

Move it

Pflichtenheft

<b>Move it</b>	<b>0</b>
<b>Pflichtenheft</b>	<b>0</b>
Meilensteine	2
12.6 & 19.5	2
26.5 & 2.6	2
9.6 & 16.6	3
23.6 & 30.6	3
Fortlaufend	3
1. Zielbestimmung	4
1.1 Musskriterien	4
1.2 Wunschkriterien	5
2 Produkteinsatz	6
2.1 Anwendungsbereiche	6
2.2 Zielgruppen	6
2.3 Betriebsbedingungen	6
3 UserStories	7
4 Produktübersicht	8
4.1 Architekturplan	8
4.2 Use Case Diagramme	9
4 Produktfunktionen	11
4.1 Use Case	11
5 Produktdaten	11
6 Mockups	12
7 Datenmodell	20
8 API	21

# Meilensteine

12.6 & 19.5

Aufgabe	Zuständig
Arbeitsumgebung schaffen / einrichten zum gescheiten Arbeiten	Alle
Datenbank <ul style="list-style-type: none"><li>- Datenbank erstellen</li><li>- Tabellen erstellen</li><li>- Testdaten einfügen</li><li>- Zugriff gewähren für REST</li></ul>	Nils, Maxi
REST-Schnittstelle <ul style="list-style-type: none"><li>- REST via express realisiert</li><li>- Zugriff auf die Datenbank</li><li>- Zugriff gewähren für Frontend</li></ul>	Tobias, Eugen
Pages mit Navigation, Routing, GUI	Sasha
Login-Register GUI	Sasha

26.5 & 2.6

Aufgabe	Zuständig
Map API / Funktion in ionic einbinden ( 1 / 2 )	
Event erstellen <ul style="list-style-type: none"><li>- Informationen eingeben</li><li>- Standort abgreifen</li><li>- In Datenbank eintragen</li></ul>	
Event teilnehmen <ul style="list-style-type: none"><li>- Event auf Karte einsehbar<ul style="list-style-type: none"><li>- Anklicken: Popup mit teilnehmen Aktion einsehbar</li></ul></li><li>- In Datenbank als teilgenommen eintragen</li></ul>	
Login-Register Logic	

## 9.6 & 16.6

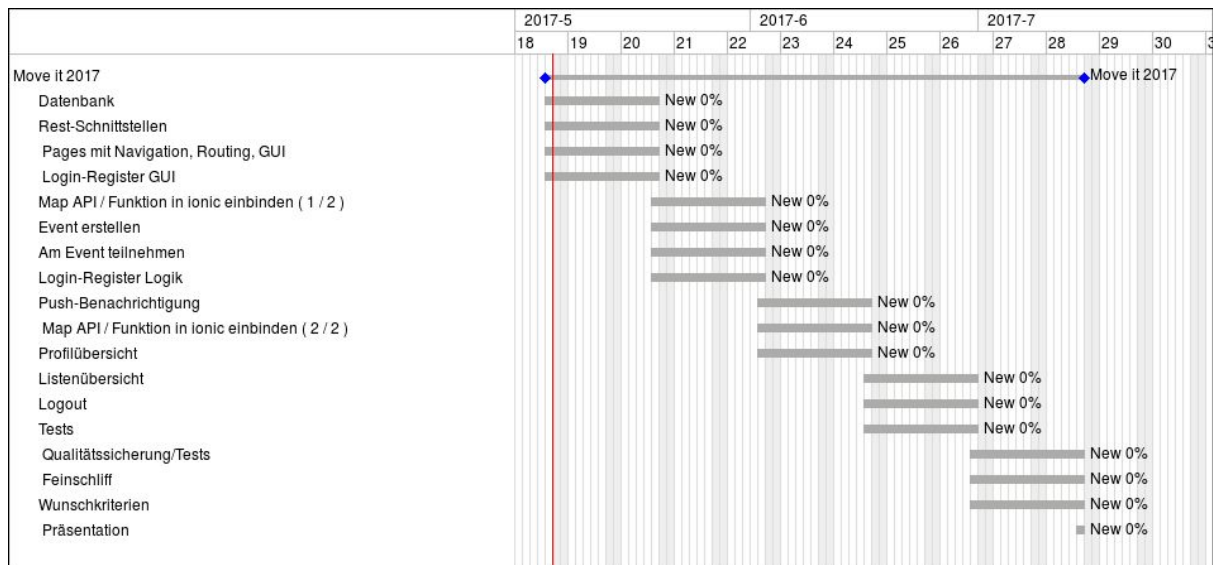
Aufgabe	Zuständig
Push-Benachrichtigung - Wenn jemand an meinen Event teilnimmt ("User X nimmt an deinem Event Y teil!")	
Map API / Funktion in ionic einbinden ( 2 / 2 )	
Profilübersicht	

