

# Matching Activities

---

Eine Präsentation von Florian Berghahn & Maximilian Feldmann

Auf Basis des Feedbacks der ersten beiden Audits und den Open Spaces haben wir beschlossen, sämtliche Artefakte neu zu iterieren. Dabei haben wir die Diagramme und Analysen in der Reihenfolge der klassischen Projektstruktur überarbeitet.

Die überarbeiteten Versionen wurden im GitHub in der selben Reihenfolge als Iteration hochgeladen.  
Die Links befinden sich in den jeweiligen Folien-Notizen.

## Reihenfolge:

1. Re-Evaluierung des Domänenmodells
2. Stakeholderanalyse anhand des Feedbacks aus dem Open Space
3. Erfordernisse & Anforderungen mit der Hilfe der MCI-Schablone überarbeiten
4. Weiterentwickelte Projektrisiken

erweitern

5. POC's auf Basis der neuen Iteration  
entwickeln



Das Domänenmodell als Einstieg in das Projekt. Nach der Entwicklung der „deliverables“ für den Audit Nr. 3 waren keine Änderungen am Domänenmodell Version 6 nötig.

## 2. Iteration: Stakeholderanalyse

---

- Kategorisierung geändert von internen/externen Stakeholdern zu Primär- und Sekundärstakeholdern
- Hobbygruppen aus der Kategorie der Sekundär-Stakeholder entfernt, da sie nur als Primär-Stakeholder relevant für das System sind
- Die Stakeholdergruppe der Teilnehmer, die nicht direkt vom System betroffen sind, entfernt

4



GitHub-Link zur alten Version:

[https://github.com/fberghahn/EPWS2122Feldmann-Berghahn/blob/main/Artefakte/WS2122\\_FeldmannBerghahn\\_02\\_stakeholderanalyse.pdf](https://github.com/fberghahn/EPWS2122Feldmann-Berghahn/blob/main/Artefakte/WS2122_FeldmannBerghahn_02_stakeholderanalyse.pdf)

GitHub-Link zur überarbeiteten Version der Stakeholderanalyse:

[https://github.com/fberghahn/EPWS2122Feldmann-Berghahn/blob/main/Artefakte/WS2122\\_FeldmannBerghahn\\_02\\_stakeholderanalyse\\_V2.pdf](https://github.com/fberghahn/EPWS2122Feldmann-Berghahn/blob/main/Artefakte/WS2122_FeldmannBerghahn_02_stakeholderanalyse_V2.pdf)

### 3. Iteration: Erfordernisse & Anforderungen

---

- Systemspezifische Informationen aus dem Nutzungskontext und den Erfordernissen entfernt
- Nutzungskontext neu formuliert und damit auf die Stakeholderanalyse bezogen
- Erfordernisse auf definierte Stakeholder-Gruppen bezogen
- Anforderungen überarbeitet und nach Vorbild der Schablone neu formuliert

GitHub-Link zur alten Version:

[https://github.com/fberghahn/EPWS2122Feldmann-Berghahn/blob/main/Artefakte/WS2122\\_FeldmannBerghahn\\_03\\_Erfordernisse\\_Anforderungen\\_V2.pdf](https://github.com/fberghahn/EPWS2122Feldmann-Berghahn/blob/main/Artefakte/WS2122_FeldmannBerghahn_03_Erfordernisse_Anforderungen_V2.pdf)

GitHub-Link zur überarbeiteten Version der Erfordernisse und

Anforderungen:

[https://github.com/fberghahn/EPWS2122Feldmann-Berghahn/blob/main/Artefakte/WS2122\\_FeldmannBerghahn\\_03\\_Erfordernisse\\_Anforderungen\\_V3.pdf](https://github.com/fberghahn/EPWS2122Feldmann-Berghahn/blob/main/Artefakte/WS2122_FeldmannBerghahn_03_Erfordernisse_Anforderungen_V3.pdf)



