Web Scraping

Progettazione e sviluppo di un applicazione per il web-scraping

Sommario

1 INTRODUZIONE

Il web scraping è una tecnica informatica di estrazione di dati da un sito web per mezzo di programmi software. Lo scopo di questo progetto è quello di analizzare, progettare e sviluppare uno di questi software mediante l'utilizzo di Java, in particolare per l'estrazione di dati riguardanti determinati tag HTML. I dati estratti verranno memorizzati in maniera permanente, per permettere all'utente di mantenere i dati di tutte le pagine analizzate.

Il processo di analisi si è concentrato principalmente sulla necessità di trovare una metodologia che sia il più possibile efficiente ed efficace per estrarre le principali informazioni dalle pagine web, indipendentemente dal tipo di pagina.

Il problema principale nell'estrazione automatizzata dei dati da una generica pagina web, sia nello specifico caso di questo progetto, sia in ambito generale, è spesso rappresentato dalla mancanza di una struttura standardizzata della pagina web. Pertanto si è scelto di andare a estrarre le informazioni in base al contenuto dei tag e dei relativi attributi, in quanto tag e attributi sono l'unica linea guida per estrarre tutti i contenuti rilevanti di una pagina web casuale.

Una pagina web può essere composta da diversi elementi come titolo, links, testi, immagini, media… Quindi in ogni risorsa web che l'utente passerà in input al programma , il sistema dovrà rilevare tutti i contenuti ritenuti importanti presenti all'interno della pagina, recuperando poi le informazioni di interesse in modo strutturato, trascurando elementi decorativi. Tale struttura recuperata verrà mantenuta in un database.

In quanto tale progetto, o parte di esso, deve avere la possibilità di far parte di un sistema software più grande, è fondamentale strutturare il codice sorgente tenendo conto della riusabilità di quest'ultimo in altri contesti.

Inoltre l'applicativo non ha un preciso target di utenti in quanto non richiede particolari conoscenze in ambito web, tuttavia è consigliabile l'utilizzo solo a chi ha una base di HTML, poiché lo scraper agisce proprio sul markup delle pagine.

2 DOCUMENTAZIONE UTENTE

2.1 FUNZIONALITÀ

Come descritto nell'introduzione, l'applicazione deve svolgere il web-scraping correttamente e mantenere i dati recuperati dallo scraping in maniera persistente perciò le sue funzionalità:

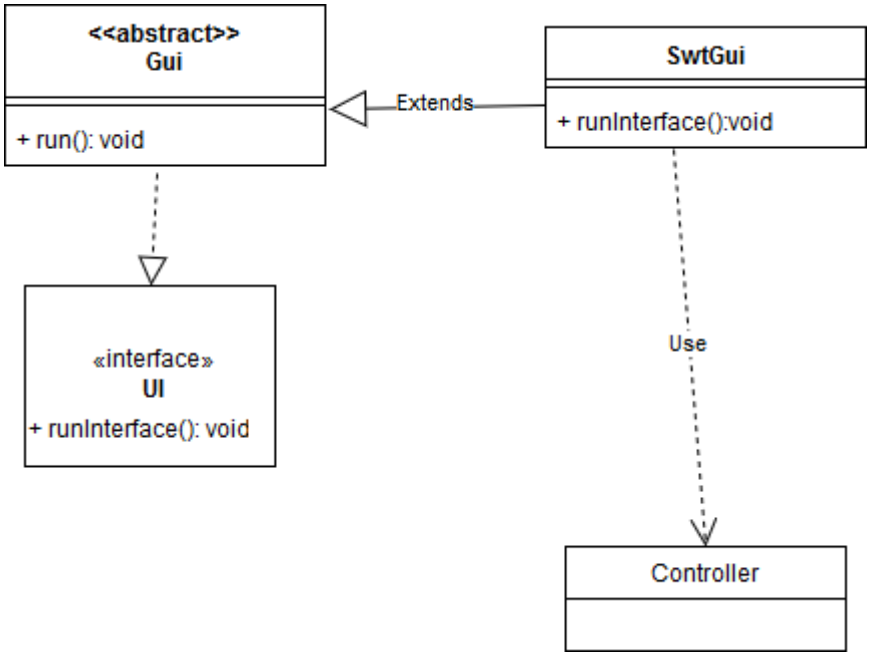
1. Recupero di una pagina web;
2. Memorizzazione di una pagina web;
3. Visualizzazione pagina web salvata nel repository;
4. Visualizzazione degli url relativi ad un dominio salvato nel repository;