## 23.6 & 30.6

Aufgabe	Zuständig
Listenübersicht	
Logout	
Tests anfangen	

## Fortlaufend

Aufgabe	Zuständig
Qualitätssicherung durch automatische Tests	
Feinschliff	
Wunschkriterien	
Präsentation	



# 1. Zielbestimmung

## 1.1 Musskriterien

Funktional:

- Anwender kann ein Event erstellen, um Teilnehmer zu finden
- Anwender kann Events auf einer Karte oder Liste einsehen, damit er teilnehmen kann
- Anwender kann ein Profil erstellen, damit er die App benutzen kann
- Anwender möchte sich einloggen können, um auf die Funktionen zuzugreifen
- Anwender möchte eine Push-Benachrichtigung sobald jemand teilnimmt
- Sobald der Anwender ein Event erstellt hat, ist dieses auf dem Event erstellen Reiter wiederzufinden
- Anwender möchte sich ausloggen können

Nicht Funktional:

- Performanz: vergleichbar mit gängigen Web-Applikationen
- Zuverlässigkeit: Anwendung muss zuverlässig funktionieren
- Benutzerfreundlichkeit: Bedienung muss benutzerfreundlich und verständlich sein
- Weiterentwickelbarkeit: Software muss änderungs freundlich sein, sodass sie leicht weiterentwickelt werden kann

## 1.2 Wunschkriterien

- Anzahl der Teilnehmer der Events einsehen
- Vom Event abmelden
- Minimale/maximale Anzahl an Teilnehmer
- Permanentes Event
- ((Beschwerde Button))

## **2 Produkteinsatz**

### **2.1 Anwendungsbereiche**

Öffentlicher Anwendungsbereich. Um weitere Teilnehmer für den sportlichen Alltag zu finden.

### **2.2 Zielgruppen**

Männer und Frauen. Sportler jeden Alters. Ab 18.

### **2.3 Betriebsbedingungen**

Hybrid App. Jederzeit über Internet verfügbar.

### 3 UserStories

Ich als Benutzer möchte mich registrieren, damit ich mich einloggen kann.

Akzeptanzkriterien:

- Account erstellt

Ich als Benutzer möchte mich einloggen können, damit ich auf die Funktionen der App zugreifen kann.

Akzeptanzkriterien:

- Eingeloggt sein
- Weitergeleitet werden

Ich als Benutzer möchte ein Event erstellen, damit ich weitere Teilnehmer finde.

Akzeptanzkriterien:

- Event erstellt
- Event in der Nähe sichtbar

Ich als Benutzer möchte auf einer Karte Events in der Nähe einsehen können, damit ich meine Teilnahme besser einschätzen kann.

Akzeptanzkriterien:

- Events in der Umgebung mit Namen anzeigen

Ich als Benutzer möchte auf ein Event klicken, damit ich mehr Informationen erhalte oder teilnehmen will

Akzeptanzkriterien:

- Popup erscheint mit
  - Button zum teilnehmen
  - Bild und Username des Erstellers

Ich als Benutzer möchte bei einem Event zusagen, um teilzunehmen.

Akzeptanzkriterien:

- Ersteller erhält Push-Benachrichtigung
- Wird als Teilnehmer im Event Datenbank eingetragen

Ich als Benutzer möchte mich ausloggen können, sodass niemand anderes an meinem Handy die App in meinem Namen benutzen kann.

