

# Roam Network - Community-Driven Network Transformation

## Bachelor-Thesis



Studentin/Student	Elena Algaria, Studiengang Informatik – iCompetence, <a href="mailto:elena.algaria@students.fhnw.ch">elena.algaria@students.fhnw.ch</a> Yannick Hohler, Studiengang Informatik, <a href="mailto:yannick.hohler@students.fhnw.ch">yannick.hohler@students.fhnw.ch</a>
Experte	Prof. Dr. Norbert Seyff, <a href="mailto:norbert.seyff@fhnw.ch">norbert.seyff@fhnw.ch</a> , IIT FHNW
Fachbetreuer	Dr. Nitish Patkar, <a href="mailto:nitish.patkar@fhnw.ch">nitish.patkar@fhnw.ch</a> , IIT FHNW
Auftraggeber	Michail Angelov, <a href="mailto:michail@romdao.network">michail@romdao.network</a> , Roam Network
Projektnummer	24FS_IIT19
Vereinbarung	vertraulich

Fachhochschule Nordwestschweiz, Hochschule für Technik

Windisch, August 2024

## Abstract

### **Abstract**

Im Rahmen dieser Arbeit wurde eine Android-Applikation weiterentwickelt, die das globale Mobilfunknetzwerk durch kontinuierliche Datenerfassung verbessern soll. Die Applikation erfasst im Hintergrund Messwerte der Mobilfunkverbindung. Im Gegenzug erhält der Nutzer Punkte.

Ziel dieser Untersuchung war es, die grössten Motivationsfaktoren, basierend auf praktischem Nutzen, Gamification, UX und UI, zu identifizieren, um die langfristige Nutzung der Applikation sicherzustellen.

Anhand von A/B-Testings und einer Umfrage konnte aufgezeigt werden, dass ein praktischer Nutzen, wie personalisierte Providervorschläge, in Kombination mit dem Motivationsfaktor Geld, über die Auszahlung von Roamcoins, die Nutzer am meisten motiviert. Des Weiteren konnte anhand von Usability-Testings gezeigt werden, dass eine klare UX und ein ästhetisch ansprechendes UI weitere wichtige Motivationsfaktoren sind, um den Nutzer zur Installation und langfristigen Nutzung der App zu motivieren.

Mit diesen Ansätzen kann Roam Network eine erfolgreiche Applikation veröffentlichen. Um jedoch neben einer technisch affinen Zielgruppe auch andere Zielgruppen erreichen zu können, wäre ein stärker gamifizierter Ansatz erfolgsversprechender. Das Roamgochi-Spiel würde eine jüngere oder auch technisch weniger affine Zielgruppe besser erreichen. Des Weiteren sind Roamcoins, die man traden könnte, nicht sehr erfolgreich, solange sie nur einen kleinen Wert hätten. Ebenfalls bieten die Providervorschläge nur einen kurzfristigen Nutzen.

### **Keywords:**

Netzwerkdaten, Mobile-Applikation, Android-Geräte, Mobilfunknetz, Gamification, User Experience, User Interface

Inhaltsverzeichnis

<b>Abbildungsverzeichnis</b> .....	<b>v</b>
<b>Tabellenverzeichnis</b> .....	<b>vi</b>
<b>1 Einleitung</b> .....	<b>1</b>
1.1 <i>Problemstellung und Ziele</i> .....	1
1.2 <i>Methoden und Materialien</i> .....	2
1.3 <i>Struktur dieses Dokumentes</i> .....	3
1.4 <i>Projektmanagement</i> .....	3
<b>2 Grundlagen und Hintergründe</b> .....	<b>5</b>
2.1 <i>Roam Network</i> .....	5
2.2 <i>RoamApp</i> .....	6
2.2.1 <i>Funktion</i> .....	7
2.2.2 <i>Technische Analyse</i> .....	8
2.1 <i>Anforderungen und Umgebung</i> .....	9
2.2 <i>SOTA</i> .....	10
2.2.1 <i>Marktanalyse</i> .....	10
2.2.2 <i>Workshop</i> .....	13
2.2.3 <i>Literatur und Websuche</i> .....	14
2.2.4 <i>Umfragen und Testings</i> .....	19
<b>3 Theoretische Umsetzung</b> .....	<b>21</b>
3.1 <i>Zielgruppen</i> .....	21
3.2 <i>Figma Hi-Fi-Versionen</i> .....	23
3.2.1 <i>Hi-Fi Provider Version</i> .....	24
3.2.2 <i>Hi-Fi Roamgochi Version</i> .....	25
3.2.3 <i>Roamcoins</i> .....	26
3.2.4 <i>UX und UI-Verbesserungen</i> .....	27
3.3 <i>Mobilfunkanbietervergleich</i> .....	30
3.3.1 <i>Signalbewertung</i> .....	30
3.3.2 <i>Preisbewertung</i> .....	30
<b>4 Praktische Umsetzung</b> .....	<b>32</b>
4.1 <i>Grundlagen</i> .....	32
4.2 <i>Allgemeine Änderungen</i> .....	33
4.3 <i>Erklärungsscreens</i> .....	33
4.4 <i>Push-Benachrichtigungen</i> .....	34
4.5 <i>Kryptowährungsintegration</i> .....	35
4.6 <i>Providervergleich</i> .....	35
4.6.1 <i>Messungen</i> .....	35
4.6.2 <i>Datenbank</i> .....	38
4.6.3 <i>Preisbewertung</i> .....	38
4.6.4 <i>Rangliste</i> .....	38
4.7 <i>Roamgochi Game</i> .....	39
4.7.1 <i>State Pattern</i> .....	39
4.7.2 <i>Levelsystem</i> .....	40
<b>5 Testings</b> .....	<b>41</b>
5.1 <i>Usability-Testings Figma</i> .....	41

## Inhaltsverzeichnis

5.2	<i>Usability-Testing Provider Prototyp</i>	42
5.3	<i>A/B-Testings</i>	44
5.3.1	Umfrage	44
5.3.2	Interviews	47
<b>6</b>	<b>Diskussion</b>	<b>50</b>
6.1	<i>Ergebnisse</i>	50
6.2	<i>Herausforderungen</i>	51
<b>7</b>	<b>Schlussbemerkungen</b>	<b>52</b>
	<b>Quellenverzeichnis</b>	<b>53</b>
	<b>Glossar</b>	<b>55</b>
	<b>Eigenständigkeitserklärung</b>	<b>56</b>

## Abbildungsverzeichnis

### Abbildungsverzeichnis

Abbildung 1 RoamApp Stand Juli (Version: 1.0.4)	7
Abbildung 2 MVVM Struktur	8
Abbildung 3 Onion Modell der Stakeholder	10
Abbildung 4 Kategorisierung von ähnlichen Produkten auf dem Markt	11
Abbildung 5 Vergleich der Features der ähnlichen Apps	12
Abbildung 6 Auswertung der KPIs der RoamApp	12
Abbildung 7 Ausschnitt Mindmap aus Workshop mit Roam Network	14
Abbildung 8 Field Test Mode - Dashboard	18
Abbildung 9 Field Test Mode - LTE Serving Cell Meas	18
Abbildung 10 Field Test Mode - All Metrics	18
Abbildung 11 Persona Zielgruppe 1	22
Abbildung 12 Pointsscreen mit Badges und Glücksrad	23
Abbildung 13 Leaderboard mit Friends und Worldwide Tab	23
Abbildung 14 Provider Vorschläge Location	24
Abbildung 15 Provider Filter Kontext	24
Abbildung 16 Provider Vorschläge Details	24
Abbildung 17 Roamgochi Full	25
Abbildung 18 Mini-Game	25
Abbildung 19 Roamgochi Hungry	25
Abbildung 20 Roamcoins Send Screen	26
Abbildung 21 Overlay Erklärung Daily Points	27
Abbildung 22 Login 1	27
Abbildung 23 Push-Notifikationen Einstellungen	28
Abbildung 24 Design Roamcoins	28
Abbildung 25 Log-out Bestätigung	29
Abbildung 26 Referral Screen	29
Abbildung 27 CryptoViewModel MutableLiveData	32
Abbildung 28 CryptoFragment Observer ViewModel	33
Abbildung 29 CryptoFragment Erklärungsanzeige	34
Abbildung 30 Klassendiagramm Erfassen von Providermessungen	36
Abbildung 31 Funktion für die Berechnung der Signalbewertung	37
Abbildung 32 Screen Detailed Filter	39
Abbildung 33 Erfassen von Detailangaben	39
Abbildung 34 Klassendiagramm State Pattern Roamgochi	40
Abbildung 35 Umfrage RoamApp Bewertung	45
Abbildung 36 Umfrage Roamgochi Bewertung	45
Abbildung 37 Umfrage Provider Bewertung	46
Abbildung 38 Umfrage Bewertung Provider, Roamgochi, RoamApp	46

## Tabellenverzeichnis

### **Tabellenverzeichnis**

Tabelle 1 Kategorisierung und Referenzierung der identifizierten Studien	3
Tabelle 2 Gewichtung der Messgrößen für die Berechnung der Signalstärkebewertung	30
Tabelle 3 Usability-Testing Resultate	42
Tabelle 4 Provider Feature Usability-Testing	43
Tabelle 5 A/B Testing Interviews	49

## 1 Einleitung

### 1 Einleitung

Das Ziel der Bachelorarbeit bestand darin, die Fragestellungen zu beantworten und infolgedessen eine Prototypen-Applikation für Roam Network zu implementieren. Diese Applikation soll zur Messung von Netzwerkdaten dienen und ist speziell für Nutzer mit technischem Verständnis geeignet. Roam Network hat bereits eine erste Version der Anwendung für das Android-Betriebssystem veröffentlicht im Februar 2024.

#### 1.1 Problemstellung und Ziele

Die kürzlich veröffentlichte mobile Applikation "RoamApp" befindet sich in der Anfangsphase.<sup>1</sup> Das Hauptziel von Roam Network besteht darin, möglichst viele Downloads zu generieren und aktive Nutzer zu gewinnen. Dieses Projekt beschäftigt sich damit Wege zu finden und umzusetzen, um dieses Ziel zu ermöglichen. Dabei geht es um die Implementation von neuen Features, Designs und um die Möglichkeit die Applikation in iOS umzusetzen.

Die Fragestellungen lauten wie folgt:

RQ1. Was sind die effektivsten Strategien, um die Anzahl der aktiven Nutzer der aktuellen RoamApp zu steigern?

Um einen Überblick zu schaffen, wurde zuerst eine Marktanalyse durchgeführt. Dabei wurden die Themen mit dem grössten Einfluss auf die Steigerung der Benutzerzahlen identifiziert. In Zusammenarbeit mit den Kunden wurden Ideen generiert und die Projektgrenzen seitens der Studierenden definiert. Durch das Analysieren von KPIs der RoamApp, konnten die Bereiche identifiziert werden, die unterdurchschnittliche Werte aufweisen. Diese Analyse half dabei, Schwerpunkte zu bestimmen, die das grösste Verbesserungspotential haben.

