e IOAnalysis per ottenere i dati necessari per le domande. Questo permette di separare la logica di business (la generazione delle domande e la gestione dei quiz) dalla logica di navigazione e interfaccia utente.

4.4 Wireframe

Di seguito sono riportati i wireframe delle view principali dell'applicazione.

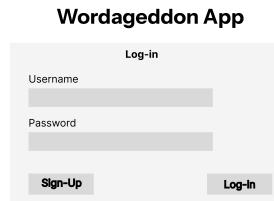


Figure 7: Login.fxml

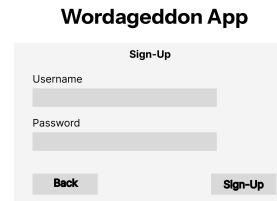


Figure 8: SignUp.fxml

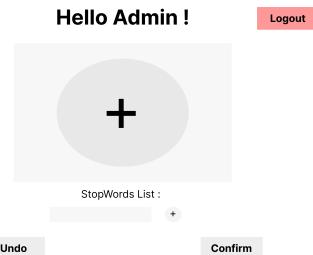


Figure 9: AdminPanel.fxml



Figure 10: Home.fxml

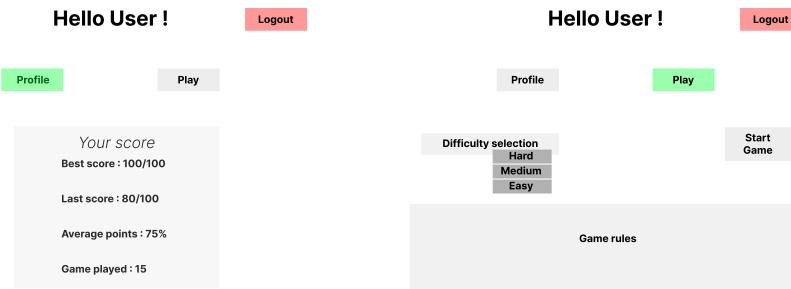


Figure 11: Home.fxml - Sezione Profile

Figure 12: Home.fxml - Sezione Play



Figure 13: Reading.fxml

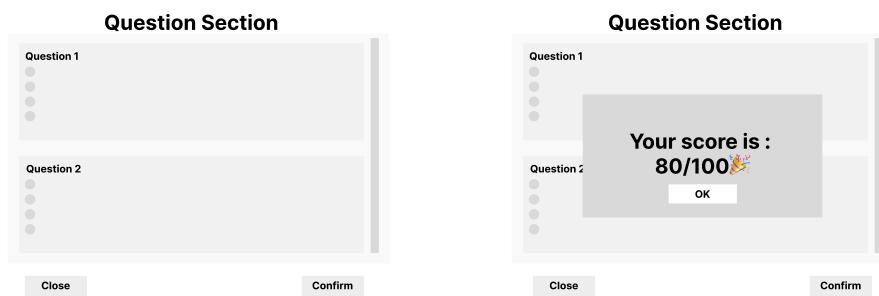


Figure 14: Question.fxml

Figure 15: Question.fxml

4.5 Wireflow

Di seguito è riportato il Wireflow dell'applicazione.