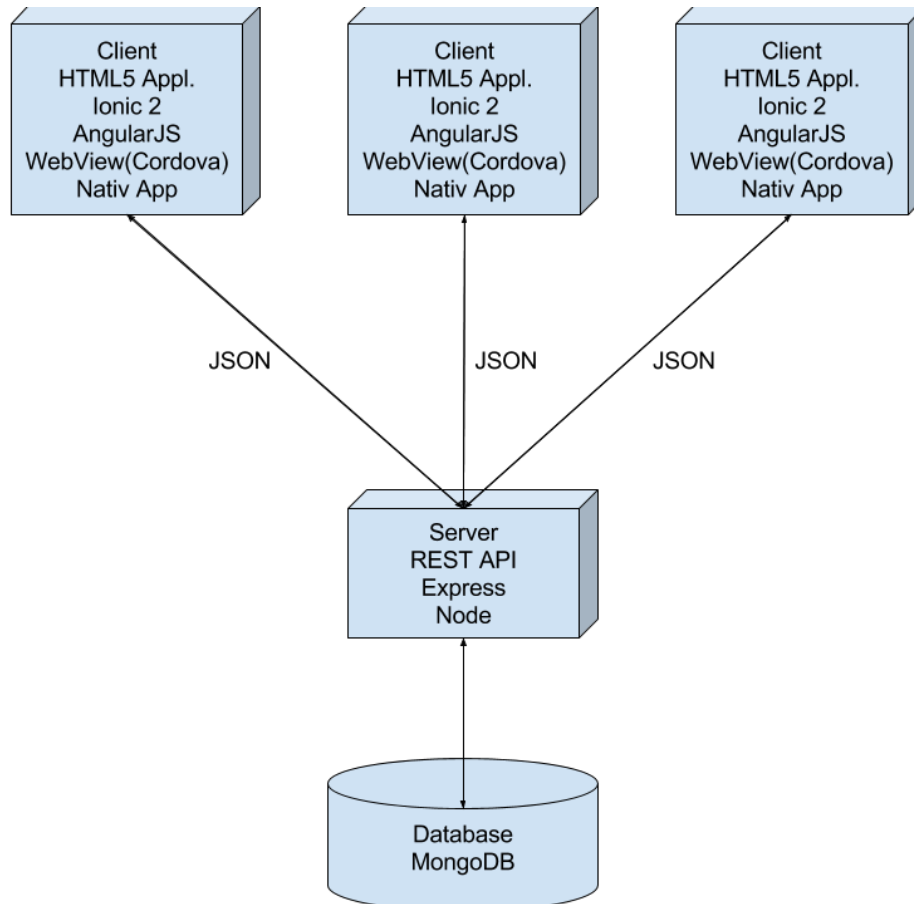
Akzeptanzkriterien:

- Ausgeloggt sein
- App startet beim Login Bildschirm



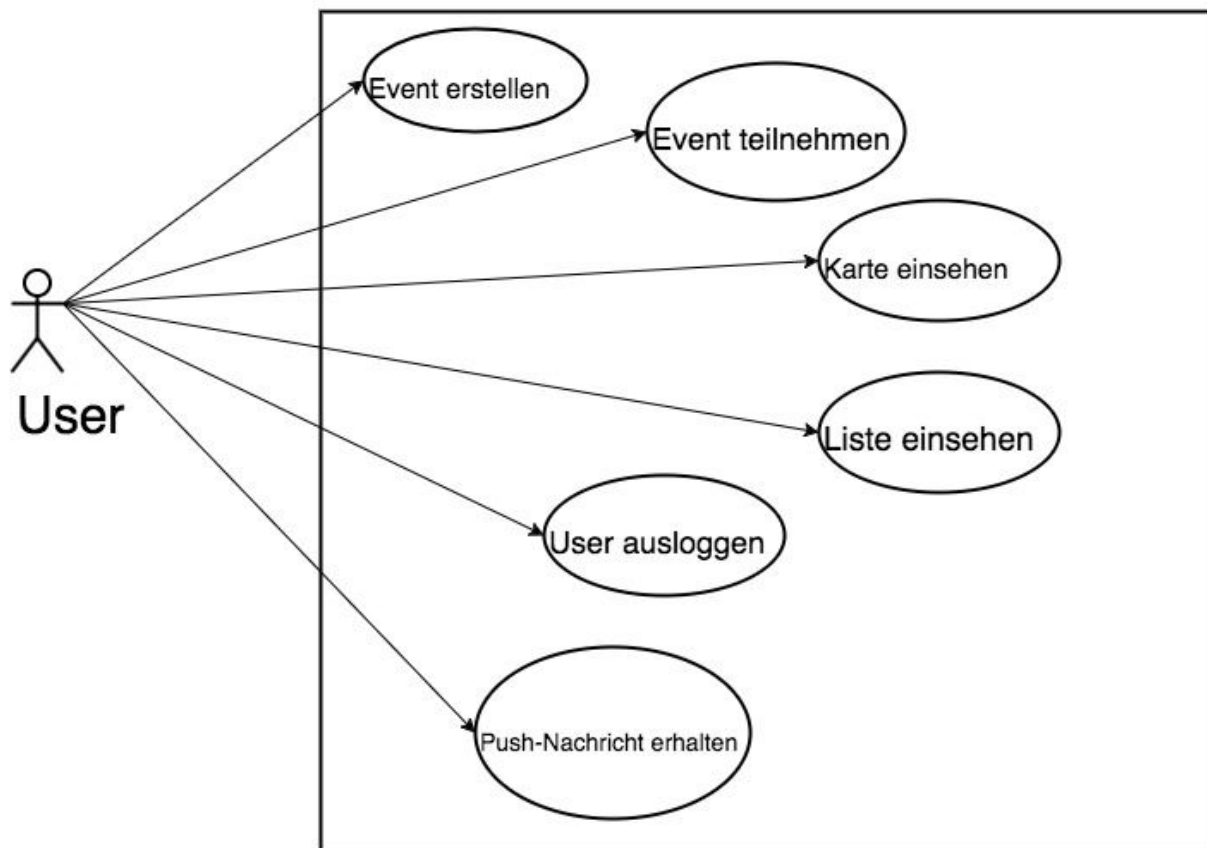
## 4 Produktübersicht

### 4.1 Architekturplan

