

Wordageddon

Relazione Tecnica del Progetto per il corso "Programmazione Java Avanzata"

Gabriele Lupo , Vincenzo Ragone , Silvio Ventura , Gianmarco Vitolo

10 giugno 2025

Indice

1	Introduzione	2
2	Obiettivi del Progetto	2
3	Architettura del Sistema	2
3.1	Pattern MVC	2
3.2	Diagramma dell'Architettura	3
4	Database PostgreSQL	3
4.1	Schema del Database	3
4.1.1	Tabelle Principali	3
4.2	Tecnologie Utilizzate	4
4.3	Struttura del Progetto	5
5	Funzionalità Principali	6
5.1	Gestione Utenti	6
5.2	Gestione Documenti	6
5.3	Sistema di Gioco	6
5.4	Filtraggio Stopwords	6
5.5	Gestione degli Errori	6
6	Conclusioni	7
6.1	Risultati Ottenuti	7
7	Casi d'Uso	7
7.1	UC1 - Registrazione Utente	7
7.2	UC2 - Avvio Sessione di Gioco	8
7.3	UC3 - Gioco a Domande Multiple Choice	8
7.4	UC4 - Caricamento di un Documento	9

1 Introduzione

Wordageddon è un'applicazione Java sviluppata come progetto universitario che implementa un sistema di gioco basato su domande e risposte. Il progetto utilizza il pattern architetturale Model-View-Controller (MVC) per garantire una separazione delle responsabilità e una struttura modulare del codice.

L'applicazione integra un database PostgreSQL per la persistenza dei dati e offre un'interfaccia utente intuitiva per gestire sessioni di gioco, utenti e contenuti.

2 Obiettivi del Progetto

Gli obiettivi principali del progetto Wordageddon sono:

- Implementare un sistema di gioco interattivo basato su domande e risposte
- Integrare un database relazionale per la persistenza dei dati
- Gestire utenti e sessioni di gioco
- Implementare un sistema di filtraggio delle stopwords
- Creare un'interfaccia utente user-friendly
- Garantire specifici permessi ad un amministratore

3 Architettura del Sistema

3.1 Pattern MVC

Il progetto adotta il pattern Model-View-Controller per separare le responsabilità:

Model: Rappresenta i dati e la logica di business dell'applicazione. Include le classi per la gestione di:

- Utenti
- Documenti
- Domande e risposte
- Sessioni di gioco
- Stopwords

View: Gestisce l'interfaccia utente e la presentazione dei dati. Responsabile di:

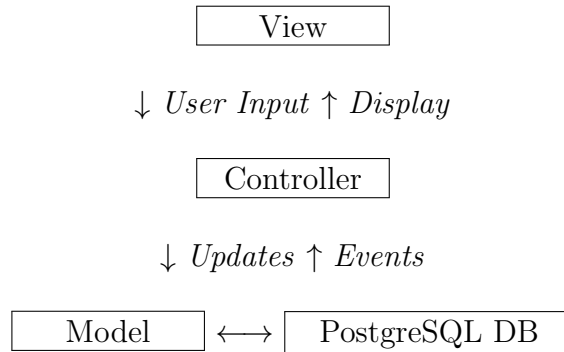
- Visualizzazione delle schermate di gioco
- Gestione dell'input utente
- Presentazione dei risultati

Controller: Coordina le interazioni tra Model e View, gestendo:

- Logica di controllo del gioco
- Validazione dell'input
- Aggiornamento del Model e della View

3.2 Diagramma dell'Architettura

Architettura MVC - Wordageddon



4 Database PostgreSQL

4.1 Schema del Database

Il database PostgreSQL è strutturato per supportare tutte le funzionalità dell'applicazione:

4.1.1 Tabelle Principali

Utenti

- user_id (PRIMARY KEY)
- username
- password_hash
- email
- created_at
- last_login

Documenti

- document_id (PRIMARY KEY)
- title
- content
- upload_date
- user_id (FOREIGN KEY)

Domande

- question_id (PRIMARY KEY)
- question_text
- difficulty_level
- document_id (FOREIGN KEY)
- created_at

Risposte

- answer_id (PRIMARY KEY)
- answer_text
- is_correct
- question_id (FOREIGN KEY)

Sessioni di Gioco

- session_id (PRIMARY KEY)
- user_id (FOREIGN KEY)
- start_time
- end_time
- score
- questions_answered
- correct_answers

Stopwords

- stopword_id (PRIMARY KEY)
- word
- language

4.2 Tecnologie Utilizzate

- **Linguaggio:** Java 8
- **Database:** PostgreSQL 13+
- **Driver JDBC:** PostgreSQL JDBC Driver
- **GUI:** JavaFX
- **Build Tool:** Maven