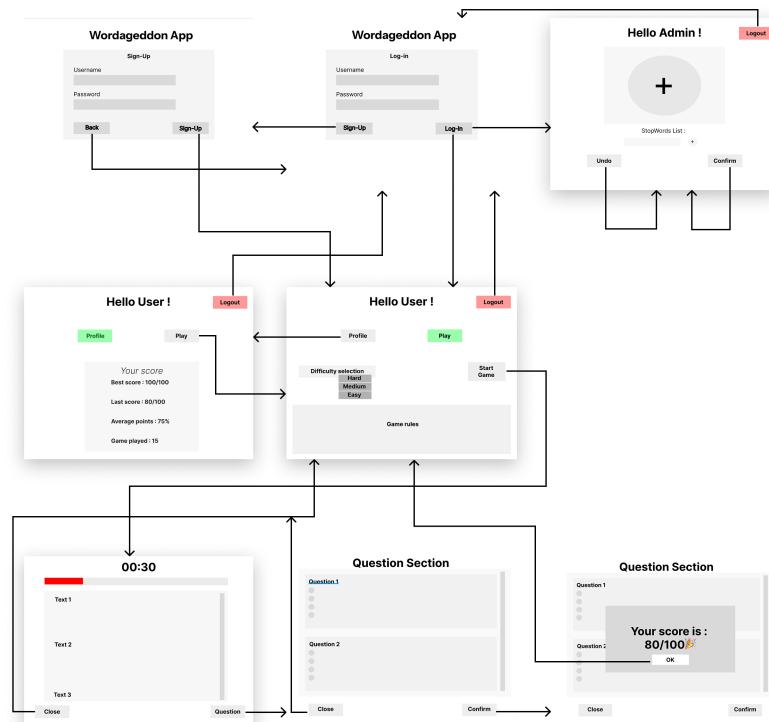


Figure 16: Wireflow

5 Process View

5.1 Sequence Diagrams

In questa sezione sono riportati i diagrammi di sequenza che mostrano l'interazione tra gli oggetti nel sistema durante l'esecuzione di specifici scenari.

5.1.1 SignUp

Il diagramma seguente descrive il flusso di operazioni per la registrazione di un nuovo utente nel sistema. Mostra il processo da quando l'User si trova nella pagina di Login e clicca sul bottone di 'SignUp' fino a quando la registrazione non è completata.

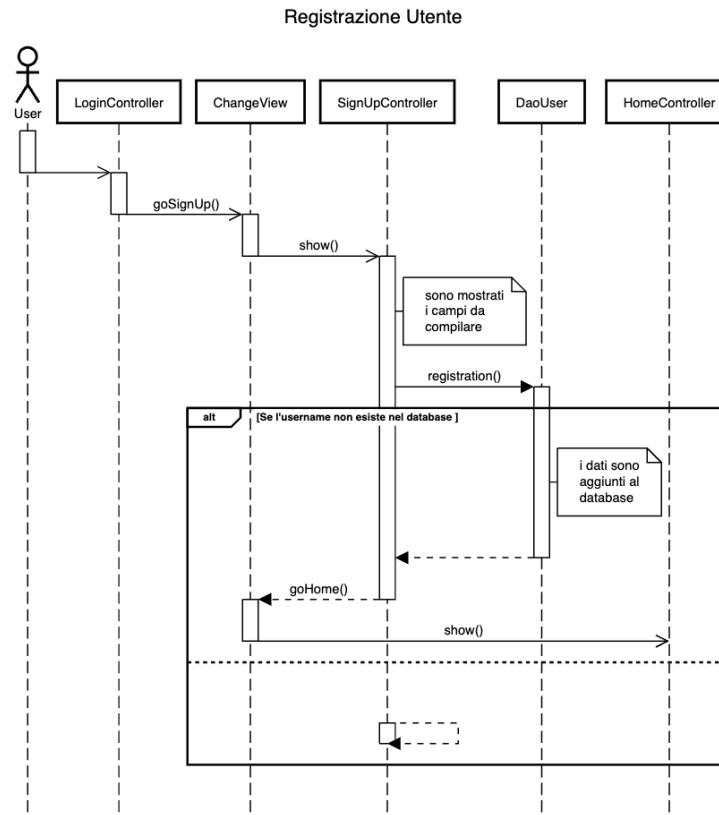


Figure 17: Diagramma di sequenza per la registrazione dell'utente

5.1.2 User Login

Il diagramma seguente descrive il flusso di operazioni che avvengono quando un utente tenta di accedere al sistema. Mostra il processo di verifica delle credenziali e l'accesso all'area riservata partendo da quando l'User inserisce le credenziali e clicca il bottone 'LoginIn' fino all'accesso alla schermata Home.