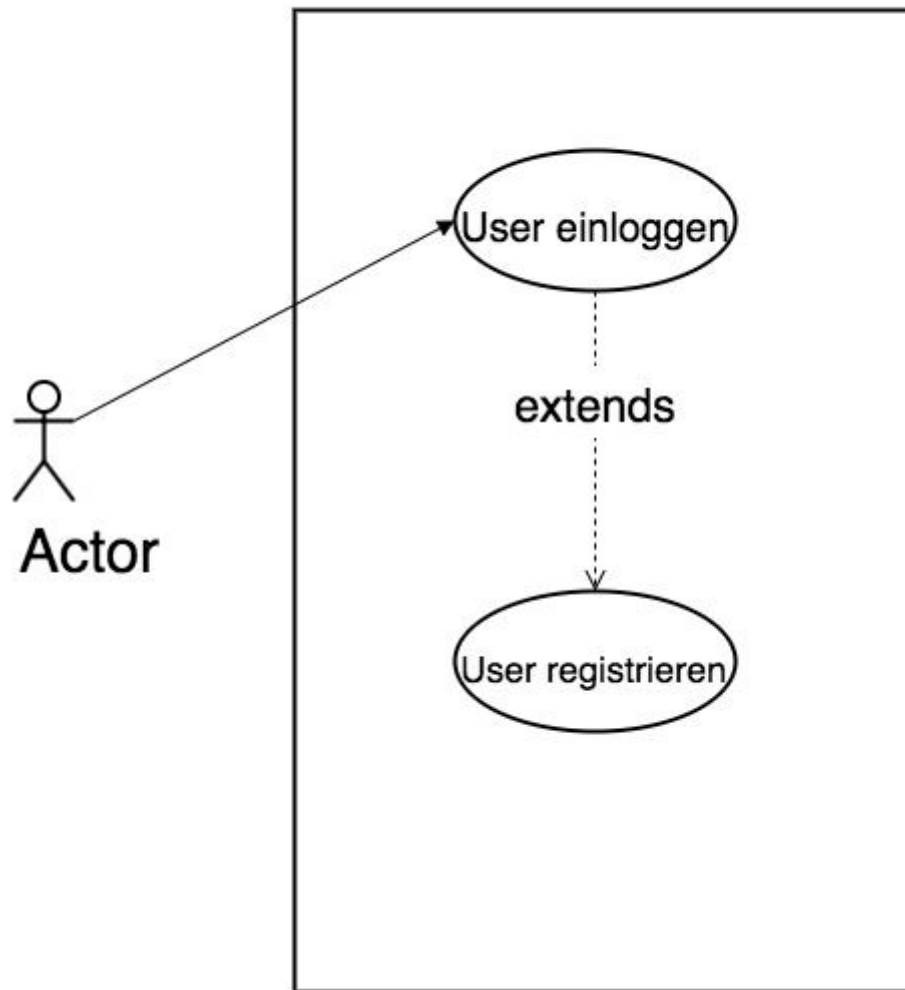


## 4.2 Use Case Diagramme

### Move It App



## Move It Login



## **4 Produktfunktionen**

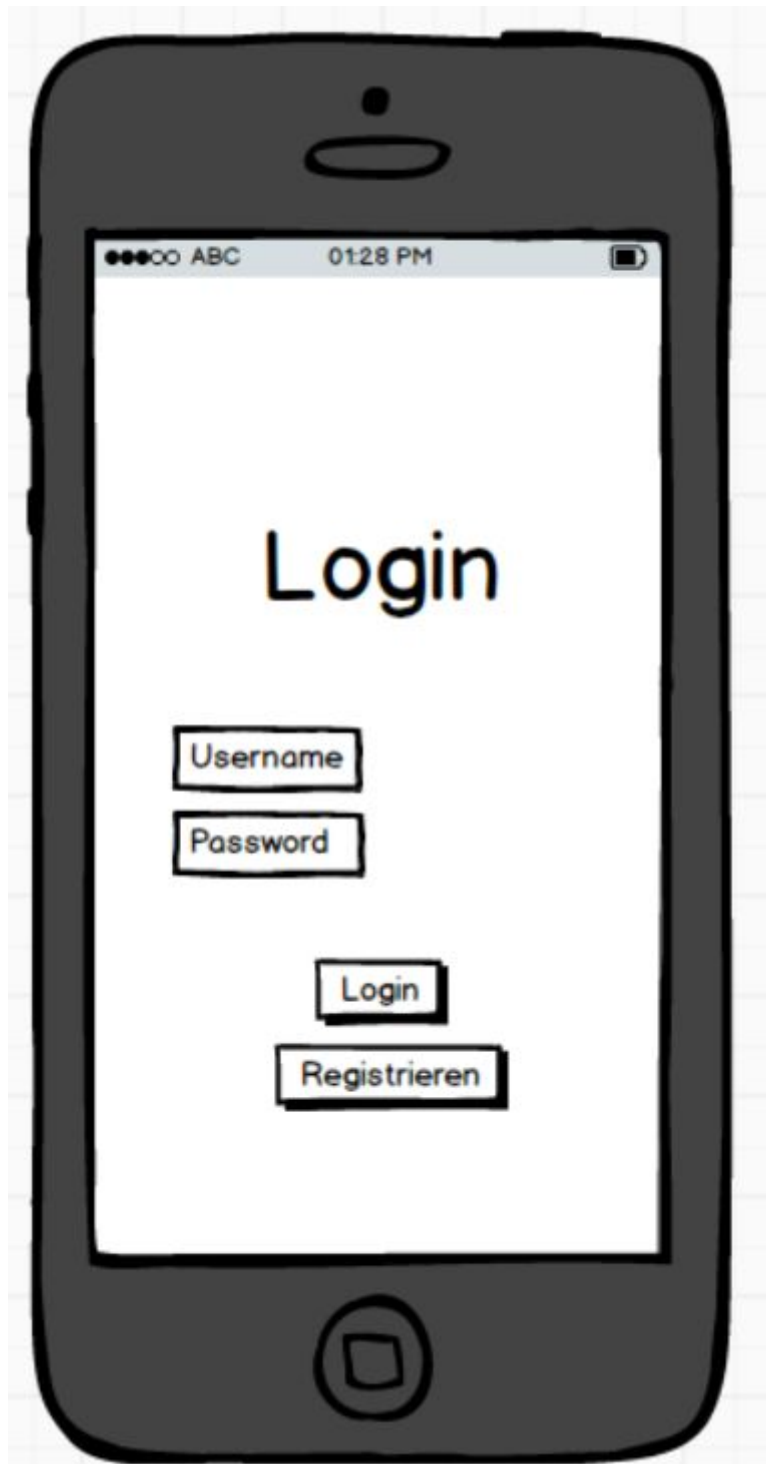
