Guten Appetit

Team Mitglieder:

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

1.Projekt-Idee

Im Rahmen des Moduls "Java 1" haben wir ein Lieferdienst-Software entwickelt.
Die Software heißt **Guten Appetit**, die viele gut programmierte Optionen enthält, und mit denen Sie schnell und einfach orientalische Speisen bestellen können.
Die App, ist sowohl für die Kunden als auch für unsere Angestellte gedacht.

2.Ausgangssituation

2.1.Musskriterien:

Was muss die App dem Benutzer anbieten,

- 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. (machen wir in Java 2)
- zwischen verschiedenen Zahlungsoptionen auswählen können.
- Zahlungsmethoden verwalten (Hinzufügen bzw. ändern)
- Rechnungsübersicht haben können. (machen wir in Java 2)
- 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.

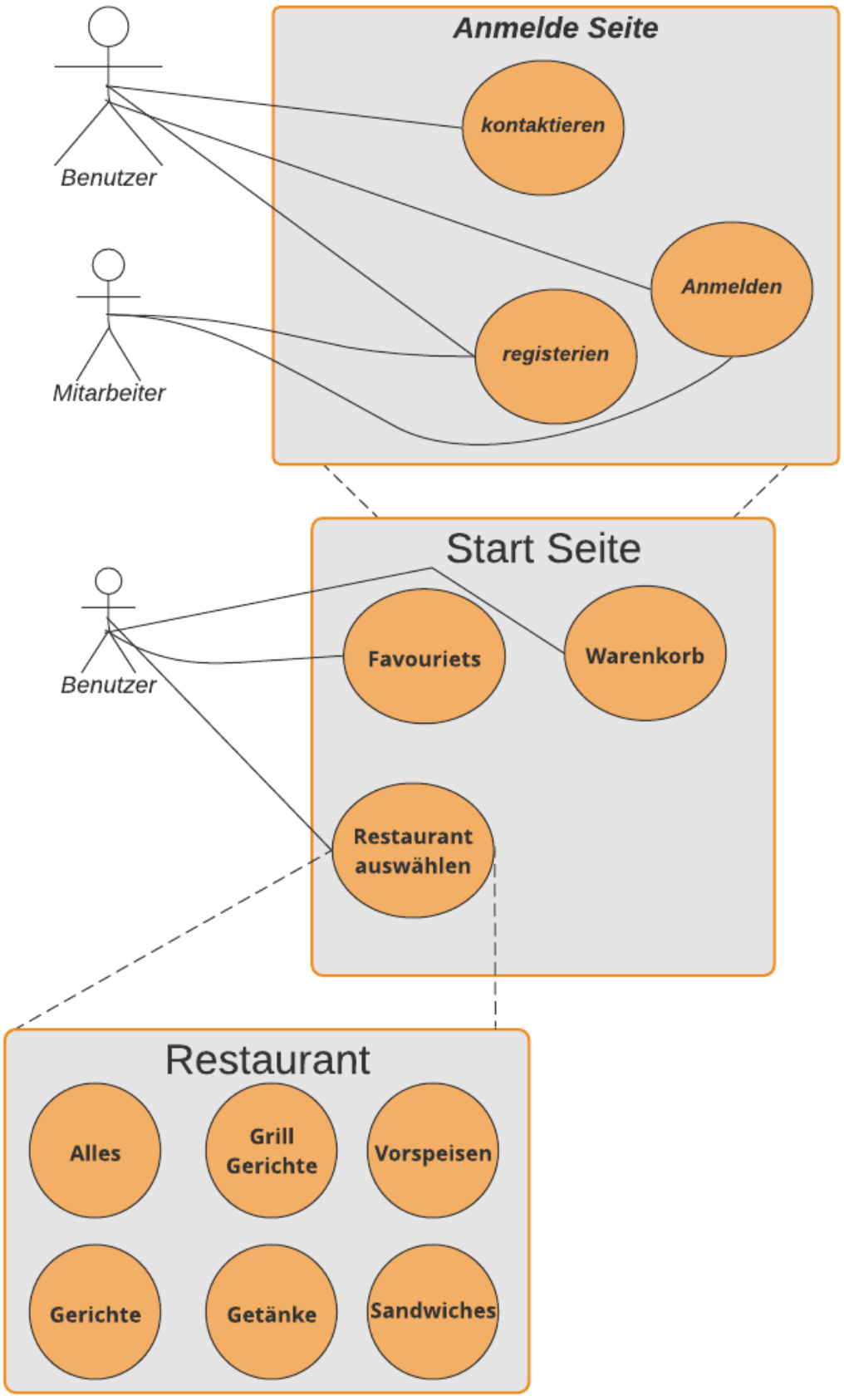
Usability

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

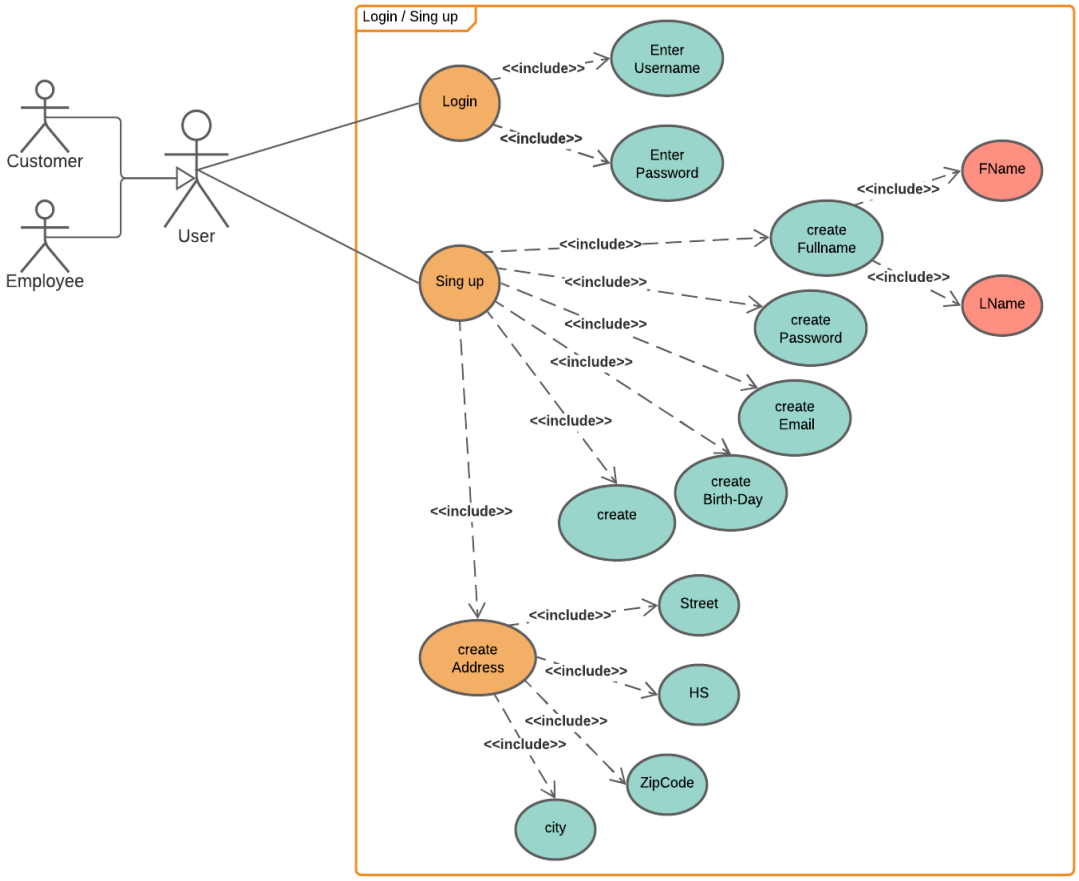
Das System muss bei fehlerhaften Benutzereingaben stabil bleiben und aussagekräftige Fehlermeldungen erzeugen.

