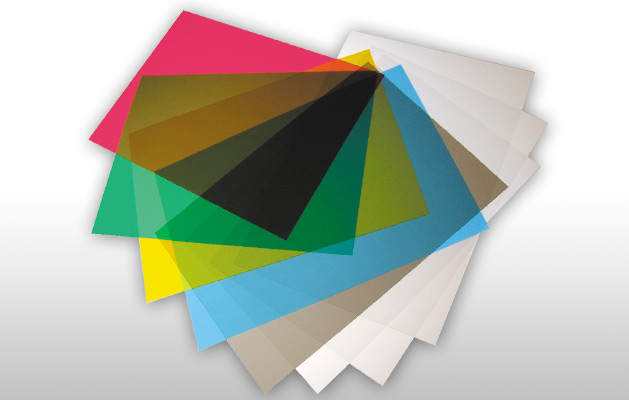
Pflichtenheft

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Version* | *Autor* | *Datum* | *Status* | *Kommentar* |
| 1.0 | AS | 7. März 2017 | Entwurf | Version 1.0 |
| 2.0 | AS | 27. Dez. 2016 | Fertigstellung | Version 2.0 |

# Pflichtenheft Flitzer Ges.m.b.H

**Simon Appel**



27.12.2016

Inhalt

[Pflichtenheft Flitzer Ges.m.b.H 1](#_Toc476486013)

[1 Zielbestimmung 2](#_Toc476486014)

[1.1 Musskriterien 2](#_Toc476486015)

[1.2 Wunschkriterien 3](#_Toc476486016)

[1.3 Abgrenzungskriterien 3](#_Toc476486017)

[2 Produkteinsatz 3](#_Toc476486018)

[2.1 Datenbankprogramm 3](#_Toc476486019)

[2.2 Webseite 3](#_Toc476486020)

[2.3 Schulungsprogramm 4](#_Toc476486021)

[3 Produktumgebung 4](#_Toc476486022)

[3.1 Software 4](#_Toc476486023)

[3.2 Hardware 4](#_Toc476486024)

[3.3 Orgware 4](#_Toc476486025)

[3.4 Produkt-Schnittstellen 4](#_Toc476486026)

[4 Produktfunktionen 5](#_Toc476486027)

[4.1 Aktivitätsdiagramm 5](#_Toc476486028)

[4.2 Datenbankprogramm 7](#_Toc476486029)

[4.2.1 Kundendaten speichern 7](#_Toc476486030)

[4.2.2 Autodaten speichern 8](#_Toc476486031)

[4.2.3 Verfügbarkeit speichern 9](#_Toc476486032)

[4.3 Webseite 10](#_Toc476486033)

[4.3.1 Kundenregistration annehmen 10](#_Toc476486034)

[4.3.2 Autodaten anzeigen 11](#_Toc476486035)

[4.3.3 Verfügbarkeit anzeigen 12](#_Toc476486036)

[4.3.4 Firmeninformationen anzeigen 13](#_Toc476486037)

[4.3.5 Auto auf der Webseite mieten 14](#_Toc476486038)

[4.4 Schulungsprogramm 15](#_Toc476486039)

[4.4.1 Neue-Funktionen erklärt 15](#_Toc476486040)

[4.4.2 Alte-Funktionen erklärt 16](#_Toc476486041)

[5 Produktdaten 17](#_Toc476486042)

[5.1 Datenbankprogramm 17](#_Toc476486043)

[5.2 Webseite 17](#_Toc476486044)

[5.3 Schulungsprogramm 17](#_Toc476486045)

[6 Produktleistungen 17](#_Toc476486046)

[7 Benutzungsschnittstelle 18](#_Toc476486047)

[8 Qualitätsbestimmung 19](#_Toc476486048)

[9 Globale Testfälle 20](#_Toc476486049)

[9.1 Auto mieten 20](#_Toc476486050)

