

Projektarbeit Gruppe 3 – SNOMED-CT Matching

THU - Technische Hochschule Ulm

Data Science in der Medizin

5. Semester

18.07.2021

Betreuer: Keno März

Inhaltsverzeichnis

Projektarbeit Gruppe 3 – SNOMED-CT Matching	1
1. Projektsteckbrief	2
1.1. Überblick	2
1.2. Arbeitsplans Beschreibung	2
2. Durchführung	2
2.1. Abweichungen	4
2.2. Probleme	4
2.3. Zukünftige Arbeiten	4
2.4. Start des Programms	4
3. Beiträge der Gruppenmitglieder	5
3.1. Natalia Sokolov	5
3.2. Nha Vy Bui	6
3.3. Susie Golubowski	6
3.4. Asli Sakiz	7
3.5. Emre Baylas	7
3.6. Felix Mattes	8

1. Projektsteckbrief

1.1. Überblick

Repo-URL: <https://github.com/Felix94-hub/SNOMED-CT-Matching>

Teammitglieder: Natalia Sokolov,
Nha Vy Bui,
Susie Golubowski,
Asli Sakiz,
Emre Baylas,
Felix Mattes

Libraries:

- JSON, GroupID: org.json, Version: 20210307
- Unirest: unirest-java-1.4.7.jar
<https://jar-download.com/artifacts/com.mashape.unirest/unirest-java/1.4.7/source-code>
- CSV: commons-csv-1.8.jar
https://commons.apache.org/proper/commons-net/download_net.cgi

1.2. Arbeitsplans Beschreibung

Ziel des Projektes war es ein Programm zu schreiben, welches die am besten passendsten Matches von jeweiligen Begriffen aus der SNOMED-CT Nomenklatur auszugeben, mit der Möglichkeit, bevorzugte Mappings des Nutzers separat abzuspeichern.

Bei der Planung und Strukturierung der Realisierung des Tools, hatte die benutzerfreundliche Oberfläche eine hohe Priorität. Dies wollten wir mit einer entsprechenden GUI von Java Swing umsetzen.

Unsere Projektplanung führten wir mit Meistertask durch (<https://www.meistertask.com/app/project/dk6jKttt/snowmed-ct>). Dadurch war es uns möglich flexible Arbeitspakete zu erstellen und jedem Gruppenmitglied jederzeit einen Überblick über den aktuellen Stand und den noch zu erledigenden Aufgaben zu bieten. Da es für uns absehbar war, dass wir nicht alle Projektziele innerhalb der vorgegebenen Zeit implementieren würden, war es uns wichtig, realistische Arbeitspakete und Ziele zu bilden.

Unser Programm ist nach dem Pattern „Model-View-Controller“ strukturiert.

2. Durchführung

Zu Beginn wurde das erste Grundgerüst von Model, View und Controller erstellt. Als nächstes folgte das Einlesen der „ihCCOntology_Excerpt.csv“ Datei (package -> readcsv, class -> CSVReaderWithHeaderAutoDetection2). Parallel wurden die MVC Klassen weiter verfeinert. Daraufhin wurden die Einträge der „ihCCOntology_Excerpt.csv“ Datei, welche eine Liste von onkologischen Begriffen beinhaltet, passenden Matches von SNOMED-CT verglichen und versucht, für den Nutzer möglichst fachlich interessante Begriffe zu identifizieren, welche in der final Version des Programms angezeigt werden sollen. Dieser Schritt hat sich am Ende jedoch als überflüssig erwiesen.

Um eine bessere Vorstellung vom Ablauf unseres finalen Programms zu bekommen, wollten wir „Snomed in 5 minutes“¹ nutzen, da sich dieses jedoch als veraltet herausgestellt hat,

¹ <http://snomedin5minutes.org/>

sind wir auf „Snowstorm“² gewechselt. Daher war unser nächster Schritt, eine Serververbindung zum SNOMED-CT Server herzustellen. Gleichzeitig haben wir einen ersten Prototypen fertiggestellt, der die grundlegendsten Funktionen wie bzw. das Anzeigen von passenden Matches zum jeweiligen vom Benutzer eingegebenen Begriff ermöglicht hat. Nachdem wir eine bessere Vorstellung davon hatten, wie unser Programm aussehen sollte, wurden die folgenden Klassen implementiert: CTModel.java, CTView.java, CTController.java und startCTSNOMED.java.

