

|  |  |
| --- | --- |
| Autor(en): | Joan Künzler |
| Ausgabe: | Version 1  12.12.2016 |

Inhaltsverzeichnis

[1. Vorgehen 3](#_Toc469780370)

[1.1 Analyse der Vorgabe 3](#_Toc469780371)

[1.1.1 Ausgangslage 3](#_Toc469780372)

[1.1.2 Abgrenzung 3](#_Toc469780373)

[1.1.3 Rahmenbedingungen gemäss Vorgabe 3](#_Toc469780374)

[1.2 Grob-Konzept 4](#_Toc469780375)

[1.2.1 Grundfunktionalität 4](#_Toc469780376)

[1.2.2 Navigation 4](#_Toc469780377)

[1.2.3 Als eingeloggter Benutzer 4](#_Toc469780378)

[1.2.4 Datenstruktur 4](#_Toc469780379)

[1.2.5 Benutzerdaten 5](#_Toc469780380)

[1.2.6 Technische Objekte 5](#_Toc469780381)

[1.3 Terminplanung 7](#_Toc469780382)

[1.4 Testfälle im Voraus beschreiben 7](#_Toc469780383)

[1.5 Vorgehens-Reflexion 8](#_Toc469780384)

[2. Konzeption 9](#_Toc469780385)

[2.1 Entwurf der Funktionalität 9](#_Toc469780386)

[2.2 Realisierungskonzept 11](#_Toc469780387)

[2.3 Login Maske 11](#_Toc469780388)

[2.4 Registreren Maske 12](#_Toc469780389)

[2.5 Meine Ausleihen 13](#_Toc469780390)

[2.6 Mein Profil 14](#_Toc469780391)

[2.7 Konzept des Session-Handlings 15](#_Toc469780392)

[2.8 Formularprüfung 15](#_Toc469780393)

[2.8.1 Benutzer 15](#_Toc469780394)

[2.9 Testkonzept 15](#_Toc469780395)

[2.10 Sicherheitsrelevante Anforderungen 15](#_Toc469780396)

[3. Quellenverzeichniss 16](#_Toc469780397)

# Vorgehen

## Analyse der Vorgabe

In diesem Abschnitt wird die Ausgangslage beschrieben und die Abgrenzungen / Rahmenbedingungen   
werden gemäss Vorlage definiert und festgelegt.

### Ausgangslage

Die Ludothek „SpielGut“ besitzt bereits ein als Thick Client realisiertes Administrations-/Verwaltungssystem.

Dieses Verwaltungssystem wird als Standalone angesehen und ist nicht Teil dieses Projektes.

Die Ludothek möchte neu den Kunden eine Web-Applikation zur Verfügung stellen, mit der die im Verwaltungssystem erfassten Spiele ausgeliehen werden können.

### Abgrenzung

Die auf den Endkunden angepasste WebApplikation hat nur den Zweck Spiele Online auszuleihen.

Es wird keine Schnittstelle für die Benutzerverwaltung implementiert und ebenso keine speziellen Berechtigungen vergeben (Administrator, Moderator, Benutzer).

### Rahmenbedingungen gemäss Vorgabe

Der Kunde:

* … kann sich registrieren (eMail, Passwort, eMail-Bestätigung fakultativ)
* … kann sich an der Web-App anmelden
* … muss beim erstmaligen Anmelden zwingend seine persönlichen Daten eingeben, ohne die eine Ausleihe nicht möglich ist
* … kann sein Profil jederzeit anschauen und ändern
* … kann sein Passwort zurücksetzen
* … kann neue Ausleihen erstellen
* … kann aktive Ausleihen verlängern
* … kann all seine Ausleihen (aktiv als auch die in der Vergangenheit) anschauen

Die nachfolgenden Bedingungen dienen der Einfachheit und sind einzuhalten:

* Spiele sind serverseitig in einer Datei gespeichert (ca. 6 Spiele)
* Ausleihen werden serverseitig in einer Datei gespeichert
* Pro Ausleihe darf nur ein Spiel geliehen werden
* Ein Kunde kann mehrere aktive Ausleihen haben
* Ein Spiel kann genau einmal ausgeliehen werden (1x vorrätig in der Datenbank)

## Grob-Konzept

In nachfolgendem Abschnitt wird vor allem auf die technische Realisierung eingegangen.

### Grundfunktionalität

Im Grunde geht es darum, dass ein Benutzer Ausleihen erstellen kann. Diese Ausleihen beinhalten gemäss Rahmenbedingungen ein Spiel. Besagter Benutzer kann jederzeit eine Ausleihe verlängern.

