

# WORDAGEDDON

Programmazione Java Avanzata

GRUPPO 16:

ROSSELLA PALE - 0612707284

EMANUELE TOCCI - 0612707488

FRANCESCO LANZARA - 0612707459

CLAUDIA MONTEFUSCO - 0612707404



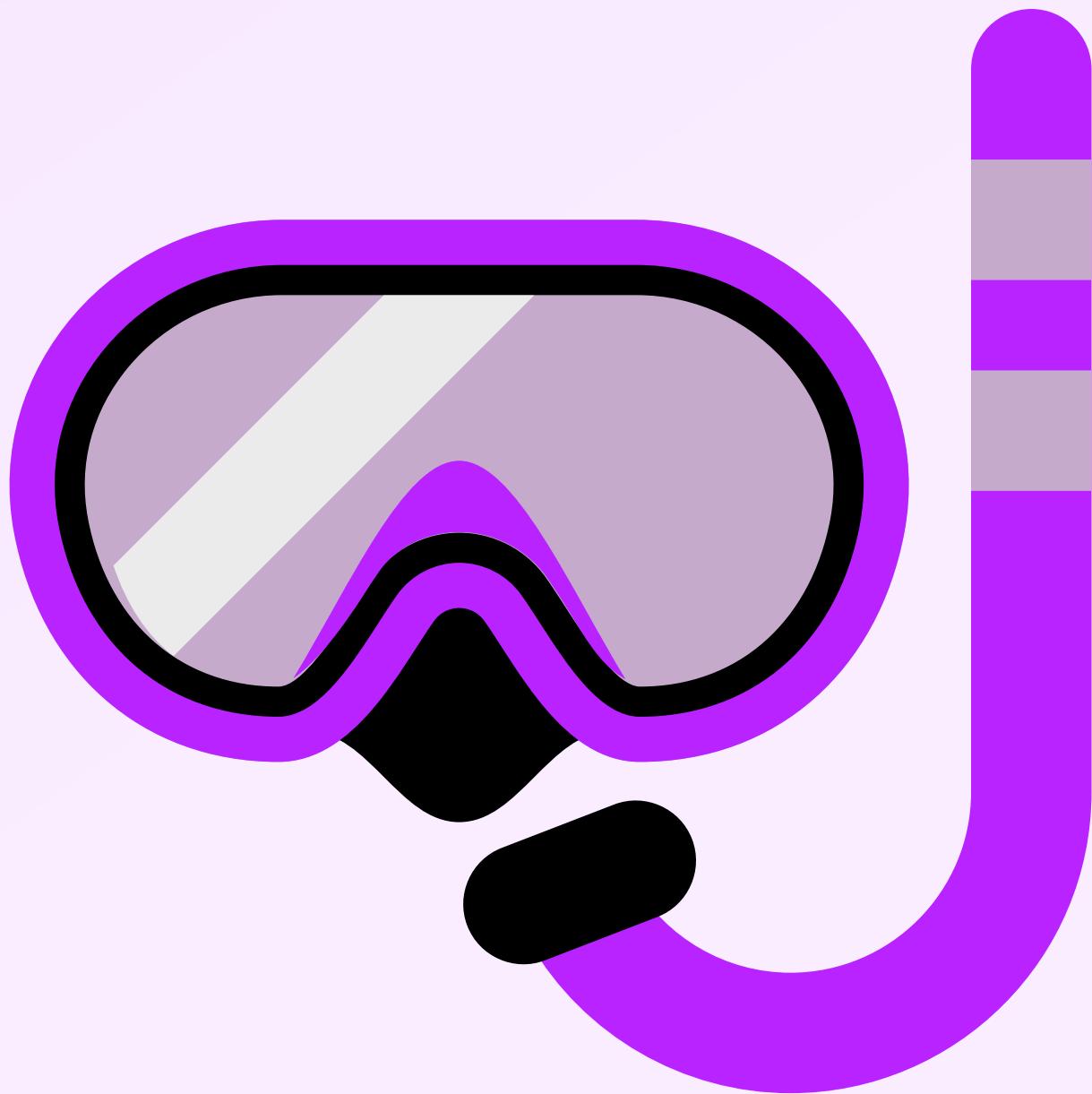
01

# Panoramica

Wordageddon è un gioco a **quiz** basato sulla memorizzazione di parole chiave presenti in documenti testuali.

L'utente legge uno o più documenti e risponde a domande sulle parole più significative.

L'obiettivo è stimolare la memoria e l'attenzione, con un sistema di punteggio che varia in base alla difficoltà.



# Fasi di sviluppo



Raccolta e analisi dei requisiti funzionali e non funzionali del sistema.

1. Diagramma delle classi per la struttura del codice.  
2. Diagramma di sequenza per visualizzare il flusso delle operazioni.

• Revisione dei diagrammi in base all'implementazione.  
• Ottimizzazione del codice e verifica finale.

**Creazione dei Mockup**  
Realizzazione delle interfacce utente per visualizzare l'esperienza finale

**Sviluppo del Codice**  
1. Implementazione del Database.  
2. Realizzazione delle funzionalità in java/javafx.

# Analisi dei requisiti

## Funzionalità individuali

- FASE DI LETTURA
  - FASE QUIZ
- SELEZIONE DELLA DIFFICOLTÀ
- AUTENTICAZIONE UTENTI
  - STOP WORDS
  - PRIVILEGI DI AMMINISTRATORE
- LEADERBOARD
- STATISTICHE POST-PARTITA
- GESTIONE SESSIONI INTERROTTE
- LOGGING

## Dati e formato dei dati

- PERSISTENZA DATI UTENTE

## Interfaccia Utente

- MENU
- PANNELLO UTENTE

## Requisiti non funzionali

- VARIETÀ DELLE DOMANDE
- STYLING CSS

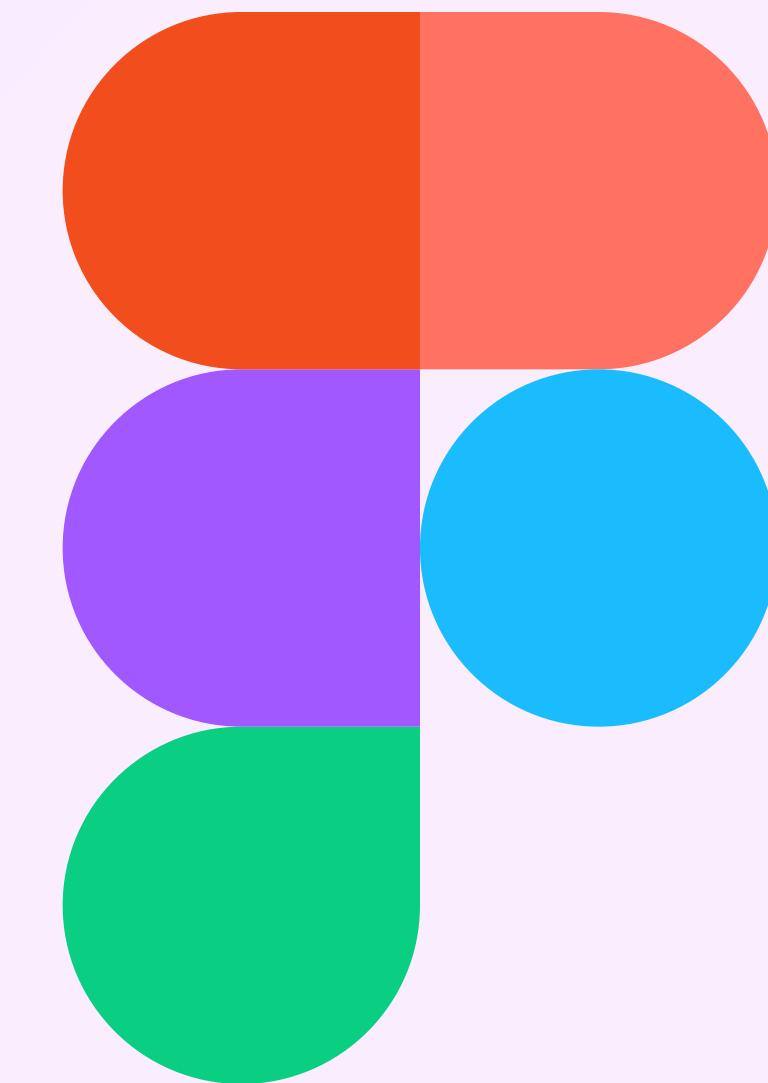
## Vincoli di progetto

- IMPLEMENTAZIONE IN JAVA/JAVAFX

# Creazione dei Mockup

Durante la fase di progettazione, abbiamo realizzato una serie di **mockup** funzionali che ci hanno guidato nello sviluppo delle interfacce.

- Schermata di Login / Registrazione
- Menu Principale
- View Lettura Documenti
- View Quiz
- Schermata di Statistiche Post-Partita
- Leaderboard View
- Pannello Utente



# Progettazione

Wordageddon adotta un'architettura modulare basata sui pattern **MVC e DAO**, con un **Service Layer centrale** per encapsulare la **logica di business** dell'applicazione.