Diese haben die folgende Funktionen:

CTModel.java: Verbindung zu Unirest mit der Ausgabe der passenden Matches aus SNOMED-CT und Dateierstellung

CTView.java: Benutzeroberfläche

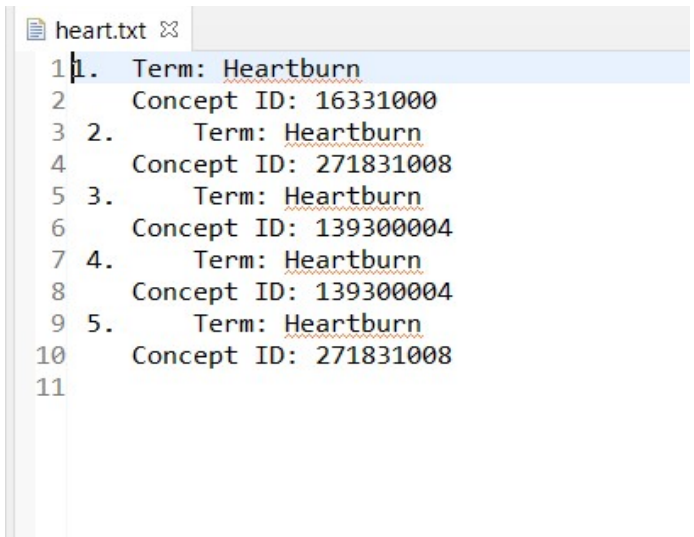
CTController.java: Schnittstelle zwischen Ctmodel.java und CTView.java

startCTSNOMED: Start der Ausführung des Programms

Zum Schluss haben wir in der Klasse CTmodel.java die Methode „ctDescriptions“ erweitert, die das Schreiben und Speichern der Matches in eine Datei ermöglicht.

Abb. 2: Inhalt der Textdatei "heart.txt"

Abb. 1: Testdateierstellung von "age.txt" und "heart.txt"



```
1 1. Term: Heartburn
2   Concept ID: 16331000
3 2.   Term: Heartburn
4   Concept ID: 271831008
5 3.   Term: Heartburn
6   Concept ID: 139300004
7 4.   Term: Heartburn
8   Concept ID: 139300004
9 5.   Term: Heartburn
10  Concept ID: 271831008
11
```

² <https://snowstorm.test-nictiz.nl/swagger-ui.html#!/Concepts/findConceptsUsingGET>

2.1. Abweichungen

Für die Nutzeroberfläche war zunächst eine Java Swing GUI geplant, da dies jedoch nicht mehr zeitgemäß realisierbar war, haben wir uns für eine Interaktion mittels Konsole entschieden. Wie zuvor bereits erwähnt sind wir für die Serververbindung von „Snomed In 5 Minutes“ auf „Snowstorm“ gewechselt, da ersteres veraltet ist. Da sich für uns insbesondere der Einstieg in das Projekt schwierig gestaltet hat und entsprechend viel Zeit beanspruchte, konnten Funktionen wie das Hinzufügen, Löschen und Bearbeiten von Mappings nicht verwirklicht werden.

2.2. Probleme

Unser erster Ansatz für eine Serververbindung war die Klasse HttpClient.java mittels einer HTTP-URL Connection, was sich als zu kompliziert herausgestellt hat. Daraufhin haben wir die Serververbindung mithilfe der Library „Unirest“ hergestellt.

Das Ziel für die Speicherung von Matches war es, dass der Nutzer jederzeit selbst entscheiden kann, ob und welche Matches in eine Datei übernommen werden. Aufgrund von zeitlicher Begrenzung des Projekts und technischen Schwierigkeiten bei der Implementierung, sind wir zu der Lösung gekommen, dass die Matches angezeigt werden und gleichzeitig in einer Datei für (jedes Match individuell) abgespeichert.

