

Dokumentation Teil 1

PartyPlanner

P

Projektteam: Lukas Aschmann, Simon Köpf & Luca Lindegger

# Abstract

Organisationstool für BYO-Veranstaltungen.

Man kann in einem geschützten Bereich einen Event erstellen und den Link via E-Mail versenden.  
Man muss nur angemeldet sein um einen Event zu erstellen.

Danach können alle mit dem Link sich in einer Liste eintragen, wer was mitnehmen will.

Man soll sich einloggen (read) und registrieren können, man soll neue Events erstellen können (create) und Benutzer sollen Dinge hinzufügen können (update). Der Eventmanager soll die Möglichkeit haben den Event zu löschen (delete).

Inhaltsverzeichnis

[1 Abstract 2](#_Toc466373117)

[2 Anforderungsanalyse 4](#_Toc466373118)

[2.1 User Stories 4](#_Toc466373119)

[3 Lieferumfang 5](#_Toc466373120)

[3.1 Applikation 5](#_Toc466373121)

[3.2 Dokumentation 5](#_Toc466373122)

[4 GUI Design 6](#_Toc466373123)

[4.1 MockUp 6](#_Toc466373124)

[5 OO Analyse- / Design 7](#_Toc466373125)

[5.1 Klassendiagramm 7](#_Toc466373126)

[5.2 ERM 7](#_Toc466373127)

# Anforderungsanalyse

## User Stories

|  |  |
| --- | --- |
| Funktionale-US-01 | Beschreibung |
| User Story | User kann sich registrieren. |
| Art | Muss-Ziel |
| Aufwand | 6 |

|  |  |
| --- | --- |
| Funktionale-US-02 | Beschreibung |
| User Story | User kann sich einloggen. |
| Art | Muss-Ziel |
| Aufwand | 1 |

|  |  |
| --- | --- |
| Funktionale-US-03 | Beschreibung |
| User Story | Eingeloggter User kann einen neuen Event erstellen. |
| Art | Muss-Ziel |
| Aufwand | 3 |

|  |  |
| --- | --- |
| Funktionale-US-04 | Beschreibung |
| User Story | Nicht eingeloggter User kann den Event sehen |
| Art | Muss-Ziel |
| Aufwand | 3 |

|  |  |
| --- | --- |
| Funktionale-US-05 | Beschreibung |
| User Story | Nicht eingeloggter User kann in sich in einem bestehenden Event eintragen als Mitbringer von etwas |
| Art | Muss-Ziel |
| Aufwand | 3 |

|  |  |
| --- | --- |
| Funktionale-US-06 | Beschreibung |
| User Story | Eingeloggter Urheber kann den Event löschen |
| Art | Muss-Ziel |
| Aufwand | 3 |

|  |  |
| --- | --- |
| Funktionale-US-07 | Beschreibung |
| User Story | Beim Erstellen des Events kann der Urheber Namen und Menge der benötigten Artikel eingeben |
| Art | Muss-Ziel |
| Aufwand | 3 |

|  |  |
| --- | --- |
| Funktionale-US-08 | Beschreibung |
| User Story | Nach dem Erstellen des Events bekommt der Urheber einen Link den er teilen kann |
| Art | Muss-Ziel |
| Aufwand | 3 |

|  |  |
| --- | --- |
| Funktionale-US-09 | Beschreibung |
| User Story | User kann einen neuen Artikel eintragen und ihn mitnehmen |
| Art | Kann-Ziel |
| Aufwand | 3 |

|  |  |
| --- | --- |
| Funktionale-US-10 | Beschreibung |
| User Story | Urheber kann den Event nachträglich bearbeiten |
| Art | Kann-Ziel |
| Aufwand | 3 |

|  |  |
| --- | --- |
| Funktionale-US-11 | Beschreibung |
| User Story | Urheber kann nach dem Erstellen des Events direkt seine Freunde (bzw. deren Email-Adressen) eingeben, damit sie die Einladung bekommen |
| Art | Kann-Ziel |
| Aufwand | 3 |

|  |  |
| --- | --- |
| Nichtfunktionale-US-01 | Beschreibung |
| User Story | Als Projektleiter möchte ich, dass wir täglich ein DailyScrum abhalten |
| Art | Muss-Ziel |
| Aufwand | 3 |

# Lieferumfang

Am Ende des Projekts müssen folgende Produkte abgegeben werden:

## Applikation

Am Ende des Projektes muss folgendes abgegeben werden:

* Der gesamte Source Code in einem EAR File
* Ein vollständiger MySQL DB Dump File (SQL)

## Dokumentation

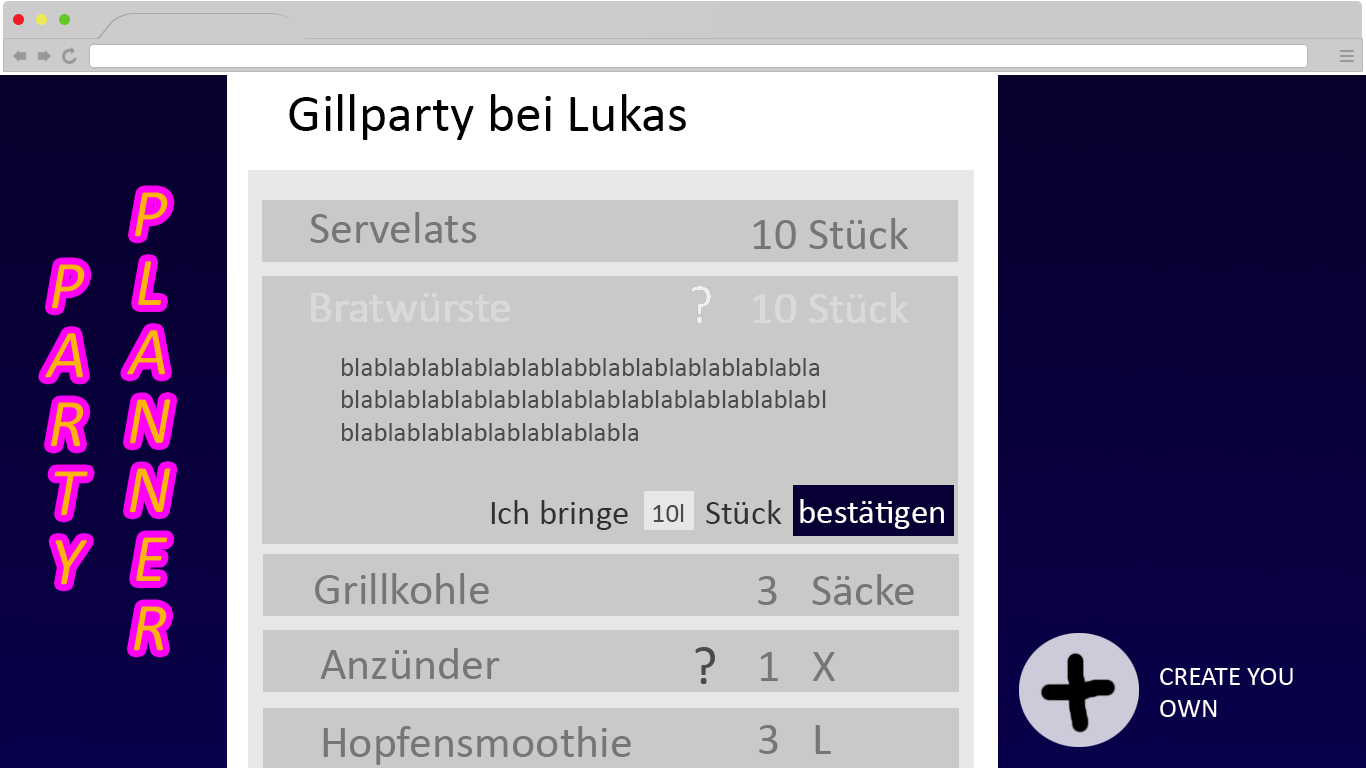
Die im Verlauf des Projekts entstandenen Dokumentationsteile 1 und 2 müssen jeweils zu den kommunizierten Terminen abgegeben werden. Alle geforderten Inhalte sind genügend genau beschrieben so, dass eine andere Gruppe das Projekt später weiterführen könnte. Zudem muss am Ende des Projektes ein Managament Summary (A4) erstellt werden, dass einen Gesamtüberblick über das Projekt gibt (Projektbeschreibung, Vorgehen, eingesetzte Hardware- und Software, bekannte Probleme, was wurde erreicht, was wurde weshalb nicht erreicht).

