

## **Entwicklungsprojekt 2023/24 - Audit 2**

### **Eine Spielersuche für ein Brettspieltreffen**

---

Frederik Hausen, Chenghua He und Esra Sancak



## Feedback vom letzten Audit

---

### Vergleich einer Webanwendungen und App

#### Vorteile von Web-Apps:

- laufen unabhängig vom Endgerät und dem eingesetzten Betriebssystem,
- funktionieren überall, wo es eine Internetverbindung und einen Browser gibt
- speichern nur sehr wenige Daten lokal (also auf Ihrem Endgerät) ab und benötigen daher nur wenig Speicherplatz
- werden nicht über einen App Store oder eine andere Quelle heruntergeladen, sondern über die URL aufgerufen,
- starten sofort, ohne vorherige Installation oder Einrichtung
- sind immer Up-to-date, denn Nutzer: innen haben stets die aktuellste Version

#### Vorteile von Apps:

- können die speziellen Software- und Hardware-Funktionen eines Gerätes nutzen wie Kamera und Sensoren
- auch Offline nutzbar
- sind oft durch die Optimierung auf ein Gerät leistungsfähiger als Web-Apps

2



Die Schlussfolgerung vom Feedback des letzten Audits: Nach dem letzten Audit kommen wir zu dem Schluss, dass unser Projekt am besten als Webanwendung umgesetzt werden sollte. Die Gruppendiskussion und die präsentierte Folie haben deutlich positive Aspekte hervorgehoben, die für eine Webanwendung sprechen. Die Entscheidung basiert auf Faktoren wie Benutzerfreundlichkeit, plattformübergreifender Zugänglichkeit und einfacher Aktualisierbarkeit und weiters. Die Vorteile einer App sind mit unserer Anwendung nicht zwingend notwendig.

Use Case

## Use Case

### Kategorie Spielsuche für Brettspiele

Use Case	Spielsuche für Brettspieltreffen	
Goal in Context	Ermöglichen, dass Spieler sich für ein Brettspieltreffen finden und verbinden können	
Preconditions	Das System ist online und funktionsfähig Der Spieler ist registriert und eingeloggt	
Postconditions	Die Teilnehmerliste für das Brettspieltreffen wurde aktualisiert Die Spieler können sich untereinander für das Treffen koordinieren	
Success End Condition	Der Spieler hat sich erfolgreich für das Brettspieltreffen angemeldet Die Teilnehmerliste ist aktualisiert, und die Spieler können sich für das Treffen koordinieren	
Failed End Condition	Falls während des Anmeldeprozesses ein technisches Problem auftritt (z.B. Serverausfall) und der Spieler die Anmeldung nicht abschließen kann, wird dem Spieler eine entsprechende Fehlermeldung angezeigt Der Spieler wird zurück zum Ausgangspunkt des Anmeldeprozesses geleitet, und der Vorgang wird abgebrochen. Der Spieler kann es zu einem späteren Zeitpunkt erneut versuchen, nachdem das technische Problem behoben wurde	
Primary, Secondary Actor	Spieler, System	
Trigger	Der Spieler möchte an einem Brettspieltreffen teilnehmen	
Description (Main Flow)	Step	Action
	1	Der Spieler navigiert zur Seite "Brettspieltreffen"
	2	Das System zeigt eine Liste der verfügbaren Brettspieltreffen mit relevanten Details (Ort, Datum, Teilnehmerzahl, Spieleangebot) an
	3	Der Spieler überlegt sich, nach welchen Kriterien er die Treffen filtern möchte (z.B. Ort, Datum, Spieleangebot)
	4	Der Spieler wendet die Filter an, um die Liste auf die für ihn relevanten Treffen zu reduzieren
	5	Der Spieler wählt ein Brettspieltreffen aus der Liste aus, das seinen Kriterien entspricht
	6	Das System zeigt eine Liste der bereits angemeldeten Spieler für das ausgewählte Treffen an

4

Die Use Case-Spezifikation für die Brettspiel-Suche beschreibt den Prozess, wie Benutzer nach passenden Spielen suchen. Trigger können die Initiierung einer neuen Suche oder das Anwenden spezifischer Filter sein. Preconditions beinhalten die Anmeldung des Benutzers, und Postbedingungen könnten den Erfolg der Suche und die Anzeige relevanter Ergebnisse umfassen. Die Spezifikation strebt an, eine benutzerfreundliche und effektive Suche mit Optionen wie Filtern, Bewertungen und Spielempfehlungen zu erreichen.