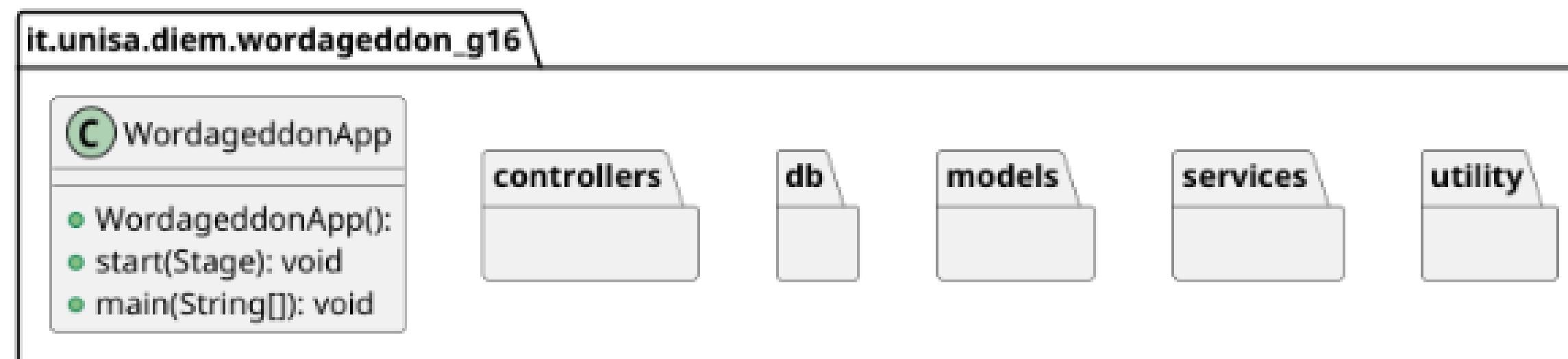
Obiettivo: separazione tra interfaccia, logica, accesso ai dati e supporti riutilizzabili.

## Model

Classi che rappresentano gli oggetti chiave (User, Document, GameReport...).

## Controllers

Gestiscono le UI, ponte tra l'interfaccia grafica (FXML) e la logica applicativa implementata nei service.



## Services

Questo livello media tra DAO e controller. Le classi encapsula operazioni complesse e logica di business.

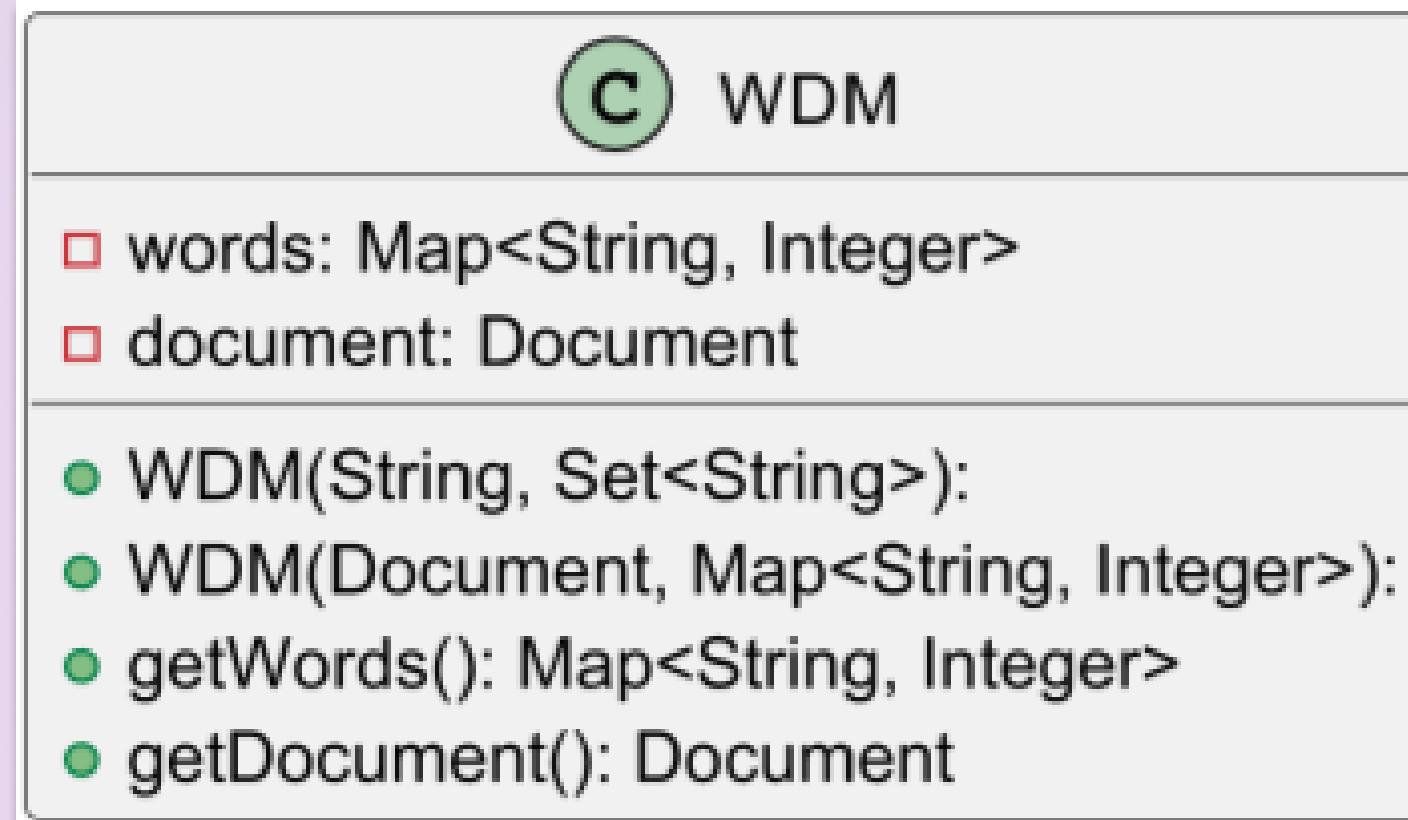
## DAO/Repository

Classi per l'accesso strutturato ai dati tramite SQLite.

## Utility

Classi di supporto (es. logger, popup, validazioni).

Document	word	occurrency



## Scelte Progettuali

Il funzionamento del programma e del gameplay si concentra su una struttura centrale chiamata **Word Document Matrix (WDM)**: una matrice che rappresenta la frequenza delle parole significative all'interno dei documenti disponibili nel sistema.

- Al caricamento di un documento, viene avviato un Thread che elabora il file in background, e crea la **WDM** associata.
- Quando cambiano le stopwords, il sistema **ricalcola** tutte le WDM, escludendo le parole inserite dagli admin. Il calcolo è effettuato in background su una serie di diversi threads.

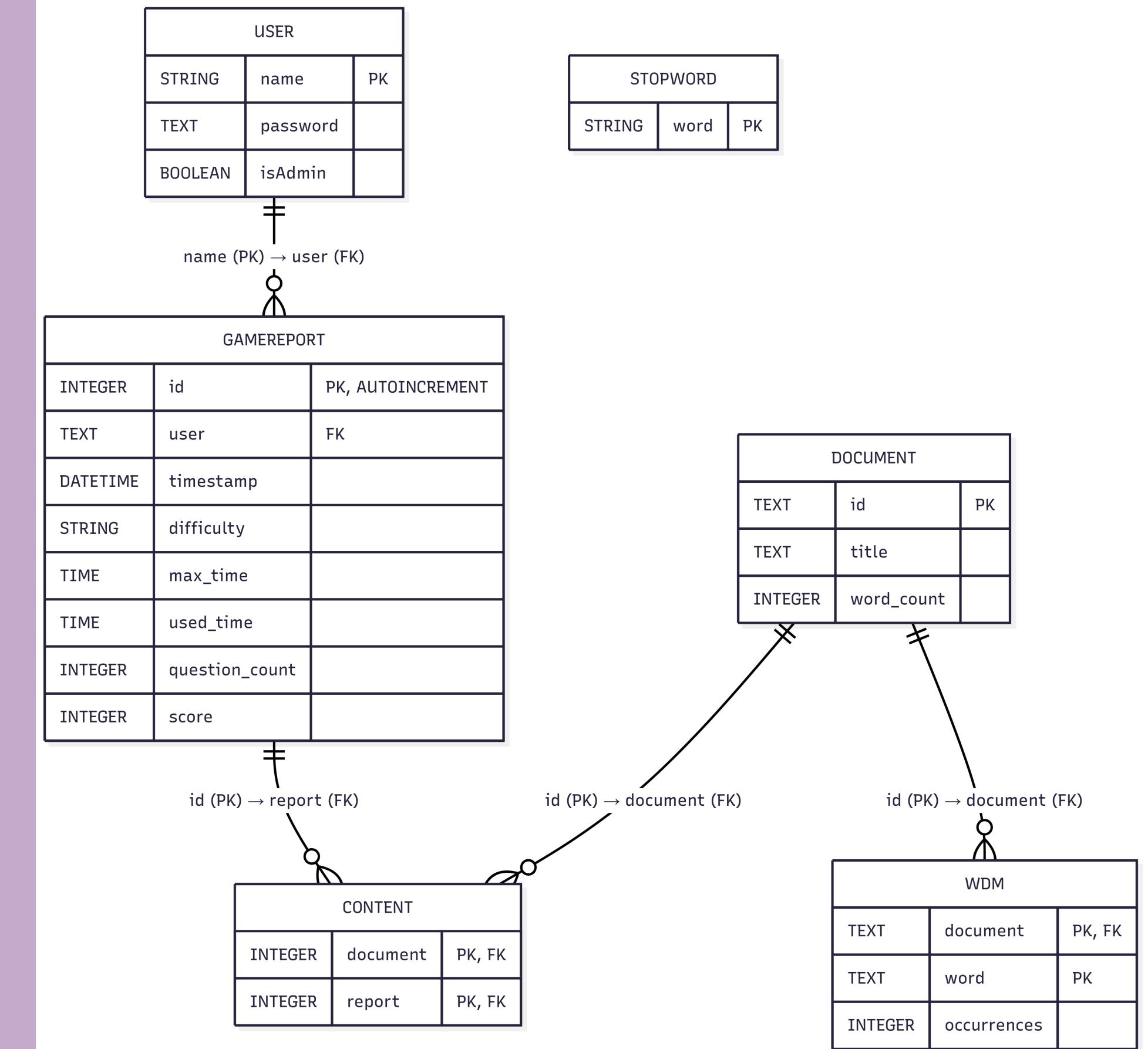
## Persistenza – Database & DAO

La logica di persistenza del progetto è stata gestita tramite una **gerarchia di DAO** (Data Access Object) basata su JDBC:

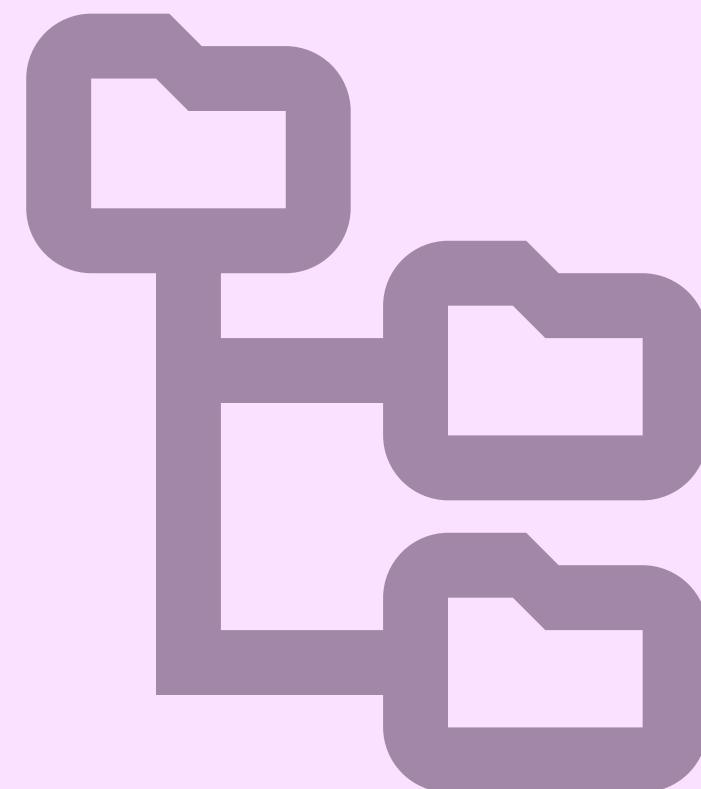
- Ogni entità (utente, documento, partita...) ha il proprio DAO specializzato.
- Tutti i DAO sono coordinati da un **JdbcRepository**, che funge da punto d'accesso centralizzato al database.

Il JdbcRepository:

- Condivide una **singola connessione** SQLite, gestita internamente.



# Persistenza - Serializzazione

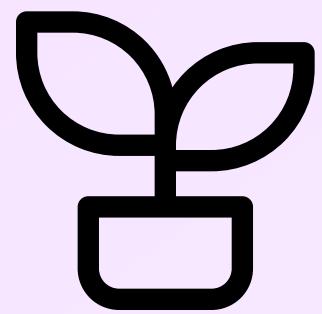


## Sessione corrente

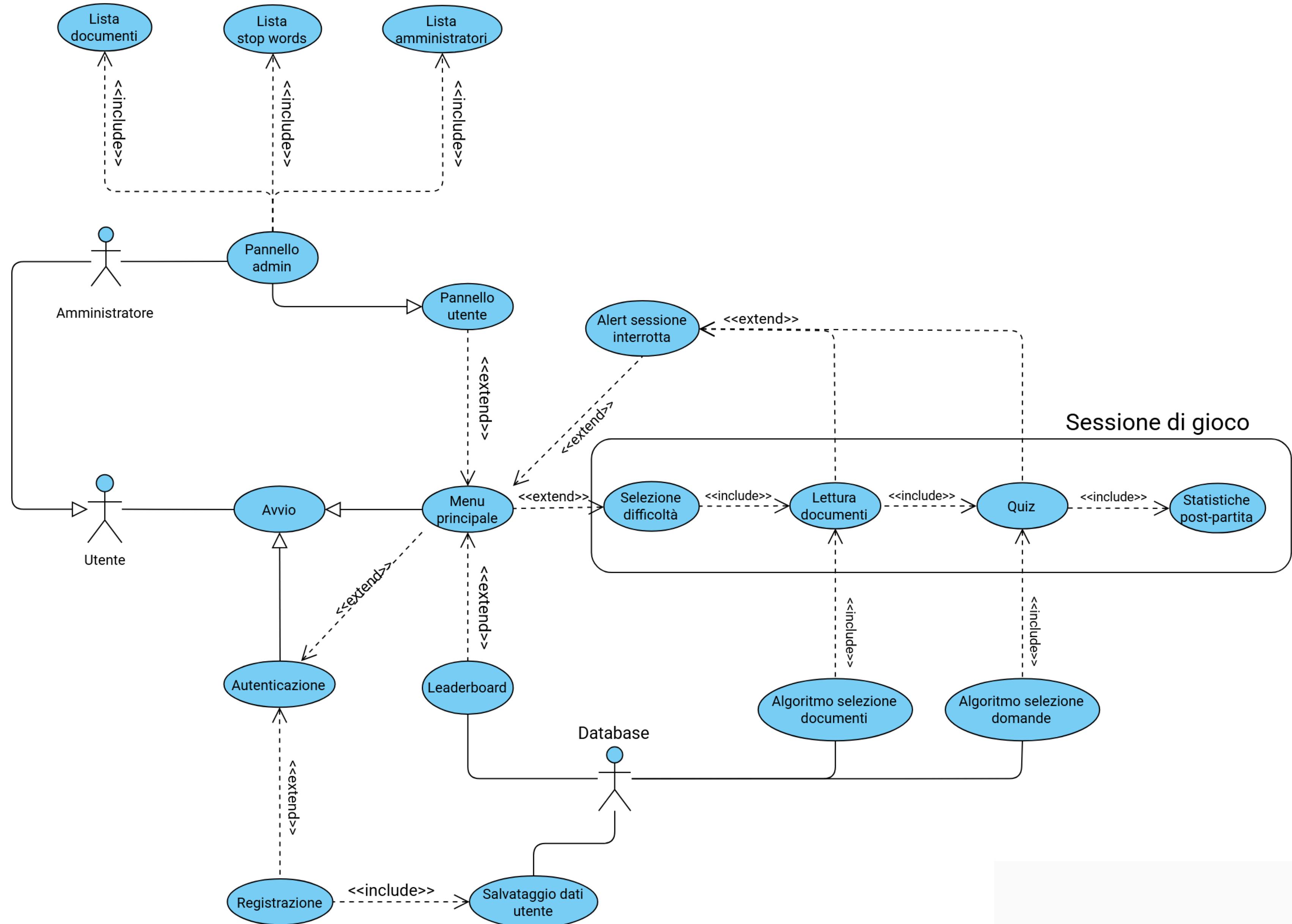
La sessione di gioco corrente è serializzata nel file "session.ser".

