Projektmanagement

CO2 Application

Ein Bild, das Grafiken, Schrift, Clipart, Logo enthält.

Automatisch generierte Beschreibung

**Contributor: Simon Balcke Tutor: Herr Röhrig**

**Inhaltsverzeichnis**

|  |  |  |
| --- | --- | --- |
| Datum | Titel | Seite |
| 21.09.2023 | Deckblatt | **1** |
| 21.09.2023 | Inhaltsverzeichnis | **2** |
| 21.09.2023 | Beginn und Planung des Projektes | **3** |
| 28.09.2023 | CO2 Daten extrahieren | **3** |
| 05.10.2023 | CO2 Daten extrahieren | **4** |
| 12.10.2023 | Datenbank Logik implementieren | **4** |
|  |  |  |

**Beginn und Planung des Projektes**

Am 21.09.2023 war der offizielle Start des Projektes im Fach Projektmanagement. Mein Projekt hat das Ziel, auf Grundlage bereits vorhandener CO2 Daten, die Luftverschmutzung grafisch aufzubereiten und die Auswirkungen auf den menschlichen Organismus, speziell die Auswirkungen auf die Lunge, ausgelöst durch den menschengemachten Klimawandel, darzustellen. Wie auch bei meinem vorherigen Projekt, habe ich mich dazu entschieden, mein Projekt mit dem Build-Tool Apache Maven zu realisieren. Zunächst habe ich erneut die Entwicklungsumgebung IntelliJ installiert und ein neues Projekt mit dem Namen CO2App erstellt. Das Projekt verwendet die OpenJDK Version 1.8 und dementsprechend dieselbe Version des Compilers. Da ich für dieses Projekt auch eine MySQL Datenbank verwenden will, habe ich die Abhängigkeit JDBC Connector for MySQL in die Datei Pom.xml eingetragen. Um den Code testen zu können, habe ich ebenfalls das JUnit5 Framework als dependency eingebunden. Alle verwendeten Hilfen, Dependencies und weitere Quellen sind in der Datei Quellen.txt in dem Ordner resources zu finden. Das Projekt ist in das Backend und Frontend unterteilt. Im Backend finden die Berechnungen, der Datenaustausch zwischen der Anwendung und der MySQL Datenbank und das web Crawling statt. Im Frontend werden die gesammelten und aufbereiteten Daten grafisch dargestellt. Zunächst habe ich im Backend die Anfänge des WebCrawlers implementiert. Hierbei wird eine TCP Verbindung zu der URL aufgebaut, welche die CO2 Daten enthält und mittels HTTP GET request das HTML Dokument angefordert. Mittels BufferedReader wird der Inhalt des erhaltenen HTML Dokuments ausgelesen. Im weiteren Verlauf werden die gesuchten CO2 Werte gesucht und herausgefiltert. Zum Ende des Unterrichts habe ich diesen Bericht geschrieben und meine Arbeitsschritte dokumentiert.

**CO2 Daten extrahieren**

Am 28.09.2023 habe ich mir Gedanken darüber gemacht, wie ich die CO2 Daten von der Website aus der Excel Datei herausfiltern kann. Nach einer kurzen Recherche bin ich auf die Java API Apache POI gestoßen, mit dessen Hilfe auf .xlsx Formate lesend und schreibend zugegriffen werden kann. Da ich die Daten nicht verändern möchte, reicht mir der lesende Zugriff bereits aus. Die Klasse ExtractCO2Data.java übernimmt diese Aufgabe und filtert die CO2 Daten für die jeweiligen Jahre heraus. Es gibt mehrere Methoden, da die Vorgehensweise sich von den Jahren teilweise unterscheidet. Die Jahre 1990-2020 haben insgesamt 5 Reihen mit jeweils 6 Spalten, in welchen sich CO2 Daten befinden. Das Jahr 2020 hat 2 Reihen mit jeweils 6 Spalten und das Jahr 2022 nur eine Reihe mit 6 Spalten. Um mit diesen Informationen einen Algorithmus zu implementieren, hat es einige Zeit und Debugging benötigt, weshalb ich in der nächsten Stunde weiter an dem Problem gearbeitet habe.

**CO2 Daten extrahieren**

In dieser Stunde ging es darum, den Algorithmus zu verfeinern und kleine Fehler auszubessern. Um sicherzustellen, dass der Algorithmus funktioniert, wie er soll, habe ich hierfür Testfälle erstellt, welche das Ergebnis mit den reellen Daten aus der Excel Datei abgleicht. Das Resultat hat ergeben, dass ich alle Daten richtig eingelesen habe, da alle Tests grün waren. Da wir im Vorfeld für die Klausur Client Server Programmierung noch einmal geübt haben, fällt dieser Bericht etwas kürzer aus.

**Datenbank Logik implementieren**

In dieser Stunde ging es darum, die Grundlogik für die Interaktion mit der Datenbank zu implementieren. Dabei gibt es verschiedene Klassen mit verschiedenen Funktionen. Die Klasse CO2Entity repräsentiert einen Datenbankeintrag und hält einen Wert, der die CO2 Konzentration beinhaltet. Die Klasse CO2Service beinhaltet SQL Statements für die Interaktion mit der Datenbank. Unter anderem werden hier Queries für SELECT, INSERT, DELETE und UPDATE vorbereitet.