

Entwurfsbeschreibung zum finalen Release

1. Allgemeines

Unser Team entwickelt im Rahmen des Softwaretechnikpraktikums eine Desktop-Anwendung mithilfe von Java, welche bei der Durchführung und Dokumentation strukturierter Literaturrecherchen Hilfe leisten soll. Unser derzeitiges Produkt umfasst einen Wizard, die Anbindung an die SpringerLink und die ACM DL Datenbank (eingeschränkt) und eine eigene Datenbank der ausgewählten Dokumente. Zudem einen BibTeX Import bzw. Export und eine Oberfläche mit diversen Optionen und Funktionen zur Organisation der gefundenen Dokumente.

2. Produktübersicht

Der Wizard wird nur einmal zu Beginn jedes Projekts ausgeführt und legt den Rahmen der Suche nach der gewünschten Literatur fest, z.B. durch Schlagwörter, den Autor oder die Zeitspanne, in welcher die Literatur veröffentlicht wurde. Durch die Festlegungen in der Cooper Taxonomie kann man sich über die Natur seiner Literaturrecherche bewusstwerden. Nach dem Schließen des Wizards werden die festgelegten Bedingungen an das Suchmodul weitergegeben. Diese grenzen dann die Suchergebnisse der Online Datenbanken ein und liefern alle Suchergebnisse, welche den gewünschten Kriterien entsprechen. Anschließend werden diese Ergebnisse in einer lokal erstellten Datenbank gespeichert. In der Main-GUI können diese Ergebnisse eingesehen werden und die Nutzer haben die Möglichkeit Ihre aktuellen Projekte zu öffnen, Kommentare und Relevanzbewertungen zu hinterlassen, sowie Dokumente unter Begründung zu entfernen und hinzuzufügen. Relevante Handlungen der Nutzer sowie alle Eingaben aus dem Wizard werden in einer Log-Datei dokumentiert.

3. Struktur- und Entwurfsprinzipien

Um ein ansprechendes Design zu verwirklichen, wurde unser Wizard mit dem JavaFX Framework erstellt. Dieser ist in mehrere Abschnitte geteilt, damit der User eine bessere Übersicht hat und nicht den Eindruck ein langes Formular ausfüllen zu müssen. So wird er Schritt für Schritt durch die Definitionsphase geführt. Die Suchbedingungen werden in dem Wizard festgelegt und mit eines Simple Query Objects an das Suchmodul weitergegeben.

Unser Suchmodul gibt diese Bedingungen ebenfalls mithilfe eines Simple Query Objekts an die SpringerLink API weiter. Dieses Objekt wird daraufhin in eines der SpringerLink kompatiblen Formate konvertiert. Die Suchergebnisse der Online Datenbank werden innerhalb der Schnittstelle über das JSON-Format empfangen, weiterverarbeitet und anschließend in die ArrayList der Dokument – Klasse gespeichert. Diese Array List wird zusätzlich ausgegeben und zurück an das Suchmodul geschickt. Unsere Datenbank wurde mit SQLite entworfen, da im Falle unserer Software eine lokale Datenbank gegenüber einer Datenbank auf einem Server zu bevorzugen ist. Diese lokale Datenbank speichert alle Suchergebnisse und eine Liste aller Benutzer. Um auf diese Datenbank zuzugreifen bzw. die zu erstellen nutzen wir den JDBC Treiber für SQLite.

Unsere Main-GUI wurde wie auch der Wizard mit Hilfe des JavaFX Framework erstellt. Diese bietet dem Nutzer eine ansprechende Oberfläche, welche das intuitive Zurechtfinden mit der Anwendung erleichtern soll.

Die Log-Datei, welche für den User leicht zugänglich im Projektordner liegt, wird durch die Java-Logger-API erstellt und von dieser auch im weiteren Verlauf des Programmes gehandelt.

