

Übungsblatt: 3

1. Abgabepartner: Matthias Wolff (458 766)

2. Abgabepartner: Anton Mende (461 328)

2. Abgabepartner: Anika Herbermann (461 655)

Aufgabe 8)

1 Zielbestimmung

Die Software soll eine graphische Benutzeroberfläche zur Bestellung von Pizzen zur Verfügung stellen und die Bestellung zusammen mit den Kontaktdaten des Kunden an die Pizzeria übermitteln.

2 Anwendungsbereiche und Zielgruppen

Die Software soll Kunden eine einfache Bestellung ihrer Pizzen ermöglichen.

Die Zielgruppe besteht aus Personen, die eine Pizza bestellen wollen

3 Produktfunktionen

- \ LF10 \ Auswahl von Pizzen und ihrer Anzahl aus einer festen Pizzaliste
- \ LF20 \ Anmelden eines Benutzerkontos
- \ LF30 \ Erstellen / Registrieren eines Benutzerkontos
- \ LF40 \ Benachrichtigung an Kunden, falls Eingaben beim Anmelden / Registrieren ungültig
- \ LF41 \ Benachrichtigung an Kunden nach erfolgreicher Bestellung
- \ LF50 \ Übermittlung der Bestellung und Kontaktdaten des Bestellers an die Pizzeria

4 wichtigste zu speichernde Daten

- \ LD10 \ Relevante Kundendaten sind zu speichern (Name, Adresse, Email, Telefonnummer,...)
- \ LD20 \ Die bestellbaren Pizzen sollen gespeichert werden
- \ LD30 \ Die erfolgten Bestellungen sind zu speichern

5 Leistungsanforderungen

- \ LL10 \ Es soll eine für eine Pizzeria übliche Anzahl an Kunden verwaltet werden
- \ LL20 \ Die Software soll auf Benutzereingaben in unter 0,5 Sekunden reagieren
- \ LL30 \ Die Übertragung der Bestellung an die Pizzeria soll in unter 10 Sekunden zuverlässig abgeschlossen sein

6 Qualitätsanforderungen

Merkmal	gewünschte Qualität
Funktionalität	normal
Zuverlässigkeit	gut
Benutzbarkeit	sehr gut
Effizienz	normal
Änderbarkeit	normal
Portierbarkeit	gut

7 Ergänzungen

-

8 Glossar

-

Aufgabe 10)

a)

context Vorfuehrung

inv:

self.start <= self.ende

self.grundpreis > 0

self.name != „“

b)

context Bestellung

inv:

Bestellung.allInstances -> select(best | best.vorfuehrung == self.vorfuehrung)

-> forAll(best | best.reservierteSitze -> intersection(self.reservierteSitze) -> size == 0)

c)

context Bestellung::cancel()

pre:

self.Zeitpunkt < vorfuehrung.start

post:

not Bestellung.allInstances -> exists(best | (best.nummer == self.nummer) and (best.vorfuehrung == self.vorfuehrung))

d)

context Bestellung

inv:

(self.vorfuehrung.saal.nummer == 42) implies (self.vorfuehrung.ende < 11.11.2020)