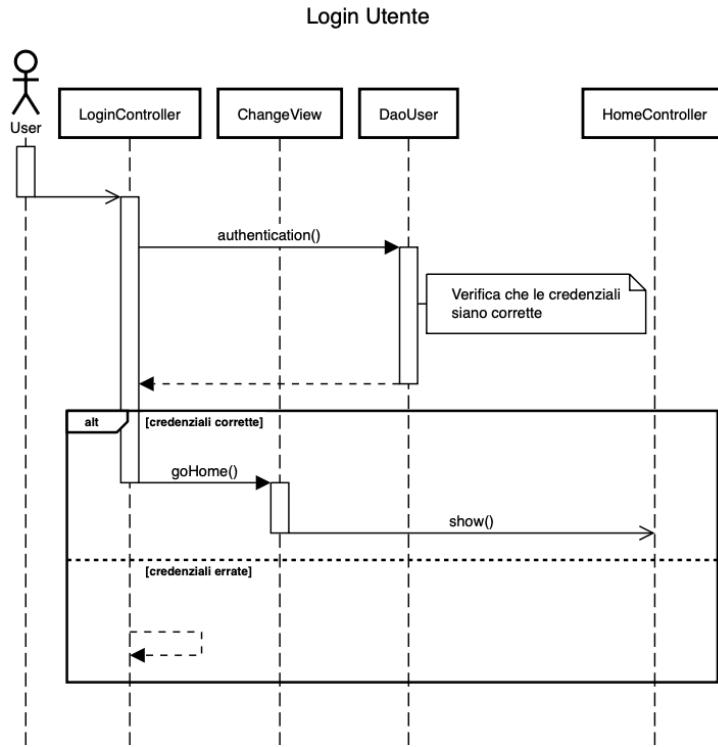


Figure 18: Diagramma di sequenza per l'autenticazione dell'utente

5.1.3 Admin Login

Il diagramma seguente descrive il flusso di operazioni che avvengono quando un amministratore tenta di accedere al sistema. Mostra il processo di verifica delle credenziali e l'accesso all'area riservata partendo da quando l'Admin inserisce le credenziali e clicca il bottone 'LogIn' fino all'accesso al pannello amministrativo.

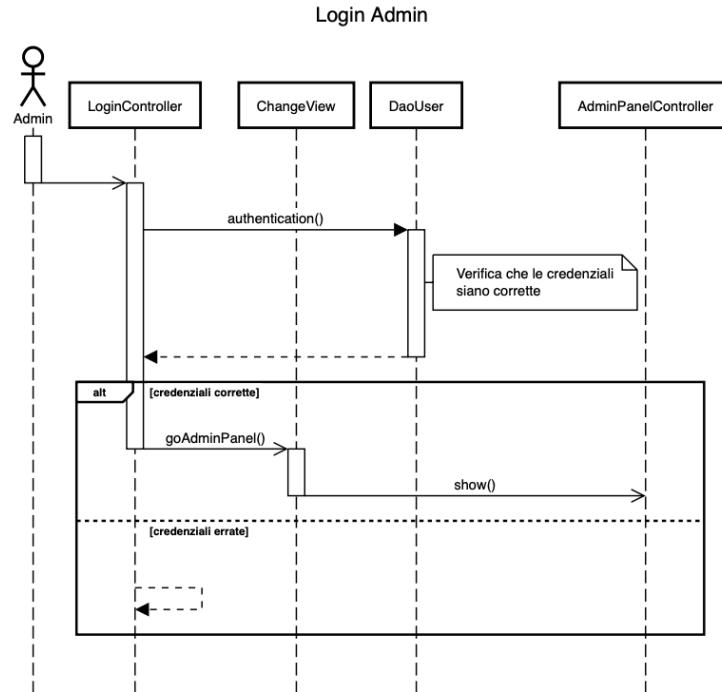


Figure 19: Diagramma di sequenza per l'autenticazione dell'amministratore

5.1.4 Adding Documents

Il diagramma seguente descrive il flusso di operazioni per l'aggiunta di un nuovo documento testuale all'applicazione. Mostra il processo, dal momento in cui l'admin è nella sua area riservata fino al momento in cui il file e la rispettiva analisi sono caricati e la pagina aggiornata.