Damit ein Benutzer sich überhaupt an der Applikation anmelden kann, muss er zwingend einen   
Benutzeraccount erstellt haben. Bei Erstellung eines solchen Accounts wird der Benutzer aufgefordert seine persönlichen Daten einzugeben, denn ohne diese, ist eine Ausleihe nicht möglich.

### Navigation

Falls ein Benutzer nicht eingeloggt ist, wird er auf die Login-Maske verwiesen.

Auf dieser kann er sich einloggen oder registrieren.

Nach dem Login kann der Benutzer auf sein persönliches Profil navigieren. Diese Seite ermöglicht es ihm, seine persönlichen Daten zu ändern und das Passwort zurück zu setzen. Der Benutzer wird sein Passwort nie im Klartext sehen, da es sich in einem „hashed“ Zustand befindet, welcher aus Sicherheitsgründen unumkehrbar ist.  
Der User kann sich auch dazu entscheiden, auf die „Meine Ausleihen“ Seite zu navigieren. Diese Seite ermöglicht es ihm, seine vergangenen und aktuellen Ausleihen einzusehen. Er hat auch die Möglichkeit,  
Ausleihen zu verlängern oder neue zu erstellen.

#### Als nicht eingeloggter Benutzer

Login

|-----Registrieren

### Als eingeloggter Benutzer

Mein Account

|-----Account Einstellungen

| |-----Ändern von persönlichen Daten

| |-----Ändern des Passwortes

|-----Meine Ausleihen

|-----Neue Ausleihe erstellen

### Datenstruktur

Die Informationen der Applikation (User Daten, Ausleihen, Spiele, usw.) werden in der JavaScript Object Notation (kurz JSON) auf dem Server in einer Datei gespeichert.

Jedes Spielzeug befindet sich nur einmal in der „Datenbank“ und kann daher auch nur einmal zur selben Zeit ausgeliehen werden.

### Benutzerdaten

Die Passwortregel erlaubt jede Kombination von Zeichen, solange sich die Anzahl zwischen

8 Zeichen und 64 Zeichen befindet. Um den Server vor DOS Attacken (erstellen vieler Benutzer in kurzer Zeit) zu schützen, werden Passwörter die länger als 64 Zeichen sind nicht toleriert.

Passwortrichtlinien werden ansonsten keine aufgezwungen, es wird vom Benutzer nicht verlangt eine bestimmte Anzahl von Sonderzeichen und oder Zahlen / Grossbuchstaben zu verwenden. Solche Richtlinien führen meist dazu, dass Benutzer Sonderzeichen und Zahlen am Anfang oder Ende des Passwortes hinzufügen. Es ist ohnehin sicherer ein Längeres Passwort zu wählen, anstelle eines welches eine komplexere Struktur beinhaltet.

### Technische Objekte

Um Daten zu verarbeiten, wird sich die WebApplikation an einen Objekt Orientierten Ansatz richten.

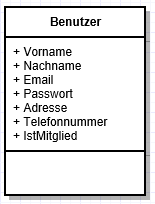
Dies bedeutet, dass mehrere Objekte eines Typs existieren können und existieren werden.

Nachfolgend werden Objekte aufgelistet welche zwingend notwendig sind, um die Funktionalität der Applikation sicher zu stellen.

Die Möglichkeit besteht, dass sich während der Entwicklung die bestehenden Objekte verändern, oder neue Objekte dazu kommen.

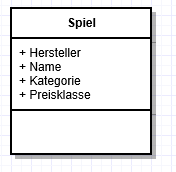
#### Benutzer

Das Benutzer-Objekt dient als Daten Struktur und beinhaltet keinerlei Business Logik.



#### Spiel

Das Spiel-Objekt dient als Daten Struktur und beinhaltet keinerlei Business Logik.



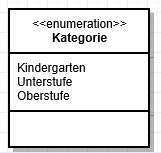
#### Preisklasse

Das Preisklasse-Objekt dient als Daten Struktur und beinhaltet keinerlei Business Logik.



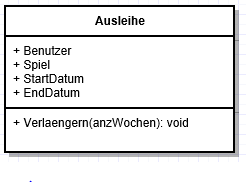
#### Kategorie

Die Klasse Kategorie ist ein Enumerator welcher die Kategorien spezifiziert.



#### Ausleihe

Das Ausleihe-Objekt verknüpft den Benutzer und das ausgeliehene Spiel. Es definiert Beginn und der   
Ausleihe. Es ist möglich, die Ausleihe zu verlängern.