### 4.1 Use Case

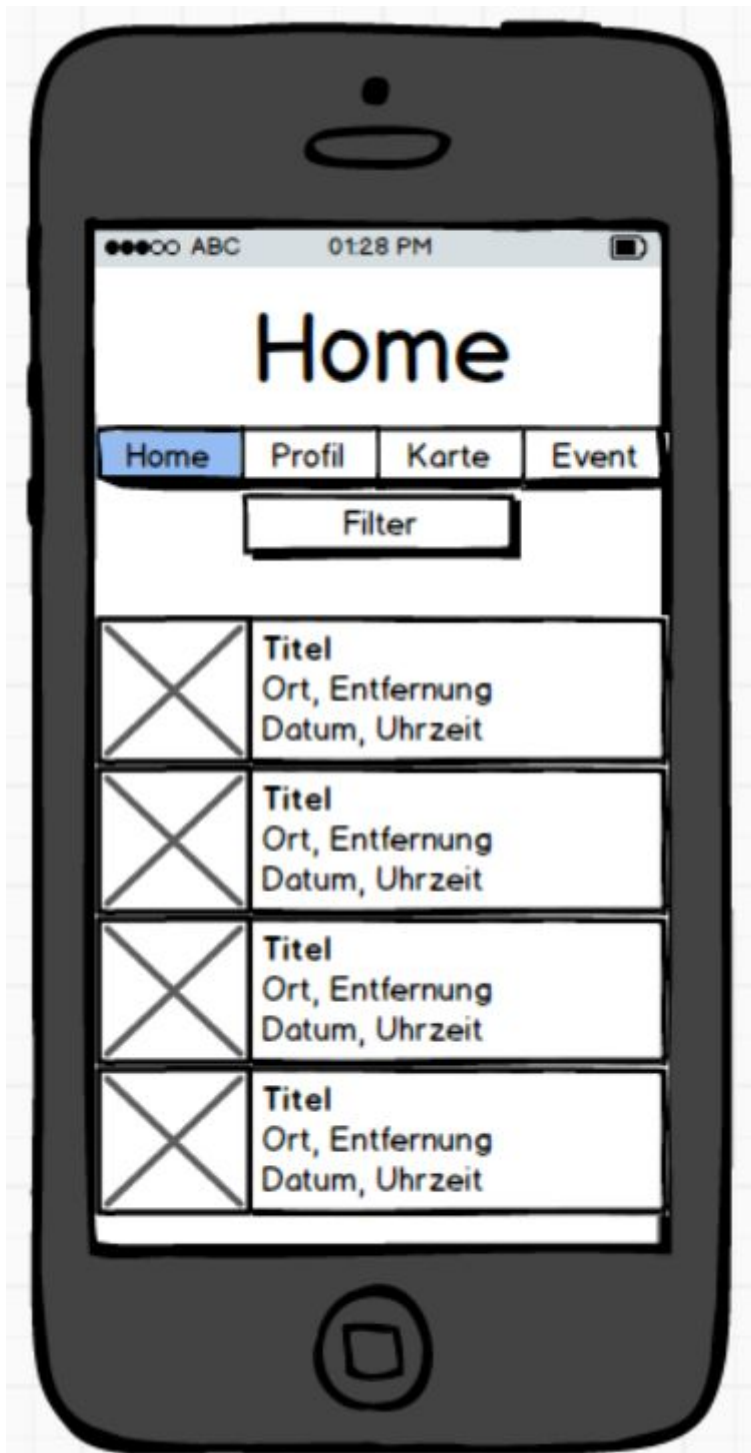
## **5 Produktdaten**

- Event (GPS, Titel)
- Profil (Username, E-Mail, Passwort, Geburtsdatum, Geschlecht)