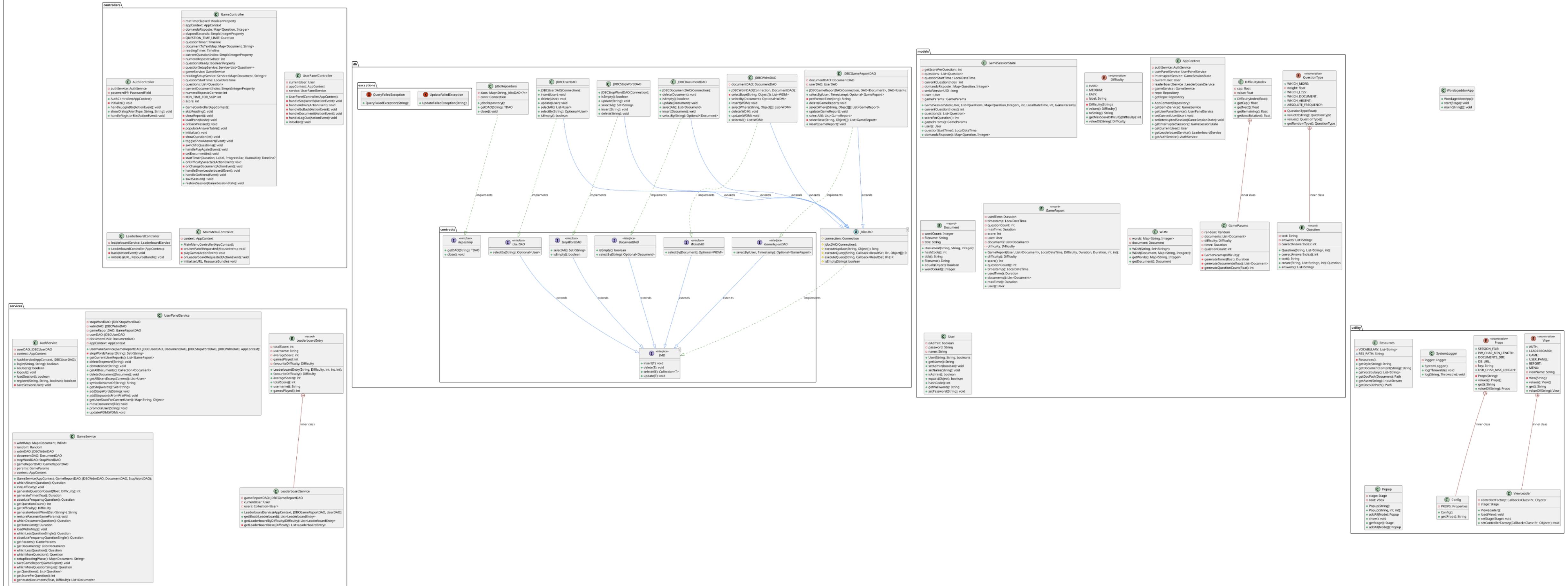
## Sessione Interrotta

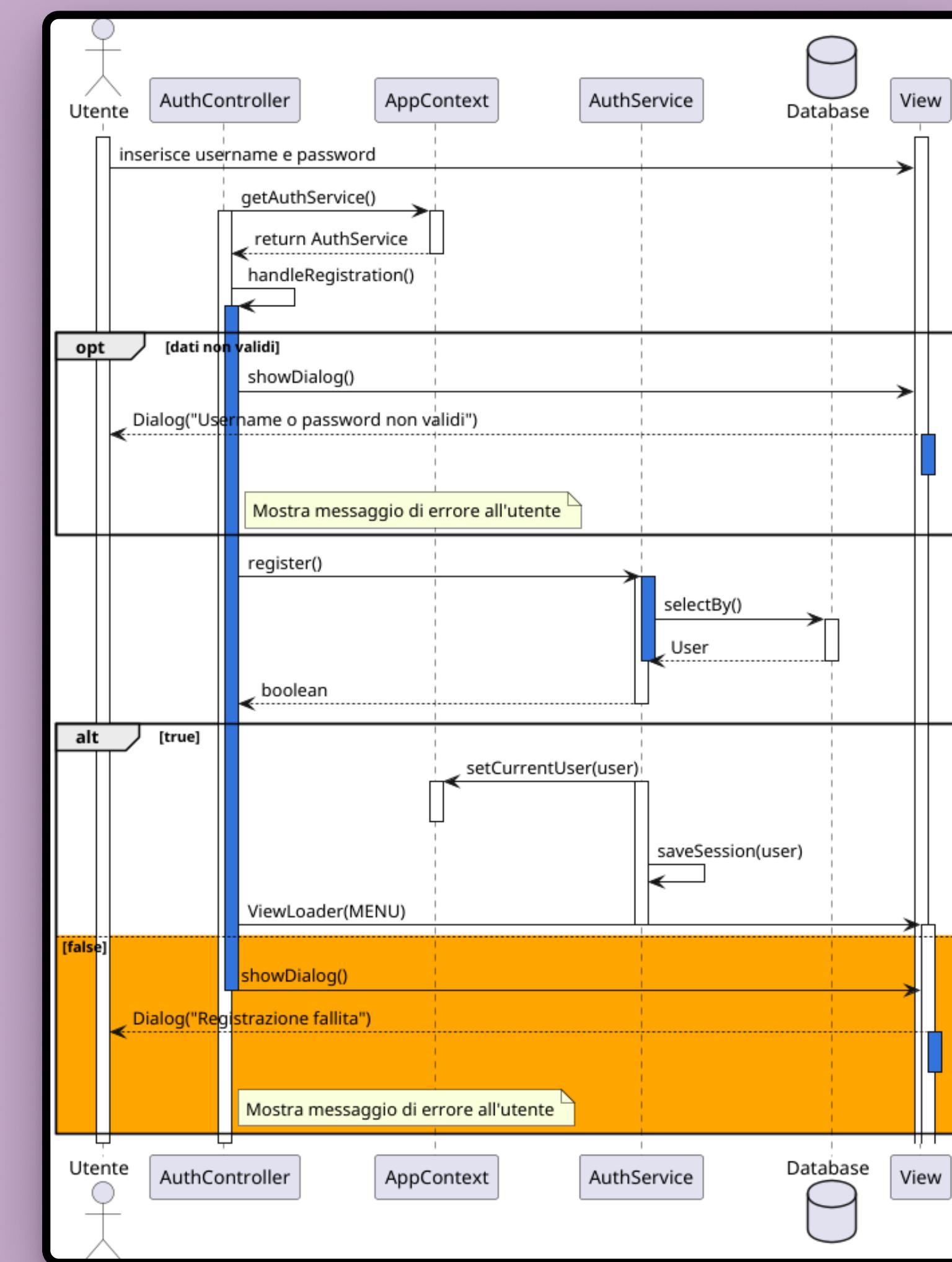
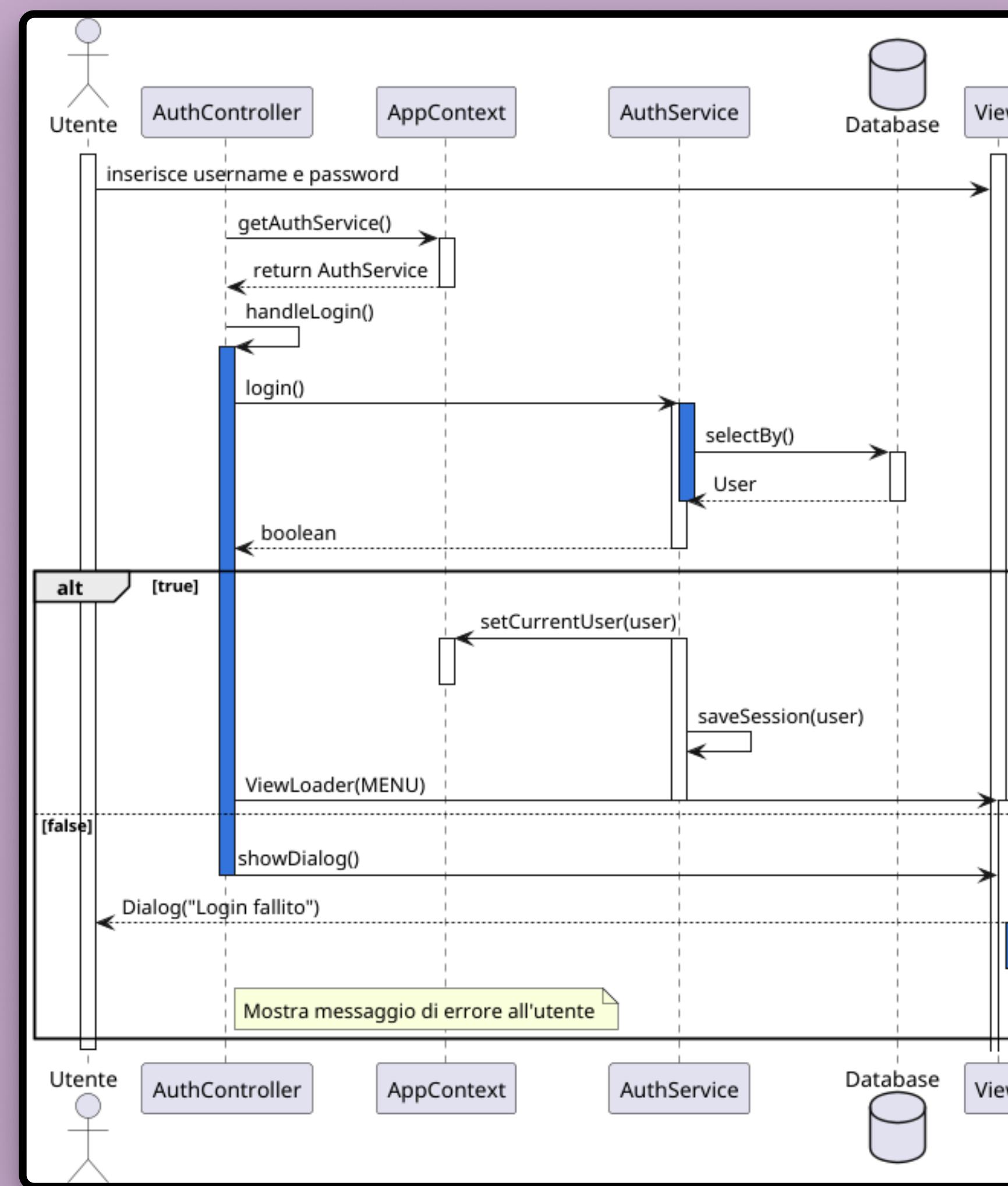
Un'eventuale sessione interrotta è serializzata nel file "interruptedSession.ser".

