

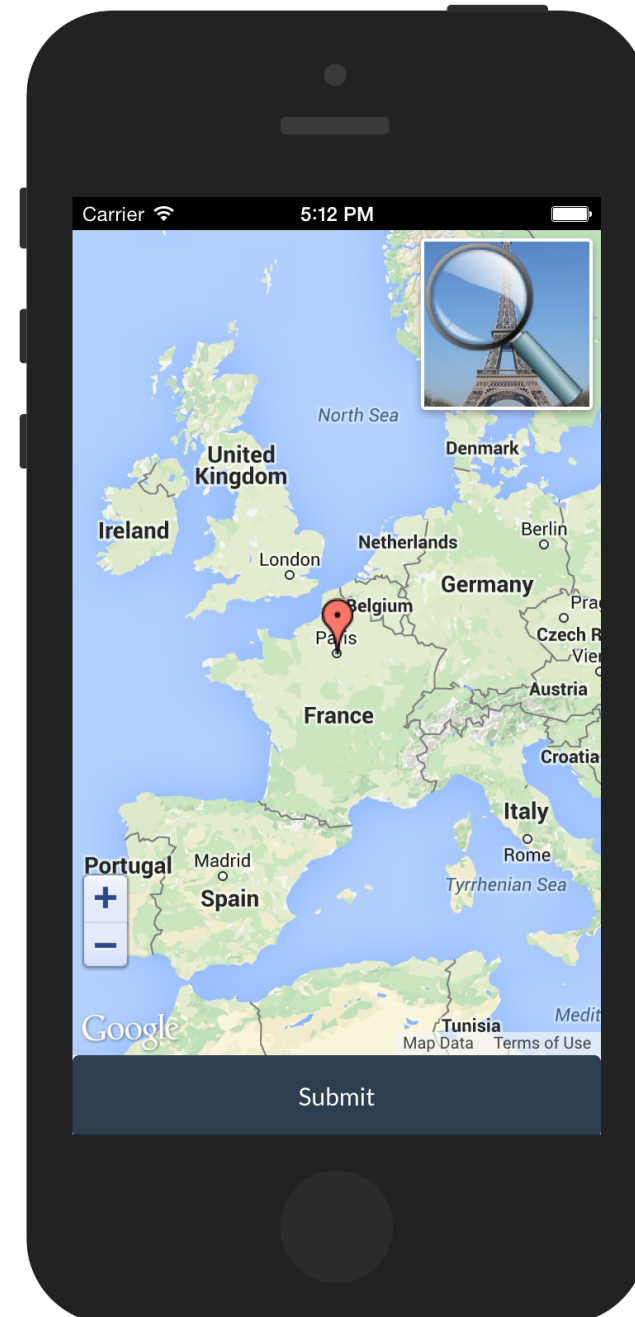
# Where Am I



Bernhard Fritz  
Thibault Gerrier  
Philipp Häusle  
Mario Zelger

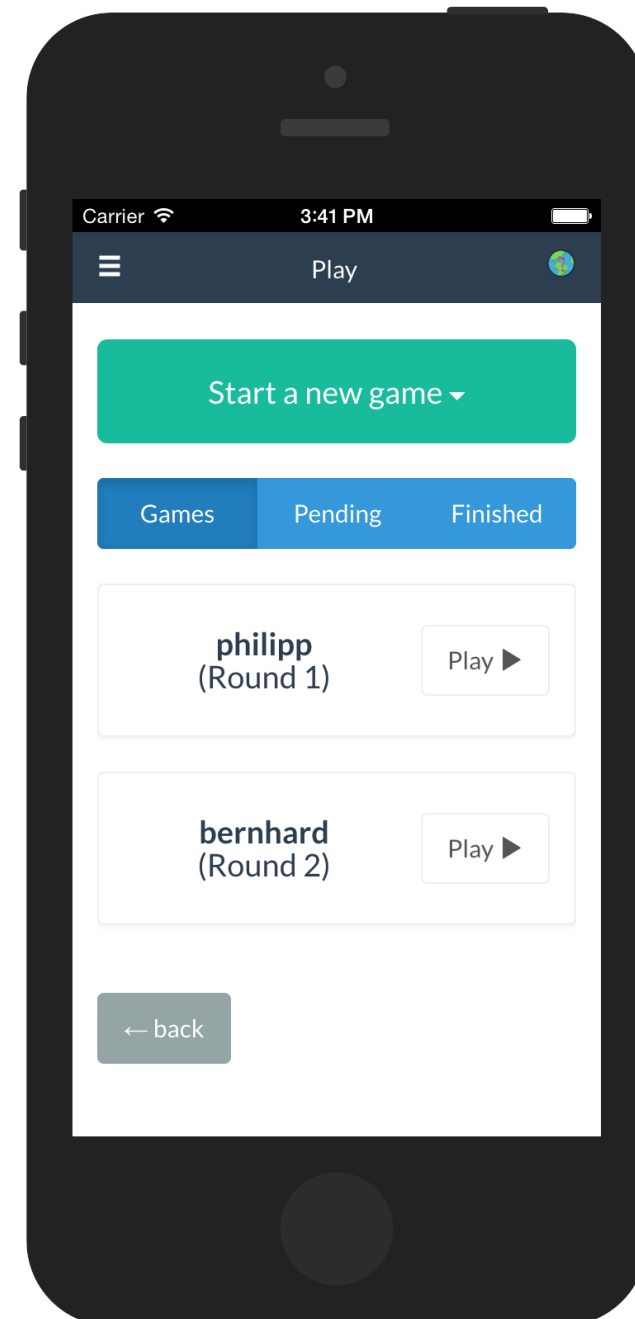
# Was ist WAI?

- WAI ist ein Geographiebasiertes Spiel
- Ziel ist es, anhand eines Bildes herauszufinden, wo dieses aufgenommen wurde



