**Pflichtenheft**

|  |
| --- |
| Caraval Webseite |

Bearbeiter: Thomas Sapelza

Betreuer: Zingerle Claudia

Pörnbacher Ulrich

Rainer Ulrich

Hvala Maximilian

Datum: 13.11.2017 (Version 0.4.0)

**Meilensteine**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Version | Datum | Bearbeiter | Kommentar | Status |
| 0.0.1 | 23.11.2016 | Thomas Sapelza | Start | WIP |
| 0.1.0 | 25.11.2016 | Thomas Sapelza | Überblick | WIP |
| 0.3.0 | 26.11.2016 | Thomas Sapelza | Muss + Kann Anforderungen | WIP |
| 0.7.5 | 27.11.2016 | Thomas Sapelza | Use Case Diagramm + Beschreibungen | WIP |
| 0.9.0 | 28.11.2016 | Thomas Sapelza | Klassenmodell | WIP |
| 0.9.5 | 29.11.2016 | Thomas Sapelza | Umsetzung | WIP |
| 1.0.0 | 30.11.2016 | Thomas Sapelza | Oberflächenprototyp | STABLE |

Inhaltsverzeichnis

[1. Anforderungen und Ziele 5](#_Toc498320260)

[1.1. Überblick 5](#_Toc498320261)

[1.2. Muss-Anforderungen 5](#_Toc498320262)

[1.2.1. Informationen über den Autor 5](#_Toc498320263)

[1.2.2. Informationen über das Genre 5](#_Toc498320264)

[1.2.3. Informationen über die Einstellung von Ort und Zeit 5](#_Toc498320265)

[1.2.4. Informationen über die Handlung 5](#_Toc498320266)

[1.2.5. Informationen über die Hauptcharaktere 5](#_Toc498320267)

[1.2.6. Informationen über die Hauptthemen 6](#_Toc498320268)

[1.2.7. Informationen über die Sprachlichen Aspekte 6](#_Toc498320269)

[1.2.8. Informationen über einige Bewertungen 6](#_Toc498320270)

[1.2.9. Interaktive Interaktion 6](#_Toc498320271)

[1.3. Kann-Anforderungen 6](#_Toc498320272)

[1.3.1. Animationen 6](#_Toc498320273)

[1.3.2. Ähnliche Bücher anzeigen lassen 6](#_Toc498320274)

[1.4. Zielgruppe und Anforderungen an die Benutzerführung 6](#_Toc498320275)

[1.4.1. Gäste 6](#_Toc498320276)

[2. Use Cases 7](#_Toc498320277)

[2.1. Akteure 7](#_Toc498320278)

[2.1.1. Gast 7](#_Toc498320279)

[2.2. Use Case Diagramm 8](#_Toc498320280)

[2.3. Use Case „Parkplätze anzeigen“ 9](#_Toc498320281)

[2.4. Use Case „Parkplatz reservieren“ 10](#_Toc498320282)

[2.5. Use Case „Ticket scannen“ 11](#_Toc498320283)

[2.6. Use Case „Ticket drucken“ 12](#_Toc498320284)

[2.7. Use Case „Auto suchen“ 13](#_Toc498320285)

[2.8. Use Case „Reservierung stornieren“ 14](#_Toc498320286)

[2.9. Use Case „Registrieren“ 15](#_Toc498320287)

[2.10. Use Case „Parkzeitüberschreitung bezahlen“ 16](#_Toc498320288)

[2.11. Use Case „Anmelden“ 17](#_Toc498320289)

[2.12. Use Case „Abmelden“ 18](#_Toc498320290)

[2.13. Use Case „Bewerten“ 19](#_Toc498320291)

[2.14. Use Case „Statistiken anzeigen“ 20](#_Toc498320292)

[2.15. Use Case „Statistiken exportieren“ 21](#_Toc498320293)

[2.16. Use Case „Bewertung löschen“ 22](#_Toc498320294)

[3. Klassenmodell 23](#_Toc498320295)

[3.1. Klassendiagramm 23](#_Toc498320296)

[4. Umsetzung 24](#_Toc498320297)

[4.1. Plattform 24](#_Toc498320298)

[4.2. Datenhaltung 24](#_Toc498320299)

[4.3. Oberfläche 24](#_Toc498320300)

