

Guten Appetit

Team Mitglieder:

Juma Assaf, George Wakji,
Mohammad Assaf, Mahmoud Matar.

1. Projekt-Idee

Im Rahmen der Module "Java 1 & 2" haben wir ein Lieferdienst-Software entwickelt. Die Software heißt **Guten Appetit**. Die ist eine Lieferdienst-Webseite, die viele gut programmierte Optionen enthält, mit denen Sie schnell und einfach orientalische Speisen bestellen können.

2. Ausgangssituation

2.1. Musskriterien:

- sich registrieren können. (Stammdaten sowie Passwort für neuen Benutzer) anlegen.
- sich einloggen können, wobei der User Name ist die E-Mail-Adresse.
- Stammdaten verwalten.
- sein Kundenkonto verwalten können
- Ein beliebiges verfügbares Produkt bzw. Speisen auswählen können.
- online Produktdetails einsehen können.
- Kontakt zum Support herstellen können.
- zwischen verschiedenen Zahlungsoptionen auswählen können.
- Zahlungsmethoden verwalten (Hinzufügen bzw. ändern)
- Rechnungsübersicht haben können.
- Status der Bestellung erfahren können.
- Die bestellten Produkte einsehen können.

- Alle Informationen rund um die Bestellung ansehen können.

Code-Dokumentation

Für Bezeichnung wurden nur ASCII letters and digits verwendet.

Die Code-Dokumentation sowie die Kommentare sind auf Englisch.

Leere Zeilen Vermeiden.

Next line Style verwendet.

Bezeichnung:

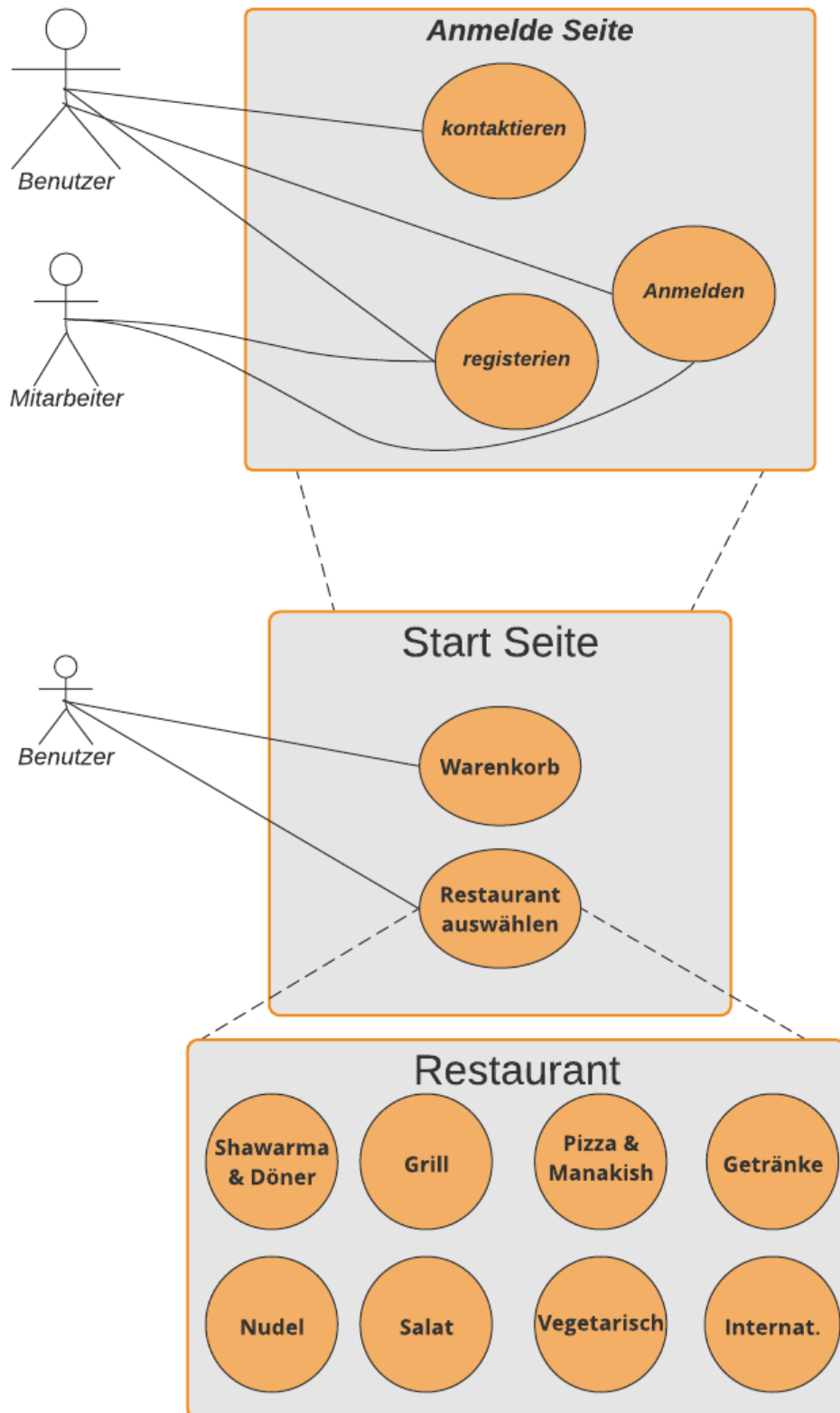
- Klassen-Bezeichnung
 - Klassen wurden nach UpperCamelCase- Schreibweise benannt.
 - TestKlassen-Namen fangen mit dem Wort „Test“ und endet mit dem Klassen-Namen.
- Methoden-Bezeichnung
 - Methoden wurden nach lowerCamelCase- Schreibweise benannt. (Beispiel: setOrderDate).
 - Bezeichnung der Methoden ist verbalisiert sprich die Bezeichnung ist ein verbaisierter Satz, wecher erklärt, warum die Methode existiert, was sie tut und wie sie benutzt wird. (Beispiel: getEmail).
- Parameter und Local Variablen-Bezeichnung
 - lowerCamelCase- Schreibweise.