Zur Beantwortung dieser Fragestellung wurden verschiedene Ansätze analysiert. Dabei wurden Studien zu den Themen UX (User Experience), UI (User Interface), Gamifizierung, Push-Benachrichtigungen analysiert. Des Weiteren wurden Usability-Testings, A/B-Testings und eine Umfrage durchgeführt, um die langfristige Bindung der Nutzer<sup>2</sup> an die Applikation zu evaluieren und um Wege zu finden, die Benutzerbindung zu erhöhen. Zusätzlich wurde ein Workshop mit dem Kunden durchgeführt, um spezifische Anforderungen und Strategien zu planen und sicherzustellen, dass diese mit den Zielen von Roam Network übereinstimmen.

RQ2. Was sind die effektivsten UI-Anpassungen, welche die User Experience verbessern?

Für die allgemeine Verbesserung der UX wurden Recherchen durchgeführt, wobei Studien und öffentliche Artikel berücksichtigt wurden. Danach konnten die konkreten Konzepte auf

---

<sup>1</sup> „RoamApp – Apps bei Google Play“. Zugegriffen: 26. Juli 2024. [Online]. Verfügbar unter: <https://play.google.com/store/apps/details?id=xyz.roamnetwork.roamapp&hl=de>

<sup>2</sup> Einfachheitshalber wird im Text „der Nutzer“ verwendet. Gemeint sind jedoch alle Geschlechter.

## 1 Einleitung

die RoamApp angewendet, mit Usability-Testings verglichen und betreffend ROI eingeordnet werden. Die Umsetzung wurde regelmässig mit dem Kunden und Betreuern besprochen. Da dieses Projekt nicht veröffentlicht werden konnte, konnte sich nur auf die Resultate dieser Tests bezogen werden.

Zur Beantwortung dieser Fragestellung wurden, basierend auf dem Vorwissen aus den Vorlesungen, Studien zum Thema UX und UI analysiert und ein Usability-Testing mit mehreren Personen durchgeführt. Die Resultate und Konzepte, die auf die RoamApp anwendbar sind, wurden entsprechend umgesetzt.

RQ3. Ist eine iOS-Version mit den vorhandenen Einschränkungen ausreichend benutzerfreundlich, um die Nutzeranzahl zu steigern?

Zur Beantwortung dieser Frage wurde eine Literatur Recherche durchgeführt Da keine Studien zu diesem Thema gefunden werden konnten, wurden diverse Webseiten und Blogs einbezogen, um relevante Informationen zu sammeln. Zusätzlich wurde Absprache mit dem Kunden abgehalten, welche bereits Erfahrungen mit den möglichen Problemen für eine Umsetzung der Applikation in iOS gemacht haben.

Im Verlauf des Projektes (ca. Juni) kristallisierte sich die übergeordnete Fragestellung: «Hat Gamifizierung oder der praktische Nutzen einen grösseren Einfluss auf die Motivation der Nutzer, eine App zu nutzen?». Der Begriff Gamifizierung bezieht sich hierbei auf das Roamgochi Spiel und der praktische Nutzen auf das Providervergleichs Feature, welche beide im Verlauf des Projektes umgesetzt wurden. Das Umwandeln der gesammelten Punkte in Roamcoins wurde als Proof of Concept umgesetzt und wird auch als praktischen Nutzen betrachtet. Dies soll die Benutzer, neben den bisherigen Features, dazu bringen die Applikation langfristig zu nutzen. Diese Fragestellung wurde mit A/B-Testings beantwortet.

### 1.2 Methoden und Materialien

Für die Literaturrecherche wurden verschiedene wissenschaftliche Portale und Datenbanken verwendet, um relevante Quellen und Studien zu identifizieren. Die folgenden Portale wurden dabei einbezogen:

- Google Scholar
- Rechercheportal Swiscovery
- SpringerLink
- Google Websuche

Es wurden Recherchen im Umfeld mobile Applikationen mit folgenden Schlagwörtern durchgeführt:

- Akquisition von Benutzern
- Key Performance Indikatoren
- Benutzerbindung
- Transparenz
- Notifikationen



## 1 Einleitung

- Gamifizierung
- Onboarding

Nachfolgend wird eine Übersicht der Studien in den jeweiligen Kategorien gegeben, unterteilt in die Gesamtanzahl der gefundenen und der in dieser Arbeit referenzierten Studien:

Tabelle 1 Kategorisierung und Referenzierung der identifizierten Studien

Kategorie	Gefundene Studien	Referenzierte Studien
Usability & UX	8	4
Gamifizierung	3	2
Benutzerbindung	1	1
Datenschutz	2	0
Gesamt	14	7

### 1.3 Struktur dieses Dokumentes

Im folgenden Abschnitt wird die Leserführung und der Aufbau des restlichen Dokuments erläutert, um einen Überblick über die weiteren Kapitel und deren Inhalt zu geben.

- Kapitel 2 widmet sich einer detaillierten Darstellung der Problemstellung, in der die Motivation hinter der Entwicklung der Applikation erläutert wird. Es werden relevante Herausforderungen und Bedürfnisse identifiziert sowie der aktuelle Stand der Technik analysiert.
- Kapitel 3 beschreibt den theoretischen Lösungsansatz und die Methodik für die Erweiterung der Applikation. Es werden die verwendeten Technologien, Algorithmen und Architekturkonzepte erläutert.
- Kapitel 4 stellt die konkrete Umsetzung, der in Kapitel 3 erwähnten Lösungen.
- Kapitel 5 befasst sich mit der Validierung und Reflexion der Umsetzungen und beinhaltet die Ergebnisse dieser.
- Kapitel 6 fasst die wichtigsten Ergebnisse zusammen, stellt sicher, dass die Projektziele beantwortet wurden, und listet Aufgaben für zukünftige Arbeiten auf.
- Kapitel 7 enthält die Schlussbemerkungen des Projekts mit einem Ausblick und die gewonnenen Erkenntnisse.

### 1.4 Projektmanagement

Dieses Projekt wird nach der agilen Projektmethodik durchgeführt. Dieser Ansatz ermöglicht es, neuer Erkenntnisse sowie Umsetzungen regelmässig vom Kunden absegnen zu lassen und so zu sicherstellen, dass das Projekt sich in die korrekte Richtung entwickelt. Dies soll spätestens nach jedem Sprint durchgeführt werden, jedoch soll bei neuen Erkenntnissen oder Problemen der Kontakt mit dem Kunden auch während der Sprints gesucht werden. Die Sprints werden zwischen drei und vier Wochen dauern. Sprint 2 umfasst die Projektwoche und ist deutlich umfangreicher, so dass diesem Sprint komplexere Aufgaben zugewiesen werden können.

Ergebnisse aus Nachforschungen sollen immer mit dem Kunden besprochen werden, damit dieser

## 1 Einleitung

die bestmögliche Beratung erhält. Umsetzungen von neuen Features müssen vorher zwingend mit dem Kunden besprochen. Dies soll sicherstellen, dass die Prioritäten immer synchronisiert sind sowie um nicht parallel am selben zu arbeiten.

Während des Inception Sprints, der vor Sprint 1 stattfindet, wird die Projektvision definiert, die Organisation festgelegt und eine Analyse des aktuellen Zustands durchgeführt. Während des ersten Sprints werden die Anforderungen spezifiziert, bisherige Erkenntnisse gesammelt und Klarstellungen vorgenommen.

Sobald möglich sollen Usability-Tests mit potenziellen Endbenutzern durchgeführt werden. Die Kommunikation im Team findet über MS Teams statt. Mindestens einmal pro Woche wird ein StandUp-Call durchgeführt, um einen regelmässigen Austausch zu gewährleisten. Die primäre Kommunikation mit Stakeholdern ausserhalb des Teams erfolgt hauptsächlich über Slack. Berichte für alle Sitzungen werden in MS Teams erstellt und gespeichert

Die Zeiterfassung wurde in folgende Kategorien aufgeteilt:

- Organisation
- Research
- Coding
- Thesis
- Prototype Testing

Es wurden insgesamt über 890 Stunden für das Projekt aufgewendet.

### 2 Grundlagen und Hintergründe

Diese Arbeit basiert sich auf die Applikation «RoamApp», welche von Roam Network im Februar 2024 veröffentlicht wurde. Es handelt sich dabei um eine Applikation, welche im Hintergrund Messwerte der Mobilfunkverbindung misst und diese an Roam Network sendet. Diese werden zur Optimierung von Mobilfunknetzen verarbeitet bzw. weitergegeben. Anders als einzelne Mobilfunkunternehmen kann Roam Network durch das Sammeln von Daten von einzelnen Benutzern ein Gesamtbild der Netzwerkstärken und -qualität erfassen. Die Daten sind dabei sehr divers und decken verschiedenen Orte, Geräte und Provider ab. Dies ermöglicht Roam Network eine umfassende Analyse aller Mobilfunkanbieter, um das mobile Netzwerk zu optimieren.

Als Gegenleistung für die geteilten Daten erhält der Benutzer Punkte. Wie die Anzahl der Punkte bestimmt werden, ist absichtlich nicht öffentlich, da dies ausgenutzt werden könnte. Die erhaltenen Punkte sollen in Zukunft in eine Kryptowährung umgewandelt werden können, um die Benutzer zu entschädigen. Dieser Coin existiert momentan noch nicht.

Die Coin Economy und das Entlohnungssystem ist kein Teil dieses Projekts und der Thesis. PR und Marketing bezüglich der Applikation werden teilweise gestreift, ist jedoch auch ein Bereich, welcher nicht in diesem Projekt behandelt wird. Der Fokus liegt auf der Verbesserung der Benutzererfahrung und die Erhöhung der Benutzerbindung mit potenziellen Features welche für die Endbenutzer nützlich und/oder motivierend sind.

Die Applikation sowie die Firma Roam Network ist im Kryptowährung-Bereich anzusiedeln. In einer späteren Phase will die Firma Roam Network einen Coin erstellen, um den Benutzern sowie Interessierten eine Möglichkeit zu bieten, Investitionen in das Unternehmen oder in das System zu tätigen. Diese Coins sollen in Zukunft einen Wert haben und in das Belohnungssystem integriert werden. Dieser Bereich wird nicht in diesem Projekt behandelt.

#### 2.1 Roam Network

Roam Network ist ein Team erfahrener Softwareentwickler, das mit der RoamApp die Telekommunikationsbranche durch Blockchain-Technologie revolutionieren möchte. Ihr Ziel ist es, das erste dezentrale, drahtlose Netzwerk zu schaffen, das die traditionelle Telekommunikationsinfrastruktur optimiert und weltweit zuverlässige und sichere Kommunikation ermöglicht. Die RoamApp wurde von Michail Angelov, dem CEO, ehemaligen Nokia-Mitarbeiter, und Hauptkunden dieses Projekts, mitgegründet.

Die Telekommunikationsbranche steht vor grossen Herausforderungen. Die Nachfrage nach Diensten mit hoher Bandbreite steigt an [17], [18]. Laut dem GSMA Mobile Economy Report 2024: "Despite talk of a slowdown in growth, monthly global mobile data traffic per connection surged from 10.2 GB in 2022 to 12.8 GB in 2023, marking the largest absolute increase since data tracking commenced in 2016" [18]. Um die Netzkapazität und -abdeckung zu verbessern, sind erhebliche Investitionen in die Infrastruktur erforderlich. Darüber hinaus bleibt die Überbrückung der digitalen Kluft durch die Bereitstellung von Internetzugang für unterversorgte Gemeinden eine drängende Aufgabe.