Um Design der Struktur

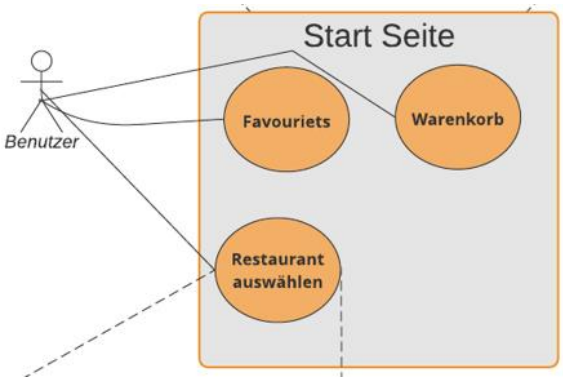
Der unter dargestellten Use-Case soll eine grafische Darstellung für die innere Struktur der App.


Anmelde Seite
kontaktieren
Anmelden
registerien
Benutzer
Mitarbeiter
Start Seite
Favouriets
Warenkorb
Restaurant auswählen
Benutzer
Restaurant
Alles
Grill Gerichte
Vorspeisen
Gerichte
Getänke
Sandwiches

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 Auswal hat, nämlich Einloggen oder Registieren.
Hat der Benutzer bereits einen Account, dann kann er sich mit seinem Zugriff-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 to generieren.
Nach Abschließen des Registrierungs-Verfahrens hat der User Vollzugriff auf die App.

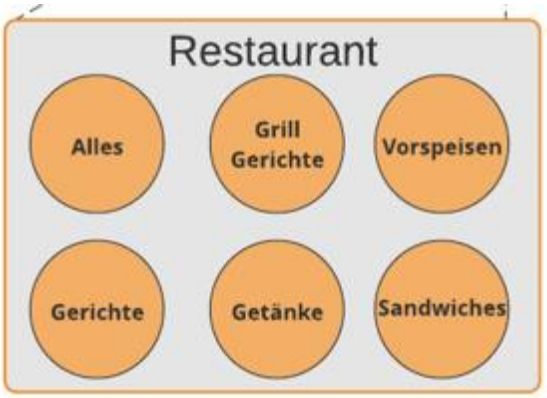


Sobald sich der Benutzer erfolgreich eingeloggt hatte, wird er auf die Startseit stoßen.

Wie es auf die grobe Darstellung zu erkennen, verfügt der Benutzer über folgende Optionen. (noch andere Optionen sind in Java 2 geplant hinzufügen.)

Die bevorzugten Produkte zu merken und speichern und dementsprechend die Favourite-Liste verwalten (ändern).

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



Außerdem steht dem Nutzer die Wah zwischen Konditorei oder Restaurant (Speisen oder Süßigkeit).

Anhand der Auswahl der Benutzer bekommt er eine Liste mit den entsprechenden Restaurants.

Nach der Auswahl eines bestimmten Restaurants werden alle relevanten Informationen angezeigt (Bild.3)

Wie ist die Struktur so entworfen und warum?

Bei dem Entwurf der Struktur wurde es so beabsichrigt, dass die Seiten bzw. Schnittstellen der App nacheinander angezeigt werden. (Bild.4)

Login→ Startseite→Favourits

→Warenkorb→usw.

→Restaaurants auswählen→ Alles

→GrillGerichte → .....

→Vorspeisen → .....

→Speisen → .....

→Getränke → .....

→Sandwiches → .....

Diese Strategie wirkt so, dass dem Benutzer keine komlexe Seite mit vielen Informationen konvruntiert, sondern er bekommt es Schritt für Schritt, was es einfacher und praktischer für den Nutzer ist. (Bild.4)