2.3. Zukünftige Arbeiten

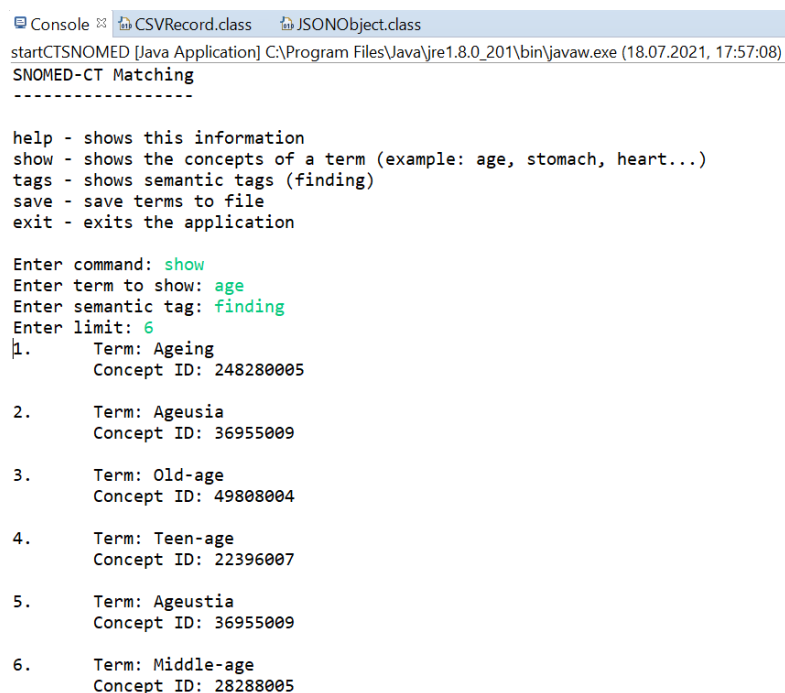
Da die Lösung für das Speichern von Matches momentan nicht ganz optimal ist, gäbe es hier Raum zu Verbesserungen bzw. eine Meldung in der Konsole, dass eine Datei erstellt wurde. Zudem fehlen Methoden, um Matches hinzuzufügen, editieren und löschen.

2.4. Start des Programms

Das Programm lässt sich auf zwei verschiedenen Arten starten:

2.4.1. Eclipse: Ausführen der Klasse „startCTSNOMED.java“ im Package „main“

Daraufhin erscheinen in der Konsole die möglichen Befehle. Mit dem Befehl „show“ und der Eingabe des gewünschten Begriffes (z.B. „age“) zeigt das Programm nach der Eingabe der gewünschten Kategorie (semantic tag) und der gewünschten Anzahl an Ergebnissen (limit), die entsprechenden Einträge aus SNOMED-CT.



```
Console  CSVRecord.class  JSONObject.class
startCTSNOMED [Java Application] C:\Program Files\Java\jre1.8.0_201\bin\javaw.exe (18.07.2021, 17:57:08)
SNOMED-CT Matching
-----

help - shows this information
show - shows the concepts of a term (example: age, stomach, heart...)
tags - shows semantic tags (finding)
save - save terms to file
exit - exits the application

Enter command: show
Enter term to show: age
Enter semantic tag: finding
Enter limit: 6
1.      Term: Ageing
        Concept ID: 248280005

2.      Term: Ageusia
        Concept ID: 36955009

3.      Term: Old-age
        Concept ID: 49808004

4.      Term: Teen-age
        Concept ID: 22396007

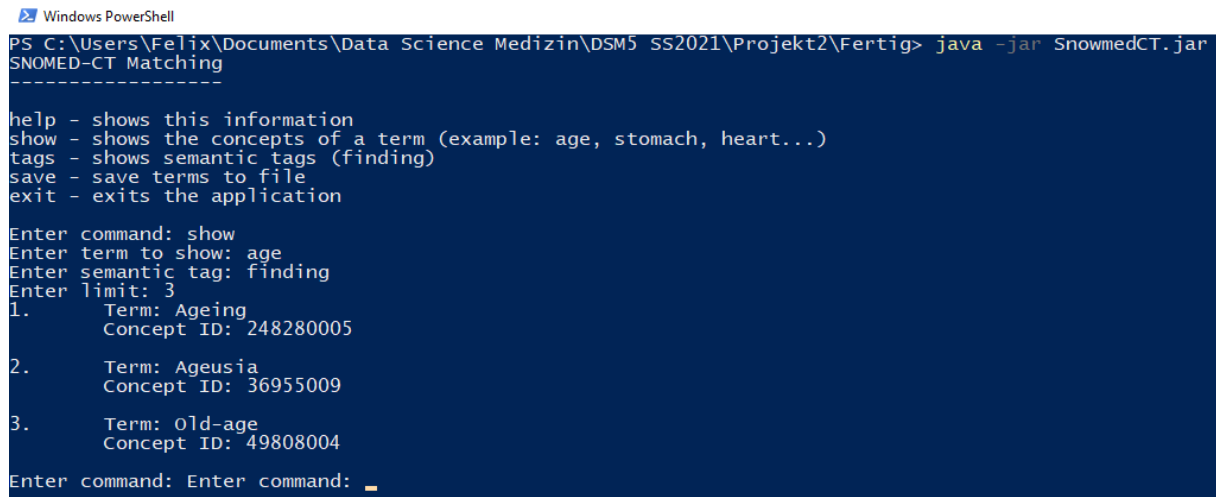
5.      Term: Ageustia
        Concept ID: 36955009

6.      Term: Middle-age
        Concept ID: 28288005
```

