

Projekt M326

Inhaltsverzeichnis

Inhaltsverzeichnis

Inhaltsverzeichnis	2
Organisation	3
Analyse & Design	4
<i>Architektur</i>	4
<i>Klassendiagramm</i>	5
<i>ERD</i>	6
<i>Userstories</i>	7
<i>Hinweise zum Code</i>	7
Testing	7
<i>Testfälle (UAT)</i>	8
<i>Testprotokoll</i>	11
Bestätigung der Eigenleistung aller Teammitglieder	12

Organisation

Team: Benjamin Raison, Jonas Gredig

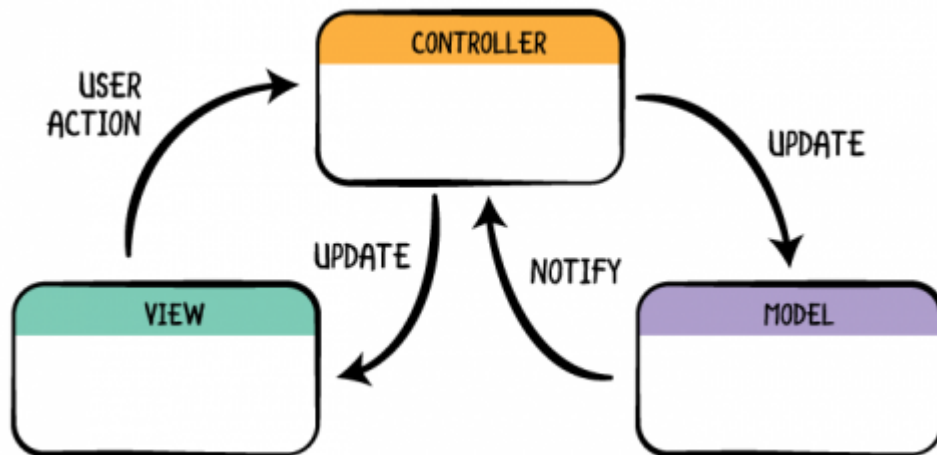
Zuständigkeiten, Terminplan (Milestones), erledigte Dinge und andere Teamplanungsbezogene Dinge auf dem ScrumBoard auf GitHub ersichtlich. Das Tagebuch ist in der GitHub-Repository unter „documentation/daily_goals“ zu finden.