[4.4. Systemarchitektur 24](#_Toc498320301)

[5. Oberflächenprototyp 25](#_Toc498320302)

# Anforderungen und Ziele

## Überblick

Es soll eine interaktive Webseite über das englische Buch Caraval von Stephanie Garber erstellt werden. Die Webseite soll einfach und verständlich Strukturiert sein und einen modernen „Look and Feel“ haben, um den Benutzer ein Eindruck über der Welt des Buches und der Autorin darzustellen. Sie sollte mindestens 3 Unterseiten mit Navigation aufweisen. Die Webseite sollte auf den gängigen Browsern heutiger Endgeräte laufen, die JavaScript unterstützen. Die Programmiersprache JavaScript steht in diesem Projekt vor allem im Fokus.

Die Webseite muss folgende Funktionen für die Benutzer beinhalten:

* Informationen über (in Englisch):
  + den Autor
  + das Genre (crime story, science fiction, love story, horror, …)
  + die Einstellung von Ort und Zeit
  + die Handlung
  + die Hauptcharaktere
  + die Hauptthemen
  + die sprachlichen Aspekte
* Bewertungen
* Interaktive Interaktion

Folgenden Kann-Anforderungen:

* Animationen
* Ähnliche Bücher anzeigen lassen
* Kommentar/Rezension System

## Muss-Anforderungen

### Informationen über den Autor

Der Benutzer hat die Möglichkeit sich Informationen über den Autor anzeigen zu lassen. Außerdem werden die wichtigsten Links (Webseiten, Buch kaufen) und Social network links des Autors verknüpft.

### Informationen über das Genre

Der Benutzer kann sich Informationen über das Genre des Buches anzeigen lassen.

### Informationen über die Einstellung von Ort und Zeit

Dem Benutzer werden Informationen über Ort und Zeit angezeigt.

### Informationen über die Handlung

Dem Benutzer wird eine zusammengefasste Handlung gezeigt, diese könnte man interaktiv realisieren.

### Informationen über die Hauptcharaktere

Dem Benutzer werden Interaktiv die Charaktere und deren wichtigsten Informationen angezeigt.

### Informationen über die Hauptthemen

Die Hautthemen des Buches werden den Benutzer aufgezeigt.

### Informationen über die Sprachlichen Aspekte

Dem Benutzer werden Informationen über die sprachlichen Aspekte dargestellt, eventuell ein paar interaktive Textstellen.

### Bewertungen

Der Benutzer kann interaktiv Bewertungen auswählen.

### Interaktive Interaktion

Dem Benutzer wird es ermöglicht Interaktiv die Webseite zu erforschen. Hinweise und Gesten geben dem Benutzer den Schluss was er machen soll.

## 

## Kann-Anforderungen

### Animationen

Die Webseite bietet moderne Animationen bei.

### Ähnliche Bücher anzeigen lassen

Dem Benutzer werden bei der Seite Genre ähnliche Bücher angezeigt.

### Kommentar/Rezension System

Den Gast wird die Möglichkeit gegeben Anonym oder mit Nickname einen Kommentar/Rezension zu schreiben, in Verbindung mit seiner E-Mail.

## Zielgruppe und Anforderungen an die Benutzerführung

Der Potenzielle Nutzer der Webseite sind Benutzer die sich über das Buch Caraval informieren wollen und im Alter von mindestens 12 Jahren befinden. Die Interaktion mit der Webseite sollte bereits bei der ersten Nutzung intuitiv bedienbar sein. Die Nutzer sollten die grundlegenden Kenntnisse in der Bedienung von Webseiten haben. Spezielle Kenntnisse werden von dieser Zielgruppe nicht vorausgesetzt. Die Zielgruppe kann nicht zwischen Benutzergruppen unterschieden werden. Spezielle Rechte sind auf der Webseite generell nicht nötig, das heißt es braucht kein Login/Registrierung. Nach dem Abschluss dieses Projekts ist die Webseite final, weil sich der Content des Buches nicht ändert und deswegen weder ein CMS noch ein Admin-Login benötigt wird.

### Gäste

Die Benutzergruppe „Gäste“ sind die Nutzer die die Webseite besuchen und begutachten. Gäste haben und brauchen keine spezielle Rechte wie oben bereits beschrieben wurde.

