Projekt M326

# Inhaltsverzeichnis

Inhaltsverzeichnis

[Inhaltsverzeichnis 2](#_Toc518373728)

[Organisation 3](#_Toc518373729)

[Analyse & Design 4](#_Toc518373730)

[Architektur 4](#_Toc518373731)

[Klassendiagramm 5](#_Toc518373732)

[ERD 6](#_Toc518373733)

[Userstories 7](#_Toc518373734)

[Hinweise zum Code 7](#_Toc518373735)

[Testing 8](#_Toc518373736)

[Testfälle (UAT) 8](#_Toc518373737)

[Testprotokoll 11](#_Toc518373738)

[Bestätigung der Eigenleistung aller Teammitglieder 12](#_Toc518373739)

# 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>

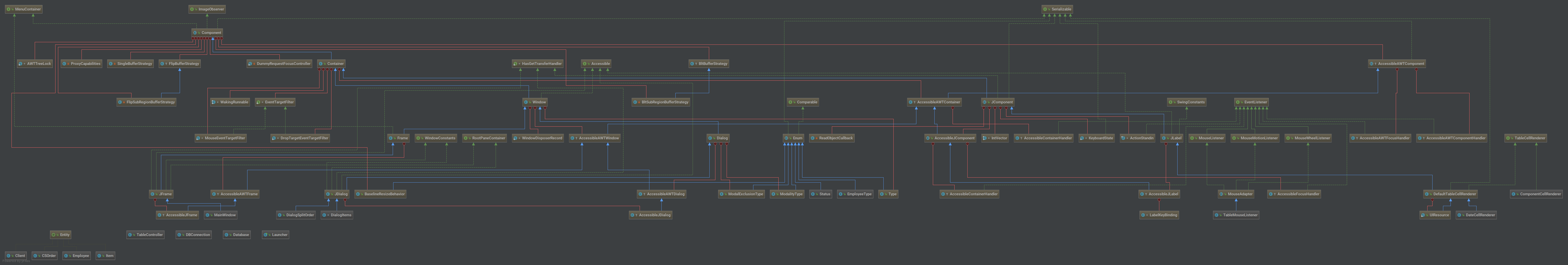
# Analyse & Design

## Architektur

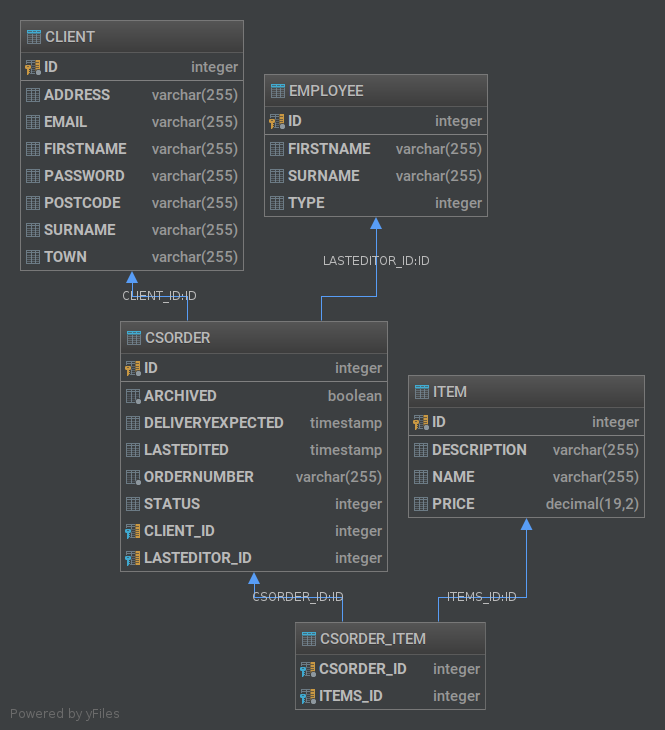
Da wir uns an das Prinzip „Keep it simple, stupid“ halten, verwenden wir als generelle Architektur MVC (Model View Controller). Um von der View auf die Datenbank zu zugreifen, muss man das ganze über einen Controller aufrufen. Wo es Sinn macht, haben wir auf diese Architektur verzichtet.



## 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. Esgibt 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**

|  |  |
| --- | --- |
| 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  Commit hash** | 1.0  4ae22695a08ee8bb0554feeea0cb395d25c5a93a |  | **Datum/Zeit** | 03.07.18 05:18 Uhr Morgens |
| **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