Usability

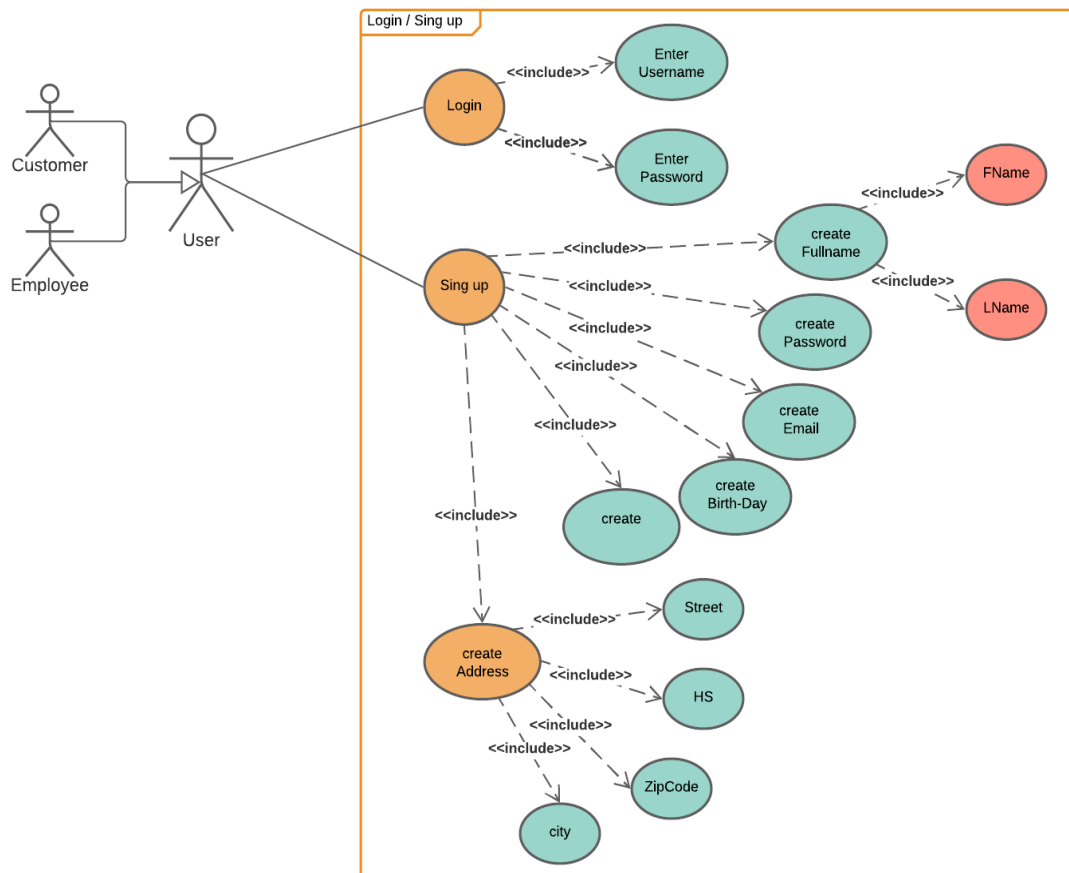
Die Webseite ist so zu gestalten, dass sie von verschiedenen Geräten mittels immer optimal dargestellt werden.