# Use Cases

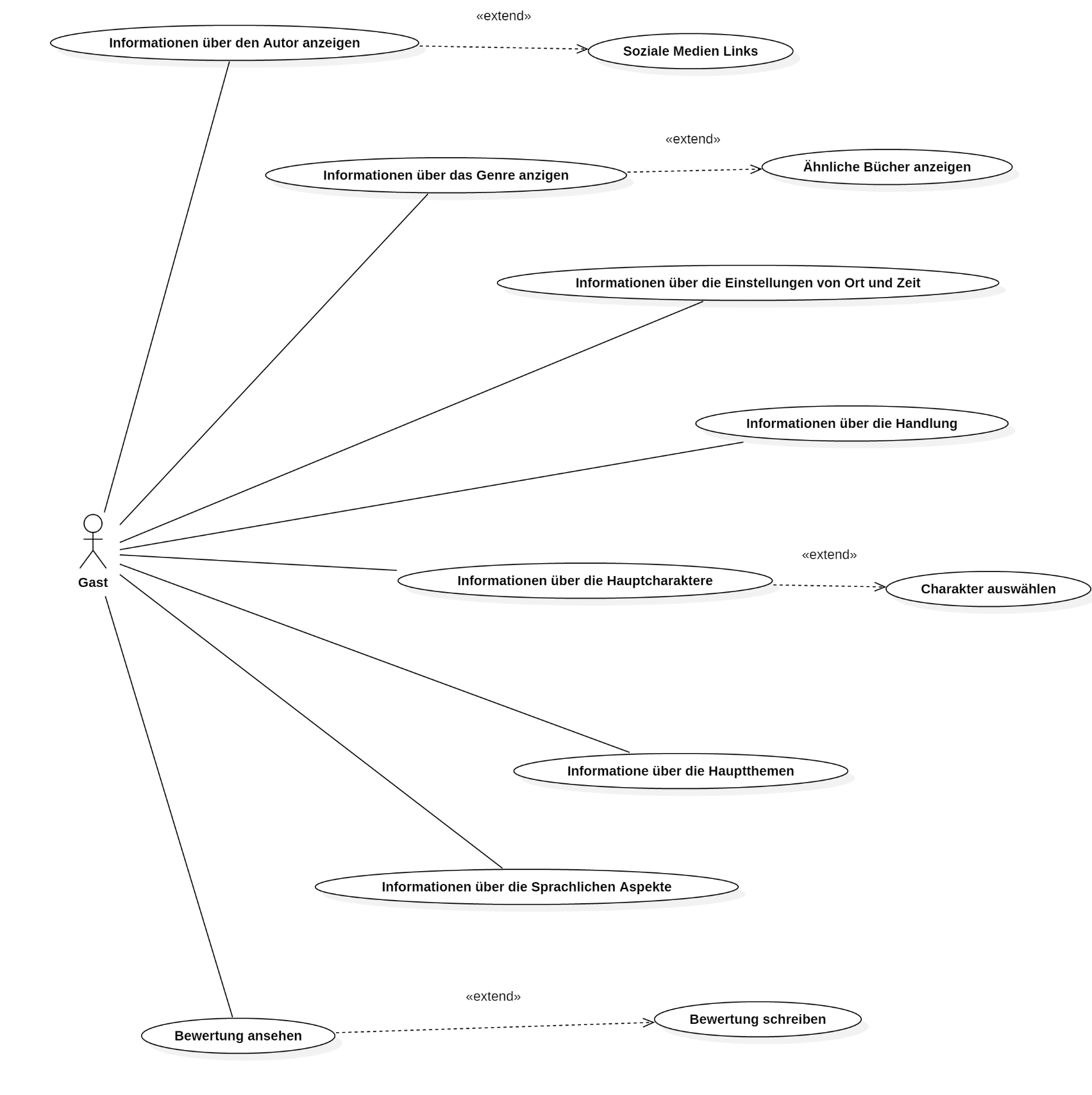
Dieses Kapitel beschreibt die Akteure bei Benutzung des Systems als auch die internen Abläufe auf der Basis der Anforderungen von Kapitel 1.