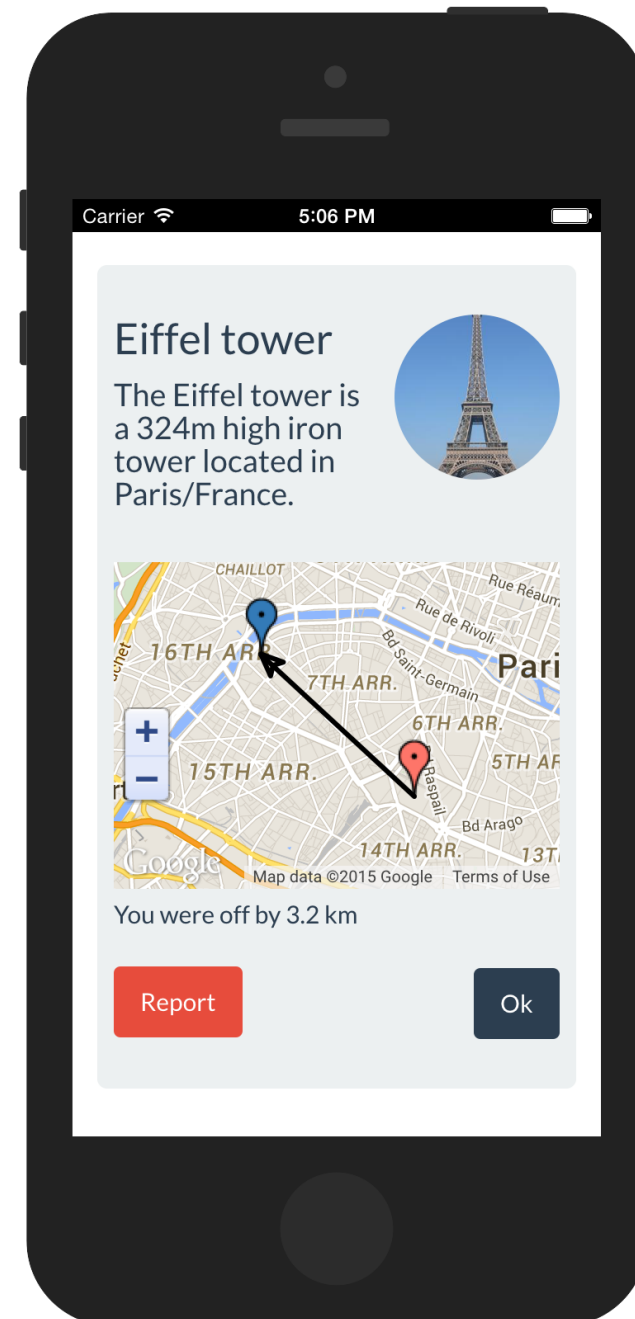
# Mehrspieler Modus

- Alleine spielen macht nur bedingt Spaß
- Seine Freunde herauszufordern macht das Ganze schon interessanter