Recupero di una pagina web: tale funzionalità è quella che esegue a tutti gli effetti il web-scraping. Esso necessita esclusivamente di un indirizzo web. Tramite esso, riesce a capire automaticamente se si tratta di un URL, si collega a quest'ultimo tramite uno user agent, recupera ogni tag rilevante e mostra tutti i dati ritornati all'utente;

Memorizzazione di una pagina web: tale funzionalità si occupa solamente dell'inserimento di un nuova pagina web nel repository. Tale funzionalità può essere utilizzata solo dopo aver effettuato lo scraping di una pagina. Se la pagina web recuperata dallo scraper esiste già nel repository, il sistema andrà a sovrascrivere i dati della medesima;

Visualizzazione una pagina web salvata nel repository: questa funzionalità si occupa del recupero di una pagina web dal repository per poi essere restituita alla user interface. Una volta che l'utente ha dato in input l'url, l'applicativo andrà a recuperare dal repository tutti dati relativi a questo, mostrandoli all'utente tramite l'interfaccia grafica;

Visualizzazione degli url relativi ad un dominio salvato nel repository: questa funzionalità si occupa solo di recuperare tutti gli url appartenenti ad un dato domino fornito in input dall'utente, recuperandoli dal repository e mostrandoli all'utente. Tale funzionalità è utile per l'utente quando vuole recuperare i dati di una certa pagina già salvata.



2.2 INTERFACCIA UTENTE

L'applicazione viene rilasciata con due tipi di interfaccia utente:

GUI (Graphic User Interface), un'interfaccia grafica per un'interazione con l'applicazione più rapida ed intuitiva.

3 DOCUMENTAZIONE TECNICA

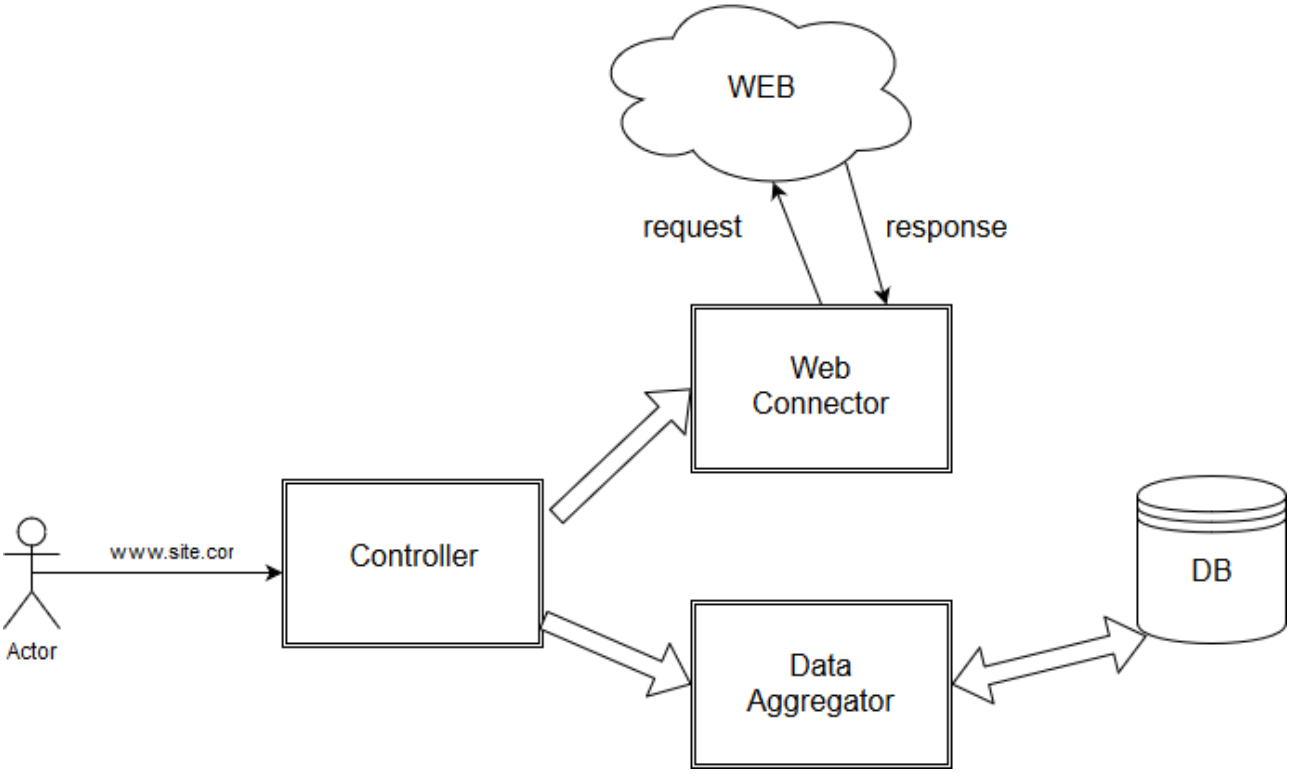
Le componenti principali dell'applicazione sono quattro:

1. **Web Connector**, per recuperare le varie informazioni dal web;
2. **Data Aggregator**, per gestire e strutturare i dati recuperati dal repository e dal web connector;
3. **Repository**, per salvare i dati in modo permanente;
4. **User Interface**, per l'interazione dell'utente con l'applicazione.

L'intera applicazione è stata sviluppata cercando di astrarre quanto più possibile questi concetti per garantire una certa elasticità nella loro gestione: per essere più precisi, sono state definite delle interfacce e classi astratte per il web-connector, la repository e la user-interface, che permettono ad esempio di utilizzare diversi tipi di repository (che possono variare da database relazionali o ad oggetti a semplici file di testo). Per fare ciò è sufficiente implementare l'interfaccia Store e definire dei metodi per il salvataggio e recupero dei dati. Oppure implementando l'interfaccia WebConnector si può effettuare un diverso tipo di scraping della pagina.

Il sistema è stato realizzato utilizzando il pattern architetturale MVC, per garantire una netta separazione delle logiche e rendere il software flessibile e sicuro.

3.1 MODELLO ARCHITETTURALE

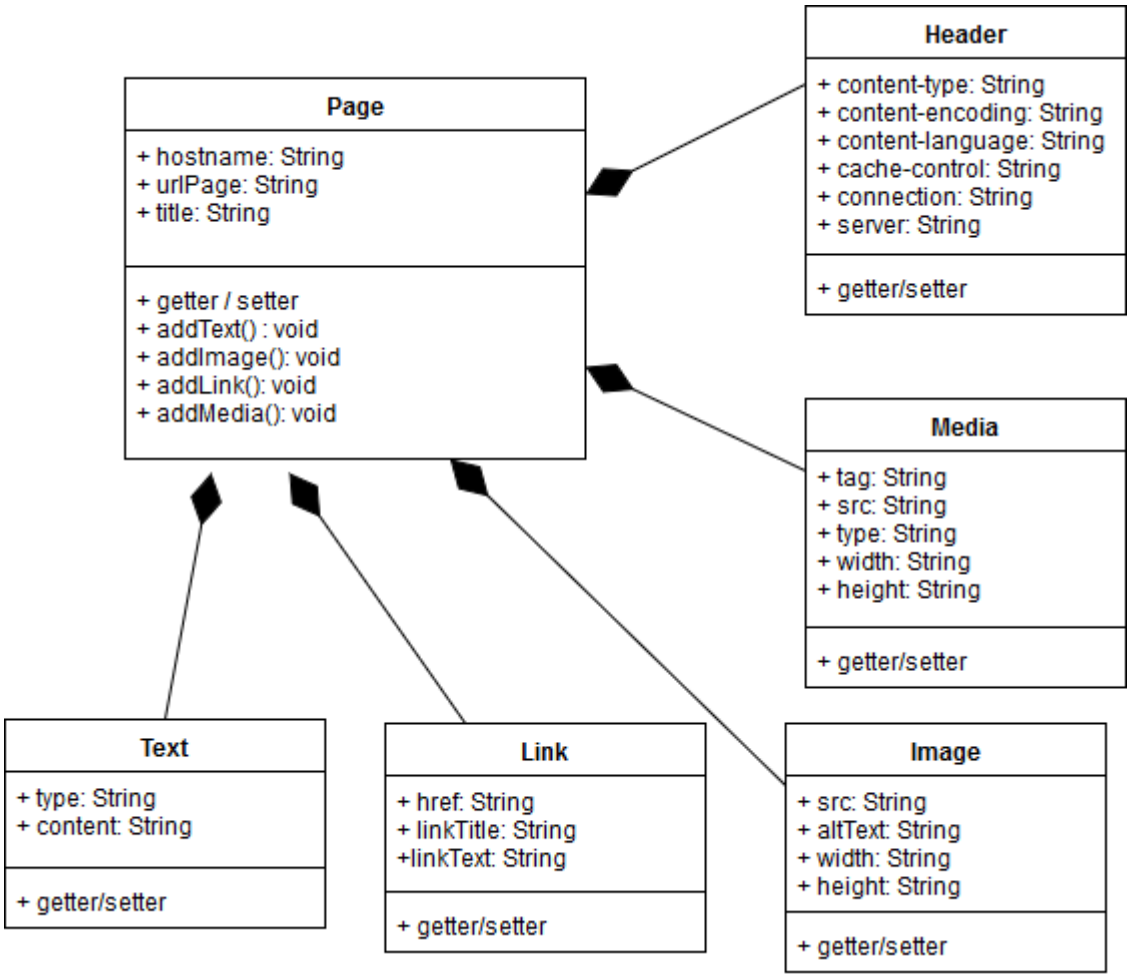


L'utente tramite l'interfaccia grafica passa l'URL della pagina web al modulo controller che aziona una funzionalità che il software deve eseguire. Pertanto il controller offre una serie di funzioni all'utente per poi delegando le altre sottocomponenti (Web Connector o Data Aggregator) all'elaborazione dell'output che verrà presentato sull'interfaccia grafica.