## Akteure

### Gast

Die Java-Applikation ist vor Ort unter minimalen Einschränkungen auch für Personen nutzbar welche sich nicht ausweisen (Registrierung/Anmeldung). Vor Ort kann ein Gast eine Reservierung abschließen. Die Website und die Smartphone-App sind für den Gast nur mit großen Einschränkungen nutzbar, sie können sich nur registrieren (Use Case „Registrieren“) oder das FAQ ansehen. Sie können eine Reservierung (Use Case „Parkplatz reservieren“) tätigen. Diese Personen fallen unter die Akteursgruppe „Gast“.

## Use Case Diagramm



## Use Case „Informationen über den Autor anzeigen“

**Kurze Beschreibung**

Der Benutzer sieht im Überblick Informationen über den Autor. Er kann auch auf die Social Media Links der Autorin klicken und wird auf die entsprechende Seite weitergeleitet.

**Auslöser**

Der Benutzer drückt auf den Button „Author“

**Akteur(e)**

Gast

**Vorbedingung(en)**

Der Browser hat die Webseite vollständig geladen und der Benutzer ist im „Hauptmenü“

**Schritte**

1. Der Benutzer klickt im Hauptmenü auf den Button „Author“.
2. Die Webseite zeigt Informationen über die Autorin an.
3. Falls der Benutzer auf die Social Network Icons (Facebook, Instagram, …) klickt, wird er auf die Sozialen Seiten der Autorin weitergeleitet.

**Alternative Schritt**

Der Benutzer scrollt zum Abschnitt „Author hinunter“

**Nachbedingung(en)**

-

**Inkludierte Use Cases**

-

**Involvierte Klassen**

-

## Use Case „Parkplatz reservieren“

**Kurze Beschreibung**

Der Benutzer hat ein Parkplatz ausgewählt, wählt den Zeitraum für die Reservierung aus und bezahlt.

**Auslöser**

Der Benutzer hat einen Parkplatz ausgewählt.

**Akteur(e)**

Gast Benutzer und registrierter Benutzer

**Vorbedingung(en)**

Der Benutzer hat einen Parkplatz vom Use Case „Parkplätze anzeigen“ ausgewählt.

**Schritte**

1. Der Gast hat ein Parkplatz gewählt.
2. Er wählt den Zeitraum für die Reservierung aus.
3. Der Benutzer wird aufgefordert den Zeitraum und den gewählten Parkplatz zu bestätigen.
4. Hat der Benutzer dies bestätigt dann wird er aufgefordert mit Bargeld zu Bezahlen.

**Alternative Schritte**

Hat ein registrierter Benutzer die Reservierung getätigt dann kann er mit Kreditkarte bezahlen, dazu muss er aber mit seinem Konto angemeldet sein.

**Nachbedingung(en)**

Der Status vom Parkplatz ist in diesem Zeitraum besetzt.

**Inkludierte Use Cases**

Parkplatz anzeigen, Ticket drucken, Anmelden

**Involvierte Klassen**

User, ParkingSpot

## Use Case „Ticket scannen“

**Kurze Beschreibung**

Der Benutzer scannt sein Ticket (QR-Code) an den QR-Code Scanner.

**Auslöser**

Der Benutzer hält sein Auto vor der Einfahrts- bzw. Ausfahrtsschranke oder will seine Reservierung stornieren (siehe Use Case „Reservierung stornieren“).

**Akteur(e)**

Gast Benutzer und registrierter Benutzer

**Vorbedingung(en)**

Der Benutzer besitzt ein gültiges Ticket.

**Schritte**

1. Das System fordert den Benutzer auf das Ticket an den QR-Code Scanner zu halten.
2. Der Benutzer hält sein Ticket an den QR-Code Scanner.

**Alternative Schritte**

-

**Nachbedingung(en)**

Hat der Benutzer das Ticket bei der Einfahrtsschranke gescannt kann er es dort nicht noch einmal scannen, er kann das Ticket nur mehr bei der Ausfahrtsschranke scannen. Sobald es bei der Ausfahrtsschranke gescannt wurde ist es kein gültiges Ticket mehr.

**Inkludierte Use Cases**

-

**Involvierte Klassen**

Ticket, Reservation, ParkingSpot

## Use Case „Ticket drucken“

**Kurze Beschreibung**

Sobald ein Benutzer die Reservierung getätigt und bezahlt hat wird das Ticket gedruckt.