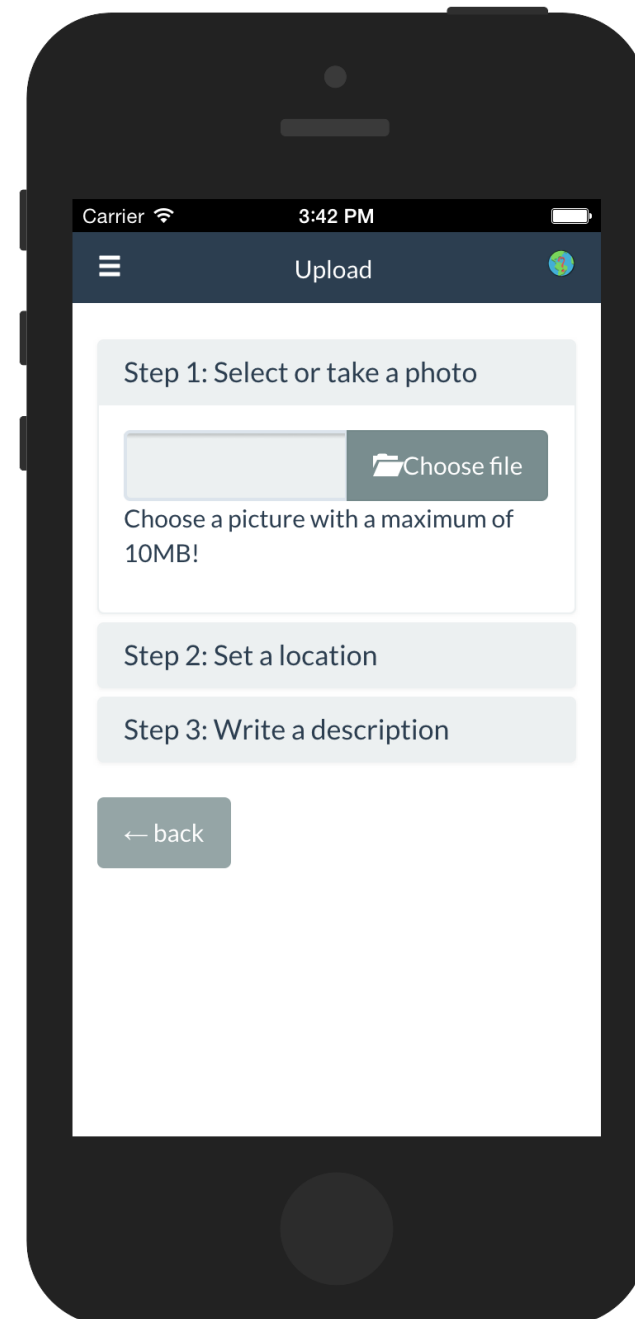
# Lerneffekt

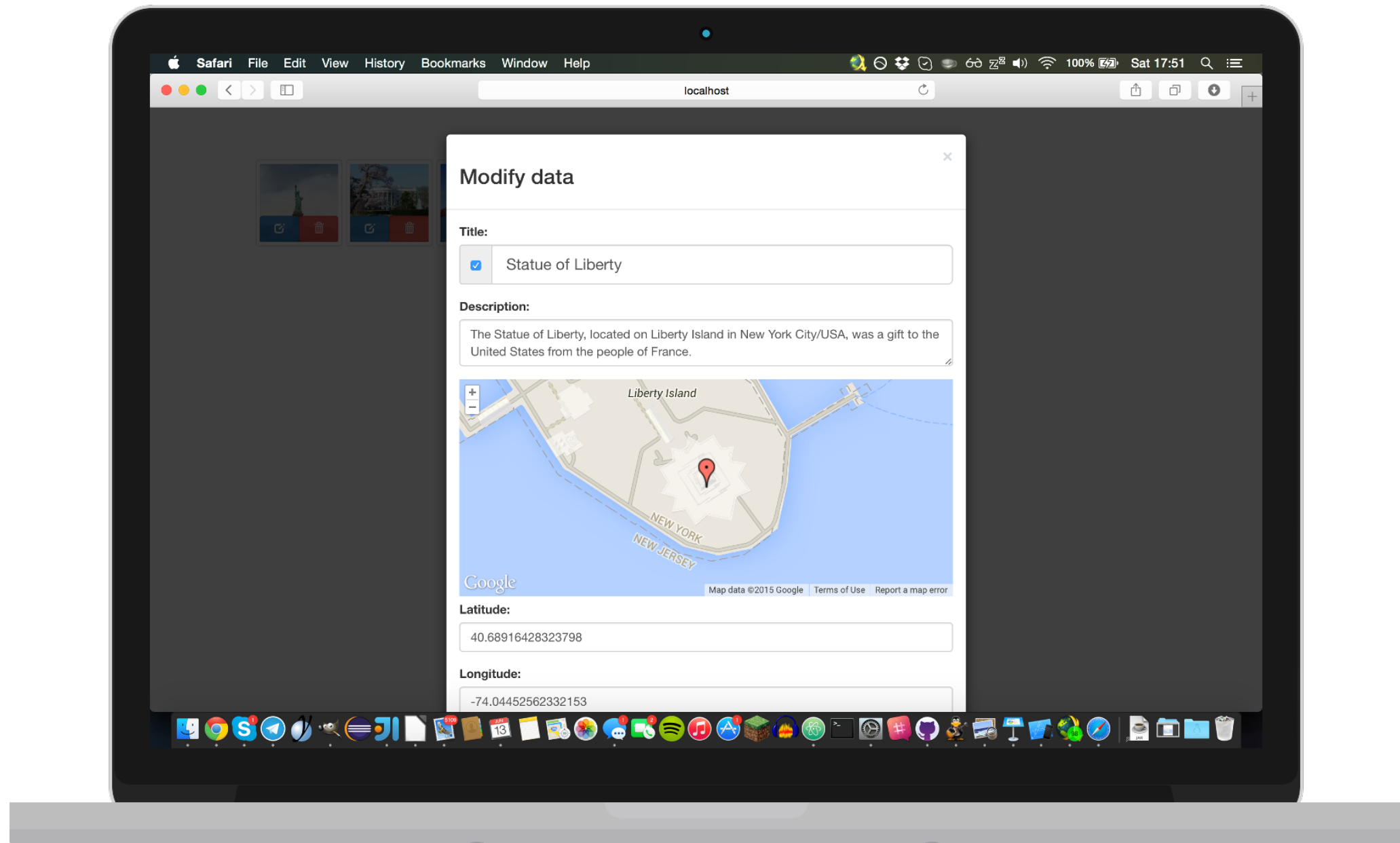
- WAI spielen macht nicht nur Spaß, sondern man kann dabei auch noch etwas dazulernen
- Nach jedem Spiel werden interessante Fakten aufgezählt



# Upload

- Registrierten Usern wird es ermöglicht eigene Bilder hochzuladen
- Nachdem ein Admin das Bild freigeschaltet hat, wird es in die Zufallsauswahl aufgenommen