## 2 Grundlagen und Hintergründe

Mitten in diesem komplexen Umfeld tritt Roam Network als unabhängige Kraft hervor und adressiert den kritischen Bedarf an Echtzeit- und Echtmessdaten zu den Netzwerkstärken verschiedener Anbieter. Roam Network nutzt die Community-Beteiligung und ermöglicht eine kollaborative Erfassung von Mobilfunknetzdaten direkt von den Nutzern und kann Daten aus verschiedenen Netzwerken und Regionen sammeln und somit eine umfassendere und objektivere Datenbasis bieten. Eine unabhängige Firma kann Daten aus verschiedenen Netzwerken vergleichen und analysieren, was es den Anbietern ermöglicht, ihre Leistungen objektiv zu bewerten und zu verbessern. Diese Vergleichbarkeit wäre schwer zu erreichen, wenn jeder Anbieter nur seine eigenen Daten sammelt. Die Crowdsourcing-Daten versetzen Netzbetreiber in die Lage, fundierte Entscheidungen über die Netzoptimierung zu treffen, was zu einer verbesserten Netzleistung und Benutzererfahrung führt.

In der aktuellen Phase der Firma und der Applikation geht es darum, die Benutzeranzahl der App zu erhöhen, um bei der Einführung der Coins eine Basis an Benutzern zu haben.

### 2.2 RoamApp

Die RoamApp ist eine reine Android Applikation, welche im Google Playstore kostenlos erhältlich ist und am Anfang dieses Projektes ca. 500 Downloads aufwies.

Die Applikation beinhaltete folgende Features bei der Übernahme:

- Ansehen der aktuelle Verbindungsqualität.
- Sammeln von Punkten.
- Ansehen der Historie der erhaltenen Punkte.
- Ansehen von Informationen der verbundenen Mobilfunkzelle.
- Möglichkeit jemanden auf die App zu verweisen mit einem Referral Code.

Die Screens dieser Features sind in **Fehler! Verweisquelle konnte nicht gefunden werden.** dargestellt.

## 2 Grundlagen und Hintergründe

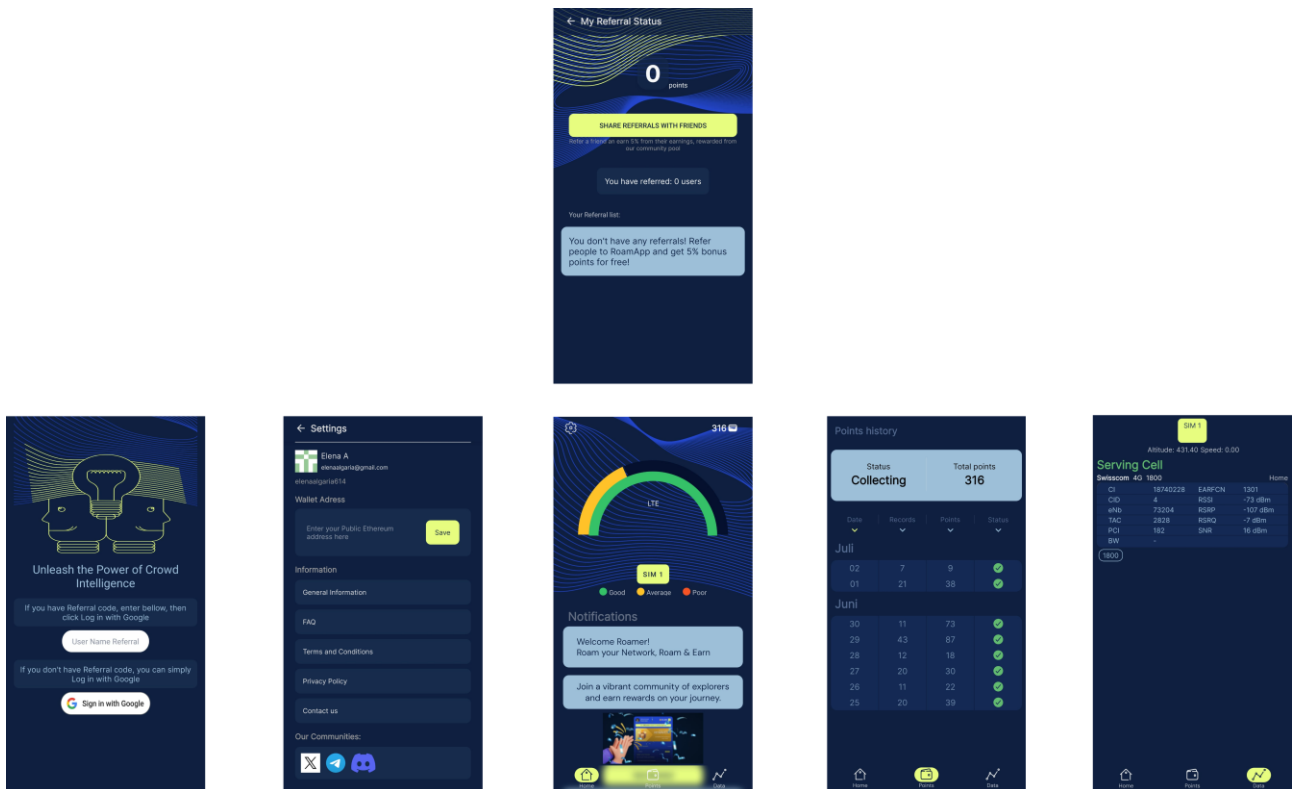


Abbildung 1 RoamApp Stand Juli (Version: 1.0.4)

### 2.2.1 Funktion

Die Hauptfunktion der Applikation für Endbenutzer besteht darin, Punkte für durchgeführte Messungen zu sammeln. Die Menge der erhaltenen Punkte variiert. Die Messungen dienen der Bewertung der Signalstärke und werden verwendet, um die Reichweite und Zuverlässigkeit des Mobilfunknetzes zu erfassen.

Es ist zu beachten, dass es sich dabei nicht um eine Messung der Datenübertragungsrate oder der Geschwindigkeit einer Internetverbindung handelt, welche in Mbps (Megabit pro Sekunde) angegeben wird. Dies wird zwar in der App angezeigt, jedoch nicht als Messdaten weitergegeben und geben an, wie viele Megabit an Daten pro Sekunde übertragen werden können. Je höher der Mbps-Wert, desto schneller ist die Datenübertragung. Beide Messungen sind wichtig, um die Qualität und Leistungsfähigkeit einer Mobilfunkverbindung zu beurteilen, jedoch werden unterschiedliche Aspekte gemessen.

## 2 Grundlagen und Hintergründe

### 2.2.2 Technische Analyse

In diesem Kapitel werden der verwendete Tech-Stack sowie die Code-Struktur der ursprünglichen Applikation erläutert, die weiterhin verwendet und weiterentwickelt wurden.

#### Frontend

- Kotlin

Kotlin ist die offizielle Programmiersprache für Android-Entwicklung, entwickelt von JetBrains und unterstützt von Google. Sie bietet eine moderne Syntax und ermöglicht die Erstellung effizienter Anwendungen. Kotlin vereinfacht die Entwicklung durch weniger Boilerplate-Code und eine bessere Null-Sicherheitsprüfung, was zu einer stabileren und fehlerfreien App führt.

- Injektion Model Hilt

Hilt ist ein Dependency Injection-Framework für Android [1], das auf Dagger basiert. Es vereinfacht die Verwaltung von Abhängigkeiten und ermöglicht eine klare Trennung von Verantwortlichkeiten in der App. Somit können Abhängigkeiten in verschiedenen Teilen der App wiederverwendet werden.

- MVVM-Struktur

Die RoamApp verwendet das MVVM (Model-View-ViewModel) -Architekturmodell [2] welche in Abbildung 2 zu sehen ist. Diese Struktur erleichtert die Trennung von Logik und UI-Komponenten und ermöglicht eine bessere Handhabung des Lebenszyklus der einzelnen Klassen.

Beispielsweise wird ein Fragment in Verbindung mit einem ViewModel verwendet. Das Fragment ist für die Darstellung der UI zuständig, während das ViewModel die Geschäftslogik und die Datenverarbeitung übernimmt. Durch die Verwendung von LiveData in Kombination mit ViewModel wird sichergestellt, dass die UI automatisch aktualisiert wird, wenn sich die zugrunde liegenden Daten ändern.

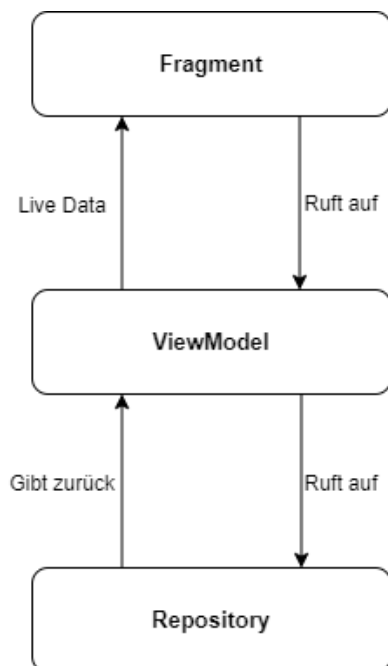


Abbildung 2 MVVM Struktur

## 2 Grundlagen und Hintergründe

- Activity

Eine Activity stellt in der Regel einen einzelnen Bildschirm in einer Android-Anwendung dar. Sie dient als Einstiegspunkt für die Interaktion des Nutzers mit der App und kann andere UI-Elemente starten und verwalten. Ein typisches Beispiel ist die MainActivity. Activities können neue Bildschirme öffnen, Daten zwischen verschiedenen Bildschirmen senden und Hintergrundservices starten. Sie spielen eine zentrale Rolle im Lebenszyklus einer Android-App, indem sie die Zustandsänderungen und die Benutzerinteraktionen verwalten. Jede Activity durchläuft mehrere Zustände, darunter onCreate, onStart, onResume, onPause, onStop, und onDestroy, die man überschreiben kann, um spezifische Verhaltensweisen zu implementieren. Diese Phasen ermöglichen es, den Status der Activity zu steuern und auf Benutzerinteraktionen oder Systemereignisse zu reagieren.

- Fragment

Fragments sind modulare Teile einer Benutzeroberfläche, die innerhalb einer Activity eingebettet werden können. Sie ermöglichen eine flexible und dynamische Gestaltung von UIs, indem sie die Oberfläche einer Activity in kleinere, wiederverwendbare Komponenten aufteilen. Fragments können verwendet werden, um zusätzliche Funktionen zu einer bestehenden Activity hinzuzufügen oder um eine gemeinsame Benutzeroberfläche für mehrere Activities bereitzustellen. Es verwendet Methoden wie onCreateView, onViewCreated, onStart, onResume, onPause, onStop, und onDestroyView, die Kontrolle über die Erstellung und Verwaltung der UI ermöglichen.