3.2 ORGANIZZAZIONE CLASSI

3.2.1 Package Model

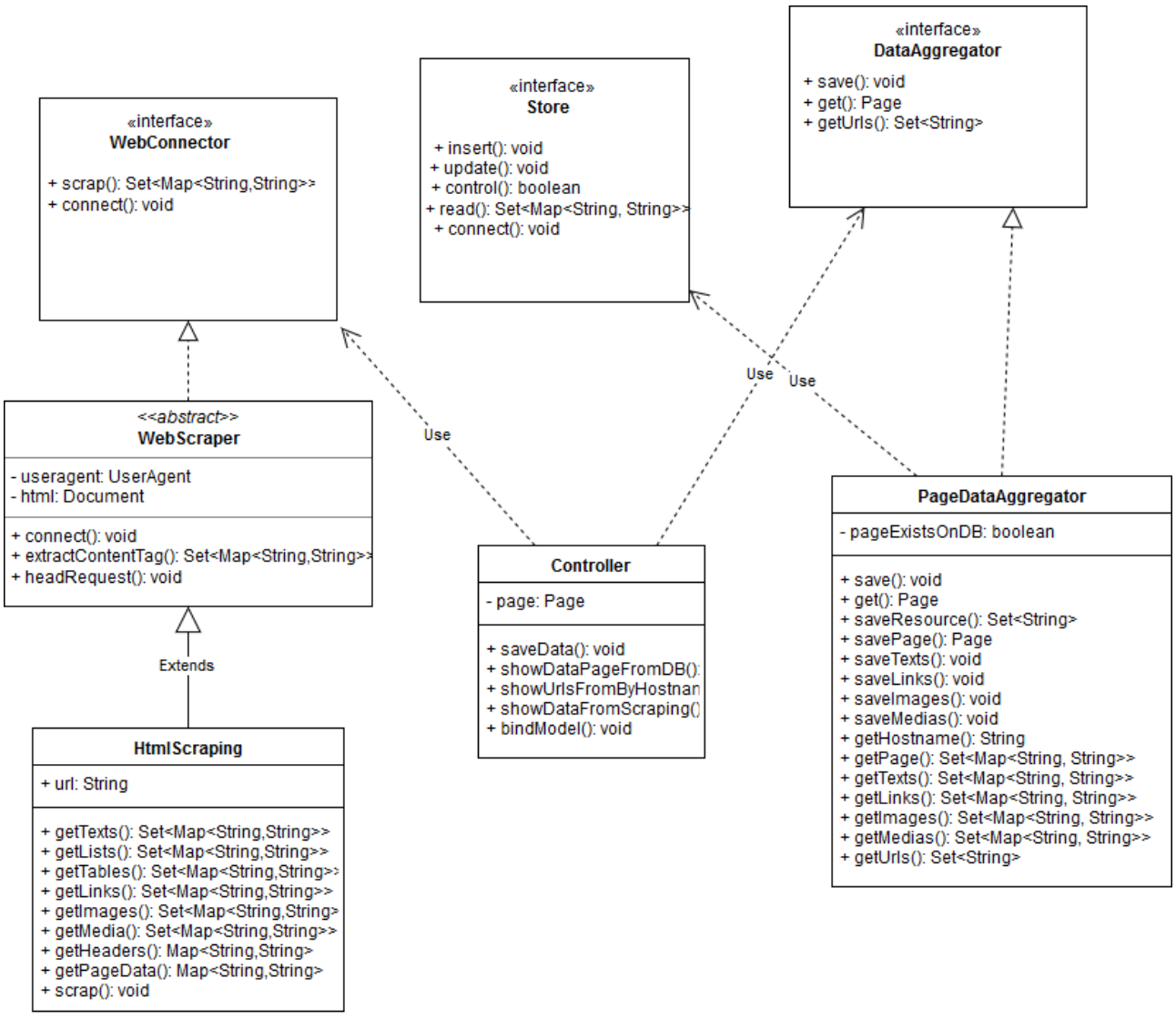
Il package model contiene tutte le classi che rappresentano gli oggetti che vengono manipolati dall'applicazione.