## Terminplanung

|  |  |
| --- | --- |
| 8.12.2016 | Start der Projektarbeit |
| 12.12.2016 | Initialisierung des Dokuments |
| 12.12.2016 | Analyse(Rahmenbedingungen, Abgenzungen) |
| 12.12.2016 | Grob-Konzept(Planung, Mockups, Technische Objekte) |
| 13.12.2016 | Umsetzung; Benutzerverwaltung und Layout gemäss Mockups |
| 14.12.2016 | Umsetzung; Ausleihe Implementieren |
| 15.12.2016 | Überarbeitung des Designs |
| 17.12.2016 | Überrabeiten des Dokuments und Abschluss der Arbeit |

## Testfälle im Voraus beschreiben

|  |  |  |
| --- | --- | --- |
| Nummer | Beschreibung | Erwartetes Resultat |
| 1 | Registrieren des Users „Karl“ mit 16 Stelligem Passwort „0123456789ABCDEF“ | Benutzer wird in der Datenbank mit „hashed“ Passwort erfasst. |
| 2 | Login mit Benutzer Karl | Weiterleitung auf Maske „Mein Profil“ |
| 3 | Ändern des Wohnorts | Wohnadresse wird dementsprechend in der Datenbank auf den neuen Wert angepasst |
| 4 | Navigieren auf „Meine Ausleihen“ | Da keine Ausleihen vorhanden sind, werden keine angezeigt. |
| 5 | Erstellen einer Ausleihe | Erstellte ausleihe ist in der Datenbank vorhanden |
| 6 | Verlängern einer Ausleihe | Ausleihe wird in der Datenbank um angegebene Anzahl Wochen verlängert |
| 7 | Ausleihe zurückgeben / abschliessen | Auflistung der Ausleihe unter „Vergangene Ausleihen“ |
| 8 | Restarten des Browsers und erneutes öffnen der Webseite | Benutzer ist ausgeloggt und sieht Login Maske |
| 9 | Im eingeloggten zustand auf Ausloggen drücken | Abmelden aus der Applikation |

## Vorgehens-Reflexion

Da die Dokumentation vor der Umsetzung verfasst wurde, werden sich noch einige Dinge ändern.

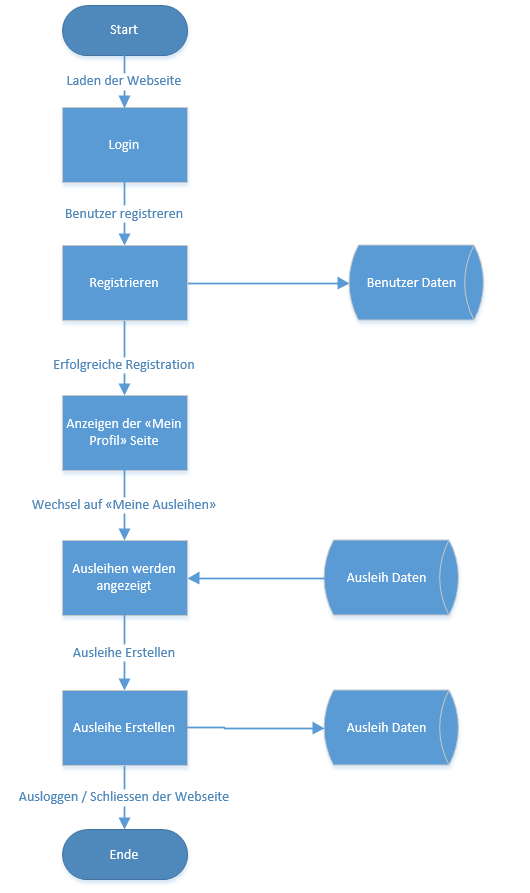
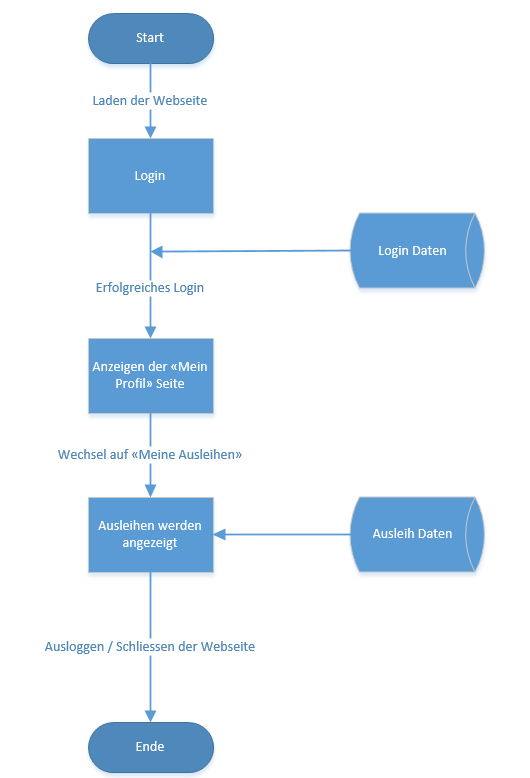
Meiner Meinung nach ist es seh von Vorteil ein Projekt nach IPERKA zu richten.