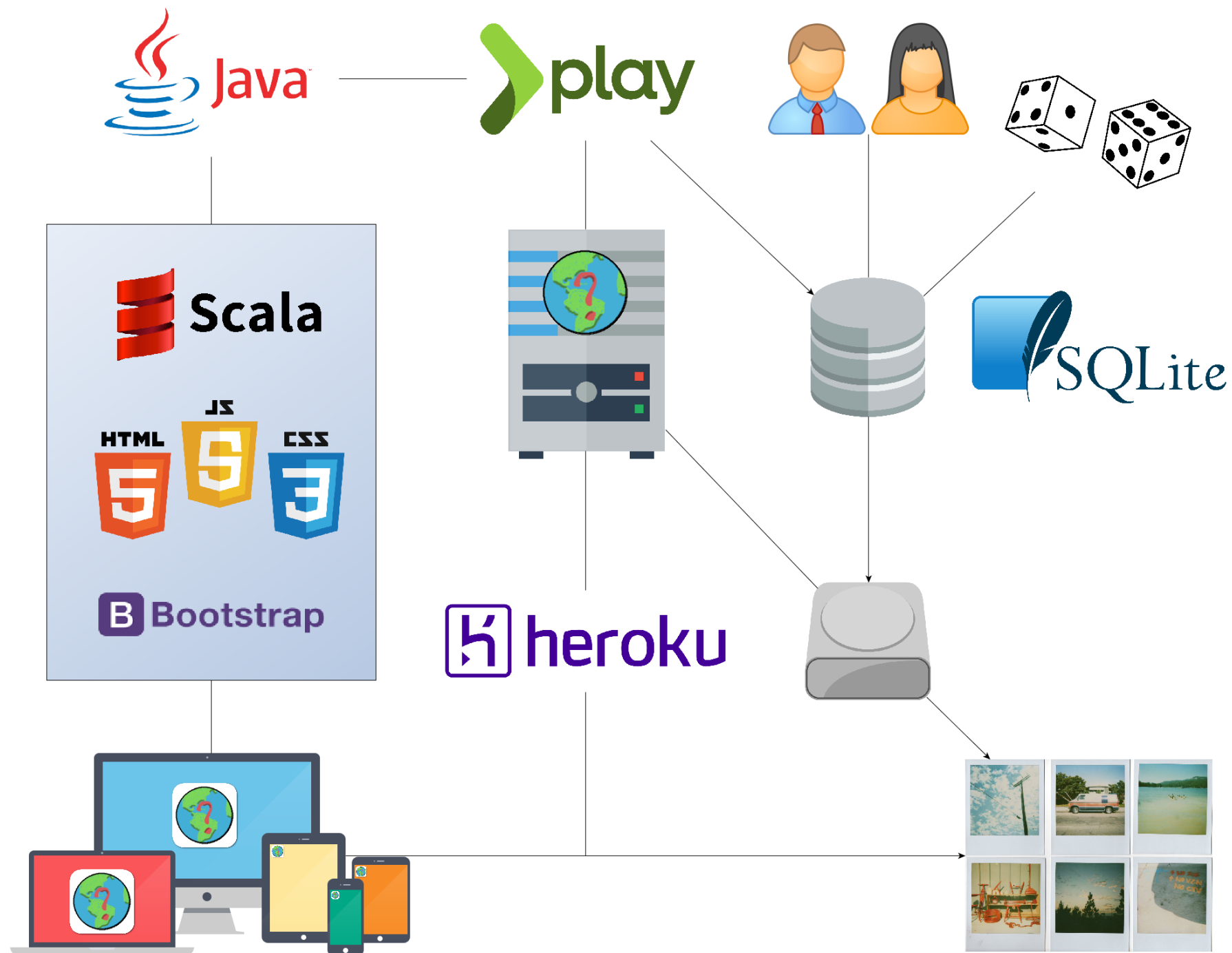




# Admin Modus

Als Admin ist es möglich hochgeladene Bilder freizuschalten und deren Eigenschaften zu verändern.