Das System muss bei fehlerhaften Benutzereingaben stabil bleiben und aussagekräftige Fehlermeldungen erzeugen.

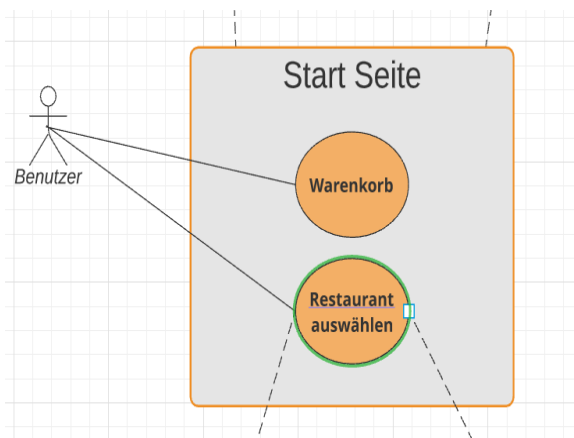
Der unter dargestellten Use-Case soll eine grafische Darstellung für die innere Struktur die Webseite.



Im Folgenden wird das Login bzw. Registrierung mithilfe eines



Diagrammes erklärt:

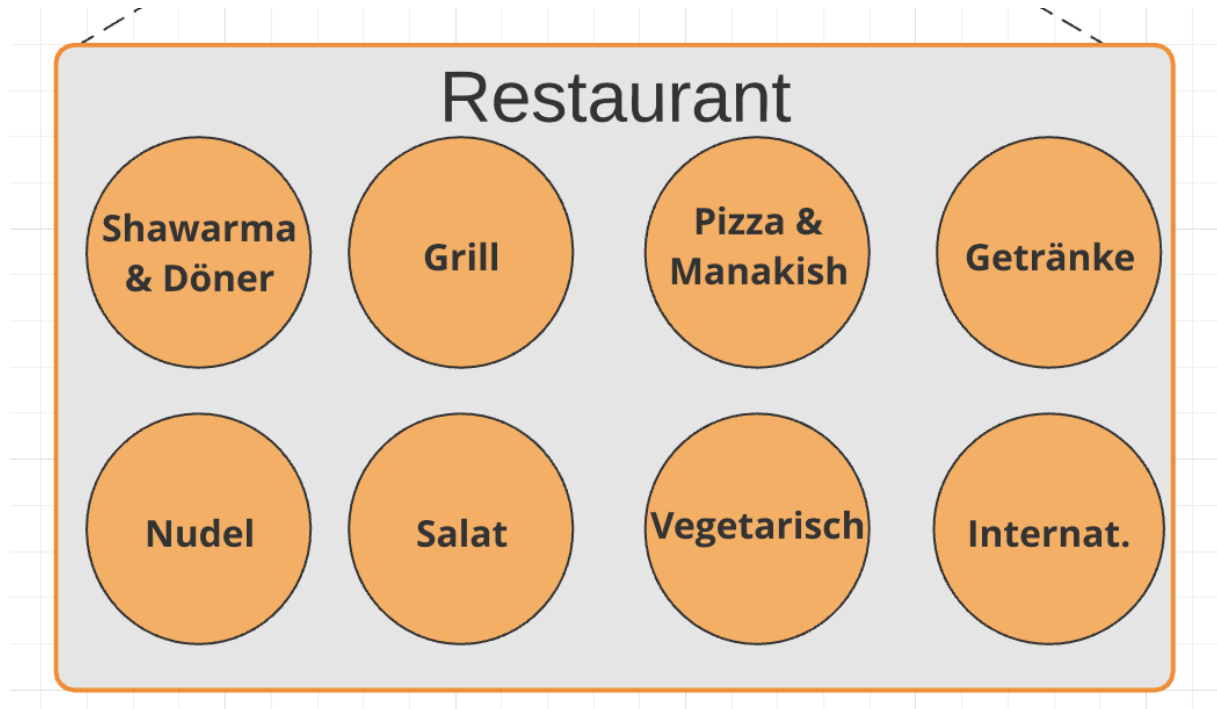


Die Software erkennt sowohl Customer als auch Employee als User.
 Wenn der Benutzer die App öffnet, dann bekommt eine Seite, wo er zwei Möglichkeiten zur Auswahl hat, nämlich Einloggen oder Registrieren.
 Hat der Benutzer bereits einen Account, dann kann er sich mit seinem Zugriffs-Daten problemlos einloggen.
 Im anderen Fall wird er gebeten, einen Account einzulegen.
 Dafür sind die Stammdaten einzugeben.
 (firstName, lastName, birthDay, phoneNumber, Adresse).

Dazu sind auch die E-Mail sowie ein selbst ausgesuchtes Passwort einzugeben, um einen Account zu generieren. Nach Abschließen des Registrierungs-Verfahrens hat der User Vollzugriff auf die App.

Sobald sich der Benutzer erfolgreich eingeloggt hatte, wird er auf die Startseite stoßen.

Wie es auf die grobe Darstellung zu erkennen, verfügt der Benutzer über folgende Optionen.



Die zur Bestellung gewünschte Produkte in den Warenkorb hinzufügen. Den Warenkorb verwalte. (Produkte hinzufügen bzw. löschen)

Außerdem steht dem Nutzer die Wahl zwischen verschiedenen Restaurants. Anhand der Auswahl der Benutzer bekommt er eine Liste mit den entsprechenden Restaurants mit den entsprechenden Kategorien.

Nach der Auswahl eines bestimmten Restaurants werden alle relevanten Informationen angezeigt.

Wie ist die Struktur so entworfen und warum?

Bei dem Entwurf der Struktur wurde es so beabsichtigt, dass die Seiten bzw. Schnittstellen der Seite nacheinander angezeigt werden.

Diese Strategie wirkt so, dass dem Benutzer keine komplexe Seite mit vielen Informationen konfrontiert, sondern er bekommt es Schritt für Schritt, was es einfacher und praktischer für den Nutzer ist.