3.2.2 Package Controller

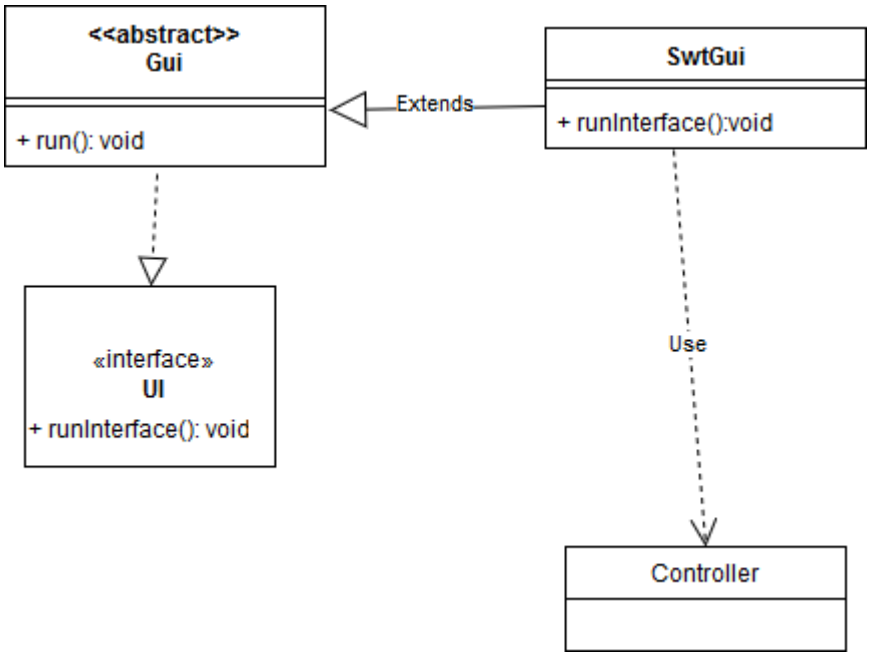
Il package controller contiene tutte le classi necessarie a gestire I dati e la logica di business dell'applicazione. Contiene inoltre la classe "controller", che rappresenta l'intermediario tra l'interfaccia grafica e le altre componenti dell'applicativo.

Da notare che l'implementazione dell'interfaccia Data Aggregator cambia a seconda dei dati di business utilizzati.



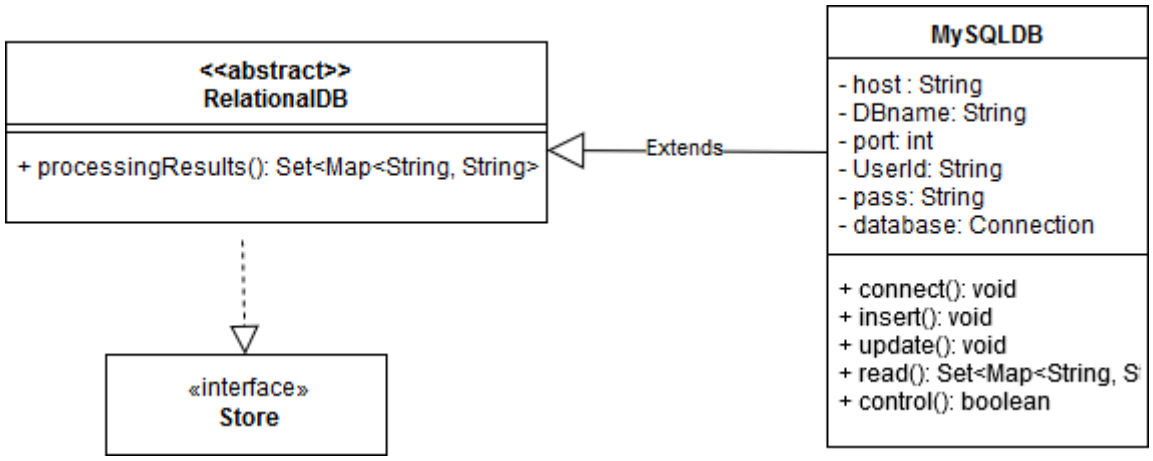
3.2.3 Package View

Il package view implementa l'omonima componente del design pattern MVC. Contiene la classe astratta UserInterface e le sue implementazioni. Inoltre notiamo che l'interfaccia grafica utilizza il controller per mostrare l'output desiderato all'utente.