[10 Entwicklungsumgebung 20](#_Toc476486051)

[10.1 Software 20](#_Toc476486052)

[10.2 Hardware 20](#_Toc476486053)

[10.3 Orgware 20](#_Toc476486054)

[10.4 Entwicklungs-Schnittstellen 20](#_Toc476486055)

[11 Termine 21](#_Toc476486056)

[11.1 Meilensteintermine 21](#_Toc476486057)

[11.2 PSP 21](#_Toc476486058)

[12 Sonstiges 22](#_Toc476486059)

# Einführung

Mit unserer App bieten wir unseren Kunden die Möglichkeit mit einem Klick Dönerstände in ihrer Nähe zu finden. Dies ist hilfreich bei einer spontanen Essentscheidung. Man kann auch mithilfe der App Freunde zum Essen einladen.

# Zielbestimmung

## Musskriterien

### Datenbank

Die in der Machbarkeitsstudie beschriebenen Funktionen /F10/ bis /F30/ müssen implementiert werden.

/LF10/ Kundendaten speichern

Die Benutzerdaten können gespeichert werden.

/LF20/ Autodaten speichern

Alle Daten der Autos müssen gespeichert werden.

/LF30/ Verfügbarkeit speichern

Verfügbarkeit muss gespeichert werden können.

**1.1.2 Webseite**

/LF110/ Kundenregistration annehmen

Kunde muss sich registrieren können.

/LF120/ Autodaten anzeigen

Alle Daten der Autos sind sichtbar.

/LF130/ Verfügbarkeit anzeigen

Verfügbarkeit muss sichtbar sein.

/LF140/ Firmeninformationen anzeigen

Informationen über die Firma sind sichtbar.

**1.1.3 Schulungsprogramm**

/LF210/ Neue-Funktionen erklärt

Neue Funktionen des Datenbankprogramms werden erklärt.

## Wunschkriterien

**1.2.1 Webseite**

/LF150/ Auto auf der Webseite mieten

Mieten eines Autos auf der Webseite

**1.2.2 Schulungsprogramm**

/LF220/ Alte-Funktionen erklärt

Alte Funktionen werden noch einmal erklärt.

## 

## Abgrenzungskriterien

1. **Vermietung durch eine App**

An die Vermietung durch eine App wird noch nicht gedacht, da zuerst einmal alles auf informationsunterstützt umgesetzt wird. Die App könnte man in 2 Jahren mit einem neuen Projektantrag umsetzten.

# Produkteinsatz

## Datenbankprogramm

Zielgruppe: Mitarbeiter der Firma Flitzer

Anwendungsbereich: Mitarbeiter bedienen das Programm, um die Daten in die Datenbank

einzutragen.

Betriebsbedingungen: Das Programm läuft auf den neuen PCs der Firma Flitzer.

Erläuterung: Die Mitarbeiter bedienen das Programm um für die Kunden Autos zu mieten oder die Daten und Verfügbarkeit der Autos in die Datenbank einzutragen.

## Webseite

Zielgruppe: Mitarbeiter der Firma Flitzer und Kunden

Anwendungsbereich: Alle Personen können sich die Daten der Autos ansehen.

Betriebsbedingungen: Die Webseite ist verfügbar für alle Geräte mit Internetverbindung.

Erläuterung: Dadurch, dass man eine Webseite hat, kann sich jeder über die

verschiedensten Angebote der Firma informieren. Weiters ist auch die Geschichte der Firma „ansehbar“.

## Schulungsprogramm

Zielgruppe: Mitarbeiter der Firma Flitzer

Anwendungsbereich: Mitarbeiter bedienen das Programm, um sich mit den Funktionen des Datenbankprogramms vertraut zu machen

Betriebsbedingungen: Die Mitarbeiter nutzen dieses Programm bevor sie mit dem echten Datenbankprogramm arbeiten.

Erläuterung: Durch dieses Programm können die Mitarbeiter alte und neue Funktionen des Datenbankprogramms kennenlernen.

