Scuola Politecnica e delle Scienze di Base
Corso di Laurea in Ingegneria Informatica

Relazione di Software Architecture Design – Task #6

Anno Accademico 2022/2023

Gruppo:

Guarino Nicola – M63001472
Niola Vittorio – M63001429
Russo Giovanni – M63001415

Indice

Introduzione

La seguente relazione ha l'obiettivo di illustrare, in maniera sintetica, i requisiti, le tecnologie e gli approcci implementativi individuati per l'esecuzione del Task #6 il quale prevede la realizzazione di un Text Editor per linguaggio Java da configurare all'interno di un'applicazione Web attraverso l'utilizzo del framework CodeMirror.

Sezione 1: Analisi dei Requisiti

Classificheremo i requisiti individuati in due categorie: funzionali e non funzionali.

1.1 Requisiti funzionali

- Creazione di un nuovo documento di testo;
- Apertura di un documento di testo esistente;
- Salvataggio di un documento di testo con un nome specificato dall'utente;
- Operazioni di modifica del testo come inserimento e cancellazione;
- Possibilità di annullare (undo) e ripetere (redo) le azioni di modifica del testo.
- Copia, taglia e incolla il testo selezionato;
- Ricerca e sostituzione di testo all'interno del documento;
- Gestione degli errori di input, ad esempio la gestione di file non validi o caratteri non validi nel testo;
- Possibilità di compilare ed eseguire il codice realizzato;
- Suggerimenti e possibilità di autocompletamento del codice;
- Operazioni di I/O con la console;
- Colorazione sintattica.

1.2 Requisiti non funzionali

- Interfaccia utente intuitiva e facile da usare;
- Compatibilità con diverse piattaforme, ad esempio Windows, macOS e Linux;
- Gestione affidabile dei file;
- Gestione adeguata degli errori, inclusi messaggi di errore chiari e comprensibili per gli utenti;
- Gestione corretta e sicura dei dati sensibili e la protezione contro le vulnerabilità di sicurezza note;
- Manutenibilità del codice, ad esempio la scrittura di codice pulito, ben documentato e facilmente manutenibile.

Sezione 2: Progettazione del sistema

Di seguito sono proposti alcuni diagrammi realizzati con *Mermaid*: un tool – installato come estensione di Visual Studio Code – che permette la creazione di ogni tipologia di diagramma.

2.1 Process Flow Diagram

Figura 2.1: Process Flow Diagram

2.2 Class Diagram

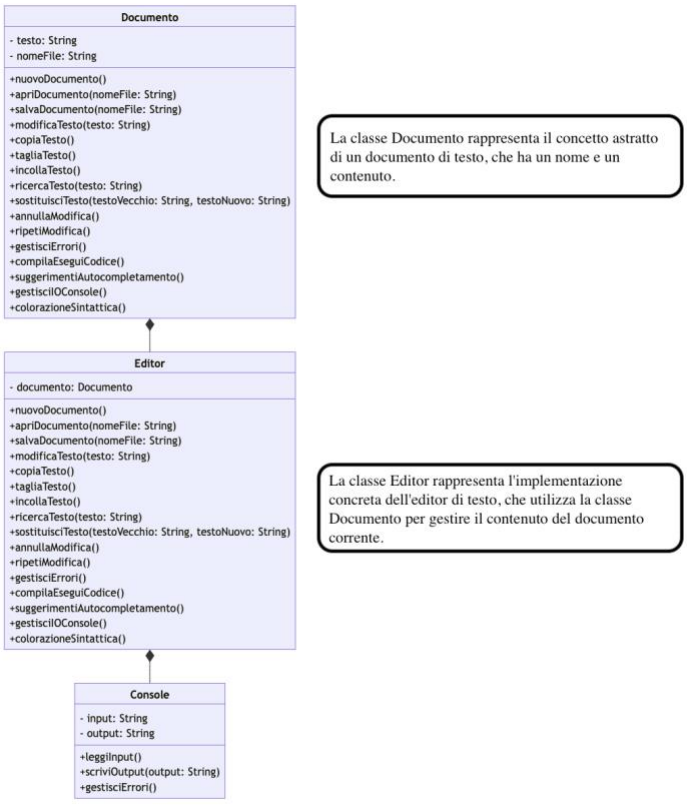

Figura 2.2: Class Diagram

2.3 Sequence Diagram

Tra i diagrammi di sequenza generati si è scelto di riportare nella seguente relazione quello inerente all'operazione di *creazione di un nuovo file*.