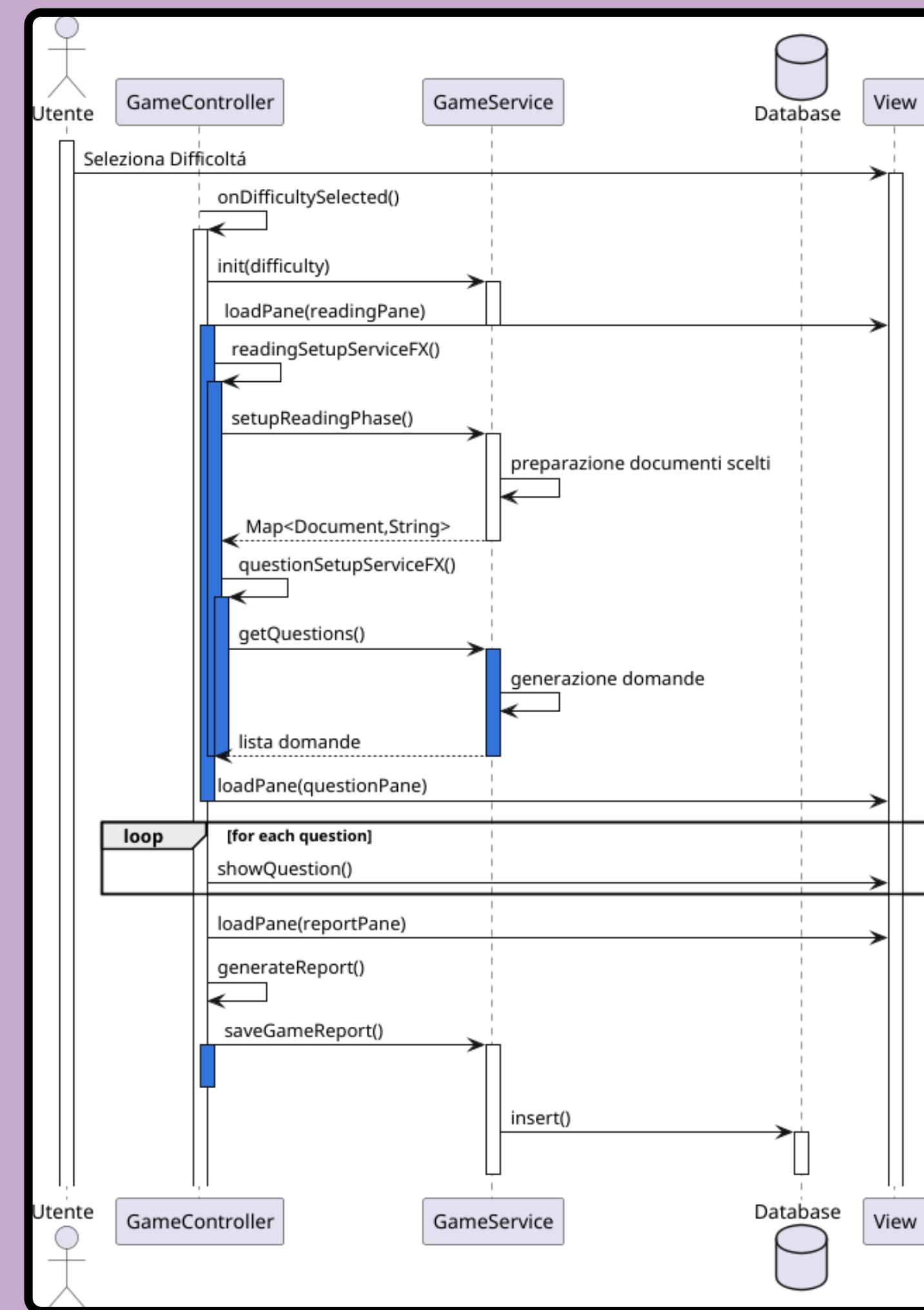
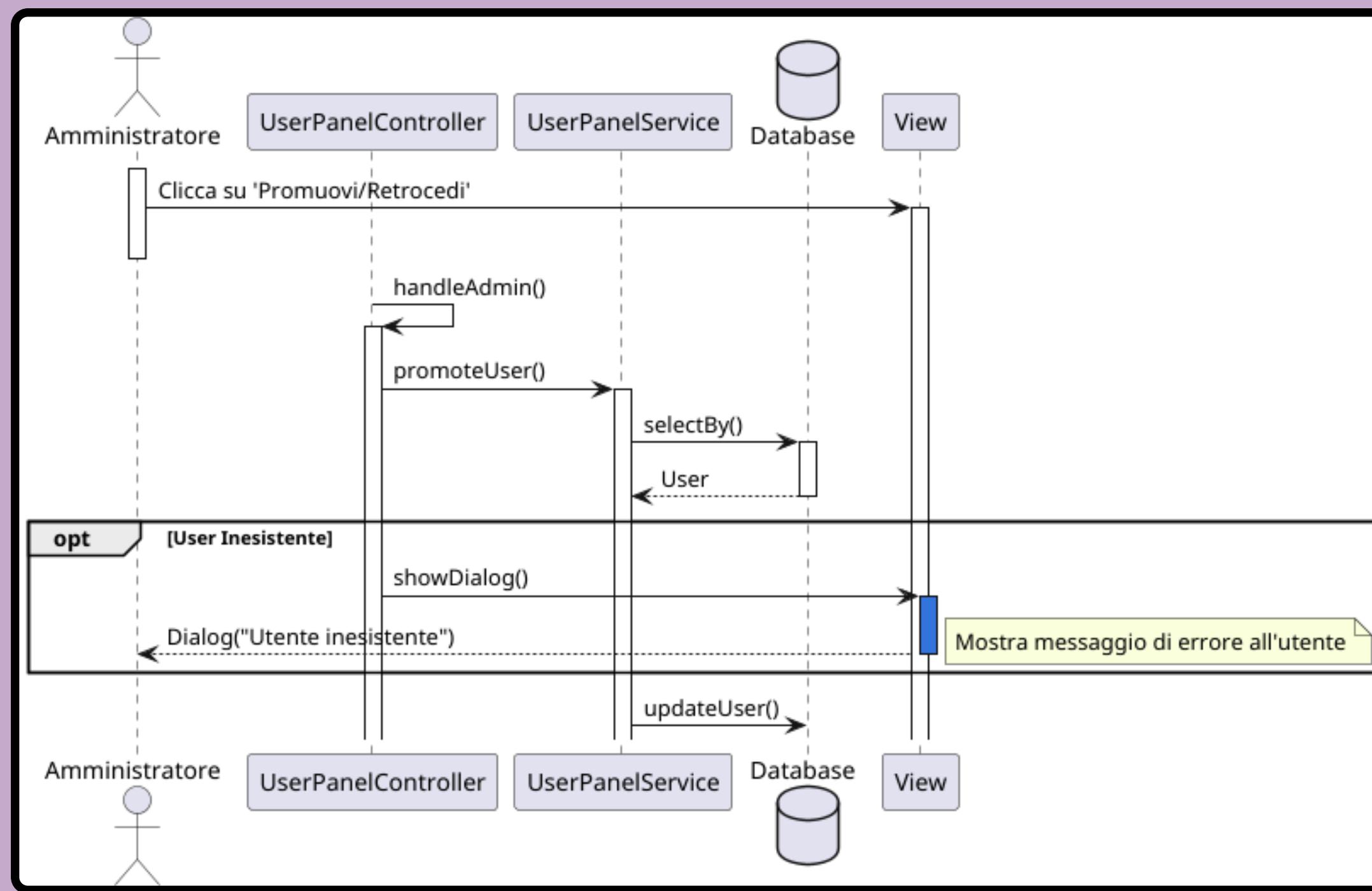


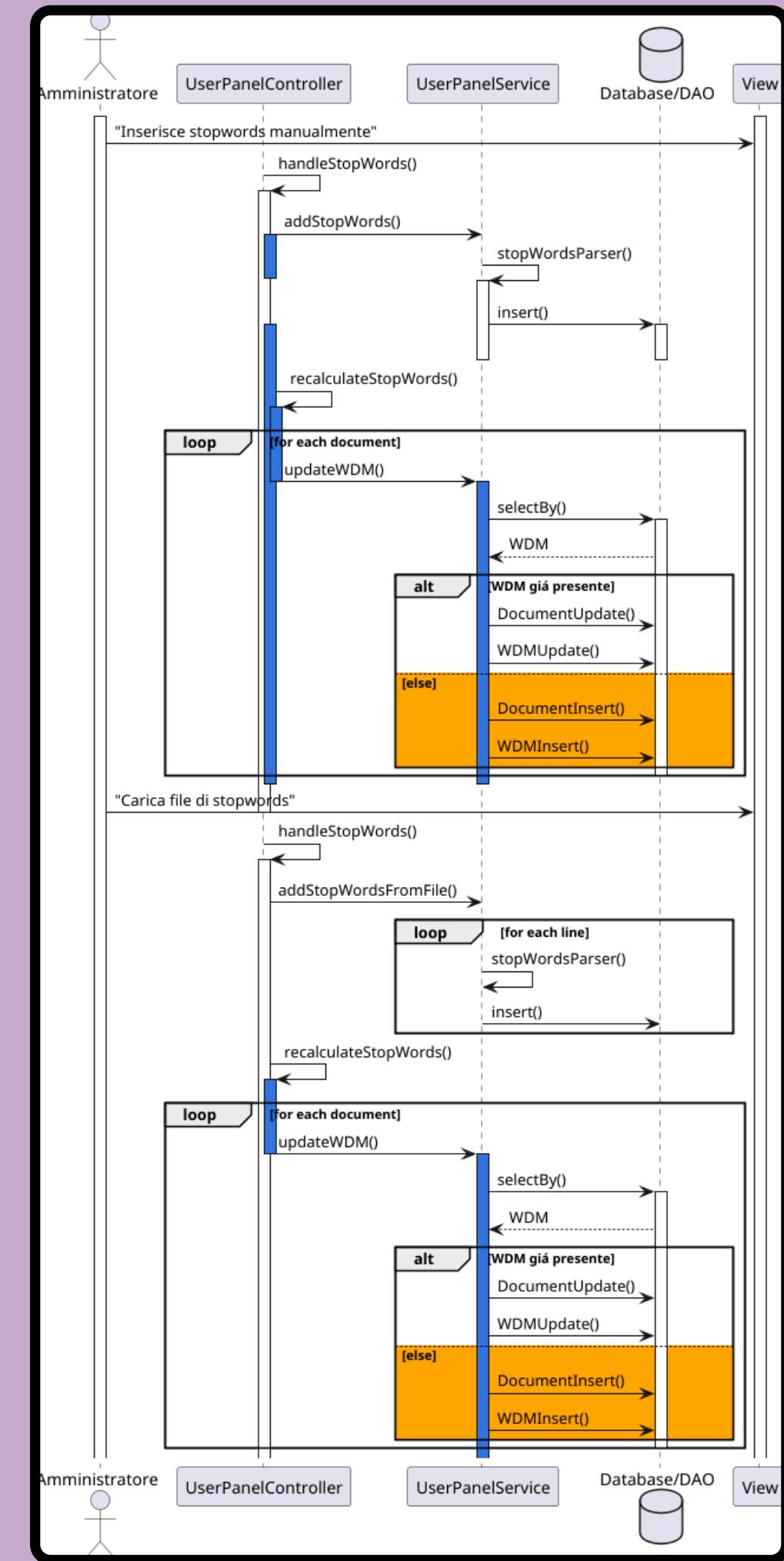
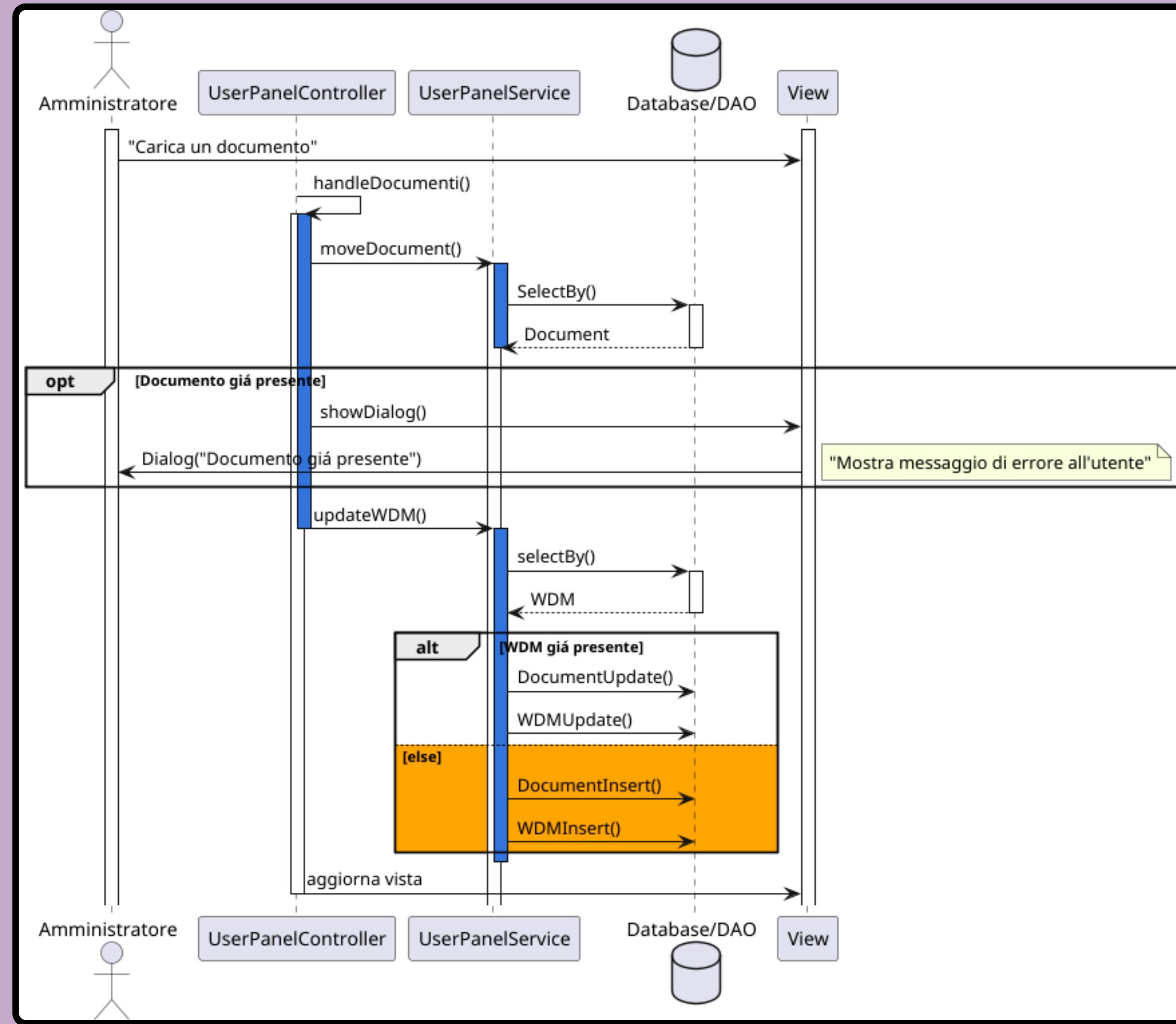
# Diagrammi UML