- SharedPreferences

SharedPreferences [3] ist eine Schnittstelle in Android, die es ermöglicht, einfache Daten als Schlüssel-Wert-Paare lokal auf einem Gerät zu speichern. Diese Daten werden in einer XML-Datei gespeichert und sind für die gesamte App zugänglich. SharedPreferences ist besonders nützlich, um Benutzereinstellungen, Konfigurationen oder andere kleine Datenmengen, die zwischen verschiedenen Sitzungen persistent gespeichert werden müssen, zu speichern.

In diesem Projekt wird SharedPreferences über die Wrapper-Klasse *sharedPrefs* verwendet.

### Backend

- PostgreSQL, BigQuery

BigQuery ist ein vollständig verwalteter, serverloser Data-Warehouse-Dienst von Google Cloud, der speziell für die Verarbeitung und Analyse grosser Datenmengen entwickelt wurde. Es bietet Vorteile wie Hochskalierbarkeit, Echtzeit-Analysen, einfache Integration mit anderen Google Diensten

- Benutzung von Google Accounts und Google Analyse Tools

Durch die Nutzung von Google Accounts können weitere Google-Dienste wie Google Analytics nahtlos integriert werden, was eine umfassende Analyse des Nutzerverhaltens und gezielte Marketingmassnahmen ermöglicht.

- Node Express Server mit JavaScript

- Docker

### 2.1 Anforderungen und Umgebung

In der folgenden Abbildung 3 werden Stakeholder mit direktem und indirektem Systemkontakt als Onion Modell dargestellt. In direktem Systemkontakt steht die Benutzergruppe Benutzer, welche

## 2 Grundlagen und Hintergründe

die RoamApp benutzt, um Messungen durchzuführen und Punkte erhält. Administratoren haben die Möglichkeit, die Applikation und die darin enthaltenen Benutzerkonten zu verwalten. In indirektem Systemkontakt stehen die Netzbetreiber von Mobilfunknetzen, welche die Messdaten von Roam Network kauft und diese verwerten, um ein flächendeckendes Netz aufzubauen oder das bestehende Netz zu optimieren.

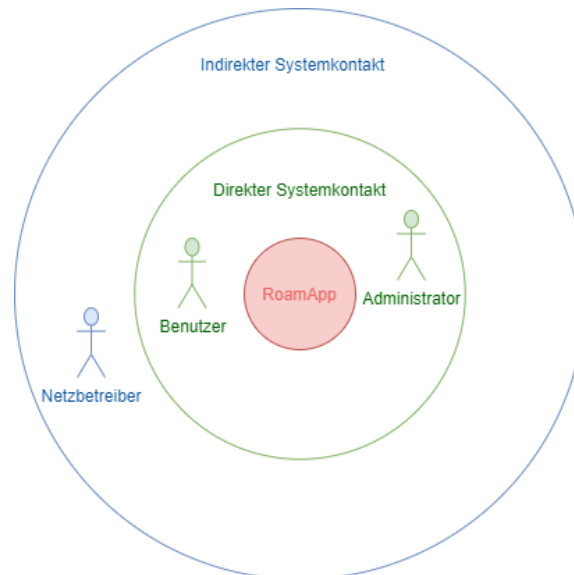


Abbildung 3 Onion Modell der Stakeholder

### 2.2 SOTA

In diesem Kapitel wird der aktuelle Stand der Technik (State of the Art) im Bereich mobiler Applikationen analysiert, die Mechanismen enthalten welche ähnlich zur RoamApp sind. Ziel ist es, einen umfassenden Überblick über bestehende Lösungen auf dem Markt zu geben und die RoamApp im Kontext dieser Applikationen zu positionieren.

#### 2.2.1 Marktanalyse

Die Marktanalyse<sup>3</sup> des aktuellen Apps stellt ein Schritt bei der Bewertung des Wettbewerbsumfelds und der Positionierung im Markt dar. Sie liefert Einblicke in den aktuellen Stand des Marktes, die Wettbewerbslandschaft und die Kundenbedürfnisse. Durch eine Marktanalyse können Stärken, Schwächen, Chancen und Risiken identifiziert werden, um die Wettbewerbsfähigkeit der App zu verbessern.

In dieser Analyse werden verschiedene Aspekte des Marktes betrachtet, darunter die bestehende Konkurrenz, deren Leistungen, Funktionen und Positionierung. Es werden potenzielle Zielgruppen und Kundenpräferenzen erfasst, um die Marktnachfrage besser zu verstehen.

---

<sup>3</sup> Die Zusammenfassung der Marktanalyse ist als Excel File abgelegt. Link: [Marktanalyse.xlsx](#)



## 2 Grundlagen und Hintergründe

### Untersuchung ähnlicher Applikationen auf dem Markt

Es wurden Zehn vergleichbare Applikationen untersucht. Die betrachteten Applikationen wurden durch Wissensaustausch mit Roam Network, Internetsuchen über Google und Recherchen im Google Play Store identifiziert.

Folgende Schlagwörter wurden für die Suche im Internet und im Play Store benutzt:

- Connection Test App
- Signal Strength Test App
- Move-to-earn App
- Speed Test App
- Mobile Network Coverage App

Die untersuchten Applikationen können bezüglich deren Hauptfunktionen kategorisiert werden. Folgende Kategorien wurden definiert:

- Geschwindigkeitsmessung
- Signalstärkemessung
- Erhalt von Coins

Teilweise gehören Applikation zu mehreren Kategorien gleichzeitig wie in Abbildung 44 **Fehler! Verweisquelle konnte nicht gefunden werden.** Fehler! Verweisquelle konnte nicht gefunden werden. zu sehen ist.

<b>Nr.</b>	<b>Name</b>	<b>Downloads</b>	<b>Kategorie</b>	<b>Store</b>
1	nPerf	5 Mio.+	Geschwindigkeitsmessung	Android & iOS
2	Silencio	100K+	Erhalt von Tokens	Android & iOS
3	OpenSignal	10M+	Geschwindigkeitsmessung, Signalstärkemessung	Android & iOS
4	Internet Speed Test by Meteor	5M+	Geschwindigkeitsmessung	Android & iOS
5	Speedtest by Ookla	100M+	Geschwindigkeitsmessung	Android & Amazon
6	weBoost	10.000+	Signalstärkemessung	Android & iOS
7	Network Cell Info Lite	10M+	Signalstärkemessung	Android
8	Signal Strength	1M+	Signalstärkemessung	Android
9	LTE Discovery	1M+	Geschwindigkeitsmessung, Signalstärkemessung	Android
10	sweatcoin	100M+	Erhalt von Tokens	Android & iOS
11	Roam App	500	Erhalten von Tokens, Signalstärkemessung	Android

Abbildung 4 Kategorisierung von ähnlichen Produkten auf dem Markt

## 2 Grundlagen und Hintergründe

Nr.	Features		Target Group
1	Bitrate speed, latency, browsing speed and video streaming quality	Coverage & Speed Map	Technically savvy individuals
2	Accumulate tokens	Noise exposure map	Noise polluted person
3	Best Network suggestion	Coverage & Speed Map	Not technically savvy individuals
4	Shows how smoothly different App as TikTok, Netflix and so on are running with actual network.		Not technically savvy individuals
5	Monitor your popular apps	Coverage & Speed Map	Customers of Ookla
6	Compass nearest cell tower		Technically savvy individuals
7	Map with cell towers		Technically savvy individuals
8	See the measurements constantly		Not technically savvy individuals
9	Overlay ability on the top of s	Identifies bandwidth for different internet suppliers in USA	Technically savvy individuals
10	Earn coins while walking		Fitness enthusiasts, crypto enthusiasts
11	Accumulate Points		technically savvy individuals

Abbildung 5 Vergleich der Features der ähnlichen Apps

In Abbildung 5 werden die Features der analysierten Apps verglichen. Die Einfärbungen stellen ähnliche Features dar. Man sieht, dass die Zielgruppe hauptsächlich unterschieden werden kann aus technisch versierten Personen und technisch nicht versierten Personen, die jeweils für etwa die Hälfte aller Downloads verantwortlich sind.

Ein Alleinstellungsmerkmal für die RoamApp kann sein, mit einer Kryptowährung erhalten zu können. Die einzigen Apps, die diese Funktionen beinhalten, sind Silencio [4], bei der es um die Messung von Lautstärke geht, und Sweatcoin [5], bei der man für Bewegung belohnt wird. Im Bereich Signalmessung werden derzeit keine solchen Belohnungen vergeben.

### Analyse der aktuellen App anhand von KPIs

Die Verwendung von Key Performance Indicators (KPI) [6] ist ein effektiver Weg, um die Leistung und den Erfolg einer App zu messen. KPIs liefern klare und messbare Werte. Durch die Analyse von KPIs können verschiedene Aspekte der App-Leistung genau überwacht und bewertet werden. Wir wollen damit Schwachstellen aufdecken und besser verstehen, wo Verbesserungspotential existiert. Die Zahlen wurden aus Google Firebase entnommen.

<b>RoamApp</b>	29.04.24
Mobile app net promoter score (NPS)	5/5
App downloads	500+
Monthly active users (MAU)	844
Daily active users (DAU)	230
Stickiness	0.272511848
Daily sessions per daily active user	1.15
Average session length	2 minutes
Average retention rate (ARR)	3.40%
Average churn rate (ACR)	96.60%

Abbildung 6 Auswertung der KPIs der RoamApp

Die Auswertung der KPIs in Abbildung 6 zeigen, dass die durchschnittliche Retention-rate bei 3.4% ist. Dies deutet darauf hin, dass die Benutzer wenig Motivation haben die Applikation nach der Installation weiter zu benutzen. Die durchschnittliche Zeit, die die Benutzer in der App verbringen ist mit 2 Minuten ebenfalls sehr tief. Dies zeigt, dass die Benutzer wenig an die Applikation gebunden

## 2 Grundlagen und Hintergründe

sind. Um die Benutzerbindung und die Nutzung der App zu verbessern, können Massnahmen umgesetzt werden:

- **Verbesserung der Benutzererfahrung:** Intuitive Benutzeroberfläche: Eine benutzerfreundliche und ansprechende Gestaltung der Benutzeroberfläche kann die Nutzererfahrung verbessern und die Verweildauer erhöhen.
- **Onboarding-Prozess:** Ein effektiver Onboarding-Prozess [7], [8], der die Funktionen und Vorteile der App klar vermittelt, kann dazu beitragen, neue Nutzer besser zu integrieren.
- **Belohnungssysteme:** Durch die Implementierung von Gamification-Elementen [9] wie Belohnungen und Fortschrittsanzeigen können Nutzer motiviert werden, die App häufiger und länger zu nutzen.
- **Herausforderungen und Wettbewerbe:** Regelmässige Herausforderungen oder Wettbewerbe können den Spassfaktor erhöhen und die Nutzerbindung stärken.
- **Personalisierte Inhalte:** Die Möglichkeit, Inhalte nach den individuellen Vorlieben der Nutzer anzupassen, kann die Relevanz der App erhöhen und die Nutzung intensivieren.
- **Personalisierte Benachrichtigungen:** Zielgerichtete Benachrichtigungen, die auf das Verhalten und die Interessen der Nutzer abgestimmt sind, können die Nutzeraktivität steigern.

