|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Projekttitel  **Programmierung eines Liefershops mit Datenbankanbindung** | | | | | | Datum  **01.12.2016** |
| GitHubprojektnummer  **06** | Name  **PizzaPizza** | Projektbeginn  **02.11.2016** | | | Projektende  **10.01.2017** | |
| Auftraggeber/in  **Prof. Dr. Oliver Braun** | | Projektleitung  **Yannic Fraebel, Nils Engelbrecht** | | | | |
| Statusbericht erstellt durch  **Yannic Fraebel, Nils Engelbrecht** | | Meilenstein / Projektphase  **2** | | | | |
|  | | **Projektstatus insgesamt** | | | | |
| **Ampel** |  | starke Probleme, besondere Aktivitäten erforderlich | | |
|  | Probleme vorhanden aber im Griff | | |
|  | kein Handlungsbedarf | | |

1. **Kurze Zusammenfassung und Überblick über das Projekt**

Das Projekt umfasst die Analyse, die Planung, und die Umsetzung einer Softwarelösung für einen neuen Lieferservice. Es wurde eine Webseite zur Verfügung gestellt, welche den Anforderungen und Spezifikationen des Auftraggebers entspricht.

Hauptsächlich soll die Webseite Kunden die Möglichkeit geben sich zu registrieren und Produkte online zu bestellen. Mitarbeiter haben zudem weitere Rechte und Features, wie z.B. das Verwalten, Anlegen oder Ändern von Daten.

Das Projekt wurde in 2 Meilensteine gegliedert.

**Meilenstein 1: Die Applikation beinhaltet folgende Funktionen:**

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)

**Meilenstein 2: Folgende Funktionen wurden hinzugefügt:**

Auf der „Startseite“:

* Login Möglichkeit über freies Textfeld mit einem Passwort für Kunden und Mitarbeiter.
* Kurzes Impressum über die Ersteller der Webseite

Die „welcomeUser“ und „welcomeAdmin“-Seite:

* Bei Pizzen können Extras für eine individualisierte Bestellung hinzugefügt werden.
* Produkte können in Mengen und in einer variablen Größe (in cm bzw. l) bestellt werden. Es ist möglich, mehrere Produkte zusammenfassend zu bestellen. Der Preis wird mengen- und größenabhängig verbucht.
* Kunden können sich bei ihren bisherigen Bestellungen zudem die Lieferzeit der Bestellung ausgeben lassen. Es werden nur Kunden beliefert, die nicht weiter als 20 km entfernt wohnen.
* Mitarbeiter haben die Möglichkeit, sich alle registrierten Benutzer anzeigen zu lassen und diese zu ändern oder zu löschen. Zudem können sie alle bisherigen Bestellungen einsehen und diese nach Kunden sortieren und erhalten nützliche Infos (Gesamtumsatz, durchschnittlicher Umsatz).
* Mitarbeiter haben die Möglichkeit, dass Sortiment anzupassen, indem sie Kategorien und Produkte löschen, editieren oder hinzufügen.
* Umfassendes Errorhandling in allen Bereichen der Webseite

1. **Hinweise auf Einschränkungen und Benutzung**

* Voraussetzung ist ein Internetzugang. Eine Registrierung ist für die Auftragsabwicklung erforderlich. Bereits registrierte Kunden oder Mitarbeiter können sich einloggen und müssen sich nicht erneut registrieren.
* Wenn der Username bereits vergeben ist, kann er nicht nicht nochmal angelegt werden. Zwischen Groß- und Kleinschreibung im Usernamen wird nicht unterschieden.
* Usernamen dürfen keine Sonderzeichen oder Zahlen enthalten.
* Falls nur ein Admin vorhanden ist, kann sich dieser nicht selbst löschen, da immer ein Admin vorhanden sein muss.
* Der Nutzer der Webseite bekommt in verschiedenen Bereichen Feedback (Success/ Error) durch ein PopUp.