# Produktumgebung

Hier wird auf das Datenbankprogramm eingegangen.

## Software

Windows 10 mit SQL Datenbank.

## Hardware

Bildschirm,20GB Speicherplatz.

## Orgware

LAN

## Produkt-Schnittstellen

Schnittstelle zum Server

# Produktfunktionen

## Aktivitätsdiagramm



## Benutzerverwaltung

### Benutzer registrieren

Bei der Erstbenützung der App, soll sich der Benutzer registrieren. Dafür wird ein einmaliger Benutzername, eine E-Mail-Adresse und ein Passwort mit mindestens 8 Zeichen (min. 1 Großbuchstabe, min. 1 Zahl) als Anmeldedaten verwendet. Nach erfolgreicher Registration sollen alle Dienste der App für den Benutzer erreichbar sein.

|  |  |
| --- | --- |
| Systemanwendungsfall |  |
| Name | /LF10/ Benutzer registrieren |
| Art | Anwendungsfall |
| Kurzbeschreibung | Bei der Erstbenützung der App, soll sich der Benutzer registrieren. Dafür wird ein einmaliger Benutzername, eine E-Mail-Adresse und ein Passwort mit mindestens 8 Zeichen (min. 1 Großbuchstabe, min. 1 Zahl) als Anmeldedaten verwendet. Nach erfolgreicher Registration sollen alle Dienste der App für den Benutzer erreichbar sein. |
| Auslöser | Benutzer möchte sich bei der App registrieren |
| Ergebnis | Benutzer hat ein eigenes Konto |
| Akteure | Neuer Benutzer |
| Eingehende Informationen | Benutzername, Passwort und E-Mail |
| Vorbedingungen | Benutzer darf noch nicht am System registriert sein |
| Nachbedingungen | Benutzer kann sich bei der Software anmelden |
| Ablauf | 1. User gibt seine Daten ein. 2. Daten werden überprüft 3. User wird in die Datenbank eingetragen |
| Risiko | gering |
| Aufwand | mittel |
| Verbindlichkeit | hohe Priorität, unverzichtbar |
| Dringlichkeit | Release 0.5 |

### Benutzer anmelden

|  |  |
| --- | --- |
| Systemanwendungsfall |  |
| Name | /LF20/ Benutzer anmelden |
| Art | Anwendungsfall |
| Kurzbeschreibung | Nach erfolgreicher Registration (/LF0010/) kann sich der Benutzer anmelden, um auf die App zugreifen zu können. Dafür wird die Eingabe des Benutzernamens und des Passwortes benötigt. Wenn die eingegebenen Daten mit denen die in der Datenbank gespeichert sind übereinstimmen, soll der Benutzer alle Tasks der App ausführen können. |
| Auslöser | Benutzer möchte sich bei der App anmelden |
| Ergebnis | User kann Funktionen der App benützen |
| Akteure | registrierter Benutzer |
| Eingehende Informationen | Benutzername, Passwort |
| Vorbedingungen | Benutzer muss registriert sein |
| Nachbedingungen | Benutzer kann die Funktionen der App benutzen |
| Ablauf | 1. Benutzer gibt seine Daten ein 2. Eingabe wird mit vorhanden Usern am Server verglichen 3. Falls eine Übereinstimmung existiert, kann sich der Benutzer anmelden und alle Funktionen der App verwenden |
| Risiko | gering |
| Aufwand | mittel |
| Verbindlichkeit | hohe Priorität, unverzichtbar |
| Dringlichkeit | Release 0.5 |

