Protokoll

Lernstandsgespräch LF9

**Datum**: 13.03.2025

**Gruppenmitglieder**: Lennart Dreier, Patrick Jenkner, Jörn Hiller, Jascha Mundt

**Gruppenmitglieder anwesend**: alle

**Protokollführer**: Lennart Dreier

# Was wurde seit dem letzten Gespräch geschafft?

**Aufgabe 1:** Wir haben die Aufgabe in Arbeitsteilung über GitLab bearbeitet: Nach einer kurzen gemeinsamen Besprechung haben Lennart, Jascha und Jörn jeweils für eine der Klassen die Statements implementiert; und Patrick hat sie überprüft, reviewt & gemerged.

**Aufgabe 2:** Die setEntity Methoden wurden für die Klassen Priority und Project schnell implementiert. Für die Task-Klasse haben wir länger gebraucht, da die Erzeugung und Verknüpfung von Project und Priority umständlich war.

**Aufgabe 3:** Die Umwandlung zwischen JSON und Objekten wurde umgesetzt. Das JSONObject von Java war für uns neu, aber nach kurzer Einarbeitung konnten wir damit ohne Probleme arbeiten.

# An was arbeiten Sie gerade?

Einige unserer Methoden benötigen noch besseres Exception Handling: Statt einem allgemeinen Catch wollen wir hilfreiche Fehlermeldungen für verschiedene Arten von Exceptions ausgeben.

Es sind noch nicht alle Methoden für alle Klassen getestet worden.

# Wo sind aktuell ihre Probleme?

Die Implementierungen für die Task-Klassen sind deutlich umständlicher als für die anderen beiden Klassen, da die assoziierten Objekte mit beachtet werden müssen.

Da manche aus unserer Gruppe vor diesem Berufsschulblock noch nicht mit Git und GitLab gearbeitet haben, müssen wir erstmal einen Exkurs machen, wie wir mit diesen Tools arbeiten wollen. Anschließend müssen wir uns darauf einigen, wer auf dem GitLab Profil die Merge-Requests reviewen wird.

# Was wurde seit dem letzten Gespräch geschafft?

Die Aufgaben 1-3 haben wir abgeschlossen. In Aufgabe 4 haben wir das Delete Statement fertig gestellt. Das Get Statement und Post Statement ist in Arbeit. Das Post Statement ist bis auf Task fertig.

# An was arbeiten Sie gerade?

Aktuell sind wir dabei, die in dem Problem Statement definierten Probleme zu beheben. Nebenbei prüfen wir gegenseitig unsere Merge-Requests und geben uns Tipps, was am Code verbessert werden kann. Auch diskutieren wir darüber, welche Herangehensweise die beste ist, um die Probleme zu beheben.

"Die POST-Methode für Tasks (Tasks sind immer mit einer Priorität und einem Projekt verknüpft) funktioniert derzeit nicht wie gewünscht. Obwohl im Payload die Felder title, date, projectid und priorityid übergeben werden, werden in unserer Datenbank an diesen Stellen nur NULL-Werte gespeichert, statt der übermittelten Daten."

Da wir besonders effizient arbeiten wollen und auch jeder etwas zu tun haben soll, haben wir unsere vierergruppe in zwei zweiergruppen aufgeteilt, während Patrick und Jörn an der Implementierung und Fehlerkorrektur der POST-Methode arbeiten, befassen sich Jascha und Lennart mit den Methoden, welche für das Read- und Get-Statement verantwortlich sind.

*Zuletzt wurde an restlichen Fehlern gearbeitet. Für korrektes Error Handling sind noch Return codes erforderlich, wenn es bei einer Anfrage kein Ergebnis in der Datenbank gefunden wird*

*Das Akzeptieren von null values für Fremdschlüssel in der Klasse Tasks ist noch erforderlich.  
Mittlerweile haben wir dies implementiert, indem wir bei der Erstellung des JSONObjects bereits NULL-Werte einfügen.*

# Wo sind aktuell ihre Probleme?

-Das erste Problem, was bei uns diesen Block aufgetaucht ist, waren Probleme mit der Kommunikation mit dem Server. Wenn wir einen Request an den Server gestellt hatten, haben wir keine Antwort enthalten, weil wir die erwarteten Antworten noch nicht implementiert hatten. Dies wurde gelöst, in dem die benötigten Methoden implementiert wurden.

- Wir haben das Problem, dass die POST-Methode den Body des Requests zwar entgegennimmt, die Ausgabe jedoch nicht wie gewollt formatiert wird, da die geschweiften Klammern {} fehlen. Dadurch entsteht kein schönes JSON, das den Anforderungen entspricht. Durch weitere Formatierung wurde dies gelöst.

-Auch müssen wir uns damit beschäftigen, wie wir die letzte Zahl in der URI, also die ID, extrahieren können, da wir diese für die PUT, DELETE und CREATE Funktionen benötigen.

- Wenn ein POST-Request gestellt wird, wird die ID mit dem Wert 0 vergeben, da das Auto-Increment anscheinend nicht korrekt funktioniert. Beim erneuten Senden eines POST-Requests tritt ein Fehler auf, da erneut die ID 0 vergeben wird, diese jedoch bereits existiert.

-Immer wenn eine Datenbank Connection aufgebaut wird, wird sie nicht geschlossen.

-Wir hattet einen Merge-Konflikt als wir die beiden feature Branches von unserer aufgesplitteten Gruppe miteinander mergen wollten. Das müssen wir noch beheben.

-Error wenn READ-Request gestartet wird “org.sqlite.SQLiteException: [SQLITE\_ERROR] SQL error or missing database (no such table: project)”, an der Lösung dieses Problemes wird auch aktuell gearbeitet.

*-Bei einem Post Request wird nicht das angegebene Datum nicht korrekt in die Datenbank geschrieben. Obwohl das angegebene Datum als Dateformat in Java erkannt wird, erscheint ein anderes in der Datenbank.*

*Dies wurde gelöst, in dem bei der Erstellung des SQL Statements für den Post Request fehlende Anführungszeichen eingefügt wurden.*

*-Die Rückgabe eines 404 wenn eine Priority nicht gefunden wird hat zunächst nicht funktioniert, da selbst eine „leere“ Priority das Tupel „value=0“ enthält und die Rückgabe somit nicht leer ist.*

*- Das Error Handling (also die Zurückgabe von einem 404 falls ein Eintrag nicht vorhanden ist) funktioniert mittlerweile für GET Requests und für PUT Requests. Es fehlt nur noch die Implementierung für DELETE Requests.*