

Projekttitel <b>Programmierung eines Liefershops mit Datenbankbindung</b>			Datum <b>01.12.2016</b>
GitHubprojektnummer <b>06</b>	Name <b>PizzaPizza</b>	Projektbeginn <b>02.11.2016</b>	Projektende <b>10.01.2017</b>
Auftraggeber/in <b>Prof. Dr. Oliver Braun</b>		Projektleitung <b>Yannic Fraebel, Nils Engelbrecht</b>	
Statusbericht erstellt durch <b>Yannic Fraebel, Nils Engelbrecht</b>		Meilenstein / Projektphase <b>1</b>	
Projektstatus insgesamt			
			starke Probleme, besondere Aktivitäten erforderlich
			Probleme vorhanden aber im Griff
			kein Handlungsbedarf

## 1. Grad der Zielerreichung / Status des Prototyps

Auf der „Startseite“ bietet die Applikation folgende Funktionen:

- Wahlmöglichkeit zwischen Kunde und Mitarbeiter (Rolle) anhand eines Dropdown-Menüs. Hierbei sind als Default-User ein Kunde „Emil“ und ein Admin namens „Padrone“ hinterlegt.
- Außerdem können sich neue Kunden eigenständig registrieren (dies wird in die Datenbanktabelle „Users“ übernommen).

Die „welcomeUser“ und „welcomeAdmin“-Seite beinhaltet folgende Funktionen:

- Die Kundenview bietet die Wahlmöglichkeit zwischen den vorhandene Kategorien Pizza, Getränke und Desserts.
- Einzelne Items können in Mengen und in einer variablen Größe (in cm bzw. l) bestellt werden. Der Preis wird mengen- und größenabhängig verbucht.
- Kunden können sich ihre bisherigen Bestellungen mit Vorgangsnummer, Datum und Details zur Bestellung ausgeben lassen.
- Mitarbeiter haben die Möglichkeit, sich alle registrierten Benutzer anzeigen zu lassen und alle bisherigen Bestellungen einzusehen.
- Kunden und Admins haben außerdem die Möglichkeit, sich abzumelden. (Redirect zur Startseite)
- Kategorisierung der User in Admins und Kunden erfolgt durch Setzen eines Admin Flags in der Users Tabelle.
- Preiskalkulation der Bestellung ist schon vollständig implementiert (sowohl intern als auch in JS um Benutzerfreundlichkeit zu gewährleisten)

## 2. Aufgabenverteilung im Team

Beschreibung	Bearbeiter
Erstellen des Statusberichts	Yannic Fraebel
Grad der Zielerreichung	Yannic Fraebel
Aufgabenverteilung im Team	Yannic Fraebel/Nils Engelbrecht
Projektplan für die Restlaufzeit	Yannic Fraebel/Nils Engelbrecht
Risikoanalyse	Yannic Fraebel
Aufwandsnachweis	Yannic Fraebel/Nils Engelbrecht
Erstellen des Lastenhefts	Yannic Fraebel
Erstellung und Einbindung der Tabellen	Nils Engelbrecht
Kategorisierung der User durch ein flag	Nils Engelbrecht
Erstellung der Views	Nils Engelbrecht
Erstellung der Models	Nils Engelbrecht
Erstellung der Services	Nils Engelbrecht
Erstellung der Controller	Nils Engelbrecht
Implementieren der Preisberechnung	Nils Engelbrecht
Testen der Projektfunktionalität	Yannic Fraebel/Nils Engelbrecht

## 3. Plan für die Restlaufzeit des Projekts

Geplant ist es die Restlaufzeit des Projekts maßgeblich grob dennoch präzise einzuschätzen. Nach dem SCRUM Verfahren werden nach und nach Funktionalitäten implementiert, integriert und sofort getestet, um einen weiteren sinnvollen Verlauf der weiteren zu implementierenden funktionalen Anforderungen zu gewährleisten.

Folgende Implementierungsideen sind noch angedacht:

- Anzeige von Gesamtumsatz
- Allg. Auswertungen (→ Pizzen/ Bestellung)
- MA read/write Rechtekonzept
- Login über freies Textfeld und Passwortabfrage
- Bestellbestätigung per email falls vorhanden
- Weitere Auswertungsmöglichkeiten
- View Unterstützung für Bestellaufnahme von mehreren Produkten auf einmal sowie mehrere verschiedene Extras pro Item
- (Evtl. Lösung durch Warenkorb Implementierung)
- Optimierung von User Registrierungsformular Error Handling z.B. check ob Name schon vorhanden
- Liefermöglichkeit je nach Kundenentfernung implementieren
- Lieferdauerberechnung und View Einbindung implementieren