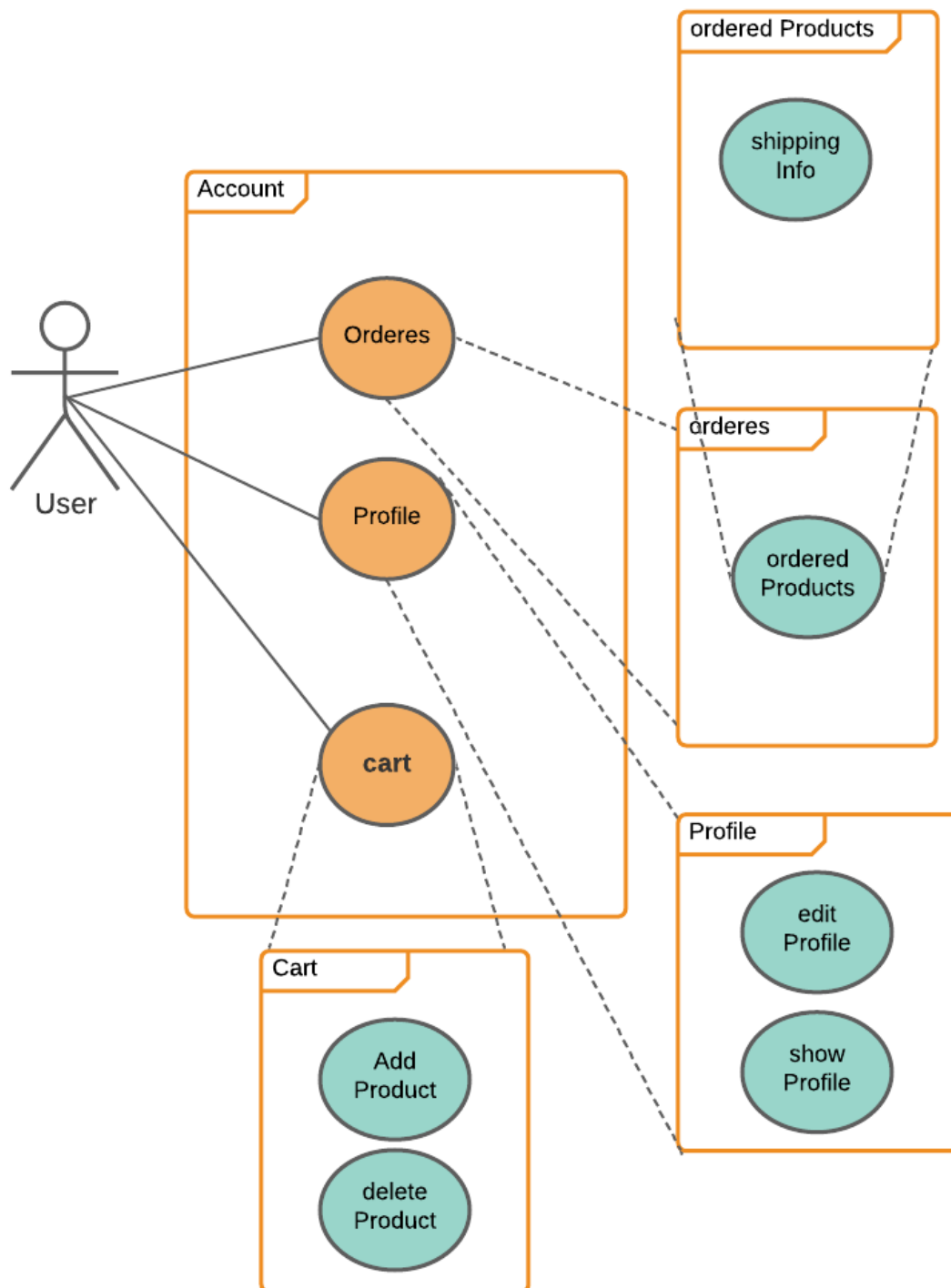