# Softwarearchitektur



Demo



# Readme Driven Development

- Readme als erstes Dokument des Projektes
- Alternative zu Document Driven Development?
- Mittelweg für Projektspezifikationen
- Gemeinsames Dokument für das gesamte Team
- Schriftliche Basis für Diskussionen

## Bernhard

- ☒ Admin Oberfläche
  - ☒ Erstellen einer Admin Oberfläche die nur durch den User admin aufgerufen werden kann. In dieser Oberfläche soll es für einen Admin ohne großen Aufwand möglich sein
  - upgeloadete Bilder zu verifizieren und freizuschalten.
  - ☒ Der Admin soll visuellen Zugriff auf die gesamte Datenbank haben
  - ☒ Der Admin soll die Möglichkeit haben böse User zu bannen.
  - ☒ Der Admin soll auch auf offene Reports zugreifen und reagieren können.
  - Einbauen eines Benachrichtigungssystems, das es ermöglicht dem Spieler mitzuteilen, dass ein anderer Spieler ihn zu einem Spiel herausgefordert hat, oder dass ein Spiel abgeschlossen wurde oder eine neue Runde beginnt:
- ☒ Nach einer abgeschlossenen Runde soll ersichtlich sein wo der Gegner seine Markierung gesetzt hat. Die Marker sollen unterschiedliche Farben aufweisen um einfacher feststellen zu können wer was getippt hat und wo das Ziel gewesen wäre.
- ☒ Erstellen einer Oberfläche zur Übersicht aller upgeloadeten Bilder eines Users. Hier soll auch ersichtlich sein ob das Bild bereits durch einen Admin akzeptiert worden ist oder nicht.
- ☒ Erstellen einer Oberfläche zur Übersicht von Statistiken/Achievements:
- ☒ Überarbeiten/Verfeinern der Oberfläche

## Thibault

- ☒ Buttons bei Index, play\_menu, new\_game\_menu sollten so breit wie der Bildschirm ist sein.
- ☒ Einfügen von Back Buttons bei play\_menu und new\_game\_menu anstatt dem Logo.
- ☒ Refactoring von search\_friend und send\_email
  - ☒ Implementieren von add\_friend bei dem man einen Freund der bereits auf WAI registriert ist zu seinen Freunden hinzufügen kann. Dafür wird ein Suchfeld benötigt das den
  - eingegeben Usenamern über AJAX mit der Datenbank abgleicht. Eine Liste der gefundenen Treffer soll auf der gleichen Seite dargestellt werden können.
  - ☒ Implementieren von invite\_friend bei dem man einen Freund über dessen Email-Adresse einladen kann auch WAI zu spielen. Der Inhalt der Email soll nicht verändert werden
  - können (hardcoded)
- ☒ Zentrieren von momentan linksbündigen Formularen
- ☒ Einbauen einer Möglichkeit zum Ändern des Passworts von der Weboberfläche. ("Forgot password?" bei Login und "Change password" im Sidebar Menü).
- ☒ Einfügen einer friends.scala.html, die alle Freunde anzeigt.
- ☒ Bei der Suche soll es die Möglichkeit geben eine neues Spiel mit einem User zu starten oder einen User als Freund hinzuzufügen.