ScrumBoard: <https://github.com/BenjaminRaison/M326-project/projects/1>

Analyse & Design

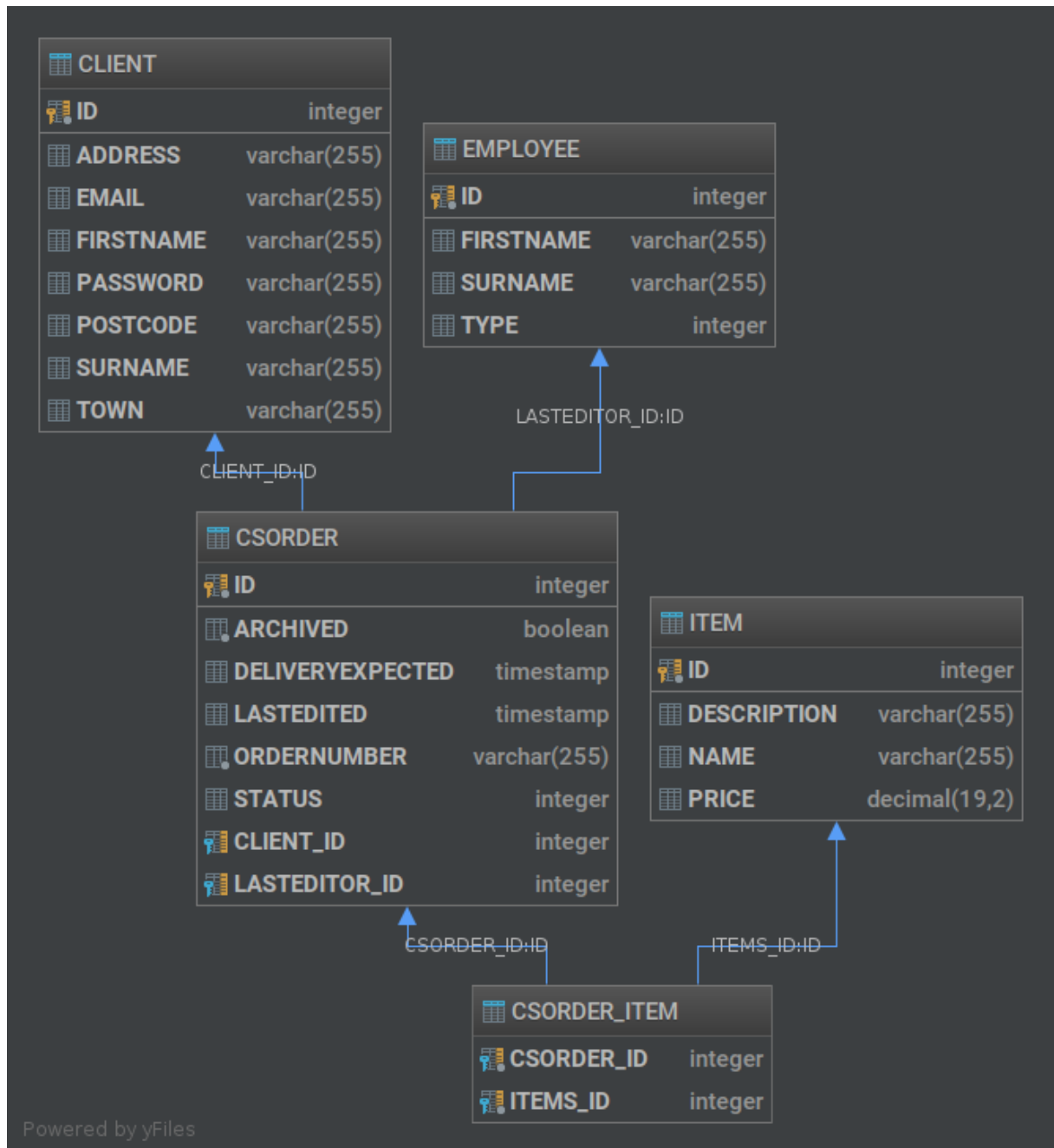
Architektur

Da wir uns an das Prinzip „Keep it simple, stupid“ halten, verwenden wir als generelle Architektur MVC (Model View Controller). Um von der View auf die Datenbank zu zugreifen, muss man das ganze über einen Controller aufrufen. Wo es Sinn macht, haben wir auf diese Architektur verzichtet.



Klassendiagramm

ERD



User stories

1. Als Benutzer möchte ich mein Konto auswählen.
2. Als Vertriebler möchte ich alle neuen Bestellungen sehen.
3. Als Lieferant möchte ich alle versandbereiten Lieferungen sehen.
4. Als Logistiker vom Lager möchte ich alle aufbereiteten Bestellungen sehen.
5. Als Benutzer möchte ich den Status einer Bestellung anpassen.
6. Als Benutzer möchte ich das Lieferdatum anpassen.
7. Als Benutzer möchte ich, dass sich das „Zuletzt angepasst“ aktualisiert, wenn ich etwas an einer Bestellung ändere.
8. Als Benutzer möchte ich auf die Bestellung aufteilen.
9. Als Benutzer möchte ich eine Detailansicht der Bestellung sehen.

Hinweise zum Code

Wir haben die komplette Architektur auf Grund der unnötigen Komplexität erneuert. Da unsere Testdaten nicht sehr umfangreich sind, wird die Datenbank beim Aufstarten jedes Mal neu erstellt und mit Daten abgefüllt.

Javadoc haben wir nur dort geschrieben, wo der Name der Methode nicht genug selbstsprechend ist.

Testing

Zusätzlich zu den UAT haben wir die wichtigste Funktionalität mit Unit- und Integrationstests getestet.