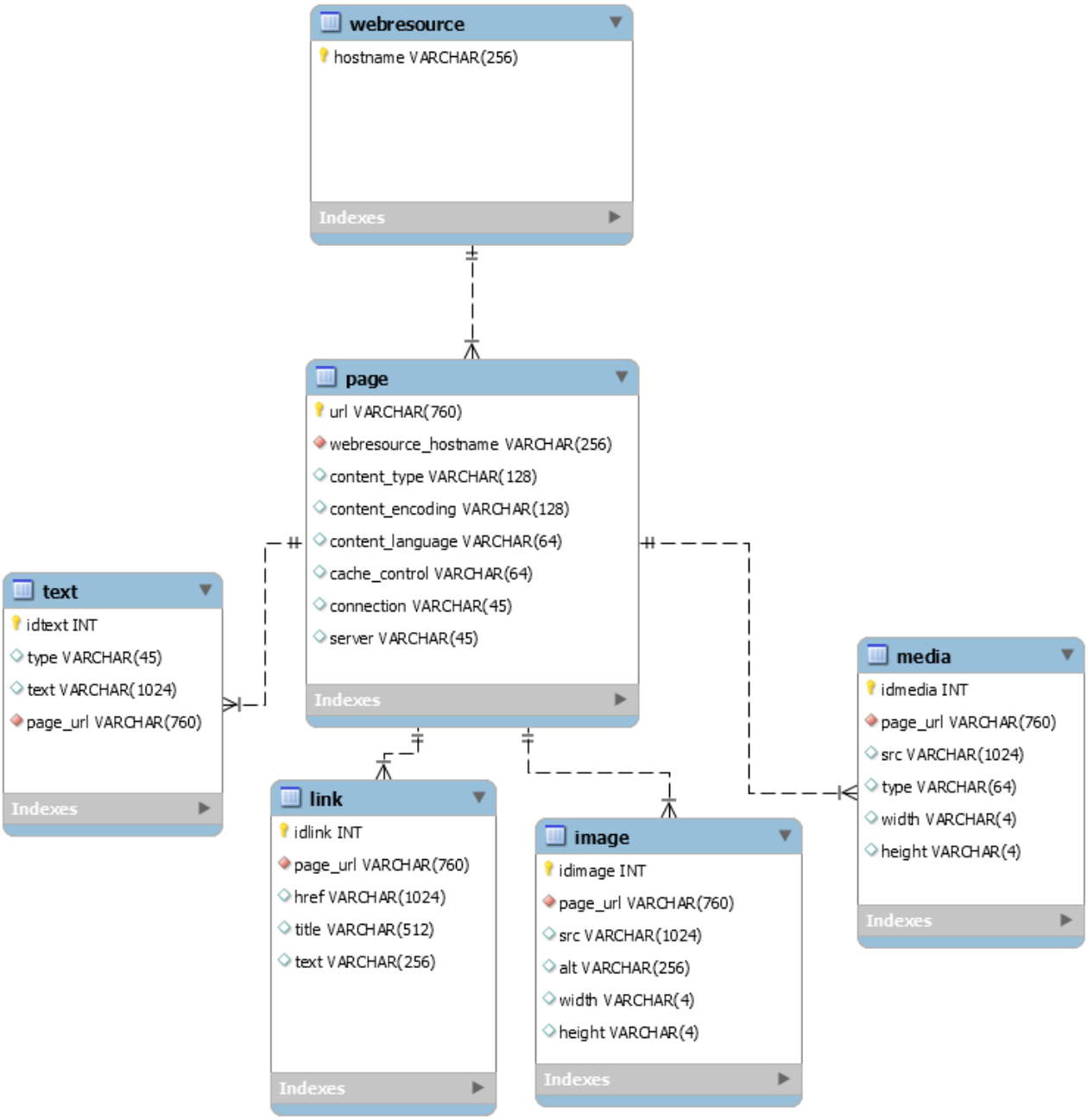


Package Repository

Il package Repository è stato inserito per fornire un'implementazione all'interfaccia Store.



3.3 MODELLO E/R DEL DATABASE



4 REALIZZAZIONE

Il software è stato realizzato tenendo presenti i requisiti descritti nel paragrafo 1, e testato per controllarne le funzionalità con i principali siti di qualunque categoria, news, ecommerce, social,….

Per la realizzazione del modulo WebConnector è stata utilizzata la libreria jaunt che implementa lo scraping vero e proprio sulla pagina web, mentre per implementare il modulo Repository si è utilizzato JDBC per connettere il sistema alla base di dati ed effettuare operazioni su di esso.