Abb. 3: Eclipse-workspace, Ausführung des Programms „SNOMED-CT Matching“

2.4.2. PowerShell

Um das Programm über die PowerShell laufen zu lassen, muss zunächst der Projektordner in Eclipse als .jar-Datei exportiert werden. Im Ordner mit der entsprechenden Datei muss die PowerShell aufgerufen werden. Sobald sich die PowerShell öffnet, kann mit dem Eintrag „java -jar SnowmedCT.jar“ das Programm gestartet werden. Die restliche Ausführung ist identisch wie in Eclipse.



```
Windows PowerShell
PS C:\Users\Felix\Documents\Data Science Medizin\DSM5 SS2021\Projekt2\Fertig> java -jar SnowmedCT.jar
SNOMED-CT Matching
-----
help - shows this information
show - shows the concepts of a term (example: age, stomach, heart...)
tags - shows semantic tags (finding)
save - save terms to file
exit - exits the application

Enter command: show
Enter term to show: age
Enter semantic tag: finding
Enter limit: 3
1.      Term: Ageing
        Concept ID: 248280005
2.      Term: Ageusia
        Concept ID: 36955009
3.      Term: Old-age
        Concept ID: 49808004
Enter command: Enter command: 
```

Abb. 4: PowerShell, Ausführung des Programms „SNOMED-CT Matching“

3. Beiträge der Gruppenmitglieder

3.1. Natalia Sokolov

Mein Name ist Natalia Sokolov und meine Aufgaben im Projekt dieses Semester waren Serververbindung aufzubauen und später das Erstellen von Dateien mit den Eingaben des Benutzers in der Konsole.

Die von mir erstellte Klasse „HttpClient.java“ baut eine Verbindung zu "https://snowstorm.test-nictiz.nl/MAIN/concepts?activeFilter=true" mithilfe von HttpURLConnection auf. Die Verbindung klappt auch, aber im Projekt verwenden wir die Verbindung mittels Unirest, die von den anderen Teampartnern erstellt wurde. Außerdem war ich für die Aufgabe zuständig, die Eingaben von dem Benutzer in eine Datei zu übernehmen. Dafür habe ich in der Klasse „CTmodel.java“ die Funktion public void ctDescriptions(...) mit dem FileWriter erweitert. Damit ist es möglich die gewünschten Ergebnisse des Benutzers in eine Datei zu übernehmen. Bevor die Funktion erstellt wurde, habe ich versucht an einem einfachen Beispiel in der Klasse „Datei.java“ Dateien zu erstellen. Dazu gibt es Commits in Github. Zu meinen besten Commits zählen diese beiden Aufgabenbereiche.

Für mich war es hilfreich, dass wir ein Arbeitsplan erstellt haben und schon am Anfang unseres Projekts versucht haben, große Aufgabenbereiche in kleinere Aufgaben aufzuteilen und zu beschreiben. Meistertask.com hat uns dabei sehr geholfen. Alle Aufgabenbereiche waren übersichtlich dokumentiert und man konnte auch sehen wer an welcher Aufgabe gearbeitet hat.

Die Kommunikation in einem Team hat auch große Bedeutung. Wir haben uns jede Woche online getroffen und haben über alle offenen Fragestellungen diskutiert und zusammen Lösungen gefunden.

3.2. Nha Vy Bui

Meine Hauptaufgabe bestand aus der Programmierung unseres Tools mithilfe des „MVC Pattern“ und den darin enthaltenen Methoden.

Am Anfang des Projektes erstellte ich die ihCCOntology.csv und implementierte die Datei in Java um den Zugriff auch unabhängig von dem lokalen Speicher zu ermöglichen. Anschließend habe ich gemeinsam mit Susie das MVC Pattern erstellt und die erste Methode programmiert um csv Dateien einzulesen. Durch Unirest haben Asli, Susie und ich anschließend weitere Methoden implementiert, um passende Matches nach bestimmten Kriterien aus SNOMED-CT zu erhalten. Matches konnten nun nach Semantic Tags gefiltert und von der Anzahl der dargestellten Begriffe begrenzt werden.

Es erfordert viel Kommunikation und eine ordentliche Strukturierung und Aufteilung von Aufgaben um eine Projektarbeit erfolgreich zu Planen. Meistertask hat es ermöglicht, verschiedene Aufgaben in große Aufgabenbereiche einzuteilen und übersichtlich darzustellen. Außerdem habe ich gelernt andere Dokumente in Java zu implementieren und mit externen Programmen zu arbeiten.

GitHub ist eine super und unkomplizierte Möglichkeit um z.B. neue Codes oder Verbesserungen schnell miteinander auszutauschen.

Mein bester Commit war die Klasse „CTmodel.java“. Unser Projekt ist erst durch die aktive und motivierte Mitarbeit aller Beteiligten mehr oder weniger ein Erfolg geworden. Deswegen bin ich auf alle Commits stolz die jeder erbracht hat.

3.3. Susie Golubowski

Für das Projekt „SNOMED-CT Matching“ umfasste sich mein Aufgabenbereich, in enger Gemeinsamkeit mit meinem Team, über das Verständnis der Struktur und Mappings der internationalen medizinischen Nomenklatur „SNOMED-CT“. Zudem strukturierte, plante und skizzierte ich mit meinen Teamkollegen Natalia, Nah Vy, Asli, Emre und Felix, wie unser Programm aussehen und funktionieren soll. Gemeinsam realisierten wir die Implementierungen entsprechend unseren Vorstellungen. Somit lag mein Hauptaufgabenbereich in der Implementierung.

In enger Zusammenarbeit mit Nha Vy Bui und Asli Sakiz, haben wir zunächst das Grundgerüst des Programms nach dem MVC-Pattern erstellt und daraufhin, das Einlesen, Abspeichern in die LinkedHashMap „map“ und die Ausgabe der vorliegenden „ihCCOntology_Excerpt.csv“ Datei implementiert (package -> readcsv, class -> CSVReaderWithHeaderAutoDetection2). Zudem fiel mit in unseren Aufgabenbereich die Herstellung der Serveranbindung zum SNOMED-CT Server mithilfe der Library „unirest-java-1.4.7.jar“. Mit dieser Verbindung ist es nun möglich nach beliebigen Begriffen (terms), mit passenden Filtern (semantic tag und limit), abzufragen und diese entsprechend auszugeben.

Mit aus dieser Projektarbeit nehme ich, die neuen gesammelten Kenntnisse über den Aufbau und der Strukturierung von SNOMED-CT, sowie auch die Verwendung der Library „Unirest“, zur Herstellung der SNOMED-CT Serververbindung. Des Weiteren war die Benutzung mit GitHub eine spannend Erfahrung. Auch wenn der Einstieg, mit der korrekten Installation und Einrichtung etwas „holprig“ war, ist es ein großartiges und angenehmes Tool und hat unsere Zusammenarbeit im Projektteam deutlich vereinfacht.

Mein bester Commit war „MVC - CTmodel & Unirest“ und der dazu gehörige Commit „Unirest Test 1“. Damit wurde die Serververbindung zu SNOMED-CT hergestellt, mithilfe der Library „unirest-java-1.4.7.jar“, sowie mit der entsprechenden Ausgabe der jeweiligen gesuchten

Begriffen („age“). Ansonsten bin ich sehr stolz auf mein Team und somit auf all unsere Commits, da jeder einzelne von uns aktiv mitgearbeitet und sich Mühe gegeben hat.

3.4. Asli Sakiz

Meine Rolle im Team war es, ein Programm zu schreiben mit meinen Teamkollegen Emre, Felix, Susie und Vy, welches helfen soll, die besten Begriffe in der Ontologie in SNOMED-CT zu finden. Außerdem habe ich auch die Funktionalitäten getestet.

Am Anfang war ich für die Zuordnung der Begriffe zuständig. Dabei habe ich mir die Csv-Datei angeschaut und für die Begriffe die passenden Findings aus SNOMED-CT raus geschrieben. Anschließend habe ich die Klasse „Ctvo“ erstellt, die leider nicht mit integriert wurde. Ich habe ebenso mit meinen Teamkollegen (Vy und Susie) zusammen gearbeitet und die Unirest und das CTModel implementiert.

Da in einem Projekt auch das Projektmanagement eine wichtige Rolle spielt, haben wir uns zunächst einen Arbeitsplan mithilfe der Internetseite Meistertask.de erstellt. Wir haben uns realistische Ziele gesetzt und konnten sie umzusetzen. Uns war bewusst, dass wir nicht alle Funktionalitäten implementieren können.

Zudem konnte ich Erfahrungen mit der Version Control Systems (git) sammeln. Und ich habe die Build Tools maven/poetry kennengelernt. Es fiel mir am Anfang schwer mit maven zu arbeiten, da es manchmal nicht auf Anhieb geklappt hat und Fehler angezeigt wurden. Als wir das Einrichten nach mehrmaligen Versuchen geschafft haben, hat es dann Spaß gemacht alle Librarys zu importieren und mit dem Programmieren loszulegen. Mir fällt das programmieren aber dennoch schwer und mir war bewusst, dass ich nicht alleine einen Aufgabenteil übernehmen kann, und deshalb bin ich dankbar, dass wir auch in Teams arbeiten konnten und einiges zusammen committed haben.

Die Commits waren alle sehr anspruchsvoll und ich finde, dass wir alle gut hingekommen haben.

3.5. Emre Baylas

Meine Rolle im Team war in diesem Semester das Programmieren vom CTSNOMED mit meinen Teamkollegen Asli, Felix, Susie, Nha Vy und Natalia. Zudem habe ich das Programm auf seine Funktionalität geprüft und war für View zuständig.

Als wir das Projekt gestartet haben, war meine Aufgabe zu dieser Zeit, die Begriffe zu zuordnen. Später war ich für die Ausgabe in eclipse verantwortlich und habe die Klasse „CTView.java“ erstellt, damit wir eine Oberfläche haben, die Ordnungsgemäß mit dem ganzen Projekt zusammenarbeitet.

In einer Projektarbeit ist das organisatorische Vorgehen am wichtigsten. Wir haben die Aufgaben auf Meistertak.com aufgeteilt. Unsere zielorientierte Arbeitsweise hat uns geholfen, die geforderten Ergebnisse bestmöglich im gesetzten Zeitplan einzuhalten. In einer Projektarbeit ist die Kommunikation sehr wichtig, um den Stand des Projekts am laufenden zu halten. Wenn es Probleme mit dem Programm gab, konnte man sich jederzeit mit Ihnen in Verbindung setzen und offene Fragen klären. Ich habe neue Programme kennengelernt wie z. B. „Slack“ und „GitKraken“.

Mein bester Commit war die Klasse „CTView.java“.

Wir sind auf jeden Commit sehr stolz, weil jeder in der Gruppe seine Arbeit zum Projekt beigetragen hat.

3.6. Felix Mattes

Mein Aufgabenbereich umfasste die grobe Planung des Programms, insbesondere was den Aufbau der View betrifft. Zu diesem Zweck habe ich die Klassen „Prototyp“ und „Mapping“ implementiert.

Auch wenn sich seitdem das Programm verändert hat und die Klassen in ihrer ursprünglichen Form nicht mehr relevant sind, zähle ich diese zu meinen besten Commits, da sie weitere Implementierungen maßgeblich beeinflusst haben.

Des weiteren war ich an der Problemlösung für Fehler an der CTVView beteiligt. Während der Nutzung des Programms, sind in der Konsole ständig Debug- und Verbindungsmeldungen aufgetaucht, was die Nutzung störend und benutzerunfreundlich gestaltet hat. Durch die logback.xml wurde dieses Problem behoben.

Aus dem Projekt habe auch auf technischer Seite mitgenommen, wie beispielsweise Github, die Arbeit am Projekt erleichtert und wie eine übersichtliche Historie zum Verständnis beiträgt, was beispielsweise ein Gruppenmitglied hinzugefügt hat. In sozialer Hinsicht wurde mir klar, wie wichtig Kommunikation im Team ist und unterschiedliche Ideen das Projekt bereichern können.