Testfälle (UAT)

Benutzer auswählen

Test Nr.	1
Vorbedingung	Ausführbare JAR ist vorhanden Java ist installiert
Testschritte	Applikation starten Oben rechts Benutzer auswählen
Erwartetes Ergebnis	Der Benutzer wird geändert und die Tabelle verändert sich dementsprechend

Daten werden angezeigt happy

Test Nr.	2
Vorbedingung	Ausführbare JAR ist vorhanden Java ist installiert Testdaten sind in der Datenbank
Testschritte	Applikation starten Benutzer auswählen
Erwartetes Ergebnis	Es sind die Felder Bestell Nr., Kunde, Status, Lieferung, Zuletzt bearbeitet und Details und darin die Daten der Bestellungen zu sehen

Daten können nach den Spalten sortiert werden

Test Nr.	3
Vorbedingung	Ausführbare JAR ist vorhanden Java ist installiert Testdaten sind in der Datenbank
Testschritte	Applikation starten Benutzer auswählen Pfeil bei Nr., Kunde, Status, Lieferung, Zuletzt bearbeitet klicken
Erwartetes Ergebnis	Die Bestellungen in der Tabelle lassen sich sortieren.

Details werden angezeigt

Test Nr.	4
Vorbedingung	Ausführbare JAR ist vorhanden Java ist installiert Testdaten sind in der Datenbank
Testschritte	Applikation starten Benutzer auswählen auf Details einer Bestellung klicken
Erwartetes Ergebnis	Ein 2. Fenster taucht auf mit den Details

Bestellung splitten happy

Test Nr.	5
Vorbedingung	Ausführbare JAR ist vorhanden Java ist installiert Testdaten sind in der Datenbank
Testschritte	Applikation starten Benutzer auswählen auf Splitten von einer Bestellung klicken
Erwartetes Ergebnis	Die Bestellung wurde gesplittet. Es gibt jetzt mehrere Bestellungen

Bestellung splitten sad

Test Nr.	6
Vorbedingung	Ausführbare JAR ist vorhanden Java ist installiert Testdaten sind in der Datenbank Testbestellung mit nur einem Item ist in der Datenbank
Testschritte	Applikation starten Benutzer auswählen
Erwartetes Ergebnis	Bei der Bestellung die nur aus einem Ding besteht ist kein Knopf für das Splitten.

Status Bestellung anpassen

DOKUMENTATION M326

Test Nr.	7
Vorbedingung	Ausführbare JAR ist vorhanden Java ist installiert Testdaten sind in der Datenbank
Testschritte	Applikation starten Benutzer auswählen Auf Status einer Bestellung klicken Einen anderen Status auswählen Enter Taste
Erwartetes Ergebnis	Der Status wurde abgespeichert

Status Bestellung anpassen

Test Nr.	8
Vorbedingung	Ausführbare JAR ist vorhanden Java ist installiert Testdaten sind in der Datenbank
Testschritte	Applikation starten Benutzer auswählen Auf die Lieferung einer Bestellung klicken Ein anderes Gültiges Datum eingeben Enter Taste
Erwartetes Ergebnis	Die Änderung wurde abgespeichert

Testprotokoll

Applikation	M326		Tester	Jonas Gredig
Version	1.0		Datum/Zeit	03.07.18 05:18 Uhr Morgens
Commit hash	4ae22695a08ee8bb 0554feeea0cb395d 25c5a93a			
Nr.	Effektives Resultat		Bewertung	Bemerkung
	Ausgabe			
1	Die Tabelle ändert sich		ok	
2	Testdaten werden angezeigt		ok	
3	Die Spalten sind sortierbar		ok	
4	Details werden angezeigt		ok	
5	Splitten funktioniert		ok	
6	Button ist ausgeblendet		ok	
7	Daten wurden gespeichert		ok	
8	Daten wurden gespeichert		ok	

Fazit: Das Programm hat die Tests bestanden!

Bestätigung der Eigenleistung aller Teammitglieder

Hiermit bestätigen wir, dass das Projekt unsere eigene Leistung ist.

Benjamin Raison

Jonas Gredig