### Rückschlüsse

Während der Vergleich mit ähnlichen Applikationen zeigt, dass die Nachfrage nach einem praktischen Nutzen vorhanden ist, kann ein Alleinstellungsmerkmal der RoamApp darin bestehen, dass Nutzer Punkte/Coins erhalten können. Eine Kombination aus beidem kann zu einem Alleinstellungsmerkmal führen, das sich positiv auf die Downloadzahlen der RoamApp auswirkt. Zudem sind nur wenige Gamification-Elemente in den Konkurrenzprodukten vorhanden.

Um die geringe Benutzerbindung zu erhöhen, können diverse bereits erwähnte Massnahmen ergriffen werden. Grundsätzlich kann durch die Kombination aus Gamification-Elementen und Alleinstellungsmerkmalen die Akzeptanz der App erhöht werden. Dafür sollte zunächst eine solide Grundlage an Benutzerbindungselementen geschaffen werden, um anschliessend Features zu implementieren, die die RoamApp einzigartig machen.

### 2.2.2 Workshop

Ein gemeinsamer Workshop (siehe Figma<sup>4</sup>) mit den Kunden wurde am 19. März 2024 durchgeführt. Der Workshop verfolgte mehrere Ziele. Zunächst wurde der aktuelle Stand der RoamApp erfasst und es wurde ermittelt, welche Features seitens Roam Network geplant sind. Dies half allen Beteiligten, ein klares Verständnis der bestehenden Bedingungen und zukünftigen Entwicklungen zu gewinnen. Darüber hinaus wurden die Projektgrenzen abgesteckt, indem die Bereiche definiert wurden, in denen die Studierenden tätig werden können.

---

<sup>4</sup> Für alle Elemente wurde ein Figma File verwendet. Dieses ist chronologisch von links nach rechts aufgebaut und beinhaltet den Workshop, Mash-Up Personas, alle Hi-Fis und das Poster. Link: <https://www.figma.com/design/M4vIGVH5N9ZEMzyH11uvxD/RoamApp?node-id=0-1&t=Ah1ZZXUBMqT4z6WZ-1>

## 2 Grundlagen und Hintergründe

Ein weiterer wichtiger Aspekt des Workshops war die Sammlung von Ideen für das Projekt. Hierbei konnten kreative Ansätze und Verbesserungsmöglichkeiten identifiziert werden. Gleichzeitig wurde ein tieferes Verständnis der Kundenbedürfnisse erlangt, um sicherzustellen, dass die Projektarbeit darauf abgestimmt ist. Schliesslich diente der Workshop dazu, konkrete Ziele für das Projekt zu definieren, sodass alle Beteiligten auf ein gemeinsames Ziel hinarbeiten konnten.

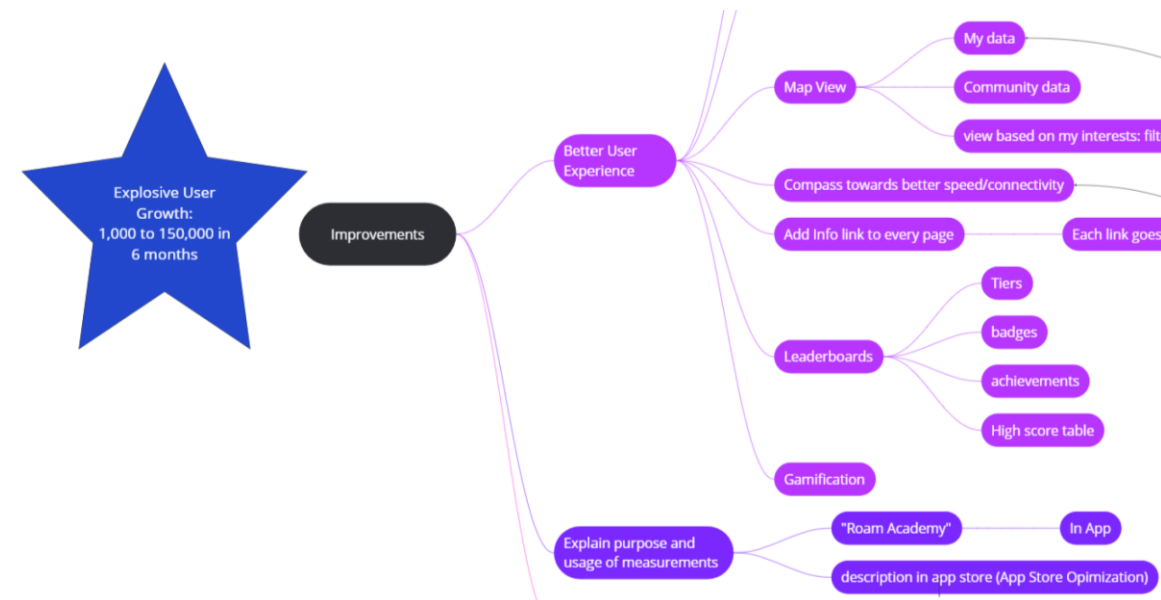


Abbildung 7 Ausschnitt Mindmap aus Workshop mit Roam Network

Für die Durchführung des Workshops wurde ein Mash-Up mit vier verschiedenen Aussagen geplant, darunter zum Beispiel „A rewarding system that benefits and motivates the user to download the app“. Die gesamte Applikation wurde ausführlich besprochen. Jeder Teilnehmer hatte die Gelegenheit, seine Meinungen sowie Verbesserungsvorschläge einzubringen, die dann in einer Mindmap Abbildung 7 erfasst wurden. Insbesondere wurden Ideen zur Verbesserung der UX und zur Steigerung des Nutzens der App notiert, wobei Vergleiche mit anderen Apps gezogen wurden.

### 2.2.3 Literatur und Websuche

#### Benutzerbindung erhöhen

Um die Benutzerbindung sowie die Identifikation eines Benutzers mit einer Marke zu erhöhen, können Gamifizierung Elemente implementiert werden. Die Ergebnisse der Studie «Enhancing user engagement: The role of gamification in mobile apps» [10], deuten darauf hin, dass die Interaktion der Nutzer mit leistungs- und fortschrittsorientierten Spielelementen die Nutzerbindung sowohl direkt als auch indirekt positiv beeinflusst. Dies geschieht durch die Erfüllung der Bedürfnisse nach Kompetenz, Autonomie und sozialer Eingebundenheit. Darüber hinaus entwickeln engagierte Nutzer stärkere Absichten, die gamifizierte mobile App weiterhin zu nutzen, sie anderen zu empfehlen, positiv über sie zu sprechen und sie eher positiv zu bewerten.

## 2 Grundlagen und Hintergründe

### User Experience (UX)

Die RoamApp wurde analysiert, wobei festgestellt wurde, dass die Applikation eine verbesserte User Experience und ein einheitlicheres User Interface benötigt. Um dies umzusetzen wurden verschiedene Studien analysiert mit dem Wissen aus diversen Modulen.

### Effekte von UI und UX auf Downloadzahlen [11]

Die Benutzeroberfläche (UI) und die Benutzererfahrung (UX) sind die wichtigsten Faktoren in einer Anwendung. Das liegt daran, dass das UI und die UX die Verbindung zwischen dem Benutzer und dem System darstellen. Im Bereich UX sind die Funktionen und die Brauchbarkeit einer App vor allem wichtig für die Nutzer. Generell gilt bei der Benutzung einer App, dass die Aufgaben schnell lösbar, und verständlich sein sollen. Ebenfalls spielt ein interessantes und übersichtliches UI eine grosse Rolle. Wenn die beiden Faktoren UI und UX beide an die Zielgruppe angepasst werden, hat eine App tendenziell mehr Nutzer.

### Gründe, warum Nutzer eine Applikation wählen [12]

In dieser Studie sind verschiedene Apps getestet worden und verschiedene Studien miteinander verglichen. Dabei ist aufgefallen, dass vor allem soziale Applikation am besten abgeschnitten haben. Sie waren am beliebtesten, wurden am längsten genutzt und wurden von den meisten nicht mehr gelöscht. Um dies zu erreichen, wurden festgestellt, dass es sechs wichtige Faktoren bezüglich der Nutzerpräferenzen und der Akzeptanz von Applikationen braucht, damit ein Nutzer eine Applikation wählt. Für unser Projekt sind fünf der sechs Punkte relevant nämlich:

1. Personalisation
2. Integrität und Vertrauen der Benutzer in die App
3. Leistung der App
4. Nutzen der App
5. Äussere Einflüsse, wie Empfehlungen, Beliebtheit und Social Media Werbung

### UI- und UX-Prinzipien für die RoamApp

Um die UX und das UI der RoamApp zu verbessern, wurden die Prinzipien aus der Literatur sowie Vorwissen aus diversen Modulen abgeleitet, die direkt auf die RoamApp anwendbar sind.

- **Einfachheit und Klarheit:** Eine intuitive und leicht verständliche Benutzeroberfläche sorgt dafür, dass die Nutzer ihre Aufgaben schnell und problemlos erledigen können. Dies kann durch ein minimalistisches Design, klare Navigationselemente sowie kurze und verständliche Erklärungen erreicht werden.
- **Personalisierung:** Die Anpassung der App an individuelle Nutzerbedürfnisse, wie zum Beispiel personalisierte Einstellungen, Filter oder Sortieren erhöht die Nutzerbindung und Zufriedenheit.
- **Konsistenz im Design:** Ein einheitliches Design über alle Screens hinweg trägt zu einer besseren Orientierung und einem angenehmeren und motivierenden Nutzungserlebnis bei.
- **Feedback und Bestätigung:** Sofortige Rückmeldungen und Bestätigungen nach Benutzeraktionen, wie Benachrichtigungen und Warnungen, steigern das Vertrauen der Nutzer in die App.

## 2 Grundlagen und Hintergründe

- **Leistung und Zuverlässigkeit:** Schnelle Ladezeiten und eine stabile Performance sind entscheidend, um Frustrationen zu vermeiden und die Nutzererfahrung zu verbessern.
- **Nutzen:** Funktionen, die dem Nutzer einen Mehrwert bieten und schnell ausgeführt werden können, führen zu einer erhöhten Nutzerbindung und Zufriedenheit.
- **Sozialer Aspekt:** Menschen sind soziale Wesen und dementsprechend glücklicher, wenn sie ihre nahegelegenen Menschen einbeziehen oder gegen sie antreten können.
- **Bedürfnisse:** Wenn die Bedürfnisse des Nutzers durch Funktionen befriedigt werden, führt dies zu mehr Nutzerbindung und Zufriedenheit.

Durch die Anwendung dieser Prinzipien kann die RoamApp ihre User Experience und das User Interface verbessern und somit eine höhere Nutzerbindung und -zufriedenheit erreichen.

### Push-Notifikationen [27]

Push-Benachrichtigungen sind ein wichtiges Kommunikationsinstrument, um den Nutzer daran zu erinnern, die App zu öffnen und ihn über wichtige Ereignisse zu informieren, was zu einer stärkeren Bindung zur App führt. Allerdings sollte man darauf achten, nicht zu viele Benachrichtigungen zu verschicken und sicherstellen, dass sie relevant und angemessen sind. Wichtig ist auch, die Benachrichtigungen zu günstigen Zeiten zu senden, zum Beispiel morgens, mittags und abends während der Arbeitswoche.