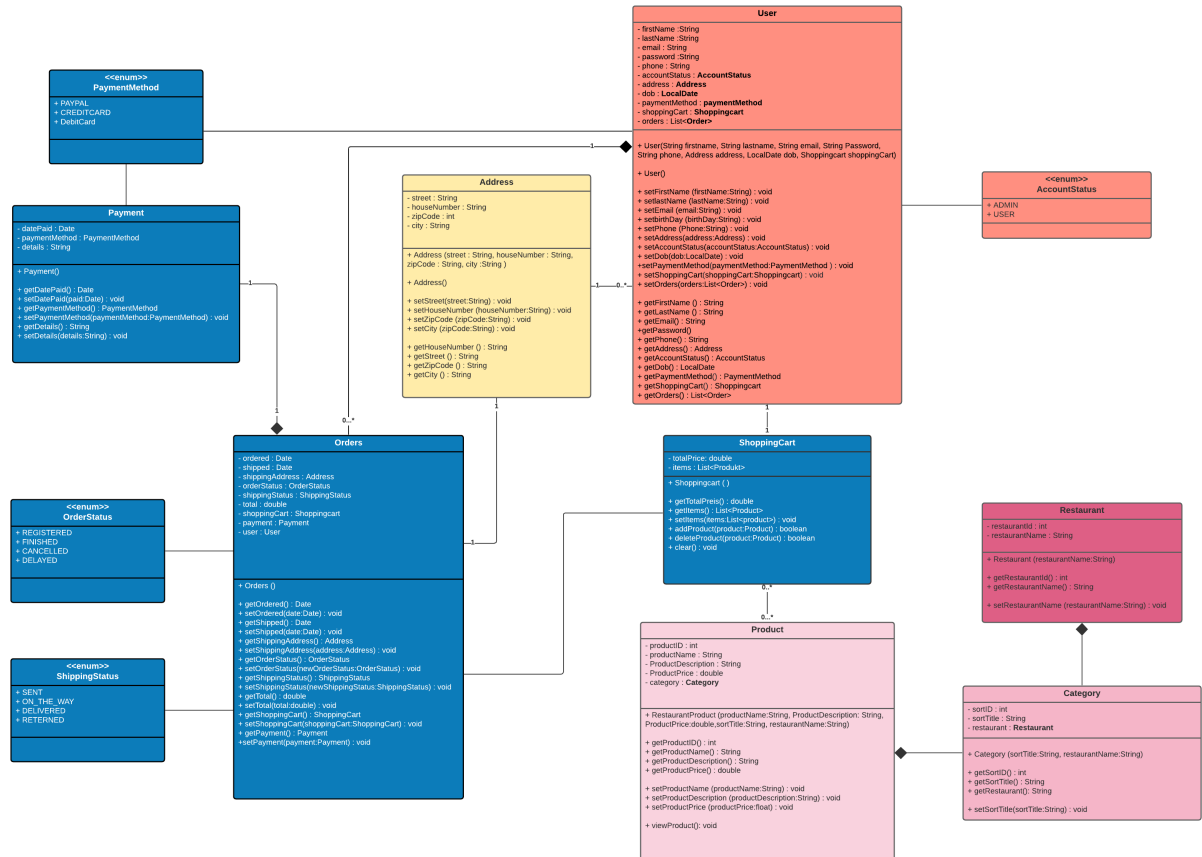