Die Phase des Informierens habe ich im Teil der Analyse, der Ausgangslage und den Ramenbedingungen untergebracht.  
Die Planung und Entscheidungen die ich traf, spiegeln sich im Rest dieses Dokumentes wieder.

Das Realisieren wird nach Terminplanung erfolgen und vorraussichtlich am Wochenende des 17.12 mit der finalen Version abgeschlossen. Die Kontrolle wird mittels Testfällen nicht nur nach sondern auch wärend der Realisierung durchgeführt. Es ist wichtig Tests nach jeder veränderung durch zu führen um ungewollte Nebenwirkungen zu minimieren.

# Konzeption

## Entwurf der Funktionalität

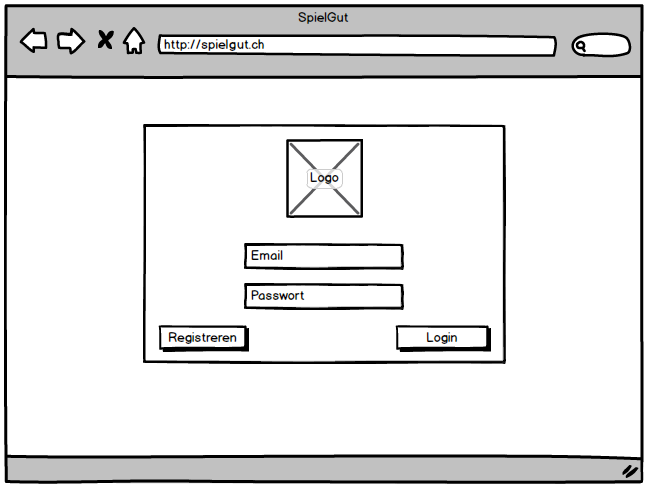


## Realisierungskonzept

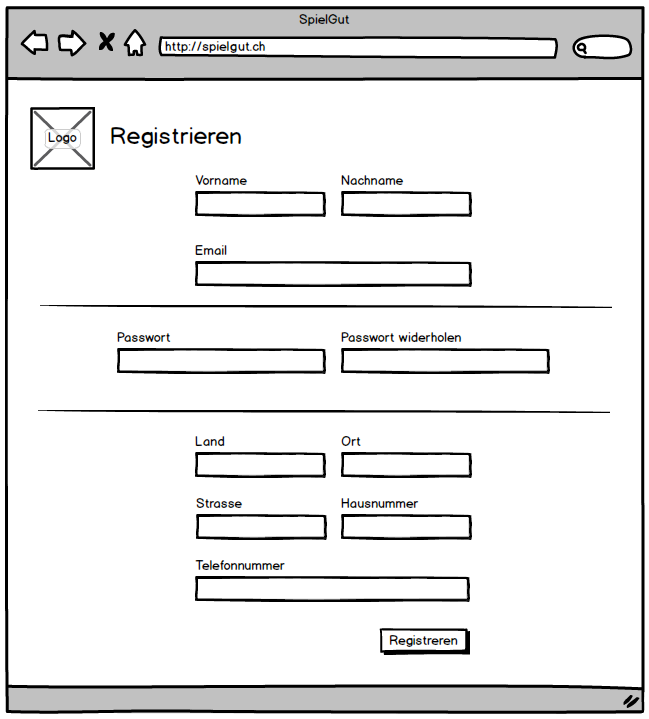
Nachfolgend werden die Mockups aufgeführt. Die Mockups dienen nur zur generellen Orientierung.

Die Realisierte Applikation wird mit sicherheit von den Mockups abweichen, aber die grunsätzliche Konzeption bleibt dieslbige.

