PS Softwareentwicklung und Projektmanagement

Code-Review

Reviewteam: 1A

Janine Pallhuber: 1315063

Johannes Ehrhart: 1517554

Softwareteam: Team 1

Proseminargruppe: Gruppe 4

Datum: 19.05.2017

Zusammenfassung

Trotz mehreren Versuchen sowohl über eine Entwicklungsumgebung also auch über die Command Line, war es uns nicht möglich die Anwendung zu starten. Durch diesen Umstand war das Testen der Applikation stark eingeschränkt.

Alles in Allem kann man sagen, dass sich das Team in der finalen Phase der Entwicklung des Backends befindet und ihre bisherige Leistung beachtlich ist.

Architektur

Die Architektur aus dem Konzept wird sehr gut im Code umgesetzt. Die einzelnen Komponenten aus dem Systemkonzept gehen klar aus dem Code hervor. Man hätte eventuell die einzelen Komponenten in Service und Controller noch einmal wie im Konzept in Admin, Parent, Employee aufteilen können. Das würde die Übersicht noch einmal verbessern. Im großen und ganzen ist die Umsetzung von Konzept zu Code sehr gut gelungen.

Man sieht auch auf einen Blick in welchen Komponenten das MVC-Pattern umgesetzt wird. Da die Konvention bei der Namensgebung sehr gut eingehalten wurde. Eine Klasse, die ScheduleView, sollte noch umbenannt werden zu ScheduleController. Die Schnittstelle zwischen den Daten und dem Controller befindet sich in der Service Komponente. Dadurch entsteht ein sauberer Zugriff von der Oberfläche hin zur Datenbank und wieder zurück.

Laut Zeitplan sollten sich alle in der Finalisierung des Systems befinden. So weit ist das Projekt noch nicht fortgeschritten, aber man ist auch nicht weit vom Zeitplan entfernt.

Code

Der Code ist aufgrund vernünftiger Benennung der Klassen- und Variablennamen verständlich gestaltet. Außerdem werden teaminterne Konventionen wie die Implementierung der Getter und Setter Methoden eingehalten.Unter unser möglichen Betrachtung konnten wir kein redundantes Codesegement finden. Die Größe der Klassen ist angemessen und auch gut struktruriert. Weiters haben die Methoden eine bestimmte Aufgabe die sie erfüllen und sind nicht mit Aufgaben überladen. Auf Design-Patterns wurde bis jetzt noch verzichtet. Des weitern gibt es keine Globalen Variblen und auch keine Gottklassen.

Die Methode getId, die sich in mehreren Klassen befindet (zb. Child etc.), sollte noch implementiert werden. Dies wurde bei jedem Aufruf bereits dokumentiert und wir möchten lediglich nochmals darauf hinweisen. Das Audit Log wurde adäquat implementiert, jedoch könnte noch mehr geloggt werden, z. B. könnte auch die Zuteilung der Tasks gelogt werden.

Sicherheit

Einige Unterordner innerhalb des Webapp Ordners werden nicht von den „antMatchers“ in der WebSecurityConfig erfasst. Wie zum Beispiel die Ordner parent, employee, calendar, general und nursery. Dies stellt ein Sicherheitsrisiko dar, da man diese Ordner ohne Authentifizierung erreichen kann.

Ein Teil der Eingabedaten an der Oberfläche werden durch Constraints geprüft. Es werden auch eigene Constraint-Klassen deklariert, aber teils nicht angewandt. Man könnte die Constraints noch erweitern, da z. B. beim Anlegen von einem neuen Mitarbeiter nicht überprüfft wird, ob der Username schon vergeben ist. Dies wird dazu führen dass in der Datenbank eine Exception geworfen wird, die nicht behandelt wird.

Dokumentation

Der Code wurde im großen und ganzen gut kommentiert. Ein kleiner Störfaktor ist die Dokumentation über die Erstellung der Klassen. Da das Team sich entschieden hat den Autor jeder Klasse preiszugeben, wäre ein einheitliches Format angebracht. Als Beispiel möchten wir hier die Erstellung durch Stefan Mattersberger nennen, all seine Klassen wurden mit Namen, Mail-Adresse und Datum versehen. Dann gibt es noch Klassen die von zerus oder root erstellt wurden und hier ist nicht klar, wer genau das sein soll. Teilwiese existieren leere Kommentarblöcke, die nur der Optik etwas schaden.

Sowohl Corner Cases als auch Libraries wurden in keiner Klasse dokumentiert, jedoch gehen die Corner Cases aus dem Code teilweise hervor und bezüglich der Libraries ist auch klar was verwendet wurde und wo. Auskommentierten Code gibt es in den Klassen, die von den Teammitgliedern erstellt wurde, nicht.

Die beste Dokumentation im Code befindet sich bei den Tests. Es ist sehr hilfreich zu wissen was getestet wird und wie. Großes Lob gilt auch der Dokumentation im Audit Log, da man auf den ersten Blick erkennt, was gelogt wird und nicht noch extra in jeder Klasse nachsehen muss.

Tests

Tests wurden als JUnit Tests implementiert, jedoch waren auch diese nicht ausführbar.

Mängelauflistung

Art	**Beschreibung**
grob	Applikation lässt sich nicht ausführen
kosmetisch	Komponenten aufteilen
kosmetisch	ScheduleView umbenennen in ScheduleController
mittelschwer	getId implementieren

kosmetisch	Audit Log ausbauen
grob	Ordner sind frei zugänglich
mittelschwer	Constraint-Klassen werden nicht angewandt
kosmetisch	Dokumentation bei Erstellung einer Klasse vereinheitlichen
grob	Tests nicht auführbar