# GUI Design

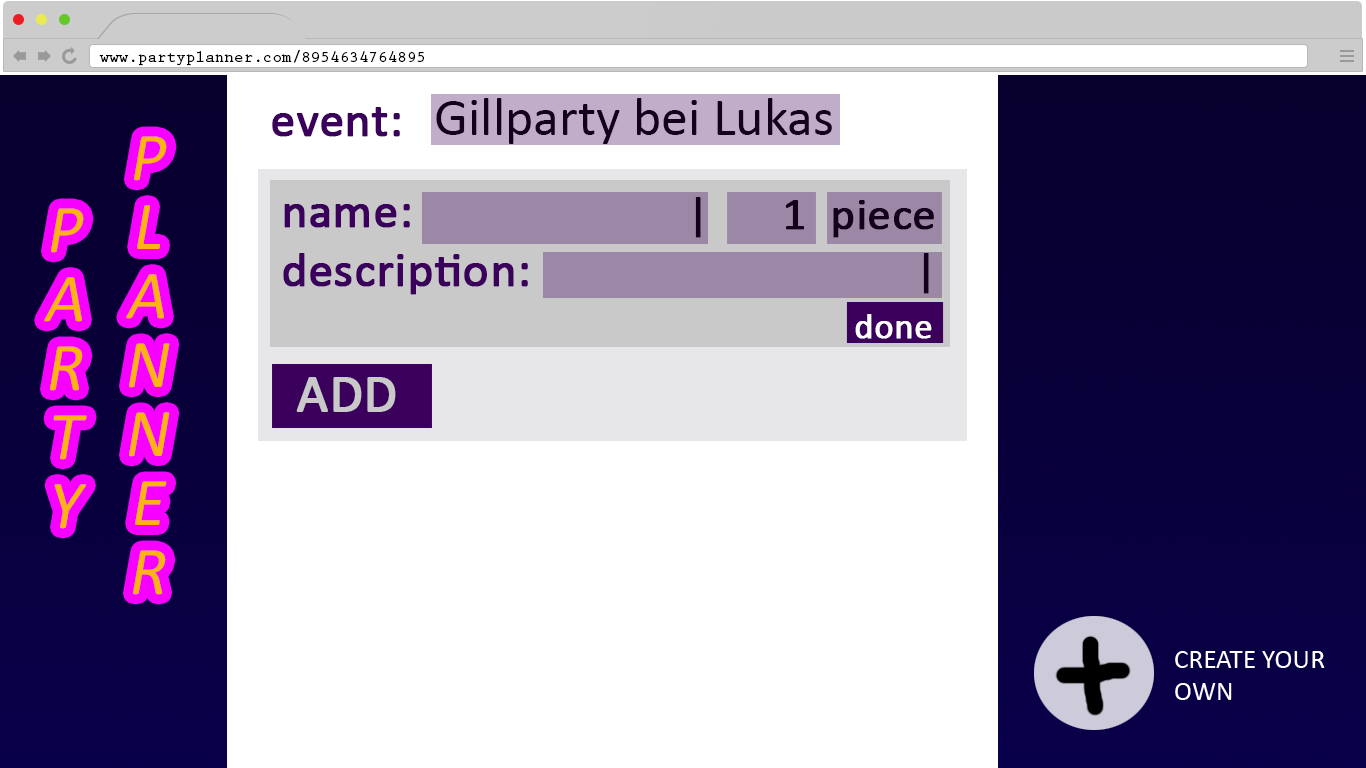
## MockUp



Event-Übersicht



Artikelauswahl (? = Beschreibung vorhanden)

 Formular zum mitbringen von Dingen

# OO Analyse- / Design

## Klassendiagramm

Vollständiges UML Klassendiagramm. Je nach Projekt kann dieses nur die Modellschicht (Abstraktion der Realität) darstellen oder aber bereits alle Klassen (technische Architektur), welche zu implementieren sind, beinhalten. Klassenbeziehung (Vererbung, Assoziation), Sichtbarkeitsmodifikatoren, static, Datentypen, Attribute und Methoden müssen vollständig sein. Bei Assoziationen müssen Richtung, Rollen und Kardinalitäten angegeben werden. Java Standard Klassen und Klassen aus verwendeten Frameworks müssen nicht modelliert werden, ausser es trägt zu einem besseren Verständnis bei.

## ERM