## Philipp

- ☒ Bilder der Sehenswürdigkeiten unskaliert überschreiben.
- ☒ Entfernen der Skalierungsfunktion beim Upload von Bildern. Bilder sind aufgrund des PictureViewers nun responsive und es nicht mehr nötig sie zu skalieren.
- ☒ Ändern der maximalen Dateigröße beim Upload von Bildern auf 5MB.
  - ☒ Umbau des PictureManagers mithilfe von Thumbnailator um die Qualität unserer Thumbnails zu verbessern. Diese Bibliothek ist in das Projekt eingebunden und kann bereits
  - verwendet werden. Info dazu hier: <https://code.google.com/p/thumbnailator/>
- ☒ Report Seite
  - ☒ Hinzufügen eines Report Buttons der auf eine report Seite verweist. Dabei solltest du die PictureID als Parameter verwenden. PictureID ist auf result bereits verfügbar und somit
  - kann bereits beim Laden der Seite der report Button mit dem richtigen Link belegt werden.
  - ☒ Von der Report Seite aus kann man falsch gesetzte Locations von bereits freigegebenen Sehenswürdigkeiten direkt dem Administrator melden. Dabei soll es auch möglich sein
  - eine Begründung anzugeben. Die Daten dieses Formulars sollen dann mithilfe einer Post-Request in Java griffbereit sein. Test mit System.out.println
- ☒ Erstellen einer Oberfläche zur Übersicht offener Spiele.
- ☒ Verbinden der Game Oberfläche mit der aktuellen aus der Datenbank abgefragten Spielrunde des jeweiligen Users.

## Mario

- ☒ Erstellen der Datenbanklogik für das Erstellen eines Spiels (Game.java). Dabei sollen 3 Runden (Round.java) beim Erstellen des Spiels mit einem zufälligen Bild angelegt werden. Das Spiel beinhaltet die konkurrierenden Spieler (User.java), sowie die jeweiligen Runden (Round.java). Es soll zu jeder Zeit ersichtlich sein welche Runde schon gespielt bzw. ob das Spiel
- bereits fertig ist.
- ☒ Erstellen einer Funktion die offene Spiele eines Users in einer List zurückgibt.
- ☒ Falls möglich, Zählfvariable der Bilder bei 0 statt 1 beginnen zu lassen um unnötige Nullpointer exceptions zu vermeiden. --> Leider nicht möglich, weil Datenbank IDs bei 1 starten.
- ☒ Erstellen einer Funktion im DBManager.java die alle Bilder als List zurückgibt.
- ☒ Erstellen einer Funktion im DBManager.java die alle akzeptierten Bilder als List zurückgibt.
- ☒ Erstellen einer Funktion im DBManager.java die es ermöglicht ein bestimmtes Bild zu akzeptieren.
- ☒ Erstellen einer Relation Freund (Friend.java), die zwei User (User.java) miteinander verknüpft.
- ☒ Erstellen einer Funktion die alle Freunde eines gewissen Users als Liste zurückgibt.
- ☒ Überlade alle deine Funktionen die Listen zurückgeben mit 2 Parametern die es erlauben die Anzahl der zurückzugebenden Elemente zu limitieren (start, end).

## ENDSPRINT

Wenn ein Task abgeschlossen ist, einfach in WAI/planning/endsprint/README.md ein [x] setzen.

## Bernhard

- ☒ Überarbeite alle GUI Elemente die noch nicht ausgereift sind
  - ☒ Überarbeite navbar
  - ☒ Überarbeite index
  - ☒ Überarbeite play\_menu
  - ☒ Überarbeite practise
  - ☒ Überarbeite game
  - ☒ Überarbeite login
  - ☒ Überarbeite register
  - ☒ Verfeinere result
  - ☒ Stelle result\_map fertig (-> LatLng Objekte)
- ☒ Verwendung von Geolocation sollte beim Uploaden von Bildern möglich sein
- ☒ Implementiere UnitTests
  - ☒ UnitTest für die Distanzberechnung (Innsbruck->Wien ca. 500km)