#### 4. Iteration: Weiterentwickelte Projektrisiken

---

Technisch		Unsichere Datenbankstruktur	Anfragen funktionieren und sind sicher	Fehlerhafte Einbindung oder Leaks von Daten	Erneute Evaluierung der Datenbankstruktur
Technisch		Einbindung der Maps-API fehlgeschlagen	Einbindung erfolgreich	Einbindung fehlgeschlagen	Alternativ-API's verwenden
Technisch		Implementierung der Chat-Funktion fehlgeschlagen	Implementierung erfolgreich	Funktioniert nicht oder fehlerhaft	Alternativ-Services verwenden
Technisch		Implementierung des Nutzerprofils fehlgeschlagen	Implementierung erfolgreich	Erstellung nicht möglich oder Daten nicht abrufbar	Neu implementieren

-> Mehr systembezogene technische Risiken hinzugefügt

GitHub-Link zur alten Version:

[https://github.com/fberghahn/EPWS2122Feldmann-Berghahn/blob/main/Artefakte/WS2122\\_FeldmannBerghahn\\_09\\_Weiterentwickelte\\_projektrisiken.pdf](https://github.com/fberghahn/EPWS2122Feldmann-Berghahn/blob/main/Artefakte/WS2122_FeldmannBerghahn_09_Weiterentwickelte_projektrisiken.pdf)

GitHub-Link zur überarbeiteten Version der Weiterentwickelten

Projektrisiken:

[https://github.com/fberghahn/EPWS2122Feldmann-Berghahn/blob/main/Artefakte/WS2122\\_FeldmannBerghahn\\_09>Weiterentwickelte projektrisiken\\_V2.pdf](https://github.com/fberghahn/EPWS2122Feldmann-Berghahn/blob/main/Artefakte/WS2122_FeldmannBerghahn_09>Weiterentwickelte projektrisiken_V2.pdf)

## 5. Iteration: Proof of Concepts

---

- Komplette neu erstellt auf Grund von fehlendem Bezug zu den weiterentwickelten Risiken
- Beim Neuerstellen wurden Risiken gewählt, die besser zur spezifischen Projektumsetzung passen
- Feedback per GitHub-issue eingeholt und darauf hin angepasst

7



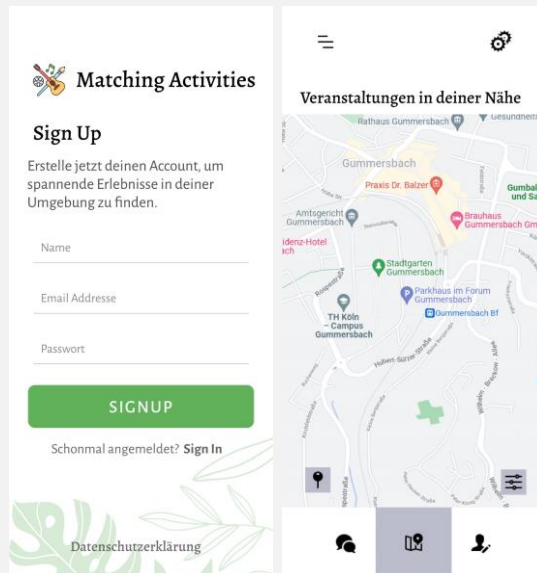
GitHub-Link zur alten Version:

[https://github.com/fberghahn/EPWS2122Feldmann-Berghahn/blob/main/Artefakte/WS2122\\_FeldmannBerghahn\\_07\\_Proof\\_of\\_Concept\\_V3.pdf](https://github.com/fberghahn/EPWS2122Feldmann-Berghahn/blob/main/Artefakte/WS2122_FeldmannBerghahn_07_Proof_of_Concept_V3.pdf)

GitHub-Link zur überarbeiteten Version der POC's:

[https://github.com/fberghahn/EPWS2122Feldmann-Berghahn/blob/main/Artefakte/WS2122\\_FeldmannBerghahn\\_07\\_Proof\\_of\\_Concept\\_V4.pdf](https://github.com/fberghahn/EPWS2122Feldmann-Berghahn/blob/main/Artefakte/WS2122_FeldmannBerghahn_07_Proof_of_Concept_V4.pdf)

## 6. Wireframes



8

(Von links nach rechts) Nr.1: -Account-Erstellung mit Verlinkung zur Datenschutzerklärung