# Stack Tecnologico



## Java/JavaFX 24

Sviluppo dell'interfaccia grafica  
e della logica applicativa



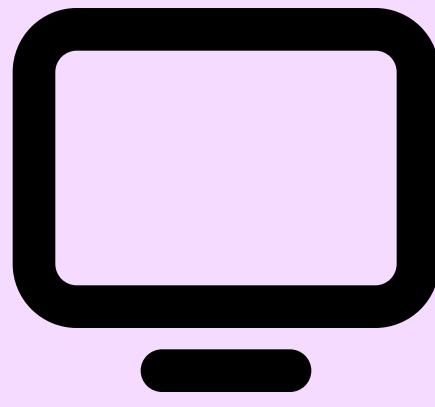
## SQLite

Gestione e archiviazione dei  
dati in locale



## Maven

Gestione delle dipendenze e  
automazione del build



# Schermate dell'app

01

# Autentificazione

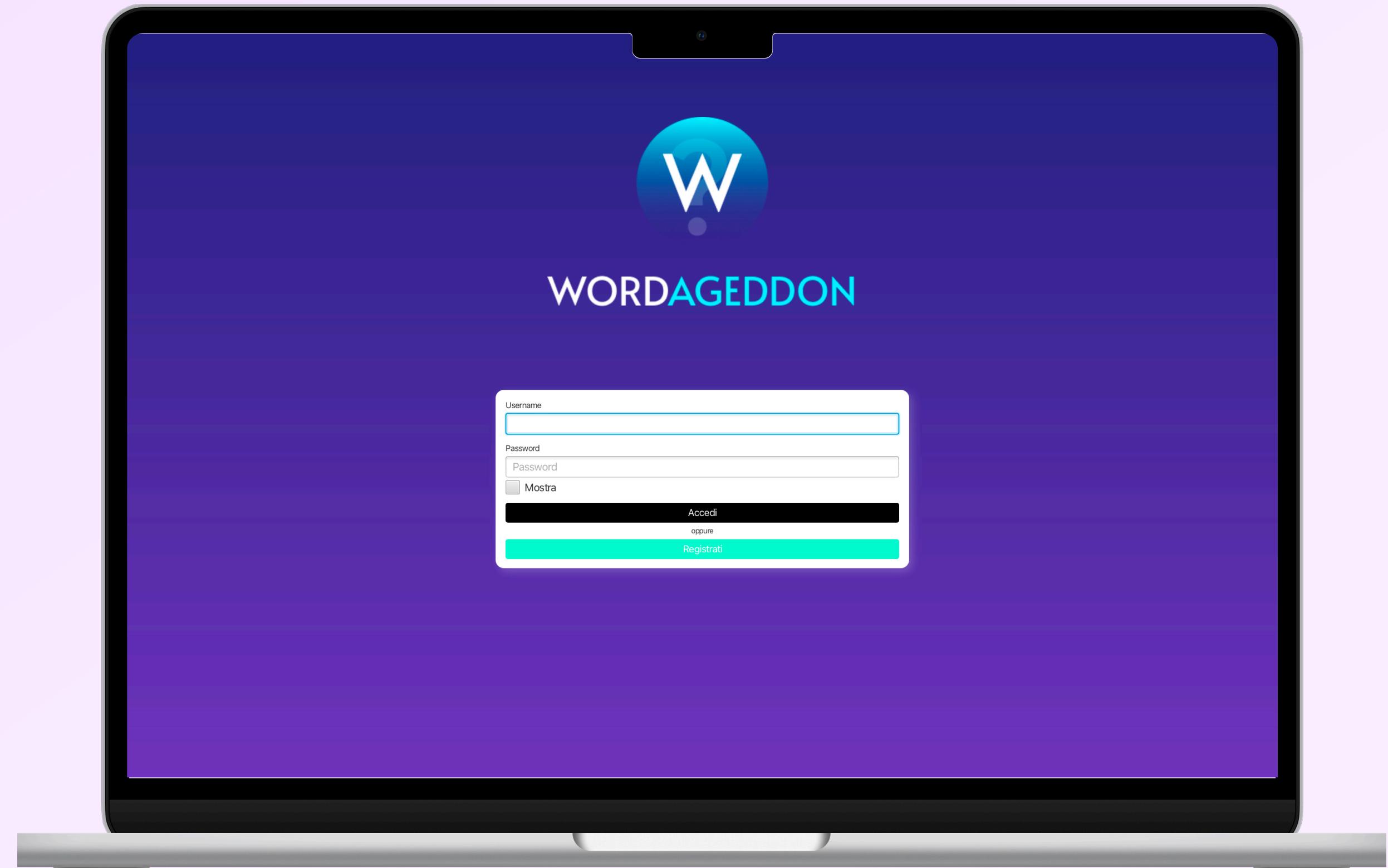
All'avvio dell'applicazione viene mostrata la schermata di login/registrazione, ma solo se non esiste già un utente (sessione) attivo.

Infatti, una volta effettuato l'accesso, l'utente viene **serializzato** localmente: al successivo avvio del programma, la sessione rimane attiva e si viene indirizzati direttamente al **menu principale**, evitando il login ripetuto.

Sono gestiti anche:

- la registrazione di nuovi utenti (con controllo univocità),
- il login con la verifica delle credenziali,

Il **primo utente** registrato è di default **admin**. Inoltre il sistema richiede sempre la presenza di **almeno un admin**, mediante un **trigger** sul database che impedisce la cancellazione dell'ultimo admin rimasto.



## 02

# Menu



Il **menu principale** è il punto centrale dell'applicazione, da cui l'utente può accedere a tutte le funzionalità:

- Avviare una **nuova partita**
- Visualizzare la **leaderboard**
- Aprire il proprio **pannello utente**

Se è presente una sessione interrotta, viene mostrato un popup che consente all'utente di **riprendere la partita precedente** dal punto in cui era stata interrotta.

Se invece l'amministratore non ha aggiunto alcun documento, viene visualizzato un messaggio d'errore

03

## Game play

Ogni partita inizia con la scelta della **difficoltà**, che determina il **numero di documenti** mostrati, il **tempo di lettura** e il **numero di domande**.

L'utente legge uno o più testi in un tempo limitato, con la possibilità di passare direttamente al quiz tramite un pulsante "Salta".

Al termine della lettura, inizia il quiz a **risposta multipla**.