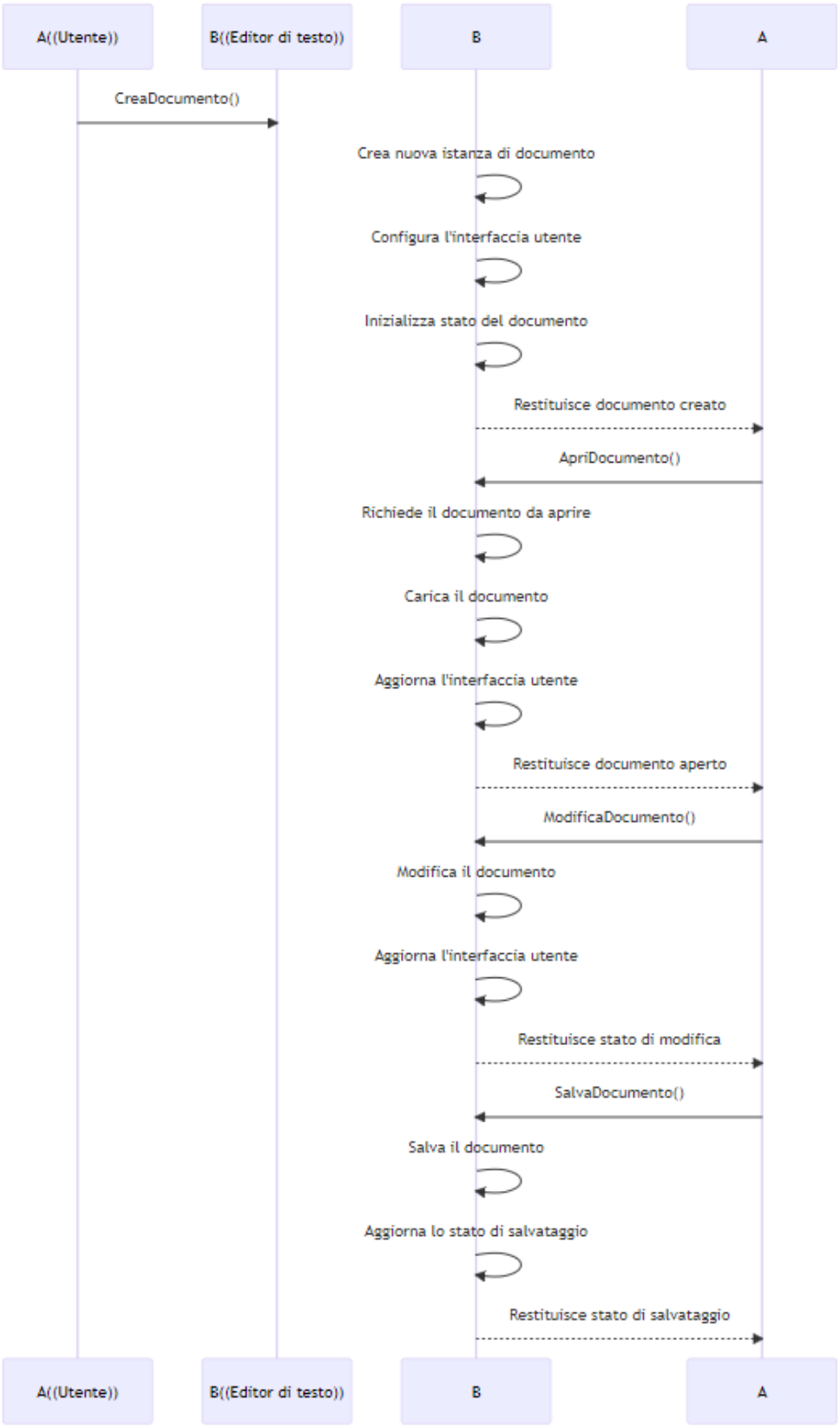

Figura 2.3: Sequence Diagram per l'operazione CreateNewFile

Sezione 3: Implementazione

Per la realizzazione grafica del TextEditor è stato utilizzato il linguaggio HTML con elementi di JavaScript e CSS. Nelle fasi successive del progetto si prevede l'adozione di PHP e SQL al fine di