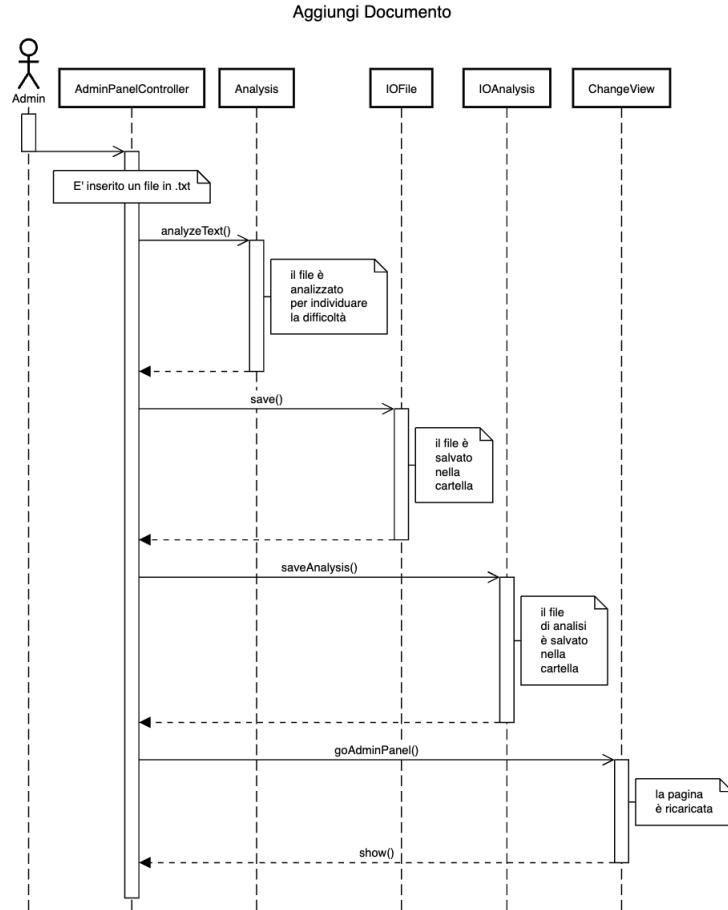


Figure 20: Diagramma di sequenza per l'aggiunta di un nuovo documento

5.1.5 Home - Profile visualization

Il diagramma seguente descrive il flusso di operazioni che avvengono quando un utente già autenticato richiede di visualizzare il proprio profilo.