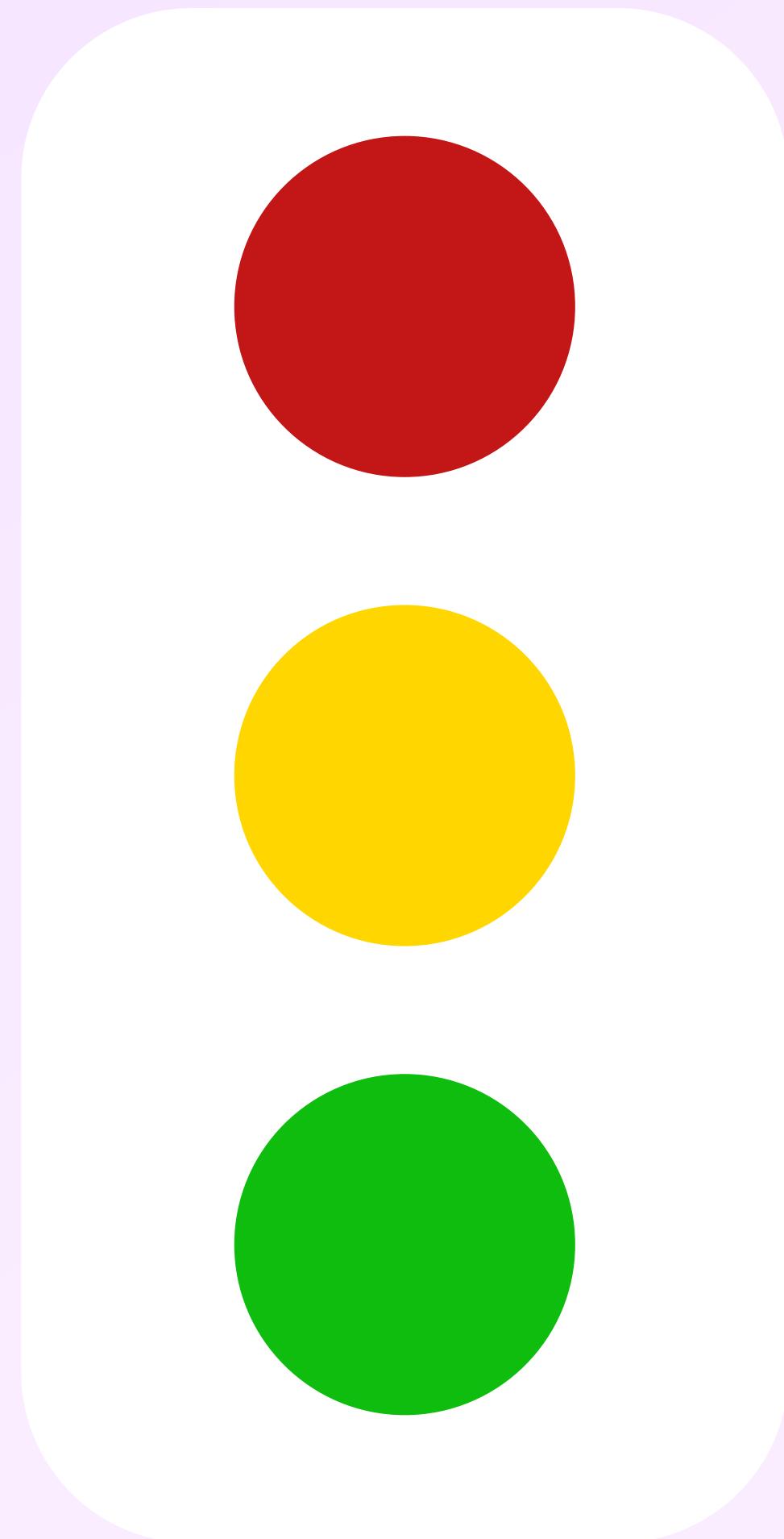
## 03

### Game play

A livello tecnico, il sistema assegna a ogni partita un **indice di difficoltà** complessivo, in base al livello scelto. Questo indice viene poi distribuito casualmente tra il numero di documenti da leggere, il numero di domande da generare e il tempo di lettura.

A seconda della difficoltà scelta verranno quindi mostrati i documenti che **rispettano la classificazione** mostrata.

DIFFICOLTA	PUNTEGGIO MAX	DOCUMENTI MAX	DOMANDE MAX
FACILE	100	3	10
MEDIA	200	5	15
DIFFICILE	300	7	20



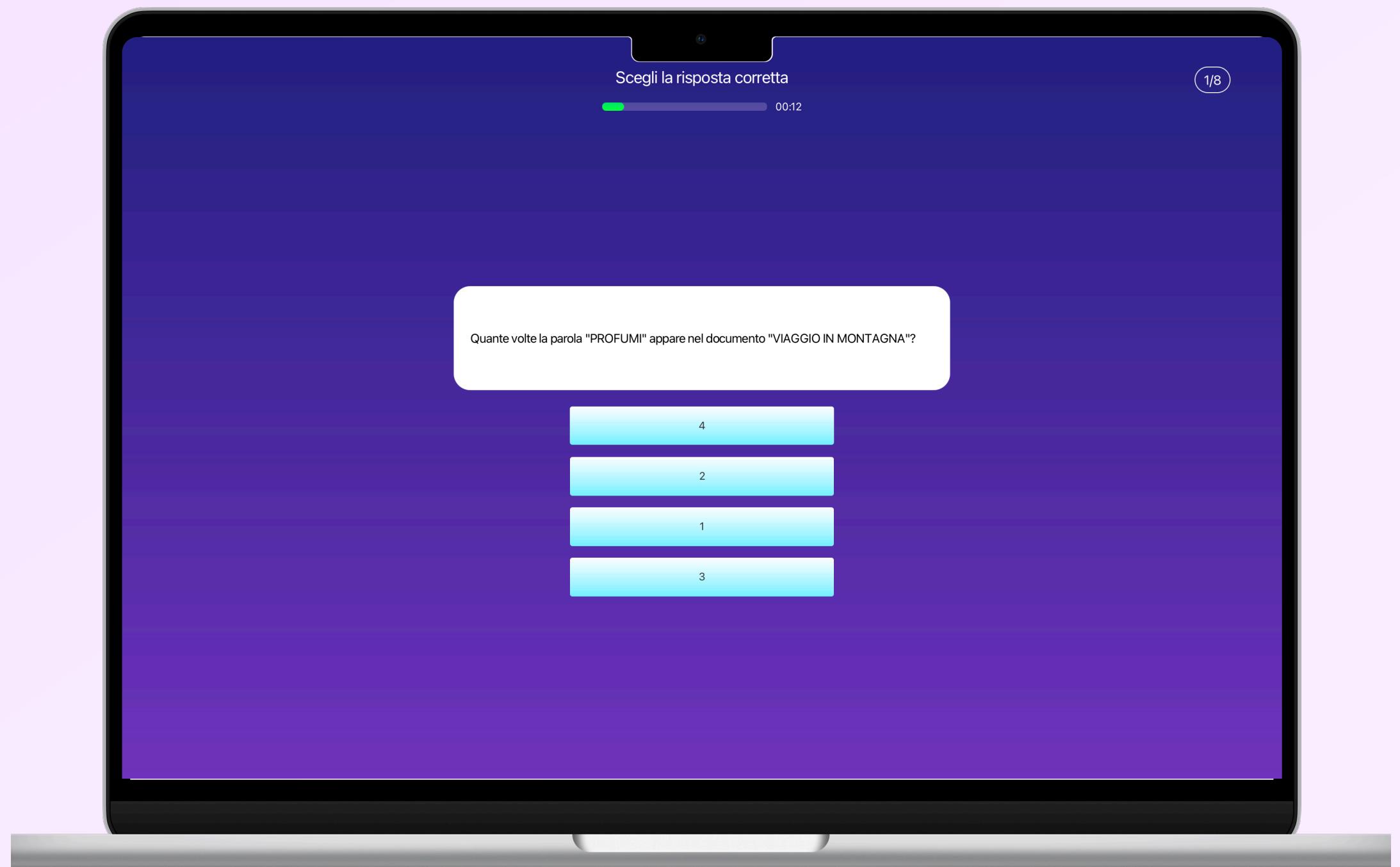
# 03

## Game play

Terminata la fase di lettura, viene avviata una serie di domande a risposta multipla che verificano la memoria e la capacità di analisi del giocatore.

Le domande possono riguardare:

- La **frequenza** con cui una parola appare in uno o più documenti
- Quale fra alcune parole è la **più o la meno frequente**
- L'**associazione** di parole a uno specifico documento
- L'identificazione di una parola che **non compare** in nessun documento.



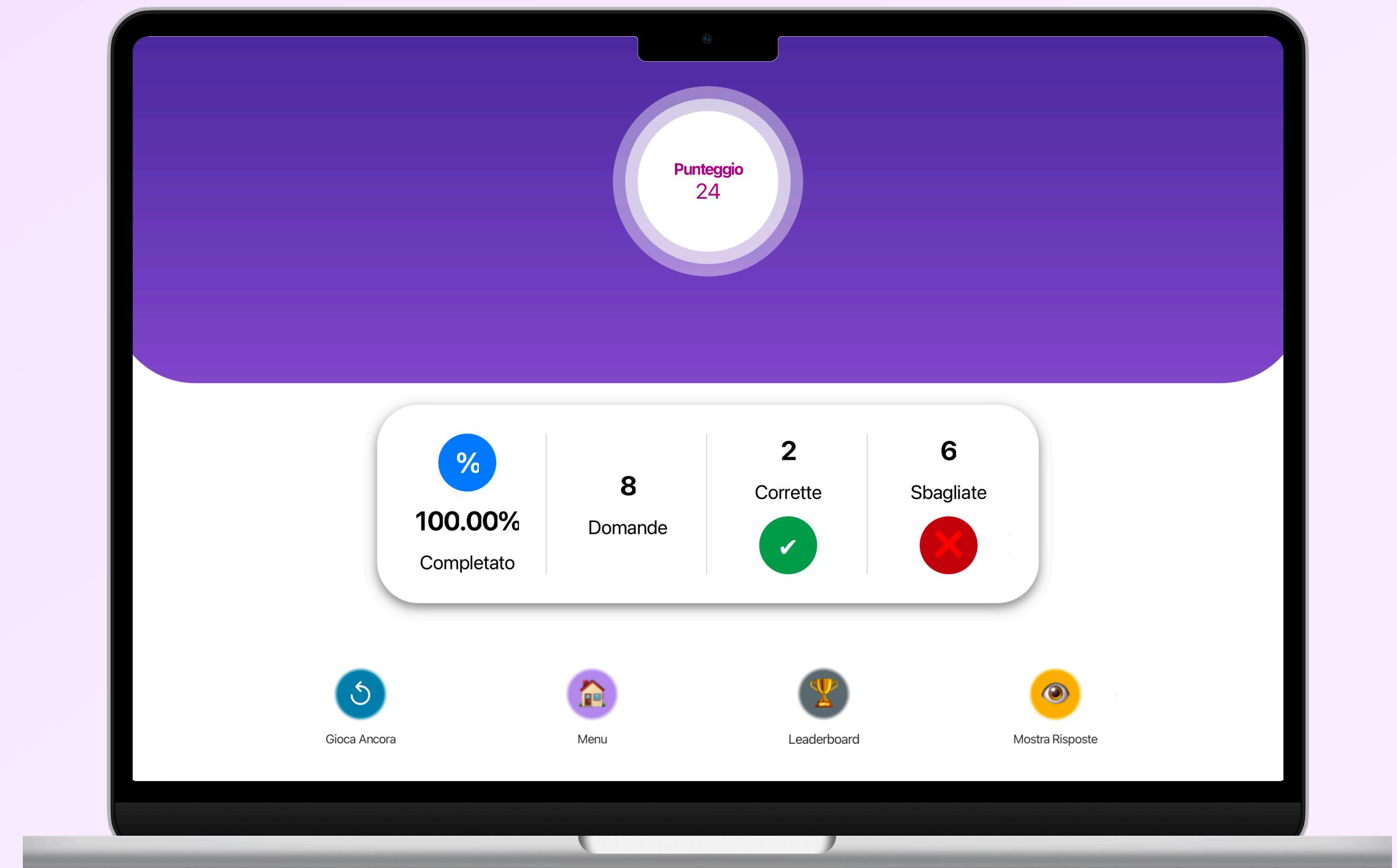
03

## Game play

Dopo il quiz, l'utente visualizza un **resoconto dettagliato** con il punteggio ottenuto ed altre informazioni relative alla sessione di gioco.