gestire correttamente l'interazione tra l'editor ed il database. Sono state utilizzate, inoltre, librerie e funzioni di CodeMirror coerentemente a quanto assegnatoci.

Di seguito viene proposta un'immagine che raffigura il prototipo finora realizzato:

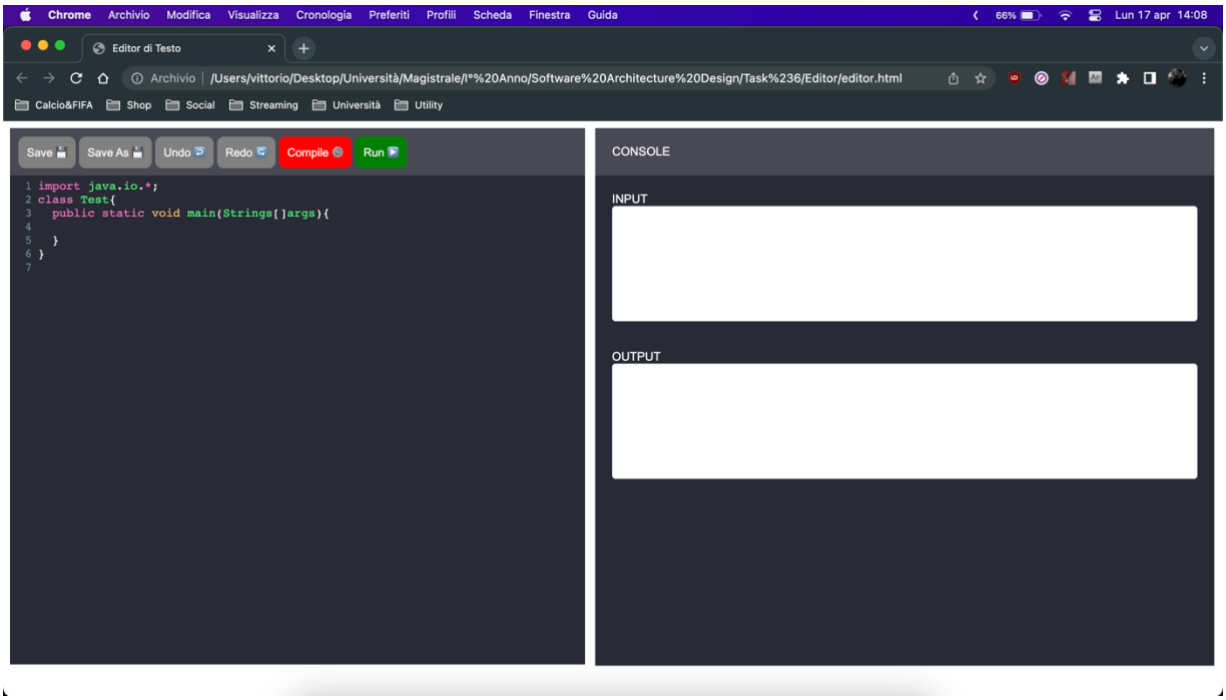

Figura 3.1: Prototipo TextEditor