Use

Case: <https://github.com/FHausen/EPWS2324HausenHeSancak/blob/main/Artefakte/PDF%20Copies%20f%C3%BCr%20Audit%203/Use%20Case-Spielsuche.pdf>

Anwendungslogik

## Anwendungslogik

---

### Profilerstellung

Nutzer müssen für die Nutzung der Anwendung Profile mit ihren Vorlieben im Thema Brettspiele und ihren Standort bestimmen. Dies ermöglicht unseren Algorithmus personalisierte Empfehlungen an die Nutzer auszurichten. Die Verwaltung der Profile kann mit Firebase ausgeführt werden.

### Gruppenerstellung

Bei der Gruppenerstellung werden Daten wie Standort, Termin, Spiele, Genres, teilnehmende Personen und weitere konkrete Informationen abgefragt, um möglichst die passenden Mitspieler für die Gruppe finden zu können.

### Spielersuche

Die Spielersuche bietet Filteroptionen, die es ermöglichen, präzise die Mitspieler zu finden, die dem individuellen Spielstil entsprechen. Dabei spielt es keine Rolle, ob nach Spielpräferenzen, Standort oder Verfügbarkeit gesucht wird.

### Matching-Algorithmus

Die Nutzung von einem Matching-Algorithmus sorgt für eine ideale Zusammensetzung von Mitspielern und Gruppen. Durch den Vergleich von verschiedenen Metadaten wie Standort, Präferenzen, Erfahrung, Alter, Verfügbarkeit und weitere Daten können passende Zusammensetzungen direkt von der Anwendung vorgeschlagen werden, ohne spezifisch danach suchen zu müssen.

### Empfehlung-Algorithmus

Die Nutzer werden anhand ihrer Vorlieben, Suchverläufe, Nutzungsverhalten und weiteren Daten zu ihrer Person passende Brettspiele und Inhalte vorgeschlagen.

### Lokalisierungs-Algorithmus

Um möglichst viele lokale Gruppen und Mitspieler finden zu können müssen die Standorte der Nutzer bestimmt werden und damit die Distanz und Zeitaufwand der jeweiligen Veranstaltungen dem Nutzer mitgeteilt werden.

### Benachrichtigungen

Benachrichtigungen helfen dem Nutzer dabei schnell auf passende Vorschläge des Matching-Algorithmus aufmerksam zu werden. Zudem werden so Nachrichten von anderen Nutzern und Neuigkeiten zu Brettspieltreffen sofort zugestellt.

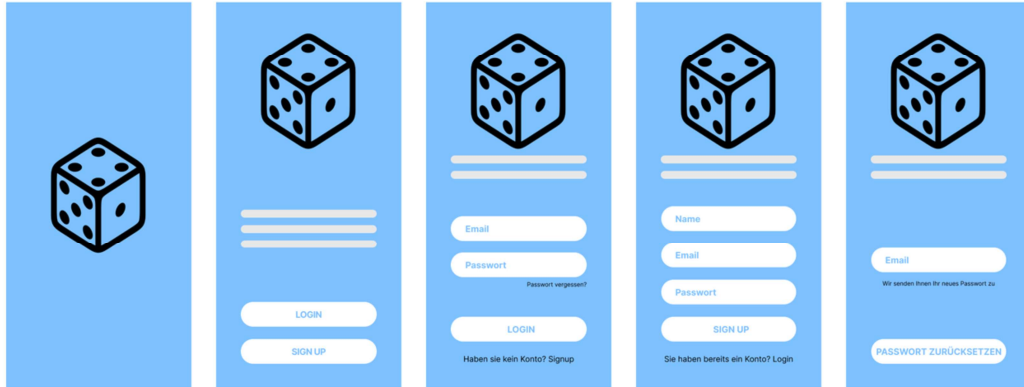
6



Die Funktionsweise unserer Anwendung wird durch die Anwendungslogik in kurzen Begriffen erklärt. In unserer Anwendung liegt besonders das Zusammenführen von Vorlieben und Präferenzen im Vordergrund, das spiegelt sich auch in den einzelnen Funktionen der Anwendung wider. Die Funktionen sorgen dafür, dass die nötigen Metadaten bereitgestellt werden, welche für unsere Algorithmen zwingend notwendig sind. Durch eine große Menge an Metadaten werden die Algorithmen bessere und präzisere Ergebnisse liefern.