1. **Aufwandsnachweis Gesamtprojekt**

|  |  |  |  |
| --- | --- | --- | --- |
| **Datum** | **Tätigkeit** | **Erledigt durch** | **Dauer** |
| 16.11.2016 | - Clonen des zur Verfügung gestellten Git-Repo’s  - Erste Einarbeitung in das Play-Framework  - Erzeugen der Projektdateien für IntelliJ.  - Erste Einarbeitung in das Play-Framework  - Einbindung eines FavIcons | Yannic Fraebel, Nils Engelbrecht | 6 Stunden |
| 17.11.2016 | - Erstellung einiger Forms  - Unterscheidung Mitarbeiter vs. Kunde | Nils Engelbrecht | X Stunden |
| 18.11.2016 | - Registrierungsvalidierung  - Datenbank getUser implementiert | Nils Engelbrecht | X Stunden |
|  | - Recherche zum Thema Statusbericht | Yannic Fraebel | 4 Stunden |
| 19.11.2016 | - Kategorien und Items eingeführt - Bestellfunktionalität  - Default Datenbanktabellen Values festgelegt  - dynamisches Kunden-Menü angelegt | Nils Engelbrecht | X Stunden |
|  | - Anlegen des Statusberichtes | Yannic Fraebel | 4 Stunden |
| 20.11.2016 | - Bestellanzeige je KundenID | Nils Engelbrecht | X Stunden |
|  | - Recherche zum Thema Lastenheft  - Anlegen des Projektplans | Yannic Fraebel | 3 Stunden |
| 25.11.2016 | - Bestellanzeige optimiert  - Re-lokation der Klassen/ Methoden je nach Funktionalität (Separation of Concerns)  - SQL Default Skripte nach Tabellen getrennt | Nils Engelbrecht | X Stunden |
|  | - Testen des Prototyps | Yannic Fraebel | 30 Minuten |
| 26.11.2016 | - Bestellvorgang Bug gelöst | Nils Engelbrecht | X Minuten |
|  | - Anlegen des Lastenhefts | Yannic Fraebel | 2,5 Stunden |
| 27.11.2016 | - Cent in Euro Anzeige erstellt  - Preiskalkulation implementiert | Nils Engelbrecht | X Stunden |
|  | - Fertigstellen und Korrekturlesen des Statusberichtes und des Lastenheftes | Yannic Fraebel | 3 Stunden |
| 28.11.2016 | - Versuch Extras Form dynamisch via JS einzufügen  - Optionale Extra fields implementiert | Nils Engelbrecht | X Stunden |
| 29.11.2016 | - vollständige backend Extra Bestellaufgabe realisiert, trotz ViewForm Problemen | Nils Engelbrecht | X Stunden |
| 30.11.2016 | - Fertigstellung Aufwandsnachweis | Yannic Fraebel, Nils Engelbrecht | 30 Minuten |
| 01.12.2016 | - Überarbeitung von Lastenheft & Statusbericht  - Anpassung der View für die  Extra-Auswahl | Nils Engelbrecht | X Minuten |
| 15.12.2016 | Erlenen von HTML5 | Yannic Fraebel | 10 Stunden |
| 18.12.2016 | - Einführung von Sessions  - Passwortfunktionalität hinzugefügt  - Login- und Registrierungsform Validierung eingeführt (bereits vorhanden/ existiert nicht etc.) | Nils Engelbrecht | X Stunden |
| 22.12.2016 | - Userverwaltung Funktionalität implementiert (Name und Admin Änderung) | Nils Engelbrecht | X Stunden |
| 23.12.2016 | - Userverwaltung Unterscheidung zwischen Kunde und MA  - Form Error Message Display erneuert | Nils Engelbrecht | X Stunden |
| 27.12.2016 | - multiple JQuery Dialog Funktionalität hinzugefügt  - User: distance hinzugefügt | Nils Engelbrecht | X Stunden |
| 02.01.2016 | - Gliederung und Erstellung des Statusberichts | Yannic Fraebel | 2 Stunden |
| 03.01.2016 | - Erstellung des Statusberichts, Analyse (Post-Mortem)  - Recherche und Erstellung des Pflichtenhefts | Yannic Fraebel | 8 Stunden |
| 04.01.2016 | - Team Besprechung und Planen der noch zu implementierenden Funktonen | Yannic Fraebel, Nils Engelbrecht | 6 Stunden |
| 05.01.2016 | - Erstellung des Pflichtenhefts und der UML-Diagramme | Yannic Fraebel | 4 Stunden |
| 06.01.2016 | - Korrekturlesen des Statusberichtes und des Pflichtenheftes | Yannic Fraebel | 2,5 Stunden |
|  |  |  |  |

1. **Rückblickende Analyse („Post-Mortem“)**

Die Post-Mortem-Analyse ist Teil des Projektmanagements und dient dazu, die Verarbeitung von Erfahrungen bei Durchführung des Projekts für die Zukunft zu festigen.

*+ Positives*

+ Praxisnahe Aufgabenstellung

+ Anwenden aller bisheriger im Studium erworbenen Kenntnisse im Bereich der Softwareentwicklung

+ Einwandfreie Planung des Projektes

+ Kompetente Teammitglieder

*-- Negativ*

- Einarbeitung in drei unbehandelte neue Themengebiete notwendig (HTML, CSS, JS)

*zukünftige Verbesserungen*

- dem Dozenten mehr Fragen stellen

- bessere Zeitplanung, da diese häufig die meisten Probleme auslöst.

*Feedback an den Auftraggeber*

- eventuell CSS & HTML auch in der Vorlesung kurz behandeln