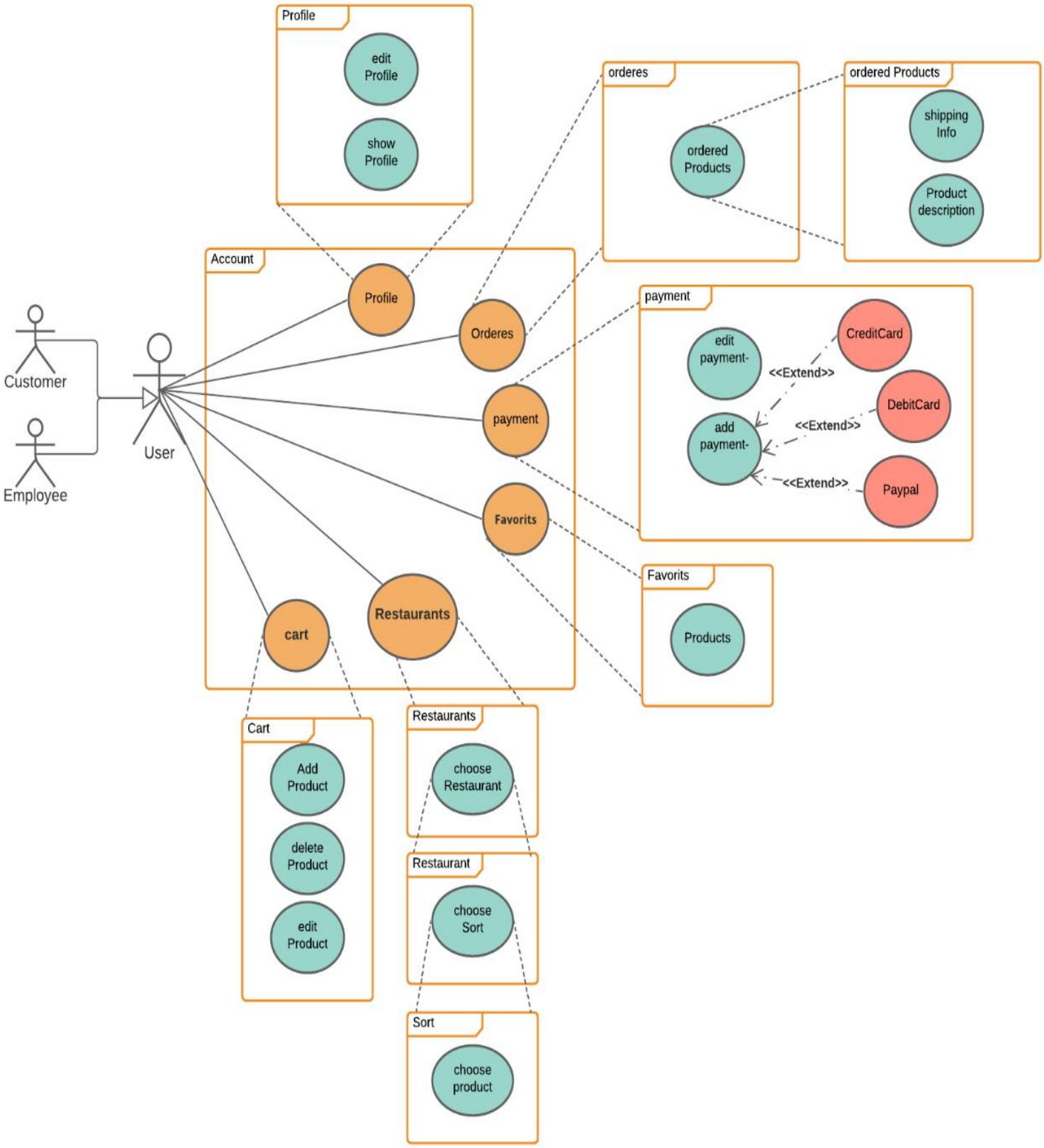