4.3 Struttura del Progetto

```
1 src/
2     main/
3         java/
4             com/QuizApp/
5                 alerts/
6                     AlertList.java
7                     AlertUtils.java
8                 controllers/
9                     LoginController.java
10                    MainController.java
11                    QuizController.java
12                    RegisterController.java
13                    RiepilogoController.java
14                    StartGameController.java
15                dao/
16                    ConnectionConfig.java
17                    documento/
18                        DocumentoDAO.java
19                        DocumentoDAOPostgres.java
20                    domande/
21                        DomandeDAO.java
22                        DomandeDAOPostgres.java
23                    risposta/
24                        RispostaDAO.java
25                        RispostaDAOPostgres.java
26                    sessionQuiz/
27                        SessionDAO.java
28                        SessionDAOPostgres.java
29                    stopWordsDAO/
30                        StopWordsDAO.java
31                        StopWordsDAOPostgres.java
32                    user/
33                        UserDAO.java
34                        UserDAOPostgres.java
35                models/
36                    User.java
37                    Documento.java
38                    Domanda.java
39                    GeneratoreRisposte.java
40                    SessioneQuiz.java
41                quizapp/
42                    WordageddonApp.java
43                difficultySettings/
44                    DifficultyEnum.java
45                    DifficultySetting.java
46                exceptions/
47                    CampiNonCompilatiException.java
48                    DatabaseException.java
49                    NessunaModificaException.java
50                    NotEnoughDocuments.java
51                    PasswordDiverseException.java
52                    PasswordNonCorrettaException.java
53                    SessioneNonCaricataException.java
54                    UsernameFormatException.java
55                    UsernameGiaPreso.java
56                timerService/
```

```
57         TimerService.java
58         utils/
59         GestoreSalvataggioSessione.java
```

Listing 1: Struttura delle directory del progetto

5 Funzionalità Principali

5.1 Gestione Utenti

- Registrazione e autenticazione
- Gestione profili utente
- Cronologia delle sessioni di gioco
- Funzionalità amministratore

5.2 Gestione Documenti

- Upload e memorizzazione di documenti
- Analisi del contenuto
- Generazione automatica di domande sulla base dell'analisi del contenuto

5.3 Sistema di Gioco

- Creazione di sessioni di gioco
- Presentazione di domande multiple choice
- Calcolo del punteggio
- Salvataggio dei risultati
- Possibilità di continuare sessione interrotta

5.4 Filtraggio Stopwords

- Database di stopwords multilingue
- Filtraggio automatico durante l'elaborazione del testo

5.5 Gestione degli Errori

- Gestione delle eccezioni SQL
- Validazione dell'input utente
- Logging degli errori

6 Conclusioni

6.1 Risultati Ottenuti

Il progetto Wordageddon ha raggiunto con successo gli obiettivi prefissati:

- Implementazione completa del pattern MVC
- Implementazione completa di tutti i requisiti richiesti
- Integrazione efficace con PostgreSQL
- Interfaccia utente funzionale e intuitiva
- Gestione robusta dei dati e delle sessioni

7 Casi d'Uso

7.1 UC1 - Registrazione Utente

Attori coinvolti: Utente

Descrizione: Un nuovo utente si registra al sistema inserendo i dati richiesti (username, password, email).

Precondizioni:

- L'utente non è ancora registrato

Flusso principale:

1. L'utente accede alla schermata di registrazione
2. Inserisce username, password, conferma password ed email
3. Il sistema valida i campi e verifica che l'username non sia già utilizzato
4. I dati vengono salvati nel database
5. L'utente viene reindirizzato alla schermata di login

Estensioni / Alternative:

- Username già esistente → messaggio di errore
- Password non corrispondenti → messaggio di errore
- Campi mancanti → eccezione `CampiNonCompilatiException`

Postcondizioni:

- L'utente è registrato nel sistema

7.2 UC2 - Avvio Sessione di Gioco

Attori coinvolti: Utente

Descrizione: Un utente autenticato può avviare una nuova sessione di gioco scegliendo un livello di difficoltà.

Precondizioni:

- L'utente è autenticato
- Almeno n documenti sono disponibili nel sistema (a seconda della difficoltà)

Flusso principale:

1. Utente sceglie il livello di difficoltà
2. Inizia la sessione di gioco

Estensioni / Alternative:

- Non sono stati caricati abbastanza documenti → eccezione `NotEnoughDocuments`
- Sessione non caricabile → `SessioneNonCaricataException`
- Sessione interrotta → Sessione serializzata e salvata su file

Postcondizioni:

- La sessione di gioco è in corso

7.3 UC3 - Gioco a Domande Multiple Choice

Attori coinvolti: Utente

Descrizione: Durante una sessione, l'utente risponde a domande a scelta multipla generate automaticamente.

Precondizioni:

- È attiva una sessione di gioco
- Domande sono già state generate

Flusso principale:

1. Il sistema presenta 4 domande con 4 opzioni
2. L'utente seleziona almeno un'opzione per ogni domanda
3. Utente conclude sessione

Postcondizioni:

- La sessione termina e viene salvata nel database

7.4 UC4 - Caricamento di un Documento

Attori coinvolti: Utente

Descrizione: Amministratore carica un nuovo documento di testo che verrà analizzato per la generazione delle domande.

Precondizioni:

- L'amministratore è autenticato

Flusso principale:

1. L'amministratore seleziona un file locale
2. Il sistema carica il contenuto nel database
3. Avviene analisi del documento

Estensioni / Alternative:

- Il documento è vuoto o non valido → errore lato GUI
- Errore nel database → `DatabaseException`

Postcondizioni:

- Il documento è salvato ed è disponibile per le sessioni di gioco