**Auslöser**

Das Ticket wurde erfolgreich bezahlt.

**Akteur(e)**

Gast Benutzer und registrierter Benutzer

**Vorbedingung(en)**

Eine Reservierung wurde bestätigt.

**Schritte**

1. Das System druckt das Ticket mit Reservierungsbestätigung aus.
2. Der Benutzer entnimmt das gedruckte Ticket.

**Alternative Schritte**

Wurde die Reservierung von einem registrierten Benutzer getätigt (Alle Plattformen) bekommt er eine E-Mail mit dem QR-Code und der Reservierungsbestätigung, diese kann er dann selber ausdrucken. Hat er die Reservierung vor Ort getätigt dann bekommt er das gedruckte Ticket und die E-Mail.

**Nachbedingung(en)**

-

**Inkludierte Use Cases**

Anmelden

**Involvierte Klassen**

User, Ticket, Reservation, ParkingSpot

## Use Case „Auto suchen“

**Kurze Beschreibung**

Der Nutzer kann mithilfe seines Tickets den Parkplatz sich den Parkplatz anzeigen lassen.

**Auslöser**

Der Benutzer drückt auf die Schaltfläche Auto finden.

**Akteur(e)**

Gast Benutzer und registrierter Benutzer

**Vorbedingung(en)**

Der Benutzer muss ein gültiges Ticket besitzen.

**Schritte**

1. Der Benutzer hält ein gültiges Ticket an den QR-Code Scanner
2. Das System zeigt den Parkplatz des Autos an

**Alternative Schritte**

-

**Nachbedingung(en)**

-

**Inkludierte Use Cases**

Ticket scannen

**Involvierte Klassen**

Ticket, Reservation, ParkingSpot

## Use Case „Reservierung stornieren“

**Kurze Beschreibung**

Der Benutzer hat die Möglichkeit bis zu 12 Stunden vor dem Eintreten des Zeitraums der Reservierung die Reservierung zu stornieren.

**Auslöser**

Der Benutzer drückt auf die Schaltfläche Reservierung stornieren.

**Akteur(e)**

Gast Benutzer und registrierter Benutzer

**Vorbedingung(en)**

Benutzer hat vorher eine Reservierung getätigt die noch gültig ist und die Stornierung wird 12 Stunden vor des eintreten des Zeitraumes der Reservierung getätigt.

**Schritte**

1. Der Benutzer muss das Ticket (QR-Code) scannen, siehe Use Case „Ticket scannen“.
2. Ist das Ticket gültig wird der Benutzer aufgefordert die Stornierung zu bestätigen.
3. Hat der Benutzer dies bestätigt wurde das Ticket erfolgreich storniert.

**Alternative Schritte**

Wurde die Stornierung von einem registrierten Benutzer getätigt dann bekommt dieser eine E-Mail wenn die Stornierung erfolgreich war.

**Nachbedingung(en)**

Der Benutzer bekommt den vollen Betrag der Reservierung rückerstattet und das Ticket ist nicht mehr gültig.

**Inkludierte Use Cases**

Ticket scannen

**Involvierte Klassen**

User, Ticket, Reservation

## Use Case „Registrieren“

**Kurze Beschreibung**

Ein Gast kann ein Benutzerkonto erstellen.

**Auslöser**

Der Benutzer drückt auf die Schaltfläche Registrieren.

**Akteur(e)**

Gast Benutzer und registrierter Benutzer

**Vorbedingung(en)**

Der Benutzer hat noch kein Konto mit seiner E-Mail erstellt.

**Schritte**

1. Der Benutzer wird aufgefordert seinen vollen Namen, seine E-Mail, seine Kreditkarte und ein Passwort anzugeben. Das selbe Passwort muss er dabei ein zweites Mal eingeben und dieses muss mit den anderen übereinstimmen.
2. Sind die Daten gültig wird das Benutzerkonto erstellt und die Daten werden in der Datenbank gespeichert.
3. Sind die Daten nicht gültig kann der Benutzer den Vorgang wiederholen.

**Alternative Schritte**

-

**Nachbedingung(en)**

Der Benutzer kann nun mit seinem Benutzerkonto Reservierungen von überall mit der Web-Applikation und Smartphone-App tätigen.