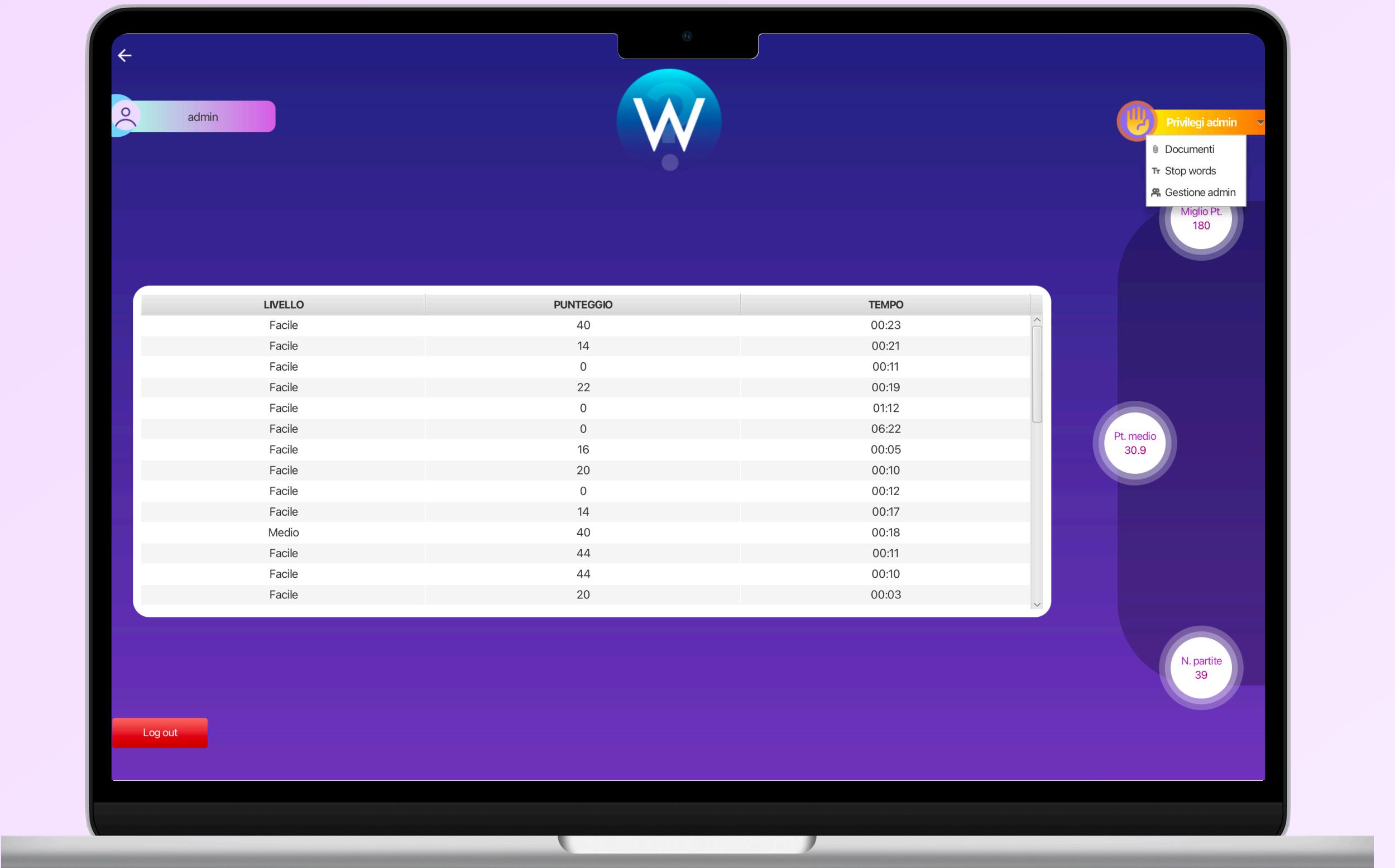
Una serie di pulsanti consentono la navigazione verso altre schermate.

L'utente può anche visualizzare lo **storico delle risposte** date per avere un confronto su quelle che erano le **risposte corrette**.



# 04

## UserPanel



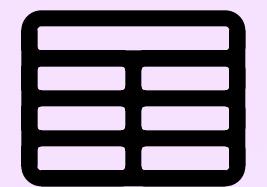
Dalla schermata "UserPanel", l'utente può visualizzare una serie di **informazioni** relative al proprio **account**, come lo storico delle partite, il punteggio medio, il punteggio migliore...

Gli **admin** hanno privilegi extra:

- Gestione documenti
- Gestione stop words
- Gestione (altri) admin

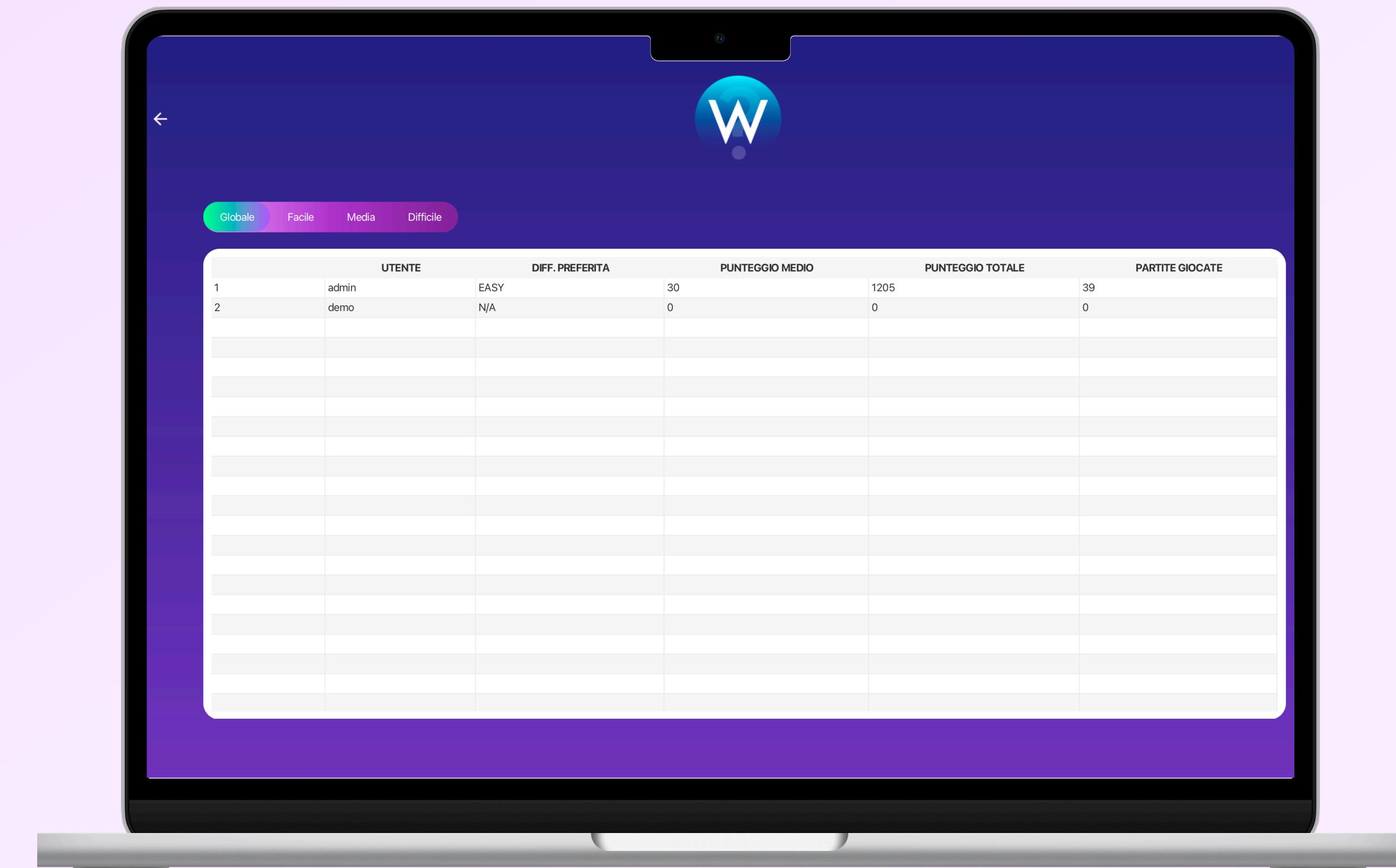
05

## LeaderBoard



La "LeaderBoard" mostra una panoramica sui **migliori risultati** ottenuti dagli utenti registrati al sistema.

Mostra la classifica degli utenti con i loro punteggi e consente di filtrare in base alle varie difficoltà.



**FINE**