## 4. Risikoanalyse

Risikoart	Wahr- scheinlichkeit	Auswirkung	Ampel	Verantwortlicher	Beschreibung	Maßnahme(n)
Datenverlust	1	5	5	Fraebel / Engelbrecht	Verlust der Projektdateien	Regelmäßige Sicherung auf GitHub
Krankheit	3	3	9	Fraebel / Engelbrecht	Teammitglied erkrankt	Telefonkonferenz zur weiteren Planung führen
Keine rechtzeitige Abgabe	3	5	15	Fraebel / Engelbrecht	Projekt wird nicht rechtzeitig abgeben	Studienarbeit stärker priorisieren
Probleme Syntax Scala/HTTP/SQL	2	4	8	Fraebel / Engelbrecht	Probleme beim Programmieren durch neue Sprachen	Tutorials lesen/schauen und Bücher im Eigenstudium
Probleme Dokumentation	2	3	6	Fraebel	Schwierigkeiten beim Erstellen des Lastenhefts und des Projektplans	Orientierung an Lastenheften, die von Unternehmen online gestellt wurden.
Anforderungen	1	2	4	Engelbrecht	Falsche oder überflüssige Programmteile werden implementiert	Funktionale Anforderungen lesen und Absprache mit Herr Prof. Dr. Oliver Braun
Mehrfachbearbeitung	2	4	8	Fraebel / Engelbrecht	Arbeitsaufträge werden parallel bearbeitet und gepusht	Verwenden der VCS Möglichkeiten
Pragmatische Programmfehler	3	2	6	Fraebel / Engelbrecht	Applikation arbeitet zu umständlich, das MVC-Modell wird nicht eingehalten	Gegenseitiges Überprüfen um Fehler zu vermeiden

## 5. Aufwandsnachweis

Datum	Nils Engelbrecht	Yannic Fraebel
16.11.2016	<ul style="list-style-type: none"> <li>- Clonen des zur Verfügung gestellten Git-Repo's</li> <li>- Erste Einarbeitung in das Play-Framework</li> <li>- Einbindung eines FavIcons</li> </ul>	<ul style="list-style-type: none"> <li>- Clonen des zur Verfügung gestellten GIT- Repo's</li> <li>- Erzeugen der Projektdateien für IntelliJ.</li> <li>- Erste Einarbeitung in das Play-Framework</li> </ul>
17.11.2016	<ul style="list-style-type: none"> <li>- Erstellung einiger Forms</li> <li>- Unterscheidung MA vs. Kunde</li> </ul>	<ul style="list-style-type: none"> <li>- Verstehen des bereits gegebenen Projektcodes</li> </ul>
18.11.2016	<ul style="list-style-type: none"> <li>- Registrierungs Validierung</li> <li>- Datenbank getUser implementiert</li> </ul>	<ul style="list-style-type: none"> <li>- Recherche zum Thema Statusbericht</li> </ul>
19.11.2016	<ul style="list-style-type: none"> <li>- Kategorien und Items eingeführt</li> <li>- Bestellfunktionalität</li> <li>- Default Datenbanktabellen Values festgelegt</li> <li>- dynamisches Kunden-Menü angelegt</li> </ul>	<ul style="list-style-type: none"> <li>- Anlegen des Statusberichtes</li> </ul>
20.11.2016	<ul style="list-style-type: none"> <li>- Bestellanzeige je KundenID</li> </ul>	<ul style="list-style-type: none"> <li>- Recherche zum Thema Lastenheft</li> <li>- Anlegen des Projektplans</li> </ul>
25.11.2016	<ul style="list-style-type: none"> <li>- Bestellanzeige optimiert</li> <li>- Re-lokation der Klassen/ Methoden je nach Funktionalität (Separation of Concerns)</li> <li>- SQL Default Skripte nach Tabellen getrennt</li> </ul>	<ul style="list-style-type: none"> <li>- Testen des Prototyps</li> </ul>
26.11.2016	<ul style="list-style-type: none"> <li>- Bestellvorgang Bug gelöst</li> </ul>	<ul style="list-style-type: none"> <li>- Anlegen des Lastenhefts</li> </ul>
27.11.2016	<ul style="list-style-type: none"> <li>- Cent in Euro Anzeige erstellt</li> <li>- Preiskalkulation implementiert</li> </ul>	<ul style="list-style-type: none"> <li>- Fertigstellen und Korrekturlesen des Statusberichtes und des Lastenheftes</li> </ul>
28.11.2016	<ul style="list-style-type: none"> <li>- Versuch Extras Form dynamisch via JS einzufügen</li> <li>- Optionale Extra fields implementiert</li> </ul>	
29.11.2016	<ul style="list-style-type: none"> <li>- vollständige backend Extra Bestellaufgabe realisiert, trotz ViewForm Problemen</li> </ul>	
30.11.2016	<ul style="list-style-type: none"> <li>- Fertigstellung Aufwandsnachweis</li> </ul>	
01.12.2016	<ul style="list-style-type: none"> <li>- Überarbeitung von Lastenheft &amp; Statusbericht</li> <li>- Anpassung der View für die Extra-Auswahl</li> </ul>	