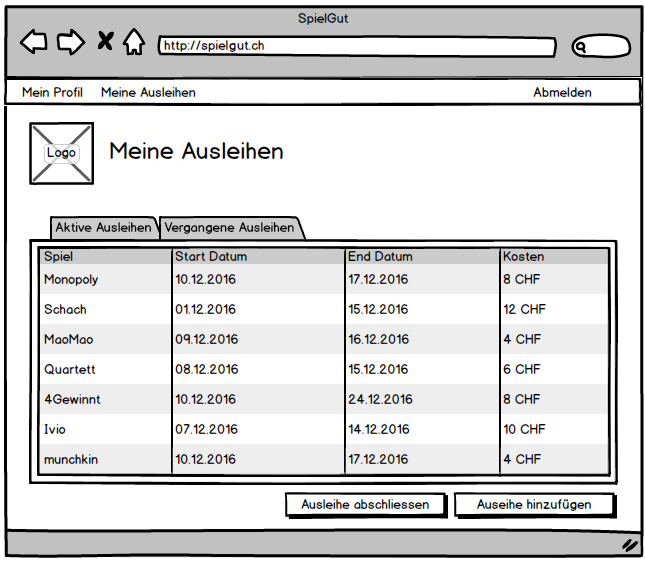
## Login Maske



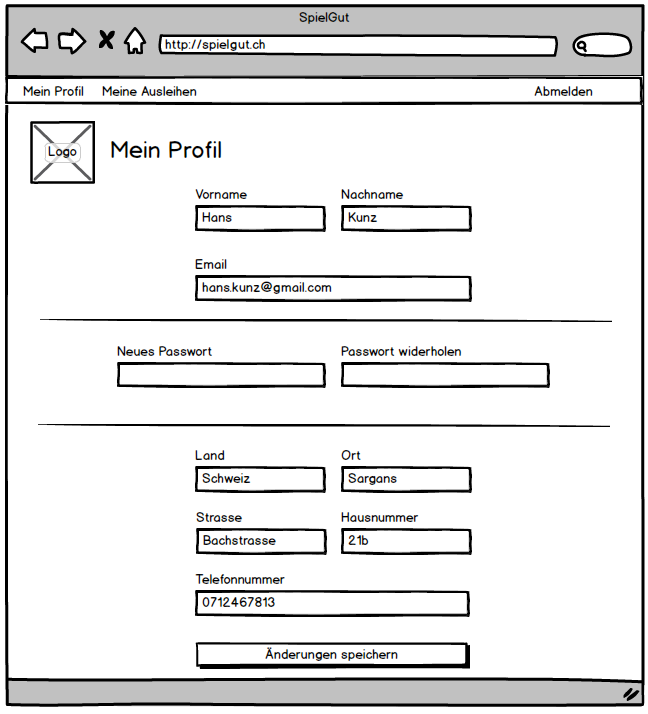
## Registreren Maske



## Meine Ausleihen



## Mein Profil



## Konzept des Session-Handlings

Beim Verbinden mit jeder Seite wird überprüft, ob die User Session Variable mit einem User in der Datenbank übereinstimmt. Falls dies zutrifft, darf der User auf die angeforderte Seite. Falls die Daten inkorrekt waren, wird der Benutzer auf die Login Seite verwiesen.

Die Session wird beim Schliessen des Browsers zerstört / gelöscht.

## Formularprüfung

Die Formularprüfung wird Clientseitig und da man dem Client nicht trauen kann, auch Serverseitig durchgeführt. Nachfolgend werden die Definitionen zu den bezüglichen Datenfeldern aufgeführt.

### Benutzer

Vorname: 40 Zeichen

Nachname: 40 Zeichen

Email: Überprüfung mittels regulärem Ausdruck max. 100 Zeichen

Passwort: Zwischen 8 und 64 Zeichen

PLZ: 4 Zeichen

Ort: 40 Zeichen

Strasse: 40 Zeichen

Hausnummer: 20 Zeichen

Telefonnummer: 20 Zeichen

## Testkonzept

Die Anfangs definierten Tests werden fortlaufend durchgeführt um zu gewährleisten das Änderungen an den einen Funktionen keine negativen / ungewollte Folgen auf andere Funktionalitäten haben.

## Sicherheitsrelevante Anforderungen

Es wird empfohlen die Datenübertragung nur mittels HTTPS durchzuführen. Dies wird implementiert sobald die Applikation Produktivgeschalten wird.

Das Passwort wird in jedem Fall verhashed in der Datenbank gespeichert.

# Quellenverzeichniss

Titelbild: <https://www.oxfordwebapps.co.uk/services/web-application-development> (12.12.2016)