4. Weiterer Ausblick

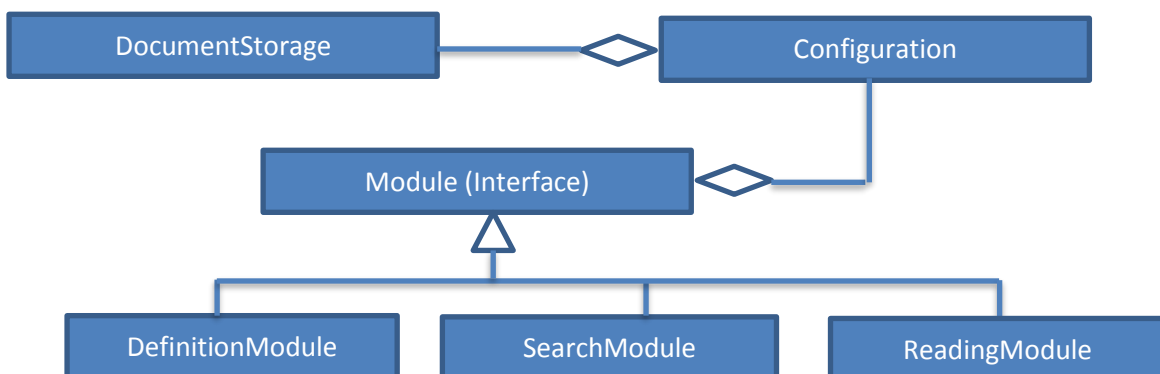
Sinnvolle Erweiterungen des Programmes wären die Verknüpfung der Suche mit zusätzlichen Datenbanken, die Erweiterung des Exports in eine größere Auswahl von Formaten und das automatisierte Herunterladen der gefundenen Dokumente. Als Suchoptionen könnte die Vorwärts- und Rückwärtssuche anhand der bereits gefundenen Dokumente hinzugefügt werden.

Es besteht zudem die Möglichkeit ein Synthesemodul zu erstellen, mit welchem man die Suchergebnisse in Konzepte klassifizieren und dann Beziehungen zwischen den Dokumenten und Konzepten festlegen und darstellen kann.

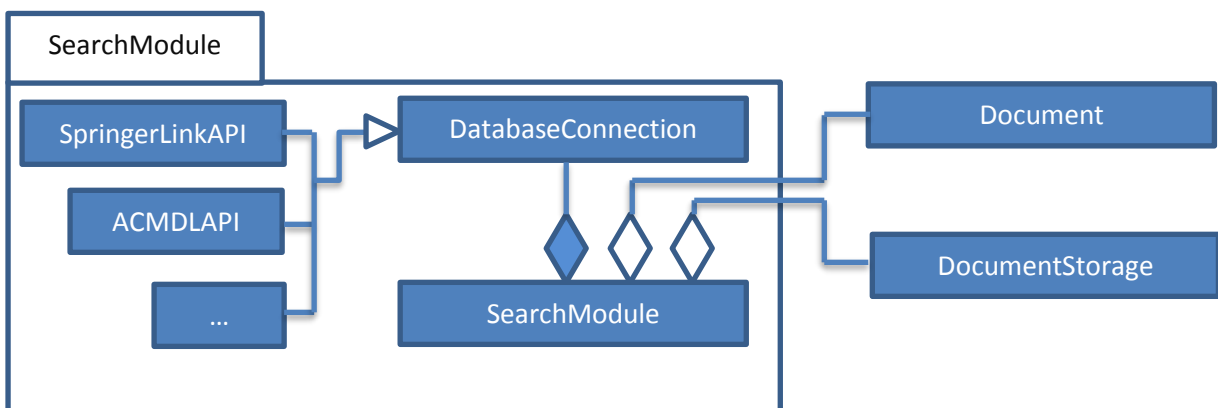
5. Datenmodell

Als vorübergehendes Datenmodell wird eine vereinfachte Form unseres derzeitigen UML-Diagramms verwendet.

Allgemeines Datenmodell:



Datenmodell des Suchmoduls:



Datenmodell des Lesemoduls:

