



TECHNISCHE UNIVERSITÄT
CHEMNITZ

CRC Cards

Softwaretechnikpraktikum - Meilenstein 2

Fakultät für Informatik
Professur Softwaretechnik

Eingereicht von: Gruppe 5
Einreichungsdatum: 11.12.2022

Betreuerin: Prof. Dr. Janet Siegmund
Betreuer: Dominik Gorgosch

Zusammenfassung

Auch in Anbetracht des zweiten Meilensteins haben wir unsere wöchentlichen Meetings fortgesetzt und untereinander über den aktuellen Stand unserer Arbeit in Kenntnis gesetzt. Wir haben uns wieder für dieselbe Aufteilung in vier Zweiergruppen, eine für je ein Systemteil, entschieden. Dabei sollte jede Zweiergruppe zusätzlich versuchen, auch das Gesamtsystem in Betracht zu ziehen sowie mögliche Kollaborationen mit anderen Systemteilklassen aufzustellen.

Für das Programmierprojekt haben wir eine Aufteilung in Front- und Backend eingeführt und jeweils den Verantwortlichen zugeteilt. Max übernimmt das Backend und Simon das Frontend. Demzufolge gab es zusätzlich zu den wöchentlichen Teambesprechungen auch Meetings in der jeweiligen Front- bzw. Backend-Gruppe. Dabei teilten die Verantwortlichen in ihrer Gruppe erste Aufgaben zu und fragten den Fortschritt regelmäßig ab.

Frontend:

- Constantin übernimmt die Landingpage.
- Simon erstellt die Navigationbar.
- Andreas fertigt einen Footer an.
- Jana erarbeitet die "404 Not Found" Seite.

Backend:

- Max kümmert sich um die Schnittstelle zwischen Backend und Frontend sowie die Datenbank.
- Martin implementiert einen Docker-Container für die Auswertung der Aufgaben, die ausgeführt werden müssen.
- Sami kümmert sich um das Login-Backend.
- Louis implementiert die Auslieferung des JWT.

Wir nutzen dabei das Kanban-Board von GitHub. In den Meetings definieren wir neue Aufgaben für das Backlog und setzen Deadlines für die Aufgaben.

Dabei sind folgende Probleme bisher aufgetreten:

Ursprünglich hatten wir uns dafür entschieden, einen Develop-Branch zu erstellen und von diesem abzuspalten, wenn man ein Feature implementieren möchte. Dieser sollte dann nach jedem Meilenstein in den Main-Branch gemergt werden. Analog dazu sollte jedes fertig-gestellte Feature in den Develop-Branch gemergt werden. Dabei wurden manche Feature-Branche vergessen und nicht gelöscht, unnötige Commits wurden nicht kombiniert und das Mergen in den Main-Branch wurde nicht wie zunächst vorgenommen durchgeführt. Das sorgte in der Gruppe für Verwirrung und verschlechterte die Übersichtlichkeit.

Sowohl im Backend als auch im Frontend gab es Fortschritt in den zu bearbeitenden Aufgaben. Beim Testen des Gesamtsystems haben wir allerdings gemerkt, dass unsere derzeitigen Fortschritte im Backend und Frontend nicht von großer Bedeutung sind, wenn die Grundstruktur im Softwareprojekt nicht richtig gesetzt ist. Der Ordner für die kompilierten Svelte-Dateien war nicht abgesprochen und führte zu einem nicht funktionierenden System. Hieraus konnten wir schließen, wie wichtig eine Grundstruktur in einem Softwareprojekt ist.

Beim Zusammentragen der CRC-Karten ist aufgefallen, dass einige, zunächst unterschiedlich wirkende Karten trotz der unterschiedlichen Aufgabenbereiche die gleichen Klassen repräsentierten. Somit hatte fast jede Zweiergruppe eine CRC Karte zur Abstraktion der Datenbank. Auch eine abstrakte Klasse für eine Aufgabe kam dreimal vor. Zudem stand oft die Frage im Raum, ob die Klasse im Frontend oder Backend implementiert wäre. Letztlich hat sich ergeben, dass die meisten CRC-Karten für das Backend erstellt wurden. Dies könnte daran liegen, dass man bessere Abstraktionen erstellen kann und man sich im Frontend auf eine Bibliothek stützt, die schon viele Abstraktionen abnimmt.

Bei der Auswertung unserer CRC-Karten haben wir zunächst alle unsere Karten auf Papier geschrieben, da es uns somit leichter gefallen ist, eine Kategorisierung zu erstellen. Nachdem wir untereinander unsere Karten vorgestellt haben, haben wir die Karten grob in Aufgabensystem/Auswertungssystem, Loginsystem und Administration/Verwaltungssystem eingeteilt. Redundante Karten oder Klassen, die die gleichen Verantwortlichkeiten aufwiesen, wie oben beschrieben, haben wir zusammengefasst.

CRC Cards

Database	-
create, read, update, delete exercise data	AbstractExercise
create, read, update, delete user data	AbstractUser

Server	-
serve web pages and related files	Database
provide API interface	AbstractUser
translate API requests into SQL queries	AbstractExercise

AbstractUser	-
reflect database structure	Database
provide database interface	Server
define API route	

User	AbstractUser
solve exercises	AbstractExercise

Administrator	AbstractUser
manipulate exercise data	Database
evaluate solutions	ExerciseCreator
	AbstractExercise
	User
	LearningProgress

AbstractExercise		-
reflect database structure provide database interface define API route		Database Server
GapTextExercise		AbstractExercise
SyntaxExercise		AbstractExercise
ParsonsPuzzleExercise		AbstractExercise
FindTheBugExercise		AbstractExercise
DocumentationExercise		AbstractExercise
OutputExercise		AbstractExercise
ProgrammingExercise		AbstractExercise
ExerciseTracker		-
track solving time		AbstractExercise
ExerciseEvaluator		-
evaluate solution store solution attempt and solving time		SourceCodeEvaluator AbstractExercise ExerciseTracker Database Administrator
SourceCodeEvaluator		-
parse user code execute user code in sandbox		ExerciseEvaluator AbstractExercise ExerciseTracker Database Administrator

LearningProgress -	
assemble learning progress create graphical representation	Database User AbstractExercise
ExerciseCreator -	
create exercise with hints and solution	Database AbstractExercise
HomePage -	
provide login	AbstractUser LoginArea
LoginArea -	
display components	TextField LoginButton SignUpButton ForgotPasswordButton
TextField -	
accept user input obfuscate password input	
LoginButton -	
trigger login process	Server
SignUpButton -	
trigger user creation process	Server
ForgotPasswordButton -	
trigger password reset process	Server