### Benutzer abmelden

|  |  |
| --- | --- |
| Systemanwendungsfall |  |
| Name | /LF30/ Benutzer abmelden |
| Art | Anwendungsfall |
| Kurzbeschreibung | Nach der Anmeldung (/LF0020/), kann sich der Benutzer wieder abmelden. |
| Auslöser | Benutzer möchte sich bei der App abmelden |
| Ergebnis | Benutzer kann nicht mehr auf die Funktionen der App zugreifen |
| Akteure | angemeldeter Benutzer |
| Eingehende Informationen | / |
| Vorbedingungen | Benutzer muss angemeldet sein |
| Nachbedingungen | Benutzer kann sich jederzeit wieder an der App anmelden |
| Ablauf | 1. Benutzer muss angemeldet sein 2. User kann sich abmelden |
| Risiko | gering |
| Aufwand | gering |
| Verbindlichkeit | hohe Priorität, unverzichtbar |
| Dringlichkeit | Release 0.5 |

## Döner-Stände suchen

### Döner-Stände per Adresse finden

|  |  |
| --- | --- |
| Systemanwendungsfall |  |
| Name | /LF110/ Döner-Stände per Adresse finden |
| Art | Anwendungsfall |
| Kurzbeschreibung | Wenn eine neue Adresse eingetragen wird, wird der Weg-Punkt der Karte zur eingegebenen Adresse gesetzt und alle Läden in der Nähe anzeigt. |
| Auslöser | Benutzer möchte nach einen Döner Laden suchen |
| Ergebnis | Benutzer finden einen Döner-Laden in der Nähe der eingegebenen Adresse |
| Akteure | angemeldeter Benutzer |
| Eingehende Informationen | Adresse |
| Vorbedingungen | Benutzer muss angemeldet sein und eine Adresse eingeben, außerdem muss eine Internetverbindung bestehen |
| Nachbedingungen | Es kann die Karte mit einer anderen Adresse angezeigt werden |
| Ablauf | 1. User ist angemeldet 2. User gibt eine Adresse ein 3. Karte mit allen Döner-Ständen, im Umfeld seiner eingegebenen Adresse, werden angezeigt. |
| Risiko | gering |
| Aufwand | mittel |
| Verbindlichkeit | hohe Priorität |
| Dringlichkeit | Release 0.8 |

### Döner-Stände in der Umgebung finden

|  |  |
| --- | --- |
| Systemanwendungsfall |  |
| Name | /LF120/ Döner-Stände in der Umgebung finden |
| Art | Anwendungsfall |
| Kurzbeschreibung | Sobald die App, nach der Anmeldung, geöffnet wird, öffnet sich eine Karte die deinen Standort und alle Döner-Läden in der Nähe auf der Karte anzeigt. |
| Auslöser | Benutzer möchte nach einen Döner Laden suchen |
| Ergebnis | Benutzer finden einen Döner-Laden in der Nähe der eingegebenen Adresse |
| Akteure | angemeldeter Benutzer |
| Eingehende Informationen | Benutzer Position |
| Vorbedingungen | Benutzer muss angemeldet sein, außerdem muss eine Internetverbindung bestehen |
| Nachbedingungen | Benutzer kann einen Döner-Laden in seiner Umgebung finden |
| Ablauf | 1. User ist angemeldet 2. User öffnet die App |
| Risiko | gering |
| Aufwand | mittel |
| Verbindlichkeit | Hohe Priorität |
| Dringlichkeit | Release 0.8 |

### Döner-Karte filtern

|  |  |
| --- | --- |
| Systemanwendungsfall |  |
| Name | /LF130/ Döner-Karte filtern |
| Art | Anwendungsfall |
| Kurzbeschreibung | Man soll die Möglichkeit haben, die Karte nach bestimmten Kriterien zu filtern, um eine genauere Suche zu gestalten. |
| Auslöser | Benutzer möchte nach einen Döner Laden in seiner Umgebung suchen |
| Ergebnis | Benutzer finden einen Döner-Laden in seiner Nähe |
| Akteure | angemeldeter Benutzer |
| Eingehende Informationen | Filter-Informationen |
| Vorbedingungen | Benutzer muss angemeldet sein und einen Filter auswählen |
| Nachbedingungen | Benutzer kann nach einem bestimmten Muster nach einen Döner-Laden suchen |
| Ablauf | 1. User öffnet Filter-Optionen 2. User wählt die Filter Optionen aus 3. Döner-Stände werden nach einem bestimmten Chema auf der Karte aufgelistet |
| Risiko | mittel |
| Aufwand | mittel |
| Verbindlichkeit | mittlere Priorität |
| Dringlichkeit | Release 0.8 |