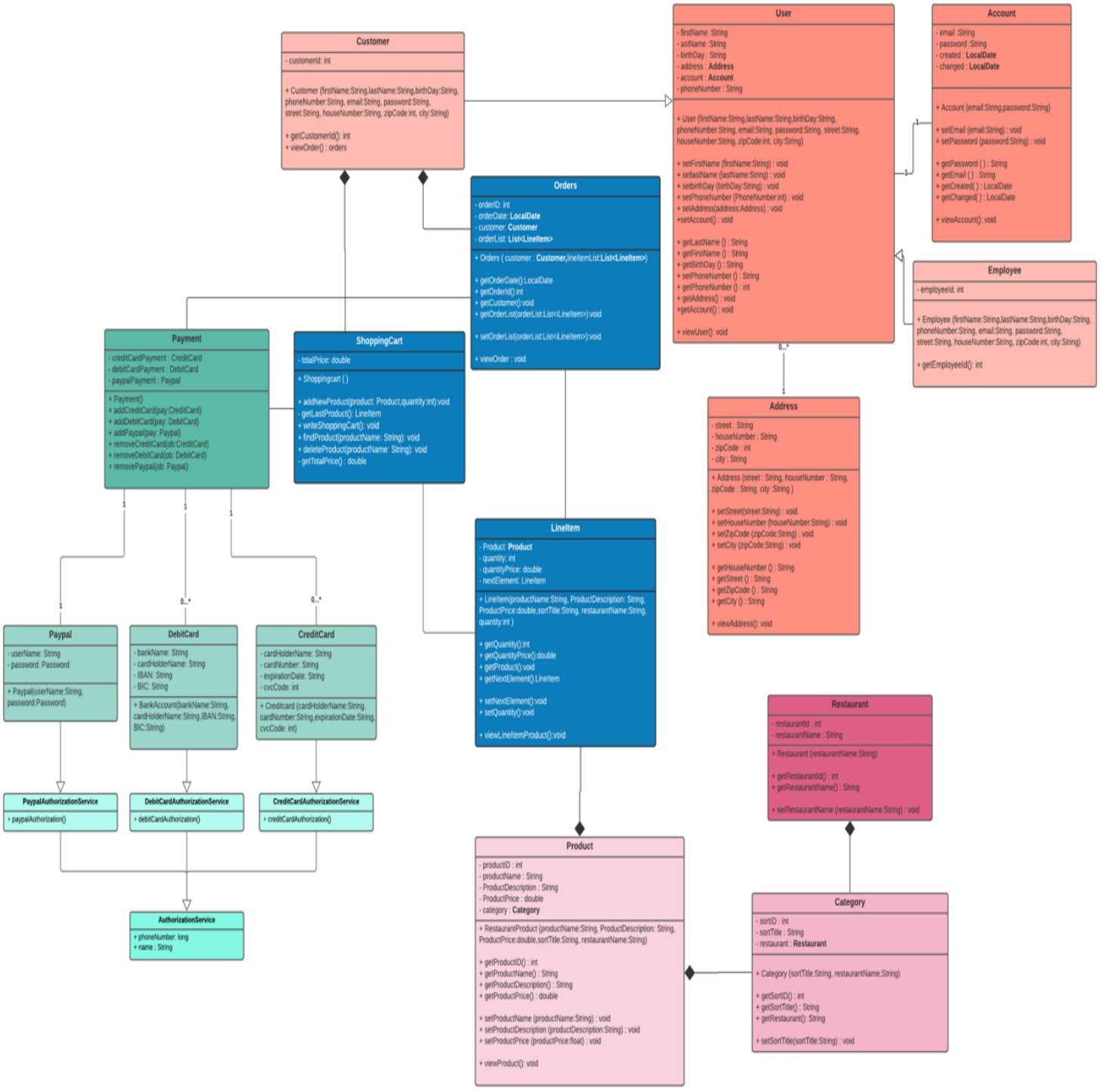