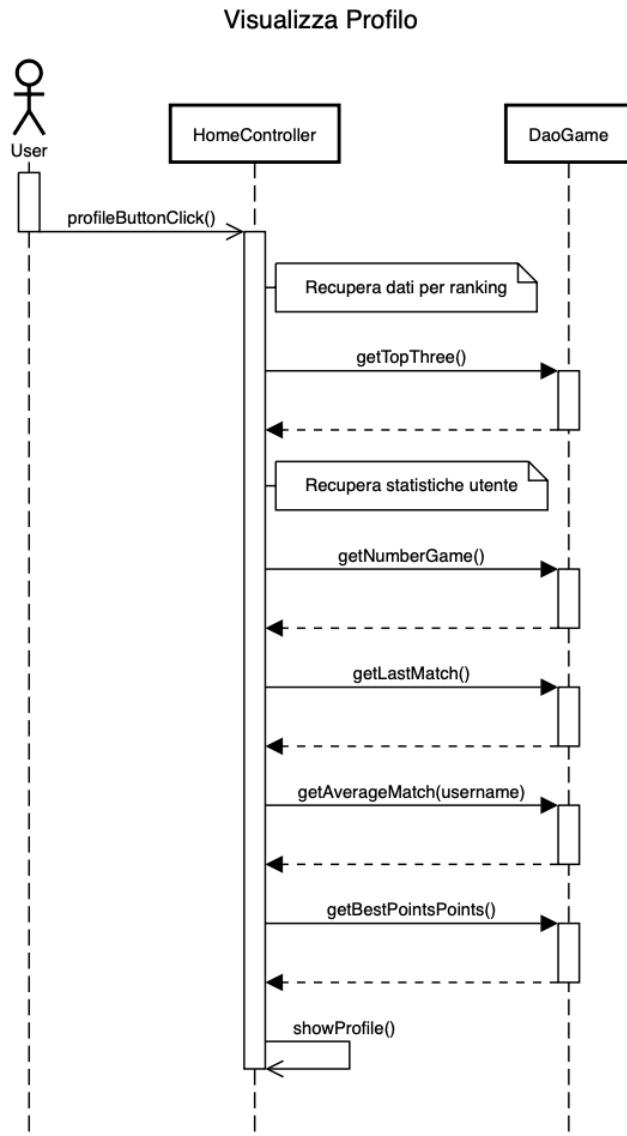


Figure 21: Diagramma di sequenza per il profilo

5.1.6 Home - Start Match

Il diagramma seguente descrive il flusso di operazioni successivamente alla richiesta di avvio di una nuova partita. Mostra il processo, partendo dal click dell'User sul bottone 'StartGame' fino ad arrivare al salvataggio della partita nel database e al successivo ritorno nella home.

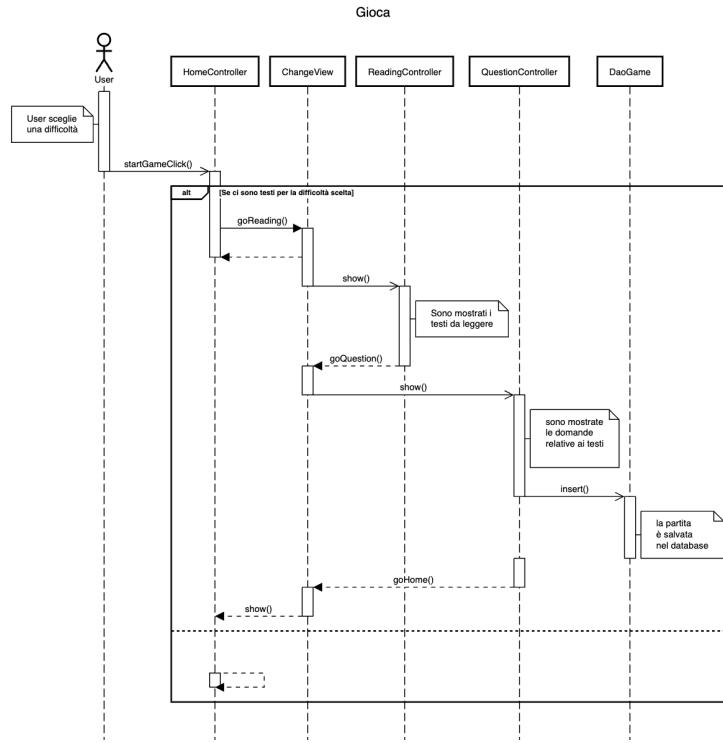


Figure 22: Diagramma di sequenza per l'avvio di una nuova partita

5.1.7 Logout

Il diagramma seguente descrive il flusso di operazioni relative che si susseguono quando un utente o un admin effettuano il logout.