## Freunde verwalten

### Freunde hinzufügen

|  |  |
| --- | --- |
| Systemanwendungsfall |  |
| Name | /LF210/ Freunde hinzufügen |
| Art | Anwendungsfall |
| Kurzbeschreibung | Man soll über eine Suchleiste Freunde finden können. Diese werden durch den  Benutzernamen gefiltert. |
| Auslöser | Benützer möchte einen Freund hinzufügen |
| Ergebnis | Benutzer fügt einen Freund zu seiner Freundesliste hinzu |
| Akteure | angemeldete Benutzer |
| Eingehende Informationen | Benutzername |
| Vorbedingungen | Benutzer sollte den Namen des Freundes kennen, damit er danach suchen kann |
| Nachbedingungen | Ein Freund wird gefunden |
| Ablauf | 1. User öffnet das Freundesfenster 2. Wählt Freunde-Hinzufügen 3. Gibt den Namen des Benutzers ein 4. Fügt den Freund zu seiner Liste hinzu |
| Risiko | Gering |
| Aufwand | mittel |
| Verbindlichkeit | mittlere Priorität |
| Dringlichkeit | Release 0.8 |

### Freundesliste anzeigen

|  |  |
| --- | --- |
| Systemanwendungsfall |  |
| Name | /LF220/ Freundesliste anzeigen |
| Art | Anwendungsfall |
| Kurzbeschreibung | Man soll sich eine Liste anzeigen lassen können, die alle Freunde anzeigt, die Online und Offline sind. |
| Auslöser | Benutzer möchte sich seine Freunde anzeigen lassen |
| Ergebnis | Benutzer sieht alle Freunde, die On-, und Offline sind |
| Akteure | angemeldeter Benutzer |
| Eingehende Informationen | / |
| Vorbedingungen | Benutzer muss Freunde haben und sich diese anzeigen lassen wollen |
| Nachbedingungen | Benutzer hat einen Überblick über all seine Freunde |
| Ablauf | 1. Benutzer geht auf Freunde anzeigen 2. Es öffnet sich ein Fenster mit allen Freunden |
| Risiko | gering |
| Aufwand | mittel |
| Verbindlichkeit | mittlere Priorität |
| Dringlichkeit | Release 1.1 |

### Freunde entfernen

|  |  |
| --- | --- |
| Systemanwendungsfall |  |
| Name | /LF230/ Freunde entfernen |
| Art | Anwendungsfall |
| Kurzbeschreibung | Man soll aus der Freundesliste Freunde entfernen können. |
| Auslöser | Benutzer möchte einen Freund entfernen |
| Ergebnis | Benutzer entfernt einen Freund |
| Akteure | angemeldeter Benutzer |
| Eingehende Informationen | Gelöschter Freund |
| Vorbedingungen | Benutzer muss Freunde haben, damit er einen entfernen kann |
| Nachbedingungen | Benutzer kann seine Freundesliste verwalten |
| Ablauf | 1. Auf Freunde anzeigen gehen 2. Freund auswählen und auf entfernen drücken |
| Risiko | Mittel |
| Aufwand | Mittel |
| Verbindlichkeit | hohe Priorität |
| Dringlichkeit | Release 1.1 |