### Benutzte Messdaten

In diesem Projekt wurden folgende Messdaten aus den Vorhandenen Messungen für die Bewertung der Signalstärke verwendet:

- RSRP (Reference Signal Received Power) [13]
  - **Einheit:** dBm (Dezibel Milliwatt)
  - **Bereich:** Zwischen -140 dBm und -44 dBm
  - **Definition:** RSRP ist die durchschnittliche Empfangsleistung der Reference Signals (RS) in einem LTE-Netzwerk.
  - **Bedeutung:** Ein niedriger RSRP-Wert bedeutet eine schwache Signalstärke, während ein höherer RSRP-Wert eine starke Signalstärke anzeigt.
  - **Verwendung:** RSRP wird hauptsächlich verwendet, um die Signalstärke in einem LTE-Netzwerk zu bewerten.
  - Misst die reine Signalstärke der Referenzsignale, ohne Rauschen und Interferenzen. Dies gibt eine klarere Indikation der Stärke des Signals. RSRP misst nur die reine Signalstärke, berücksichtigt aber nicht die Qualität des Signals. Ein starkes Signal kann immer noch durch Interferenzen oder Rauschen beeinträchtigt sein.
- RSRQ (Reference Signal Received Quality) [13]
  - **Einheit:** dB (Dezibel)
  - **Bereich:** Zwischen -19.5 dB und -3 dB
  - **Definition:** RSRQ ist ein Mass für die Signalqualität in einem LTE-Netzwerk und wird aus dem Verhältnis von RSRP zur Gesamtstärke des empfangenen Signals (RSSI) berechnet.
  - **Bedeutung:** Ein niedriger RSRQ-Wert zeigt eine schlechte Signalqualität an, während ein höherer RSRQ-Wert eine gute Signalqualität anzeigt.

## 2 Grundlagen und Hintergründe

- **Verwendung:** RSRQ wird verwendet, um die Qualität des empfangenen Signals zu bewerten, indem es die Stärke des Nutzsignals im Verhältnis zu Störungen und Rauschen misst.
- Bezieht RSRP und RSSI ein, um die Qualität des empfangenen Signals zu bewerten. Es hilft zu verstehen, wie viel des empfangenen Signals tatsächlich nutzbar ist.
- RSSI (Received Signal Strength Indicator) [14]
  - **Einheit:** dBm (Dezibel Milliwatt)
  - **Bereich:** Zwischen -90 dBm und -30 dBm
  - **Definition:** RSSI ist ein Mass für die gesamte Empfangsleistung, einschliesslich des gewünschten Signals, des Rauschens und aller Interferenzen.
  - **Bedeutung:** Ein niedriger RSSI-Wert zeigt eine schwache Signalstärke an, während ein höherer RSSI-Wert eine starke Signalstärke anzeigt.
  - **Verwendung:** RSSI wird verwendet, um die allgemeine Empfangsleistung zu bewerten, ohne zwischen Nutzsignal und Störungen zu unterscheiden.
- SNR (Signal-to-Noise Ratio) [13]
  - **Einheit:** dB (Dezibel)
  - **Bereich:** Zwischen 0 dB und +30 dB
  - **Definition:** SNR ist das Verhältnis der Leistung des Nutzsignals zur Leistung des Hintergrundrauschens.
  - **Bedeutung:** Ein niedriger SNR-Wert zeigt ein starkes Rauschen im Verhältnis zum Nutzsignal an, während ein hoher SNR-Wert eine klare Trennung zwischen Nutzsignal und Rauschen anzeigt.
  - **Verwendung:** SNR wird verwendet, um die Klarheit des empfangenen Signals im Verhältnis zum Hintergrundrauschen zu bewerten.
  - Misst das Verhältnis von Nutzsignal zu Rauschen. Ein hoher SNR-Wert zeigt an, dass das gewünschte Signal deutlich stärker als das Rauschen ist, was eine bessere Signalqualität bedeutet.
- Unterschiede und Vergleich
  - RSRP vs. RSSI: RSRP misst nur die Leistung der Reference Signals, während RSSI die gesamte empfangene Leistung misst, einschliesslich Rauschen und Interferenzen. Daher gibt RSRP eine genauere Messung der reinen Signalstärke im LTE-Netzwerk.
  - RSRQ vs. SNR: RSRQ bewertet die Signalqualität basierend auf der Signalstärke im Verhältnis zur gesamten empfangenen Leistung, während SNR die Klarheit des Nutzsignals im Verhältnis zum Hintergrundrauschen misst. SNR ist ein besseres Mass für die Signalreinheit, während RSRQ die allgemeine Signalqualität in Bezug auf Störungen und Interferenzen bewertet.

Diese Werte (RSRP, RSRQ, RSSI und SNR) bieten eine umfassende Bewertung der Signalstärke und -qualität in einem Mobilfunknetz. RSRP und RSSI messen die Signalstärke, wobei RSRP spezifischer für LTE-Netzwerke ist. RSRQ und SNR bewerten die Signalqualität, wobei RSRQ den Einfluss von Störungen und Interferenzen berücksichtigt und SNR die Klarheit des Nutzsignals im Verhältnis zum Hintergrundrauschen misst.

## 2 Grundlagen und Hintergründe

### iOS-Portation

Um die RoamApp in iOS umzusetzen, muss man zuerst an die Netzwerkdaten des iPhones gelangen. Mit den Kunden wurde besprochen, dass es Probleme gibt an die Netzwerkdaten zu kommen, da es keine entsprechende iOS API dafür gibt. Dementsprechend wurde vereinbart eine Recherche zu starten, ob man von der zukünftigen iOS RoamApp aus, Daten aus dem Field Test Mode beziehen kann und diese Daten, sich wie in der Android Version darstellen und erfassen lassen.

Um auf iOS die Netzwerkdaten anzeigen zu können, wird der Field Test Mode benötigt. Mit diesem Test kann ein iOS-Nutzer Signalstärke begutachten und sehen mit welchem Funkmast sein Gerät verbunden ist. Dieser Test wird auf dem iPhone durchgeführt, in dem man die Nummer \*3001#12345#\* anruft. Je nach Alter der iOS-Version, sieht man die Daten direkt im Dashboard oder muss zuerst unter LTE (Abbildung 10) eine Kategorie auswählen (Abbildung 9). Das Dashboard kann man selbst gestalten, in dem man Werte abspeichert mit dem Lesezeichen Icon (Abbildung 8). Bei älteren iOS-Versionen wird direkt ein Dashboard angezeigt mit allen wichtigen Daten. Obwohl die Smartphone Sprache Deutsch ist, wird alles im Field Test Mode auf Englisch dargestellt, was für Menschen, die nicht Englisch können eine Sprachbarriere darstellen könnte.

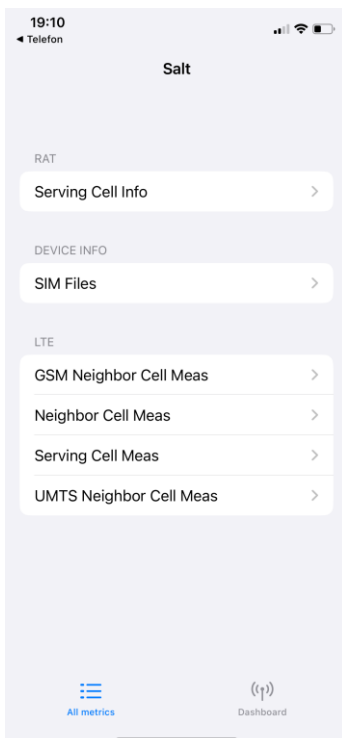


Abbildung 10 Field Test Mode - All Metrics



Abbildung 9 Field Test Mode - LTE Serving Cell Meas

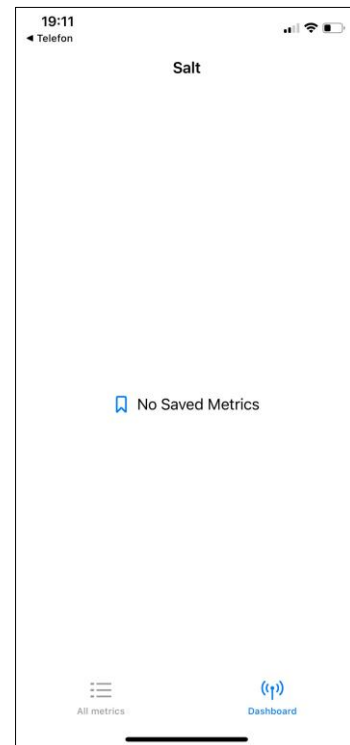


Abbildung 8 Field Test Mode - Dashboard

Für eine iOS RoamApp müsste man daher im Hintergrund automatisch die Daten aus dem Field Test Mode laden können. Eine entsprechende Recherche hat gezeigt, dass es weder Studien, Webseiten oder Blogbeiträge gibt, die beweisen würden, dass dieser Vorgang funktioniert. Im Gegenteil konnte aufgezeigt werden, dass die Field Test Mode Daten im Vergleich zu Android unzuverlässig sind für die Signalstärke und die Signalqualität [15], nicht alle wichtigen Daten mit iOS 15 gelesen werden können [15], wenn das Wi-Fi an ist, könnten die Daten nicht immer aktualisiert



## 2 Grundlagen und Hintergründe

werden [16], in iOS 17 wird die Signalstärke vom 5G nicht angezeigt [17] und es nicht möglich die Daten aus dem Field Test Mode in eine App zu laden [18], [19].

Wie in Kapitel **Fehler! Verweisquelle konnte nicht gefunden werden**. aufgezeigt, existieren Applikationen, welche die Messung der Signalstärke anbieten und auf iOS erhältlich sind (OpenSignal [20] und weeBost [21]). Die Messung der Signalstärke findet bei diesen Produkten jedoch manuell über den der Field Test Mode statt und beinhalten nur die Messung der Geschwindigkeit in der App selbst.

Somit kann unsere Fragestellung «Ist eine iOS-Version mit den vorhandenen Einschränkungen ausreichend benutzerfreundlich, um die Nutzerzahlen zu steigern?» beantwortet werden. Nein, die iOS RoamApp wäre nicht ausreichend benutzerfreundlich, um die Nutzerzahlen mit dem iPhone Marktanteil zu steigern, da eine iOS-Portierung mit den Daten aus dem Field Test Mode, nicht umgesetzt werden könnte. Dementsprechend kann eine iOS RoamApp nicht die gleichen Daten, wie bei der Android Version erfassen. Somit könnten Nutzer keinen Mehrwert aus einer iOS RoamApp ziehen, da die Applikation keine Messwerte der Mobilfunkverbindung automatisch erfassen und senden könnte, um Punkte zu erhalten. Ebenfalls könnte Roam Network von dieser Implementierung nicht profitieren, da keine zuverlässigen Daten automatisiert erhoben werden können.

### 2.2.4 Umfragen und Testings<sup>5</sup>

Für alle Testings und die Umfrage wurde als Bewertungsskala die Likert-Skala verwendet.