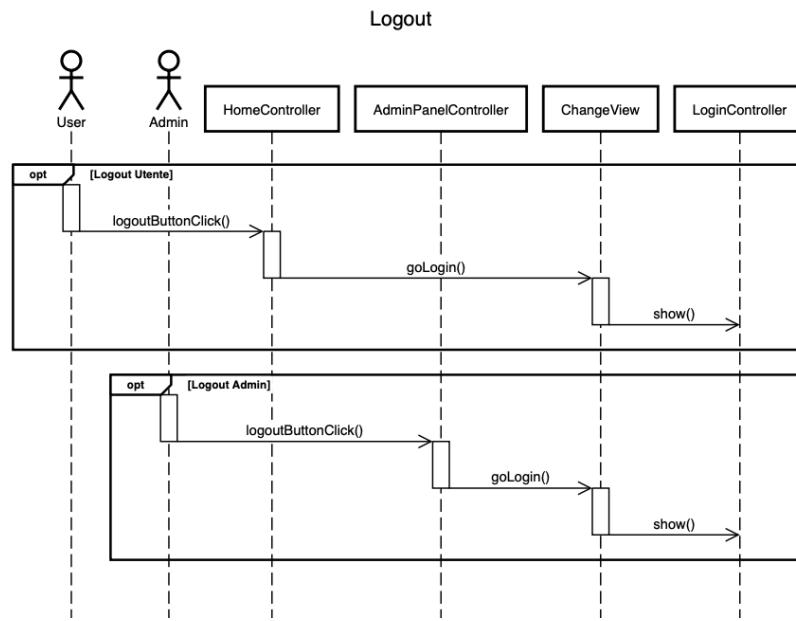


Figure 23: Diagramma di sequenza per il logout

6 Conclusion after implementation

L'applicazione sviluppata rispetta tutti i requisiti minimi richiesti e implementa diverse funzionalità aggiuntive al fine di migliorare l'esperienza dell'utente all'interno dell'applicazione. Di seguito vengono descritte le principali caratteristiche, i punti di forza e le aree di miglioramento.

6.1 Main Features

- **Requisiti minimi:** L'applicazione implementa tutte le funzionalità richieste.
- **Database:** Il sistema si avvale di un database online (hostato su Render), che, sebbene soddisfi le necessità dell'applicazione, può causare ritardi nelle operazioni di CRUD a causa dei tempi di latenza tipici di un database remoto.
- **Profilo Admin:** Il profilo dell'amministratore è preconfigurato e unico all'interno dell'applicazione. Le credenziali per l'accesso al profilo admin sono le seguenti:
 - Username: **gruppo4**
 - Password: **30L**
- **Funzionalità opzionali:** Sono state implementate diverse funzionalità opzionali che arricchiscono l'esperienza di gioco:
 - **Classifica globale:** Visualizzazione della classifica generale dei punteggi degli utenti.
 - **Progressi storici degli utenti:** Memorizzazione e visualizzazione dei progressi storici di ogni utente, con possibile visualizzazione degli stessi nella sezione Profilo.
 - **Gestione delle stopword:** L'amministratore ha la possibilità di inserire, all'atto dell'inserimento di un nuovo documento, nuove stopword, che verranno aggiunte (solo per il documento corrente) a quelle già presenti nel file '`analyticsFile/stopWordsList.txt`'.

6.2 Game Rules

Le regole del gioco variano in base alla difficoltà selezionata. Le modalità di gioco sono le seguenti:

- **Facile:** 1 testo, lunghezza massima di 250 parole, tempo di lettura di 5 minuti.
- **Media:** 2 testi, con lunghezza tra 250 e 750 parole ciascuno, tempo di lettura di 7 minuti.

- **Difficile:** 3 testi, con lunghezza tra 750 e 1000 parole ciascuno, tempo di lettura di 9 minuti.

6.3 Limitations

Sono stati introdotti dei controlli per garantire che il gioco possa essere avviato solo in determinate condizioni:

- Se non viene selezionato un livello di difficoltà, il gioco non può essere avviato.
- Se non ci sono abbastanza testi disponibili per il livello di difficoltà scelto, il gioco non può essere avviato.

6.4 Final Considerations

L'applicazione si presenta come funzionale e completa per quanto riguarda i requisiti iniziali. Tuttavia, alcune aree potrebbero essere migliorate, come la gestione dei tempi di latenza del database. Nonostante ciò, il sistema risponde adeguatamente alle necessità degli utenti e permette un'esperienza di gioco coinvolgente e interattiva.