**Rapid Prototype**

### Wireframes - Anmeldung



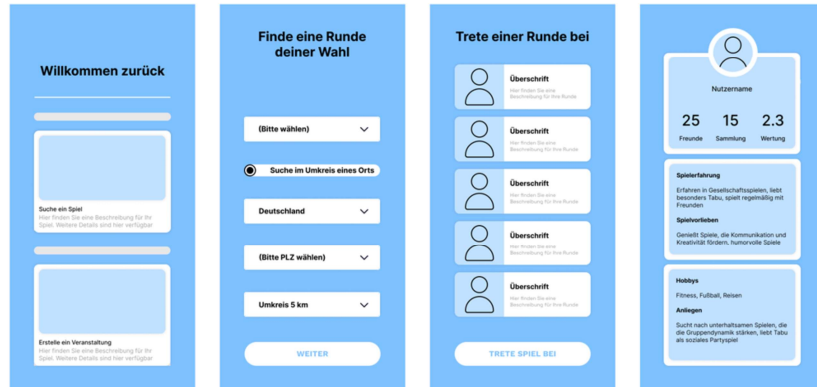
**Login:** Beim Login liegt der Fokus auf Benutzerfreundlichkeit. Ein deutlich sichtbares Anmeldeformular mit Feldern für Benutzername oder E-Mail und Passwort ermöglicht einen schnellen Zugang.

**Signup:** Das Design für die Registrierung sollte einladend und benutzerfreundlich sein. Klare Anweisungen und gut platzierte Formularfelder erleichtern den Registrierungsprozess.

**Passwort vergessen:** Die Funktion "Passwort vergessen" sollte leicht erreichbar sein, idealerweise über einen gut sichtbaren Link auf der Login-Seite. Klare Anweisungen zur Wiederherstellung des Passworts, sowie die Möglichkeit, dies über eine verifizierte E-Mail.



### Wireframes – Spielersuche / Profil



Das UI-Design für die Spielersuche bei Brettspielen bietet eine benutzerfreundliche Erfahrung. Mit klaren Suchleisten, Filteroptionen und übersichtlichen Suchergebnissen ermöglicht es Nutzern ein einfaches Finden von Mitspielern. Das Profil enthält die Informationen über den Spieler.

### Wireframes – Spielrunde erstellen

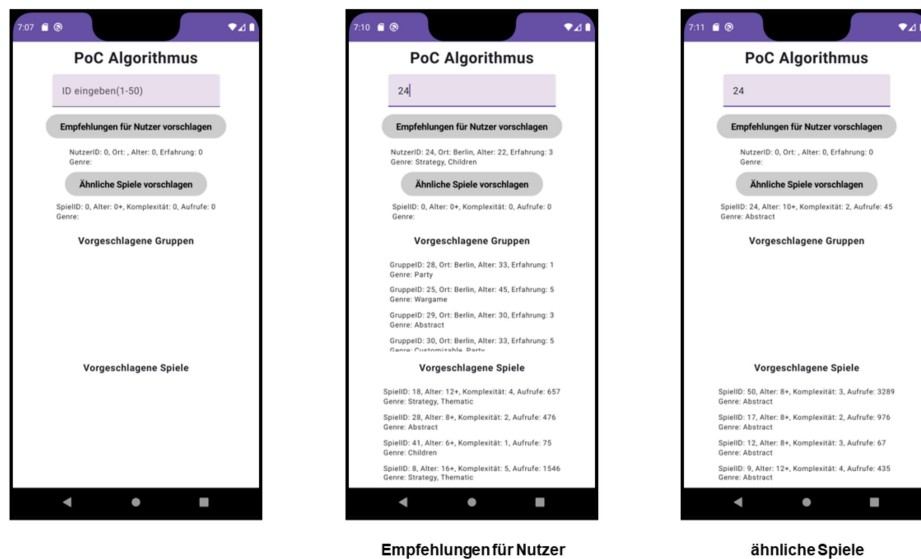
The image displays three sequential wireframe screens for a game creation interface, all set against a light blue background.