Nr.2: -Startbildschirm der App, zeigt unmittelbare Umgebung, zeigt Veranstaltungen in der Nähe an (hier nur eine Beispielkarte)

-

Links unten auf der Karte ist der Standort-Button

-

Rechts unten auf der Karte sind die Filter-Einstellungen für die Karte

- Schnellauswahlliste

unten im Bild steht auf dem Punkt „Karte“ in der Mitte

- Oben links

ist der Hauptmenü-Button

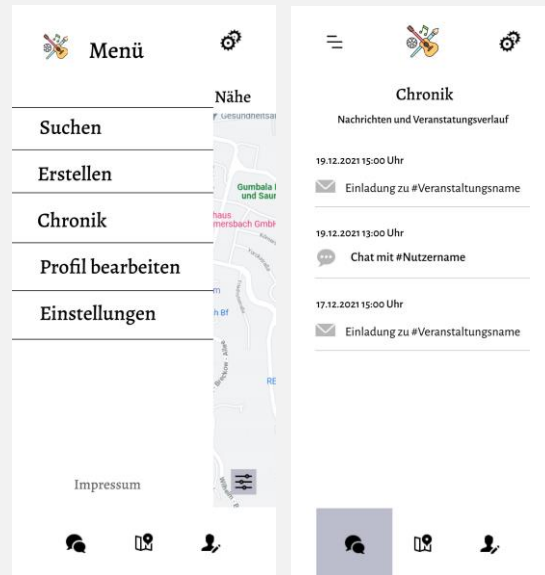
- Oben

rechts ist der Einstellungs-Button

GitHub-Link zur vollständigen Version:

[https://github.com/fberghahn/EPWS2122Feldmann-Berghahn/blob/main/Artefakte/WS2122\\_FeldmannBerghahn\\_11\\_Rapid\\_Prototype.pdf](https://github.com/fberghahn/EPWS2122Feldmann-Berghahn/blob/main/Artefakte/WS2122_FeldmannBerghahn_11_Rapid_Prototype.pdf)

## 6. Wireframes



9

Nr.3: Ausklappbares Menü mit allen Grundfunktionen und einer Impressums-Verlinkung

Nr.4: Chronik (Nachrichten und Veranstaltungsverlauf)

- Zeigt im chronologischen Verlauf alle Veranstaltungsbenachrichtigungen und Chatverläufe

GitHub-Link zur vollständigen Version:

[https://github.com/fberghahn/EPWS2122Feldmann-Berghahn/blob/main/Artefakte/WS2122\\_FeldmannBerghahn\\_11\\_Rapid\\_Prototype.pdf](https://github.com/fberghahn/EPWS2122Feldmann-Berghahn/blob/main/Artefakte/WS2122_FeldmannBerghahn_11_Rapid_Prototype.pdf)



## 6. Wireframes

**Profil erstellen**

Nutzername

Profilbeschreibung (200 Zeichen)

Interessen

- Sport
- Brettspiele
- Nightlife
- Just talk

**Veranstaltung erstellen.**

Veranstaltungsname

Datum & Uhrzeit

Location

Teilnehmeranzahl

Kategorie

Beschreibung (200 Zeichen)

Materialien (optional)

10

Nr.5: Profilerstellungs-Screen mit Namen, personalisierter Beschreibung (max. 200 Zeichen) und einer Drop-Down-Auswahl mit vorgegebenen Oberkategorien von Interessen

Nr.6: Screen zum Erstellen von Veranstaltungen mit allen relevanten Daten

- Das Feld Kategorie ist eine Drop-Down-Auswahl mit den gleichen

Möglichkeiten wie in der Interessen-  
Auswahl

GitHub-Link zur vollständigen Version:

[https://github.com/fberghahn/EPWS2122Feldmann-Berghahn/blob/main/Artefakte/WS2122\\_FeldmannBerghahn\\_11\\_Rapid\\_Prototype.pdf](https://github.com/fberghahn/EPWS2122Feldmann-Berghahn/blob/main/Artefakte/WS2122_FeldmannBerghahn_11_Rapid_Prototype.pdf)

### **Boyer-Moore-Horspool (string-searching)**

Der Boyer-Moore-Horspool-Algorithmus ist ein von Nigel Horspool entwickelter Prozess, der im Bereich der Informatik zum Identifizieren von Textzeilen verwendet wird.