## Thibault

- ☒ Überarbeite send\_email. Lass den grauen Kasten sowie die Überschrift im grauen Kasten weg. Ändere die Überschriften in der Navbar dementsprechend. Bleib konsistent.
- ☒ Überarbeite friends. Die Überschrift in der Navbar stimmt nicht. Das Design kann noch verbessert werden. Es ist noch nicht intuitiv dass mit dem blauen Button ein neues Spiel begonnen werden kann. Das "Play" schwebt in der Luft. Mach vielleicht so ähnlich wie bei search\_user einen Kasten in dem der Name des Freundes und der Button zum Spielen drin ist.
- Schreib in den Button einfach "Play" hinein statt dem Bild.
- ☒ Achte auf richtige Verlinkung
  - ☒ Nachdem man einen Freund bei search\_user hinzufügt, sollte auf friends redirected werden
  - ☒ Nachdem ein Spiel bei search\_user begonnen wurde, sollte auf play\_menu redirected werden
- ☒ Füge deine wichtigsten Funktionen in die Sidebar ein
- ☒ Überarbeite Account
  - ☒ Bei Account ein bisschen bei beiden Seiten des grauen Kastens einrücken (probiere container)
  - ☒ Username und Email linksbündig machen
  - ☒ Der Inhalt des Modals sollte auch linksbündig sein
- ☒ Überarbeite send\_email. Schaut im Moment ein wenig blank aus. Machs vielleicht so ähnlich wie Forgot Password. Von mir aus auch mit grauem Kasten.
- ☒ Funktioniert das Zurücksetzen des Passworts bereits?

## Philipp

- ☒ Überarbeite/Vereinfache dein resultGame damit der richtige User rot bzw. grün dargestellt wird
  - ☒ Im Moment sind im Result beide User grün/rot. Es sollte aber der Gewinner grün und der Verlierer rot sein. Der Name des Spielers sollte nicht ausgeschrieben werden sondern einfach mit "You" ersetzt werden.
- ☒ Implementiere UnitTests
  - ☒ UnitTest für deinen PictureManager. Schau ob das Bild nach der Skalierung wirklich die richtige Größe hat.

## Mario

- ☒ LatLng Objekte in Runde einbauen damit wir sie in der result\_map verwenden können
  - ☒ Es sollte nicht möglich sein mit sich selbst ein Spiel beginnen zu können. Bei search\_user beispielsweise kann man sich selbst finden und mit sich selbst ein Spiel beginnen. Sich mit sich selbst zu befreunden geht zum Glück nicht :p
- ☒ Verwende prettify/Distance damit Philipp seine Distanzen schön als km darstellen kann. Zur Berechnung sollte aber der exakte Wert verwendet werden.
- ☒ Stelle sicher, dass admin-only Webseiten auch wirklich nur durch den Admin aufgerufen werden können
- ☒ Implementiere UnitTests
  - ☒ Lass dir einige (min. 3) Testfälle für die Datenbank einfallen. Können auch Pseudo-Tests sein.
- ☒ Habe Bug gefunden beim game. Probiere aus dass du den counter bis auf 5 Sekunden runterlaufen lässt dann lädst du die Seite neu und gibst deinen Tipp bei z.B. 50 Sekunden ab. D.h. du hättest mehr als 60 Sekunden Zeit. Weiß nicht ob man das fixen kann, wenn es zu zeitaufwendig ist einfach weglassen.

# Markdown Syntax

markdown.md

1 Heading

2 =====

3

4 Sub-heading

5 -----

6

7 ### Another deeper heading

8

9 Paragraphs are separated

10 by a blank line.

11

12 Leave 2 spaces at the end of a line to do a

13 line break

14

15 Text attributes *italic*, **bold**,

16 `monospace`, ~~strikethrough~~.

17

18 A [link](http://example.com)(http://example.com).

19 <No space between ] and (>

20

21 Shopping list:

22

23 \* apples

24 \* oranges

25 \* pears

26

27 Numbered list:

28

29 1. apples

30 2. oranges

31 3. pears

32

markdown.md Preview

## Heading

---

### Sub-heading

---

#### Another deeper heading

Paragraphs are separated by a blank line.

Leave 2 spaces at the end of a line to do a line break

Text attributes *italic*, **bold**, `monospace`, ~~strikethrough~~.






A [link](http://example.com).

Shopping list:

- apples
- oranges
- pears

Numbered list:

1. apples
2. oranges
3. pears

markdown.md 32:1  2 deprecations UTF-8 GitHub Markdown  master  +32  1 update 

# Vielen Dank!



<http://whereamiplay.herokuapp.com/>