Bild.4

Implementierung des Codes

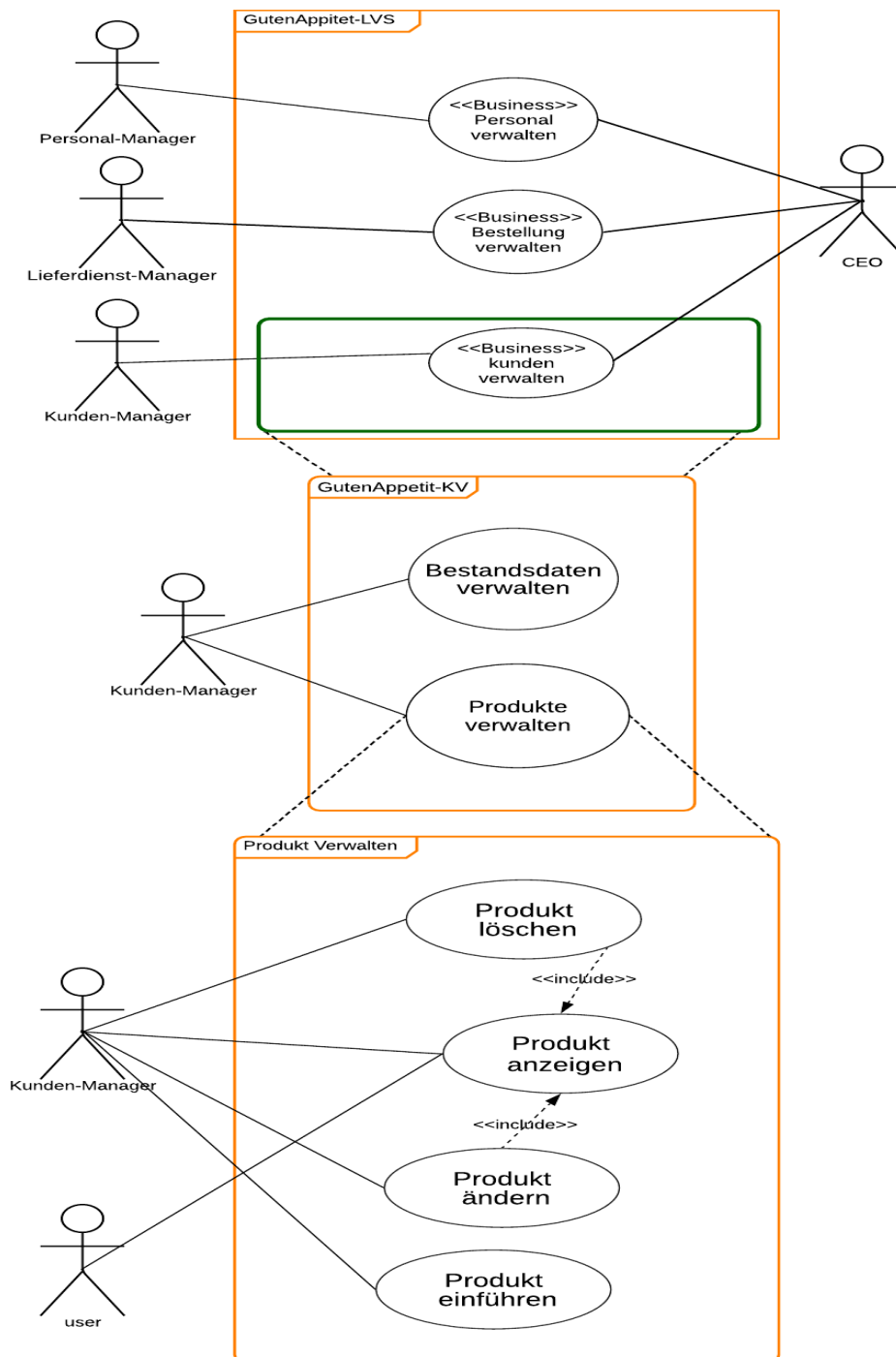
Ein Klassendiagramm zu entwerfen ist sinnvoll und erleichtert die Implementierung.



Zielgruppe

Die Software betrachtet die Kunde als der wesentliche Akteur.

Ein Bild bzw. Diagramm ist deutlicher als 1000 Worte. Das unter das dargestellte Diagramm repräsentiert



Lessons learned Java1:

- Ein wöchentlicher Termin statt alles auf einmal vor der Präsentation.
- Programmieren lernt man durch regelmäßiges Trainieren.
- Gutes Wissen bei Verwendung von Intelli-J ist von Vorteil.

- Aufgabenteilung und Planung sind immer sehr wichtig.
- Tests helfen sowohl beim Lernen als auch beim Programmieren.
- Immer Fragen stellen.
- Die größte Schwierigkeit bei diesem Projekt war die gedankliche Trennung von Datenbank und Java.
- Lernen, wie Github funktioniert, ist notwendig.
- UML Diagramm sind sehr hilfreich und vereinfachen die Implementierung des Codes.

Java2:

- Durch gute Struktur wird die Arbeit vereinfacht
- Bootstrap spart Zeit und Mühe
- Spring war am Anfang schwer zu verstehen, am Ende des Projekts hat man ein großes Verständnis geschaffen.

•

Tools:

- Webex Meetings
- Telegramm
- IntelliJ IDEA
- Lucidchart
- GitHub
- Bildverarbeitungsprogramme
- VS Code
- XAMPP
- Backend: Java Version 14
- Frontend: Bootstrap, JQuery

Installation:

Installieren Sie IntelliJ (2020.2.3)

Java Version 14

Installieren Sie XAMPP (3.2.4)

Gehen Sie zu MySQL in XAMPP und erstellen Sie ein neues DB-Schema mit dem Namen „java_projekt“. Danach sollten Sie die Select-Abfragen aus der Datei

„java_projekt.sql“ kopieren und in den erstellten Datenbanken ausführen.

Der letzte Schritt besteht darin, das Projekt über IntelliJ zu öffnen und die Anwendung zu starten.