Profile
edit Profile
show Profile
orderes
ordered Products
ordered Products
shipping Info
Product description
Account
Profile
Orderes
payment
Favorits
cart
Restaurants
Customer
Employee
User
payment
edit payment-
add payment-
<<Extend>>
<<Extend>>
<<Extend>>
CreditCard
DebitCard
Paypal
Favorits
Products
Cart
Add Product
delete Product
edit Product
Restaurants
choose Restaurant
Restaurant
choose Sort
Sort
choose product

Bild.4

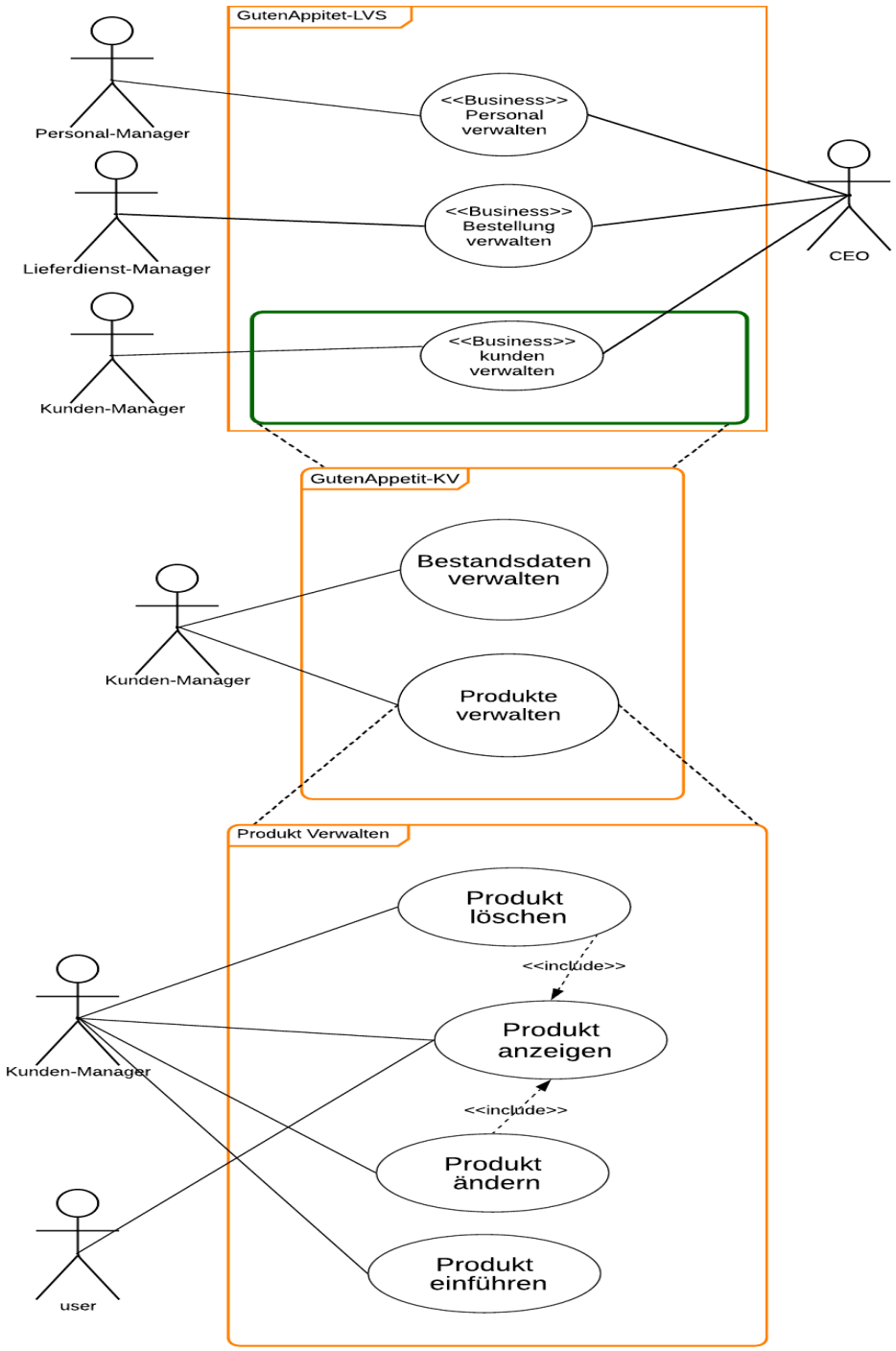
Implementierung des Codes

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.

- 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 nötwendig.
- UML Diagramm sind sehr hilfreich und vereinfachen die Implementierung des Codes.

Tools

- Webex Meetings
- IntelliJ IDEA für den Code-Entwurf
- Lucidchart (Diagramme sowie Use cases)
- GitHub
- Bildverarbeitungsprogramme