- Willkommen zurück:** Features a title bar, a horizontal separator, a large rectangular placeholder for an image, a text prompt "Suche ein Spiel/ Mitspieler" with a subtext "Some description related to your game here. Some more description here.", another horizontal separator, another large rectangular placeholder, and a text prompt "Erstelle ein Veranstaltung" with a subtext "Some description related to your game here. Some more description here."
- Wähle ein Spiel aus:** Contains a title bar, a list of game categories (Kartenspiel, Strategiespiel, Familienspiel, Würfelspiel, Geschicklichkeitsspiel, Brettspiel) each with a checkmark icon, and a "WEITER" button at the bottom.
- Erstelle ein Spiel:** Contains a title bar, four dropdown menus labeled "(Bitte Spiel auswählen)", "(Spiel Runde)", "(Spieler Anzahl)", and "(Bitte auswählen)", and an "ERSTELLE EIN SPIEL" button at the bottom.

Das UI-Design für die Erstellung einer Brettspielrunde verfolgt das Ziel, einen unkomplizierten und nutzerfreundlichen Prozess zu ermöglichen. Innerhalb eines klaren und gut strukturierten Formulars haben Spieler die Möglichkeit, essenzielle Informationen wie den Spielnamen, die gewünschte Spieleranzahl und den bevorzugten Spielmodus einzugeben. Dieser Ansatz erleichtert nicht nur die Initiierung einer neuen Spielrunde, sondern bietet auch eine klare Übersicht über die grundlegenden Parameter, um die individuellen Präferenzen der Spieler abzudecken.

PoCs

## Proof of Concept für Empfehlungen



Um Sicherzustellen, dass der Algorithmus für die Empfehlungen Umsetzbar ist haben wir mit Android Studio ein PoC erstellt, welches die Funktionen vereinfacht darstellen sollte. Für den PoC wurden Beispiel-Metadaten zu jeweils 50 Nutzer, Spiele und Spielgruppen erstellt, um den Umfang des Algorithmus möglichst genau betrachten zu können. Der PoC sucht nach Angabe des Benutzers passende Spielgruppen und Brettspiele anhand der Metadaten. Im Rahmen des PoCs wurden die Exit-Kriterien erfüllt und die Fail-Kriterien konnten verhindert werden. Bei der Erstellung des PoCs war ursprünglich auch geplant den eigenen Suchverlauf als Teil der Metadaten in die Empfehlungen einfließen zu lassen, welches sich aber als Herausforderung herausgestellt hat. Um die Begriffe der Suchverläufe vernünftig nutzen zu können bräuchte es die Verwendung von Keywords, um die Suchbegriffe im richtigen Kontext für den Vergleich der Metadaten zu bringen. Dies brachte aber im Rahmen des PoCs wahrscheinlich keinen großen Mehrwert, da bereits die Brettspielgenres als Keywords verwendet werden, um die Ähnlichkeit der Objekte zu vergleichen. Andernfalls hätte sich die Nutzung von KI für die Zuordnung des Kontextes geeignet, welches aber den Rahmen dieses PoCs gesprengt hätte. Dadurch hat sich allerdings keinen negativen Einflüsse auf die Ergebnisse des PoCs ergeben. Android Studio Projekt zum Algorithmus PoC: <https://github.com/FHhausen/EPWS2324HausenHeSancak/tree/main/PoCs/AlgorithmusPoC>

Projektplan - Deliverables für das 4.Audit

## Projektplan

---

### Meilensteine für Audit 4

29.01.2024 - Einführung in die modellierten Konzepte

05.02.2024 - Entwicklung des funktionalen Prototyps

12.02.2024 - Analyse der Implementierungsdetails

19.02.2024 - Erfüllung der Projektziele und Herausforderungen

26.02.2024 – Audit 4 + Poster und Posterslam und Showcase