## Freunde benachrichtigen

### Freunde per App einladen

|  |  |
| --- | --- |
| Systemanwendungsfall |  |
| Name | /LF310/ Freunde per App einladen |
| Art | Anwendungsfall |
| Kurzbeschreibung | Man soll nach erfolgreicher Suche, Freunde auf einen Döner einladen können. Man soll den Döner-Stand, Nachrichtentext, Uhrzeit und Datum für das Treffen angeben. Die eingeladenen Freunde bekommen eine Nachricht zugeschickt die alle wichtigen Daten enthält. |
| Auslöser | Benutzer möchte einen Freund auf einen Döner einladen |
| Ergebnis | Freund erhält eine Einladung für einen Döner |
| Akteure | angemeldete Benutzer |
| Eingehende Informationen | Adresse, Nachrichtentext, Uhrzeit und Datum |
| Vorbedingungen | Man hat sich einen Döner-Laden ausgesucht, und man hat Freunde die man einladen kann |
| Nachbedingungen | Ein Freund wird eingeladen |
| Ablauf | 1. Freundesliste anzeigen 2. Freund auswählen 3. Einladungs-Informationen eintragen 4. Einladung verschicken |
| Risiko | Mittel |
| Aufwand | Groß |
| Verbindlichkeit | hohe Priorität |
| Dringlichkeit | Release 0.8 |

### Einladung empfangen und bestätigen

|  |  |
| --- | --- |
| Systemanwendungsfall |  |
| Name | /LF320/ Einladung empfangen und  bestätigen |
| Art | Anwendungsfall |
| Kurzbeschreibung | Angemeldet: Nachdem dich ein Freund auf einen Döner eingeladen hat, soll sich ein kleines Nachrichten Fenster öffnen und die Einladung anzeigen. Man soll die Anfrage mit einem Button annehmen, oder ablehnen können. Abgemeldet: Wenn man die App nicht geöffnet hat, soll eine Nachricht am Handy angezeigt wer-den, dass man eine Einladung erhalten hat. Es soll außerdem ein Icon als Eyecat-cher angezeigt werden. |
| Auslöser | Freund möchte dich auf einen Döner einladen |
| Ergebnis | Man erhält eine Einladung |
| Akteure | Freund und man selber |
| Eingehende Informationen | / |
| Vorbedingungen | Benutzer muss dich in seiner Freundesliste haben, und alle wichtigen Informationen in die Einladung eingetragen haben |
| Nachbedingungen | Man erhält eine Einladung, die man Annehmen oder ablehnen kann |
| Ablauf | 1. Pop-Up Nachricht öffnet sich 2. Einladung öffnet sich per Druck auf die Einladung 3. Einladung kann bestätigt oder abgelehnt werden |
| Risiko | Mittel |
| Aufwand | Mittel |
| Verbindlichkeit | hohe Priorität |
| Dringlichkeit | Release 1.1 |

# Produktdaten

## Datenbankprogramm

Das Datenbankprogramm wird von den Mitarbeitern genutzt werden um die Einträge zu machen.

## Webseite

Die Webseite wird hauptsächlich vom Kunden genutzt um sich Autos vor zu mieten und um sich über die Firma zu informieren.

## Schulungsprogramm

Das Schulungsprogramm wird von den Mitarbeitern genutzt werden um sich Fähigkeiten über das neue Datenbankprogramm anzueignen. Dieses Programm wird vor dem Datenbankprogramm verwendet.

# Produktleistungen

/LL10/ Jeder Benutzername soll nur einmal vergeben werden

/LL20/ Ein Benutzer soll einen bestimmten Dönerstand nur einmal Bewerten können

/LL30/ Die App soll auf allen Android Systemen ab der Version Lollipop 5.0

/LL40/ Die Rezensionen der Kunden werden auf „Böse Wörter“ gefiltert

/LL50/ Anmeldedaten sollen auf Wunsch gespeichert werden können

# Benutzungsschnittstelle

* Bildschirmlayout: 16:9,
* Drucklayout: A4,ohne Zoom, Alle Seiten im Bereich
* Tastaturbelegung: deutsche Tastatur(qwertz)
* Dialogstruktur: Dialoge über DropDown Felder und Buttons