Es werden zwei „strings“ auf Übereinstimmung geprüft indem die gleichen Zeichen und deren Folgezeichen, ungeachtet ihrer Position, verglichen werden. Auf diese Weise können „Sub-strings“ innerhalb eines „strings“ erkannt werden.

GitHub-Link zur vollständigen  
Recherche der Matching Algorithmen:

[https://github.com/fberghahn/EPWS2122Feldmann-Berghahn/blob/main/Artefakte/WS2122\\_FeldmannBerghahn\\_12\\_MatchingAlgorithmen.pdf](https://github.com/fberghahn/EPWS2122Feldmann-Berghahn/blob/main/Artefakte/WS2122_FeldmannBerghahn_12_MatchingAlgorithmen.pdf)

## **Ablauf des Matching**

1. Standort festlegen
2. Veranstaltungen in der Nähe finden
3. Matching
4. Einladung versenden
5. Einladung annehmen/ablehnen

12



### **1. Standort festlegen:**

Standort des Nutzers feststellen oder eine Eingabe des Standorts zulassen.

### **2. Veranstaltungen in der Nähe finden:**

Das System soll in einem Radius

(veränderbarer Standardwert) um den Standort des Nutzers alle Veranstaltungen feststellen und daraus eine Liste erstellen.

(Mögliches Tool zur Radius Suche ist die Geometry Library der Google Maps API)

### **3. Matching:**

Mit allen Veranstaltungen, die im gewählten Radius des Standorts des Nutzers liegen, wird das Matching begonnen. Im Prototypen soll nur nach den gewählten Oberkategorien gematcht werden, wenn die Anzahl an Veranstaltungen zunimmt könnte man dies

durch einen String-Algorithmus erweitern.  
Dieser soll die Veranstaltungsnamen  
vergleichen, um die Matching-  
Möglichkeiten zu verbessern.  
Oberkategorie wird als Variable beim  
Veranstaltung Erstellen aus einer Reihe  
Vorgaben ausgewählt.

#### **4. Einladung versenden:**

Bei einem Match soll der Nutzer eine  
Einladung erhalten die er annehmen oder  
ablehnen kann.

#### **5. Einladung annehmen/ablehnen:**

Wenn der Nutzer die Einladung annimmt wird er der Veranstaltung hinzugefügt und die Teilnehmeranzahl wird um eins inkrementiert. Sonst wird die Einladung verworfen.

GitHub-Link zum vollständigen Pseudocode:

[https://github.com/fberghahn/EPWS2122Feldmann-Berghahn/blob/main/Artefakte/WS2122\\_FeldmannBerghahn\\_11\\_Matchingalgorithmus\\_pseudocode.pdf](https://github.com/fberghahn/EPWS2122Feldmann-Berghahn/blob/main/Artefakte/WS2122_FeldmannBerghahn_11_Matchingalgorithmus_pseudocode.pdf)

## 9. Durchgeführte POC's

---

1. Es gilt zu beweisen, dass die die Einbindung einer Karte und Feststellung des Standorts über die Google Maps API funktioniert. Exit Kriterium: Das System kann Anfragen an die API stellen, die beantwortet werden und die Erkennung des Standortes funktioniert.
5. Es gilt zu beweisen, dass die Kompetenz zur Erstellung einer App in Android Studio vorhanden sind. Exit Kriterium: Das Aufsetzen der Programmierungsumgebung ist mit erstem funktionierenden Code beweisbar.

13



Die Durchführung der POC's Nr.1 und Nr.5 sind auf den folgenden Folien durch Screenshots in Android Studio dargestellt.