## 6 Mockups



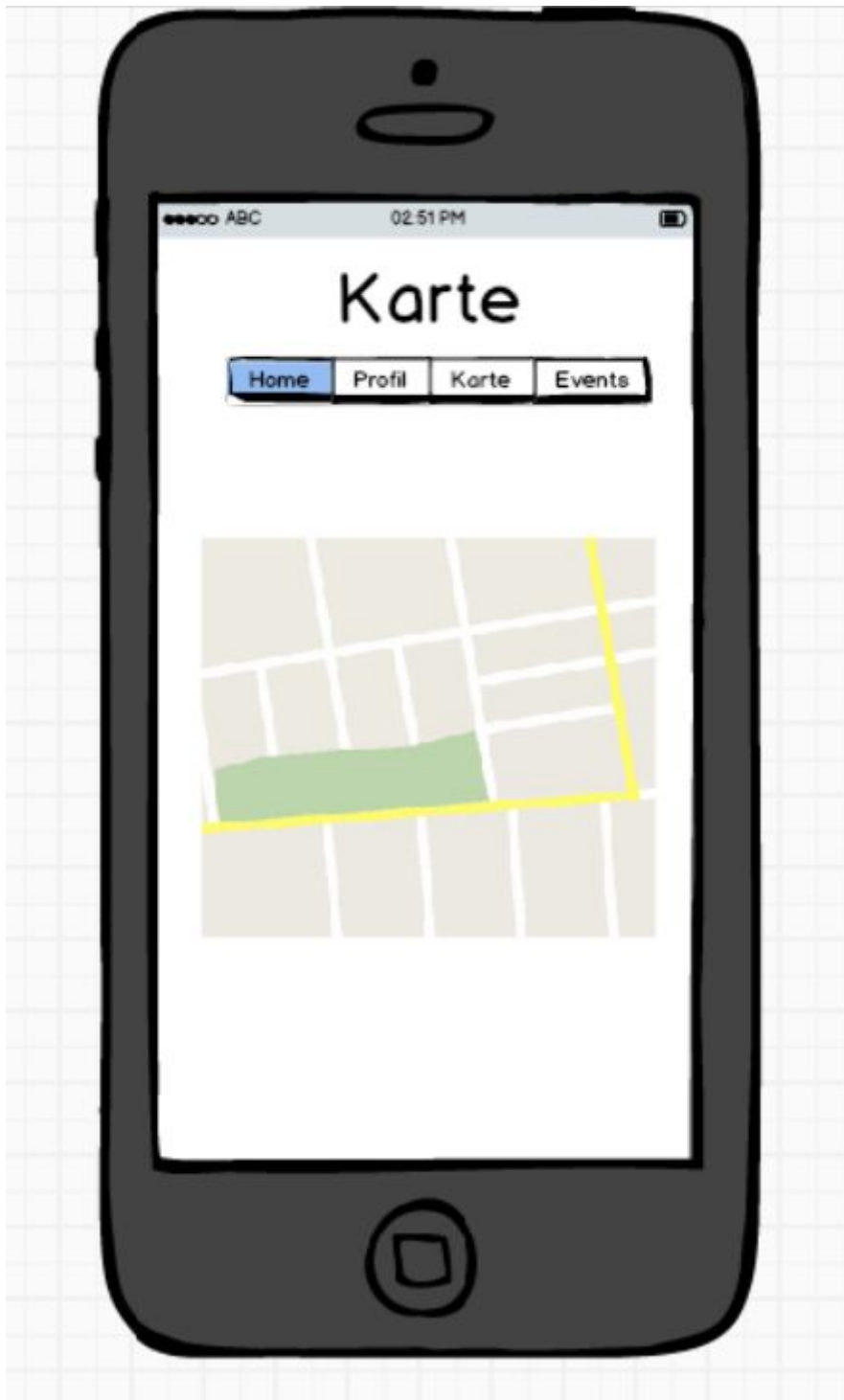


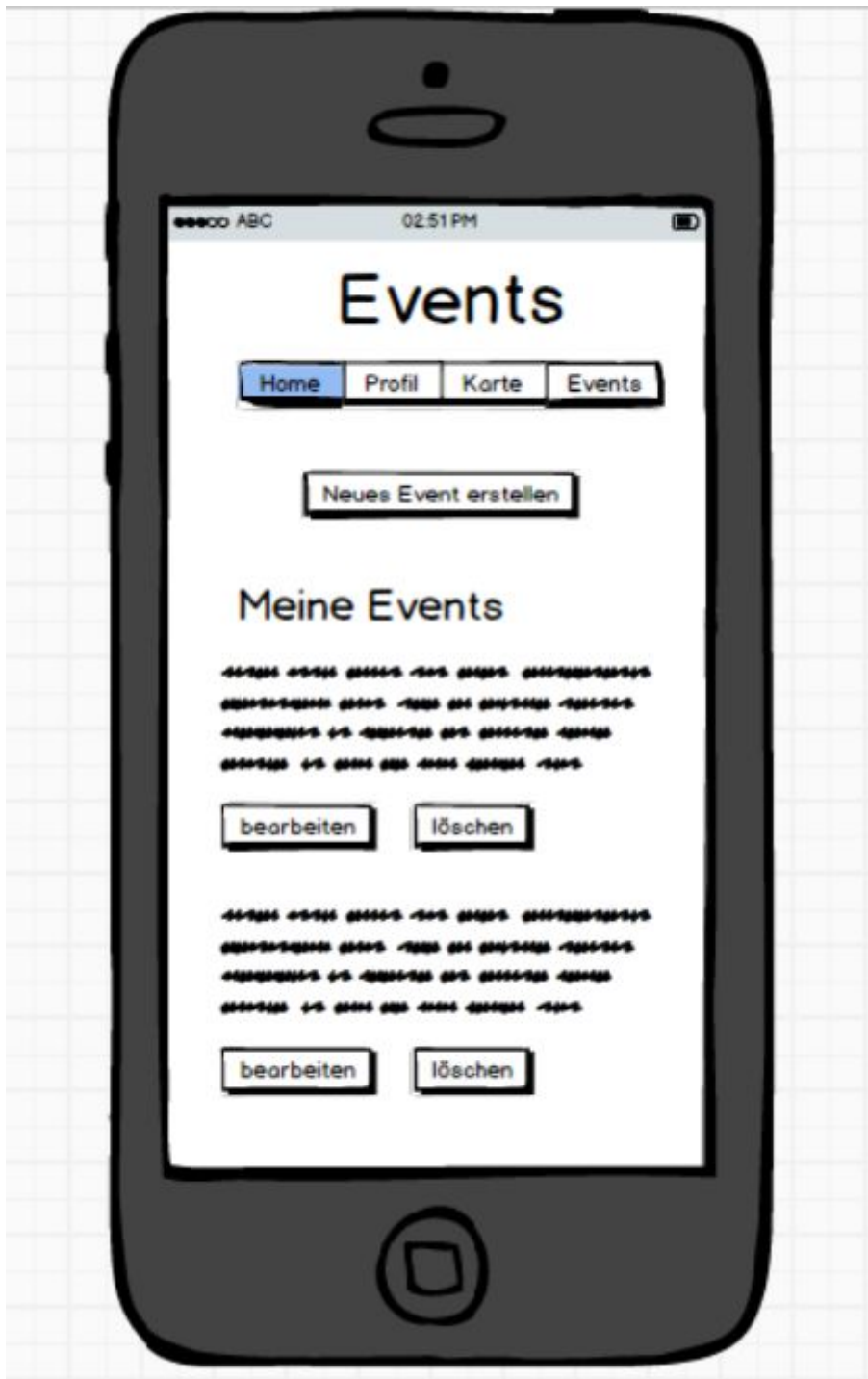














## 7 Datenmodell

### User

createdAt:	Date,
firstName:	String,
lastName:	String,
email:	String,
birthdate:	String,
sex:	String,
picture:	String,
username:	String,
password:	String,
pushToken:	String

Der username ist eindeutig. Das password steht zunächst unverschlüsselt in der Datenbank.  
Das Passwort steht jetzt gehasht in der Datenbank.

### Event

createdAt:	Date,
creator:	{ type: String, ref: 'User' },
title:	String,
keywords:	[String],
longitude:	Number,
latitude:	Number,
starttimepoint:	Date,
subscriber:	[{ type: String, ref: 'User' }]

## 8 Sites / API

/ - Einfache Willkommensseite, das Backend läuft (GET)

/signup - Einfaches Formular zum Registrieren (GET und POST)

{"message":"Missing credentials"} - Wenn Benutzername und/oder Passwort fehlen

{"message":"User Already Exists"} - Wenn Benutzername doppelt

{"message":"User Registration succesful"} - Wenn erfolgreich, dann auch angemeldet

/login - Einfaches Formular zum Anmelden (GET und POST)

{"message":"Missing credentials"} - Wenn Benutzername und/oder Passwort fehlen

{"message":"User Not found"} - Wenn Benutzername nicht vorhanden

{"message":"Invalid Password"} - Wenn Passwort falsch

{"message":"User Login succesful"} - Wenn erfolgreich, dann auch angemeldet

/home - Einfache home-Seite

/newEvent - Einfaches Formular für ein neues Event (GET und POST)

/myEvents - Alle Events, die der user erstellt hat (GET und JSON)

/myEventsSubscriber - Alle Events, an denen der user teilnimmt (GET und JSON)

/allEventsCircle?lon=<number>&lat=<number>&dis=<number> - Alle Events im Umkreis (GET und JSON)

lon: Longitude aktuelle Position

lat: Latitude aktuelle Position

dis: Distanz zur aktuellen Position (Umkreis)

/allUsers - Ein Array mit allen Benutzernamen (GET und JSON)

/user - Informationen über den eingeloggten Benutzer

/userEmail - Zum Ändern der E-Mailadresse (POST und JSON)

Parameter: email1 und email2 (Bestätigung)

/userPassword - Zum Ändern des Passworts (POST und JSON)

Parameter: password1 und password2 (Bestätigung)