# Qualitätsbestimmung

In diesem Kapitel wird festgelegt, welche Qualitäts-Merkmale das zu entwickelnde Produkt in welcher Qualitätsstufe besitzen soll. Voraussetzung für die Qualitäts-Zielbestimmung ist, dass die Qualitäts-Merkmale in operationalisierter Form vorliegen.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Produktqualität | Sehr gut | Gut | Normal | Nicht relevant |
| **Funktionalität** |  | X |  |  |
| Angemessenheit |  |  |  | X |
| Richtigkeit |  | X |  |  |
| Interoperabilität |  | X |  |  |
| Ordnungsmäßigkeit |  |  | X |  |
| Sicherheit |  | X |  |  |
| **Zuverlässigkeit** |  | X |  |  |
| Reife |  | X |  |  |
| Fehlertoleranz |  |  | X |  |
| Wiederherstellbarkeit |  | X |  |  |
| **Benutzbarkeit** |  | X |  |  |
| Verständlichkeit |  | X |  |  |
| Erlernbarkeit |  | X |  |  |
| Bedienbarkeit |  | X |  |  |
| **Effizienz** |  |  | X |  |
| Zeitverhalten |  |  | X |  |
| Verbrauchsverhalten |  |  | X |  |
| **Änderbarkeit** |  |  | X |  |
| Analysierbarkeit |  |  | X |  |
| Modifizierbarkeit |  |  | X |  |
| Stabilität |  | X |  |  |
| Prüfbarkeit |  |  | X |  |
| **Übertragbarkeit** |  |  | X |  |
| Anpassbarkeit |  |  | X |  |
| Installierbarkeit |  |  | X |  |
| Konformität |  | X |  |  |
| Austauschbarkeit |  |  | X |  |
|  |  |  |  |  |

# Globale Testfälle

## Auto mieten

Kunde will ein Auto über die Webseite mieten und ist noch nicht registriert.

/LF110/ Kundenregistration annehmen

/LF130/ Verfügbarkeit anzeigen

/LF120/ Autodaten anzeigen

/LF150/ Auto auf der Webseite mieten

# Entwicklungsumgebung

In diesem Kapitel wird die Entwicklungs-Umgebung des Produktes beschrieben. Es wird festgelegt, welche Konfigurationen bzgl. *Software*, *Hardware* und *Orgware* für die Entwicklung des Produktes benötigt wird. Diese Festlegungen sind insbesondere dann notwendig, wenn Entwicklungs- und Zielmaschine unterschiedlich sind.

Bei *Entwicklungs-Schnittstellen* ist unter Umständen aufzuführen, über welche einzuhaltenden Hard- und Software-Schnittstellen Entwicklungs- und Zielrechner gekoppelt sind.

Unter *Software* ist insbesondere aufzuführen, welche Software-Werkzeuge, z.B. Compiler, benötigt werden.

## Software

SQL

## Hardware

3 Laptops,1 Server

## Orgware

Sicherheitsanforderungen, Server-Secure

## Entwicklungs-Schnittstellen

SQL

# Termine

## Meilensteintermine

|  |  |  |
| --- | --- | --- |
| Meilenstein | Deliverable | Datum |
| Start | Dokumentation mit allen Informationen | Voraussichtlich Am 14.03.2017 |
| Prototyp | Prototyp der App mit gewolltem Design | Voraussichtlich  Am 10.04.2017 |
| RC-Version | Visualisierung mit kompletter GUI und allen Funktionen | Voraussichtlich  Am 22.05.2017 |
| Abnahme | Fertiges Produkt einsatzbereit | Am 31.06.2017 |

## PSP



# Sonstiges

Es muss genug Platz für den Server vorhanden sein.

Falls Hilfe notwendig ist, wird das in den ersten 2 Wochen von uns übernommen.