#### Likert-Skala [22]

Die Likert-Skala wird häufig in Umfragen, Marktforschung und Usability-Tests eingesetzt, um die Einschätzungen und Erfahrungen der Teilnehmer strukturiert zu erfassen und auszuwerten. Es gibt diverse Möglichkeiten die Skala zu erstellen. Die 6-Punkte-Likert-Skala ist so aufgebaut, dass sie von eins bis sechs reicht, wobei sie positiven bis negativen Auswahlmöglichkeiten abdeckt, und keine neutrale Mittelposition bietet, wodurch Die Tester gezwungen sind, eine klare Position zu beziehen.

Um genauere Daten zu erhalten und neutrale Antwortoptionen zu ermöglichen, wurde die Skala auf null bis fünf angepasst. Dies ermöglicht es, sofort zu erkennen, wenn keine Meinung vorliegt und vereinfacht die Berechnung von Durchschnittswerten.

Die Punkteskala, die zum Beispiel für die Umfrage eingesetzt worden ist, sieht wie folgt aus:

- Null = „Weiss ich nicht“
- Eins = „Schlecht“
- Zwei = „Es geht“
- Drei = „Mittelmässig“

---

<sup>5</sup> Die Usability-Testings und die A/B-Testings wurden im Interview Style geführt und in Excel erfasst. Link: [Testings.xlsx](#). Die Umfrage wurde mit Google Forms erstellt. Link: <https://docs.google.com/forms/d/1FgjE3hWxpFHnbNw9OpABL5N2xi-BUMusLt4aqBACY8kU/edit?ts=6688004c>

## 2 Grundlagen und Hintergründe

- Vier = „Gut“
- Fünf = „Super“

Durch die Verwendung der Likert-Skala wurden die Themen identifiziert, die den grössten Einfluss auf das übergeordnete Ziel haben. Um den Return on Investment (ROI) zu bestimmen, wurde der geschätzte Aufwand eines Arbeitspakets durch den erwarteten Nutzen welcher mit der Likert-Skala dargestellt ist, geteilt. Auf diese Weise konnten die Ideen anhand von Aufwandsschätzung und Auswirkungen priorisiert werden.

### 3 Theoretische Umsetzung

Dieses Kapitel beschreibt die Entwicklung der theoretischen Lösungen, um die Ziele des Projekts umzusetzen. Um diese Lösungen zu entwickeln, wurde zunächst der Stand des Prototyps analysiert und mit dem Kunden besprochen. Zusammen konnten auch teilweise Lösungen definiert werden.

Aufgrund der erzeugten Ideen im Workshop wurde, sowie der Marktanalyse in Kapitel 2.2.1, wurde entschieden zwei neue Features konkret umzusetzen, um anhand dieser die Effektivität auf die Benutzerbindung zu messen:

#### **Feature 1: Mobilfunkanbietervergleich**

Ein Wunsch von Roam Network gegenüber diesem Feature, war es einen Anbietervergleich machen zu können über einen bestimmten Zeitrahmen.

Zusätzlich wurde entschieden, dass die Möglichkeit bestehen soll, Mobilfunkanbieter mit Kontext vergleichen zu können, um ein Alleinstellungsmerkmal zu generieren. Kontext bedeutet in diesem Fall das Wetter, der genaue Aufenthaltsort (drinnen/draussen, Gebäudeart und die Art der Fortbewegung). Dies soll zu einem detaillierteren Vergleich führen welche Faktoren miteinbezieht die nicht in Vergleichen von Ähnlichen Applikationen, wie in Kapitel 2.2.12.2.1 aufgezeigt, erhältlich sind.

#### **Feature 2: Spiel**


Um ein Gamification Element zu implementieren, wurde entschieden ein Spiel als Feature hinzuzufügen. Das Spiel soll ähnlich wie das bekannte Spiel Tamagotchi [23] aufgebaut sein. Es soll darum gehen ein Avatar mit Messungen zu ernähren und Levels aufzusteigen. Die soll vor allem die Benutzerbindung erhöhen und kann auch als Alleinstellungsmerkmal angesehen werden, da die verglichenen Apps kein solches Feature besitzen. Es stellt sich die Frage wie gut dieses Feature angenommen wird von den Endbenutzern und auf welche Benutzergruppe sich dieses Feature besonders motivierend auswirkt.

#### **3.1 Zielgruppen**

Auf Basis der Aussage der Kunden, dass keine genaue Zielgruppe definiert wurden, sondern lediglich grob Nutzer von Krypto-Applikationen ansprechen wollen, wurden zwei Zielgruppen definiert. Beide besitzen gute Technikenntnisse und besitzen Krypto Applikationen. Die erste Zielgruppe (Abbildung 11) interessiert sich vorrangig für den Provider mit der besten Signalstärke in ihrer Umgebung. Die zweite Gruppe möchte passiv Geld verdienen. Beide Zielgruppen wurden im Figma umgesetzt.

### 3 Theoretische Umsetzung

**Hans Meier**



**ALTER**  
25  
**AUSBILDUNG**  
ELEKTRIKER EFZ  
**FAMILIENSTAND**  
VERHEIRATET  
**BERUF, EINKOMMEN**  
ELEKTRIKER, 6.075 CHF  
**HEIMATORT**  
ZÜRICH  
**TECHNICKENTNISSE**  
HOCH

"Ich interessiere mich für alles was sich, um Technick dreht."

**Personalität**

- Introvertiert
- Denker
- Krypto-Liebhaber
- Familienmensch

**Hobbies**

- Gaming
- Social Media
- Technologien

**Bio**

Lebt zur Zeit in Zürich mit seiner Familie. Interessiert sich hauptsächlich an alles was mit Technologien zu tun hat. Momentan ist seine Lieblingsbeschäftigung sein Krypto-Wallet zu vergrössern. Des Weiteren verbringt er viel Zeit mit seinen beiden Kindern und seiner Frau. Seine Familie ist viel am Smartphone und hat verschiedene Provider und möchte den besten und billigsten, für seine Familie finden. Insgesamt verbringt er viel Zeit an seinem Smartphone, um seinen Hobbies nachzugehen.

**Kernbedürfnisse**

- Interesse an einer Veränderung als dezentrale Gruppe/ Struktur
- Interesse an einer App die misst, mit welchem Abo man die beste Netzwerkverbindung hat, um Geld zu sparen für die Abos
- Mit Wirkung an einer Solarpunk-Welt

**Frustration/Ängste**

- Die Welt besteht aus zu wenig High-Tech
- Gibt zu viel Geld aus für die Smartphone-Abos.
- Ist besorgt, ob die App ungenaue Daten ausliest.

**Social Media/ Wo er die App findet**

- Instagram
- Discord
- Reddit

**Erwartungen und Verwendung der App**

- Möchte genaue Daten sehen in der App
- Möchte die Informationen möglichst übersichtlich dargestellt haben
- Wird die App hauptsächlich zu Hause verwenden und auch seiner Familie empfehlen
- Hat auch schon Netzwerkmessungsapps benutzt und hofft, dass diese eine bessere UX haben wird

Abbildung 11 Persona Zielgruppe 1

### 3 Theoretische Umsetzung

#### 3.2 Figma Hi-Fi-Versionen



Abbildung 12 Pointsscreen mit Badges und Glücksrad

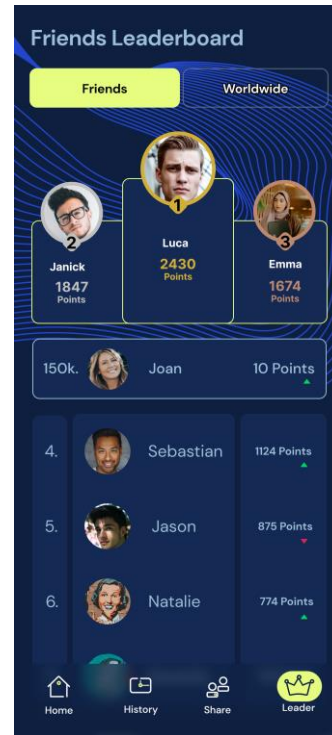


Abbildung 13 Leaderboard mit Friends und Worldwide Tab

Aufgrund der Erkenntnisse aus dem Abschnitt 3, den Usability-Tests, dem Workshop und diversen Meetings mit den Kunden wurden die Hi-Fi-Versionen Hi-Fi 1.0 und der darauffolgende durch die Usability-Tests verbesserte Hi-Fi 1.1 Leaderboard entwickelt. Die Leaderboard-Version enthält die beiden Hauptfeatures den Pointsscreen (Abbildung 12) mit den Badges, die man erhält, wenn man eine gewisse Punktzahl erreicht, sowie das Glücksrad, mit dem einmal am Tag Punkte gewinnen kann. Zudem gibt es ein Leaderboard, auf dem man sich mit Freunde oder allen anderen Nutzern messen kann (Abbildung 13). Neben der Vereinheitlichung der Benutzeroberfläche und Änderungen wie der Anpassung der Points History Tabelle, die nun nur noch aus drei Spalten besteht, wurden der Datascreen entfernt, da er der Zielgruppe keinen Mehrwert bietet, eine Suchleiste für das FAQ hinzugefügt, Push-Benachrichtigungen angepasst und weitere Elemente implementiert, die als Grundlage für die beiden im Folgenden erläuterten neuen Versionen dienen.

Auf Basis des Hi-Fi 1.1 Leaderboard wurden nach Besprechungen mit den Betreuern, zwei weitere Versionen, daraus entwickelt. Die beiden Hi-Fi Versionen, Hi-Fi 1.2.1 Provider und Hi-Fi 1.2.2 Roamgochi. Diese weisen grösstenteils die gleichen Elemente auf. Der wesentliche Unterschied besteht darin, dass sich die Provider-Version stärker auf einen praktischen Nutzen konzentriert, während die Roamgochi-Version auf eine jüngere Zielgruppe ausgerichtet ist und mehr auf Gamifizierung setzt.

### 3 Theoretische Umsetzung

#### 3.2.1 Hi-Fi Provider Version



Abbildung 14 Provider Vorschläge Location

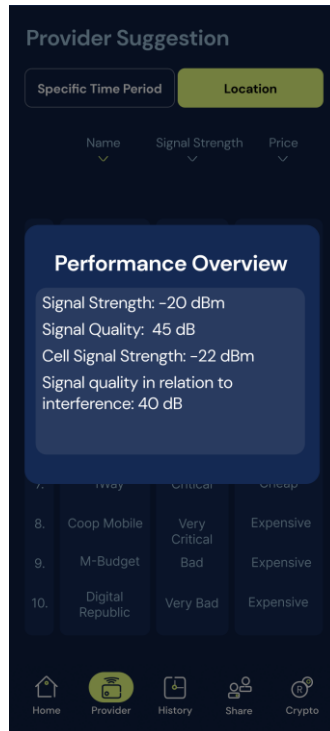


Abbildung 16 Provider Vorschläge Details



Abbildung 15 Provider Filter Kontext

Die Provider Version basiert auf dem Gedanken, dem Nutzer eine fundierte Begründung zu geben, warum man die App herunterladen und langfristig verwenden sollte. Um dem Nutzer spezielle Einblicke anhand seiner Daten zu ermöglichen, wurden die Providerscreens entwickelt. Der Nutzer kann zwischen zwei Tabs wählen: dem Time Period Tab und dem Location Tab (Details).