**Inkludierte Use Cases**

Anmelden

**Involvierte Klassen**

User, Statistics, Valuation

## Use Case „Parkzeitüberschreitung bezahlen“

**Kurze Beschreibung**

Hat der Gast bzw. Benutzer die Parkzeit überschritten muss die länger geparkte Zeit gemäß der Preisliste direkt an der Ausfahrtsschranke bezahlt werden.

**Auslöser**

Der Benutzer hat die Parkzeit der Reservierung überschritten.

**Akteur(e)**

Gast Benutzer und registrierter Benutzer

**Vorbedingung(en)**

Der Benutzer will das Parkhaus verlassen.

**Schritte**

1. Der Benutzer wird aufgefordert das Ticket zu scannen (siehe Use Case „Ticket scannen“).
2. Hat der Benutzer das Ticket gescannt meldet das System dem Benutzer, dass er die Parkzeit überschritten hat.
3. Der Benutzer wird aufgefordert den Betrag zu bezahlen.

**Alternative Schritte**

Dem registrierten Benutzer wird zusätzlich die Option gegeben die Bezahlung automatisch über seine Kreditkarte zu tätigen.

**Nachbedingung(en)**

Wurde der Betrag bezahlt wird direkt an der Ausfahrtsschranke eine Quittung ausgedruckt und die Schranke öffnet sich. Hat der Benutzer die automatische Bezahlung über Kreditkarte getätigt wird ihn zusätzlich eine E-Mail mit Rechnung geschickt. Der Status vom Parkplatz ist nun auf frei.

**Inkludierte Use Cases**

Ticket scannen

**Involvierte Klassen**

User, Ticket, Reservation, ParkingSpot

## Use Case „Anmelden“

**Kurze Beschreibung**

Der Benutzer meldet sich beim Java-Client, Web-Applikation oder Smartphone-App an.

**Auslöser**

Der Benutzer drückt auf die Schaltfläche Anmelden.

**Akteur(e)**

Registrierter Benutzer

**Vorbedingung(en)**

Der Benutzer hat ein Benutzerkonto.

**Schritte**

1. Der Benutzer wird vom System aufgefordert seine E-Mail und sein Passwort einzugeben.
2. Der Benutzer gibt diese ein und das System überprüft mithilfe der Datenbank ob die Daten übereinstimmen.
3. Sind die Daten korrekt war die Anmeldung erfolgreich.

**Alternative Schritte**

-

**Nachbedingung(en)**

Der Benutzer kann von überall mithilfe der Web-Applikation oder Smartphone-App eine Reservierung betätigen.

**Inkludierte Use Cases**

-

**Involvierte Klassen**

User

## Use Case „Abmelden“

**Kurze Beschreibung**

Der Benutzer meldet sich beim Java-Client, Web-Applikation oder Smartphone-App ab.

**Auslöser**

Der Benutzer drückt auf die Schaltfläche Abmelden.

**Akteur(e)**

Registrierter Benutzer

**Vorbedingung(en)**

Der Benutzer hat sich beim Java-Client, Web-Applikation oder Smartphone-App angemeldet.

**Schritte**

1. Das System meldet den Benutzer ab.
2. War die Abmeldung erfolgreich wird der Benutzer darauf hingewiesen.

**Alternative Schritte**

Beim Java-Client wird der Benutzer nach einer Reservierung bzw. Stornierung automatisch abgemeldet. Bei der Web-Applikation geschieht das ebenfalls, jedoch ist hier der unterschied, dass der Benutzer nur nach einer gewissen Zeit abgemeldet wird, wenn er auf der Website inaktiv ist. Der Benutzer kann sich auf der Web-Applikation selbstverständlich auch selber abmelden.

**Nachbedingung(en)**

Der Benutzer ist abgemeldet und kann keine Reservierungen mehr von der Web-Applikation bzw. Smartphone-App tätigen.

**Inkludierte Use Cases**

-

**Involvierte Klassen**

User

## Use Case „Bewerten“

**Kurze Beschreibung**

Der Benutzer hat die Möglichkeit den Parkservice zu bewerten.

**Auslöser**

Der Benutzer drückt auf die Schaltfläche Bewerten oder er wird vom System gefragt ob er eine Bewertung schreiben will.