GitHub-Link zur überarbeiteten Version des POC:

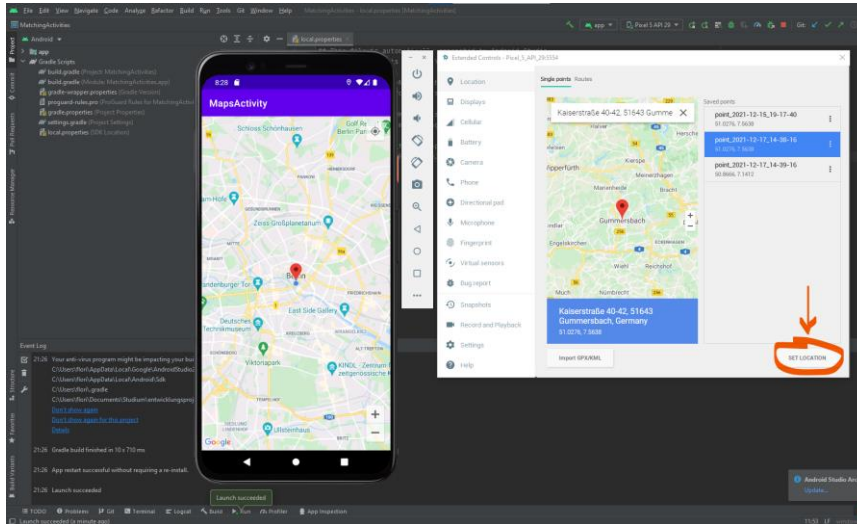
<https://github.com/fberghahn/EPWS21>



22Feldmann-

Berghahn/blob/main/Artefakte/WS2122\_FeldmannBerghahn\_07\_Proof\_of\_Concept\_V4.pdf

## 9. Durchgeführte POC's – Schritt 1



14

Bei Start der App wird ein Pin auf dem letzten Standort des Users gesetzt. Über die Einstellungen des Emulators kann die Location geändert werden.

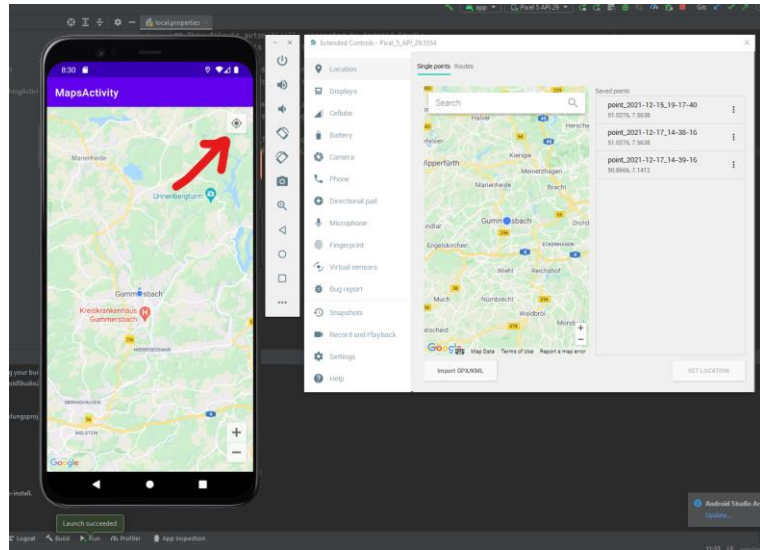
GitHub-Link zur überarbeiteten Version des POC:

<https://github.com/fberghahn/EPWS21>

22Feldmann-

Berghahn/blob/main/Artefakte/WS2122\_FeldmannBerghahn\_07\_Proof\_of\_Concept\_V4.pdf

## 9. Durchgeführte POC's – Schritt 2



15

Nun kann man durch den Button oben rechts das System den aktuellen Standort ermitteln lassen. Wir hier zu sehen ist, erkennt das System den aktuellen Standort des Users (hier: Gummersbach).

GitHub-Link zur überarbeiteten Version des POC:

[https://github.com/fberghahn/EPWS2122Feldmann-Berghahn/blob/main/Artefakte/WS2122\\_FeldmannBerghahn\\_07\\_Proof\\_of\\_Concept\\_V4.pdf](https://github.com/fberghahn/EPWS2122Feldmann-Berghahn/blob/main/Artefakte/WS2122_FeldmannBerghahn_07_Proof_of_Concept_V4.pdf)

GitHub-Link zum Code-Ordner:

<https://github.com/fberghahn/EPWS2122Feldmann-Berghahn/tree/main/MatchingActivities>