**Akteur(e)**

Gast Benutzer und registrierter Benutzer

**Vorbedingung(en)**

Der Benutzer hat den Parkservice genutzt.

**Schritte**

1. Der Benutzer klickt auf eine der fünf Smileys (Von nicht zufrieden bis zufrieden).
2. Der Benutzer kann optional eine Nachricht hinterlassen.
3. Der Benutzer bestätigt die Bewertung.

**Alternative Schritte**

Der Benutzer hat die Möglichkeit die Bewertung abzubrechen bzw. zu überspringen.

**Nachbedingung(en)**

-

**Inkludierte Use Cases**

-

**Involvierte Klassen**

User, Valuation, Statistics

## Use Case „Statistiken anzeigen“

**Kurze Beschreibung**

Der Administrator hat die Möglichkeit Statistiken anzuschauen, z.B. die Bewertungsstatistik

**Auslöser**

Der Administrator drückt auf die Schalfläche Statistik.

**Akteur(e)**

Administrator

**Vorbedingung(en)**

Der Benutzer muss mit seinem Administratorkonto angemeldet sein.

**Schritte**

1. Der Benutzer wählt die Statistik aus.
2. Das System zeigt die Statistik an.

**Alternative Schritte**

-

**Nachbedingung(en)**

-

**Inkludierte Use Cases**

-

**Involvierte Klassen**

User, Valuation, Statistics

## Use Case „Statistiken exportieren“

**Kurze Beschreibung**

Der Administrator hat die Möglichkeit Statistiken zu exportieren, z.B. die Bewertungsstatistik.

**Auslöser**

Der Administrator drückt auf die Schalfläche Statistik exportieren.

**Akteur(e)**

Administrator

**Vorbedingung(en)**

Der Benutzer muss mit seinem Administratorkonto angemeldet sein und hat bereits eine Statistik ausgewählt.

**Schritte**

1. Der Benutzer drückt auf Statistik exportieren.
2. Der Benutzer speichert sie irgendwo ab.

**Alternative Schritte**

-

**Nachbedingung(en)**

-

**Inkludierte Use Cases**

Statistiken anzeigen

**Involvierte Klassen**

User, Valuation, Statistics

## Use Case „Bewertung löschen“

**Kurze Beschreibung**

Der Administrator hat die Möglichkeit anstößige Bewertungen zu löschen.

**Auslöser**

Der Benutzer wählt eine Bewertung aus.

**Akteur(e)**

Administrator

**Vorbedingung(en)**

Der Benutzer muss mit seinem Administratorkonto angemeldet sein.

**Schritte**

1. Der Benutzer drückt auf die Schalfläche Bewertung löschen.
2. Das System fordert den Benutzer auf die Löschung zu bestätigen.
3. Hat der Benutzer die Bewertung gelöscht protokolliert das System dies.

**Alternative Schritte**

Der Benutzer hat die Möglichkeit mehrere Bewertungen mithilfe eines Elementkontrollkästchen auszuwählen und diese dann alle in einmal zu Löschen.

**Nachbedingung(en)**

Die Bewertung kann wiederhergestellt werden.

**Inkludierte Use Cases**

-

**Involvierte Klassen**

User, Valuation, Statistics

# Umsetzung

## Plattform

Das System besteht aus einem Java-Client vor Ort, der auf einem Linux-Computer läuft, einer Smartphone-App für Android und iOS und einer Website (Web-Applikation). Die App und die Website sind für die breite Masse erreichbar.

## Datenhaltung

Alle Daten werden in einer zentralen Datenbank gehalten. Einmal im Monat wird von der Datenbank mit Magnetbänder eine Datensicherung erstellt.

## Oberfläche

Die Oberfläche der Website, Java-Applikation und Smartphone-App werden aufgrund der Zielgruppe sehr minimalistisch und Benutzerfreundlich gestaltet, damit sie jedermann verwenden kann.

## Systemarchitektur

Alle Dienste laufen auf einen zentralen Server, der reichlich an Leistung und Speicher bereitstellt. Einer der Hauptaugenmerkmale ist Big Data, darum brauchen wir eine sehr schnelle Datenbank.

# Oberflächenprototyp