Im Time Period Tab kann der Nutzer einen vergangenen Zeitraum auswählen und erhält daraufhin Vorschläge, welcher Provider basierend auf der Signalstärke am besten geeignet ist für die in diesem Zeitraum besuchten Orte. Zusätzlich gibt es eine Spalte, die die Kosten der verschiedenen Anbieter auflistet.

Im Location Tab kann der Nutzer Messungen durchführen und dabei Kontextinformationen (Abbildung 15) wie zum Beispiel Wetter, Gebäudetyp oder ob man sich gerade in einem Fahrzeug befindet, angeben. Diese Informationen werden genutzt, um gezielte Providervorschläge (Abbildung 14) zu machen. Für eine manuell durchgeführte Messung erhält der Nutzer als Anreiz 10 Extra-punkte.

Wenn der Nutzer auf einen Eintrag in der Liste tippt, öffnet sich ein Fenster mit detaillierten Angaben zur Signalstärke und -qualität des jeweiligen Providers (Abbildung 16).

In dieser Version gibt es auch die Daily Points. Jeden Tag, wenn man die App öffnet, erhält man 10 extra Punkte.

### 3 Theoretische Umsetzung

Über Push-Benachrichtigungen wird der Nutzer erinnert die Provider Vorschläge anzuschauen die neuen Daten erhalten haben.

#### 3.2.2 Hi-Fi Roamgotchi Version

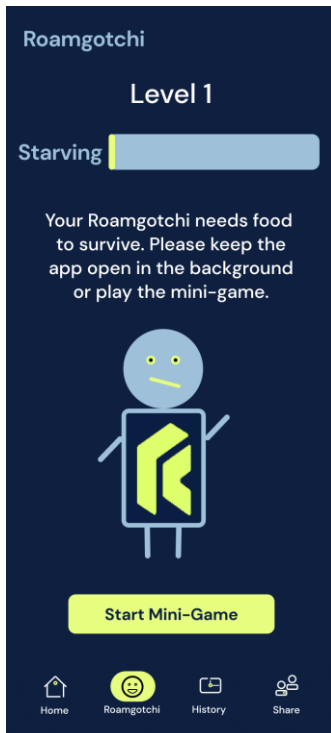


Abbildung 19 Roamgotchi Hungry

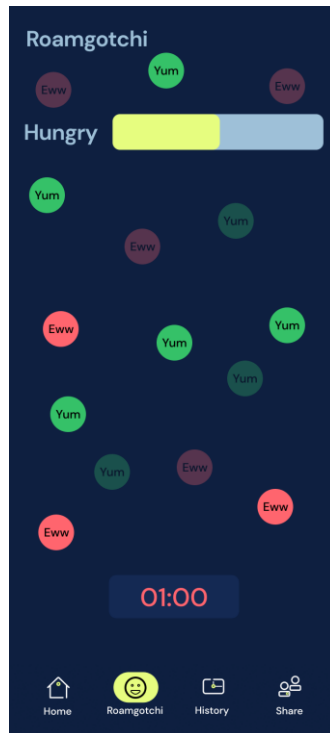


Abbildung 18 Mini-Game



Abbildung 17 Roamgotchi Full

Die Roamgotchi-Version ist ein Spiel, bei dem der Nutzer sich um ein virtuelles Wesen, das Roamgotchi, kümmern muss, nachdem es freigeschaltet wurde mit dem Tippen auf das Eis Ei. Dieses Wesen hat Bedürfnisse wie Hunger (Abbildung 19) und kann sterben, wenn es länger als drei Tage nicht gefüttert wird. Das Füttern erfolgt entweder durch das Offenlassen der App, wodurch automatisch im Hintergrund Messungen durchgeführt werden, oder durch das Spielen des Minispiels (Abbildung 18), wodurch sich die Fortschrittsleiste füllt. Im Minispiel muss der Nutzer auf alle grünen Kreise tippen, um jeweils 10 Fortschrittspunkte zu sammeln, bevor die Zeit abläuft. Tippt man jedoch auf einen roten Kreis, verliert man 10 Fortschrittspunkte. Wenn die Fortschrittsleiste vollständig gefüllt ist, ist das Roamgotchi satt, steigt ein Level auf und erhält einen neuen Skin, wie zum Beispiel einen Hut (Abbildung 17).

Über Push-Benachrichtigungen soll der Nutzer über den Status des Roamgotchi informiert werden. Beispielsweise erhält der Nutzer eine Benachrichtigung, wenn das Roamgotchi ein Level aufsteigt oder es kurz davor ist, zu sterben.

### 3 Theoretische Umsetzung

#### 3.2.3 Roamcoins

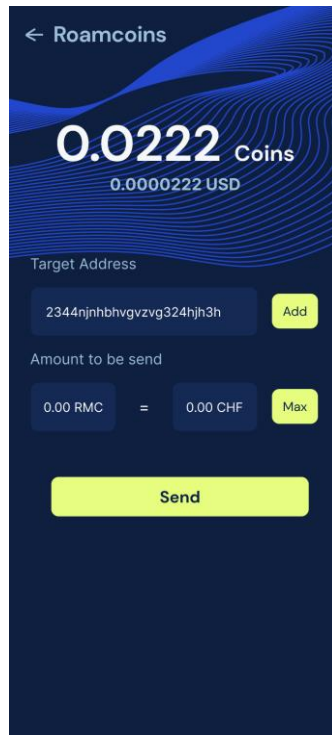


Abbildung 20 Roamcoins Send Screen

Um dem Nutzer einen weiteren langfristigen und motivierenden Nutzen zu geben, die App zu verwenden, wurden die Krypto Screens hinzugefügt. Die App hat somit mehr praktischen Nutzen und entspricht dem zukünftigen Plan vom Roam Network, die App in eine Kryptowährung App zu verwandeln. Der Krypto Screen, der in der Navigation zu finden ist, ist ein reiner Informationsscreen, der anzeigt, wie viele Coins zur Verfügung stehen und wie viel sie in der Landeswährung wert sind (Abbildung 24). Zudem gibt es eine Erklärung, was man mit den Coins machen kann.

Wenn man auf Send tippt, gelangt man zum Senden Screen (Abbildung 20). In diesem Screen sieht man dieselben eben erwähnten Angaben der Coins und der Landeswährung und kann eine beliebige Adresse angeben oder seine in den Einstellungen gespeicherte Wallet-Adresse über den Add Button hinzufügen. Danach kann man einen Betrag in RMC (Roamcoins) oder CHF eingeben oder auf Max tippen, um den gesamten Betrag zu versenden.

Dieser Proof of Concept dient dazu, Roam Network aufzuzeigen, dass die Roamcoins eine wichtige Funktion sind, die sie benötigen, um mit ihrer App erfolgreich zu machen. Anhand dieses Proof of Concept kann herausgefunden werden, ob eine Auszahlungsfunktion als praktischer Nutzen langfristig zu motiviert.



### 3 Theoretische Umsetzung

#### 3.2.4 UX und UI-Verbesserungen

Es folgt eine Auflistung und Erklärung der Elemente, bezogen auf die Literatur und deren Umsetzung die in beiden Hi-Fi Versionen (Provider / Roamgochi) verwendet wurden.

#### Einfachheit und Klarheit

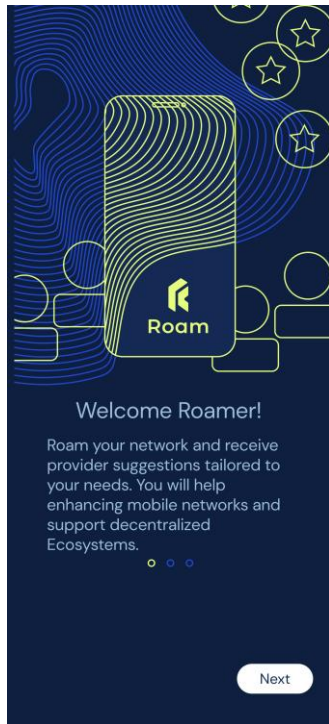


Abbildung 22 Login 1

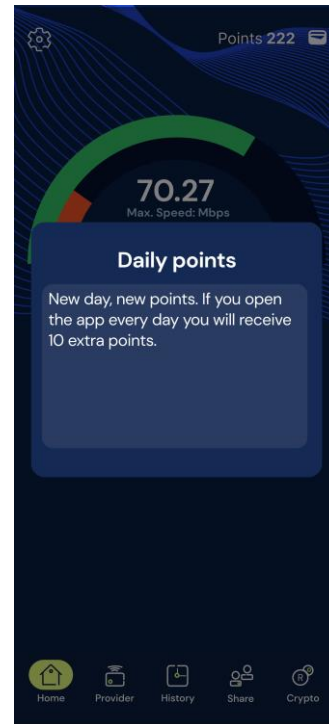


Abbildung 21 Overlay Erklärung Daily Points

Neben dem Beschreibungstexten auf der Roam Webseite<sup>6</sup> und im Google Play Store (siehe Fussnote 1, Seite 1), wurden, damit die RoamApp möglichst klar ist für die Nutzer, Onboardings hinzugefügt. Das Login wurde in drei Screens unterteilt. Als Beispiel soll der erste Login-Screen (Abbildung 22) mit dem Titel und dem Text erklären, dass die Person Teil eines grossen Ganzen ist und dazu beiträgt, das Mobilfunknetz zu verbessern sowie Anbieterempfehlungen zu erhalten. Des Weiteren gibt es für jeden Menüpunkt in der Navigationsleiste eine Erklärung (Abbildung 21). Diese Erklärungen wurden gleich wie die ursprünglichen Erklärungen für die Chart Bars in der RoamApp als Overlays mit Titel und Erklärung gestaltet. Nutzer, die kein Interesse daran haben, den gesamten Text zu lesen, können diese mit einem Tippen auf dem Screen, diesen schliessen. Es wurden möglichst kurze und informative Texte verfasst. Ebenfalls wurde darauf geachtet klare Navigationselemente zu designen wie die Buttons. Das Design wurde möglichst minimalistisch gehalten und jeder Screen hat zusätzlich noch einen Titel für mehr Klarheit erhalten. Die Push-Notifikationen wurden erweitert und angepasst, sodass der Nutzer stets informiert ist, etwa durch eine tägliche Zusammenfassung der erhaltenen Punkte oder Erinnerungen, um die Daily Points abzuholen.

<sup>6</sup> Roam Network Webseite, Link: <https://www.roamnetwork.xyz/>

### 3 Theoretische Umsetzung

#### Personalisierung

Um mehr Personalisierung anzubieten, kann der Nutzer diverse Push-Benachrichtigung ein- und ausschalten (Abbildung 23), seine Wallet-Adresse speichern, sich ausloggen, seinen Account löschen, Listen sortieren und filtern, die Anzahl der zu sendenden Roamcoins festlegen und entscheiden, wie man das Roamgochi füttert, entweder indem man die App im Hintergrund offenlässt oder aktiv das Mini-Spiel spielt. Der Nutzer sieht auch sein Google Account Profilbild, welches automatisch in der App aktualisiert wird, wenn es geändert wird.

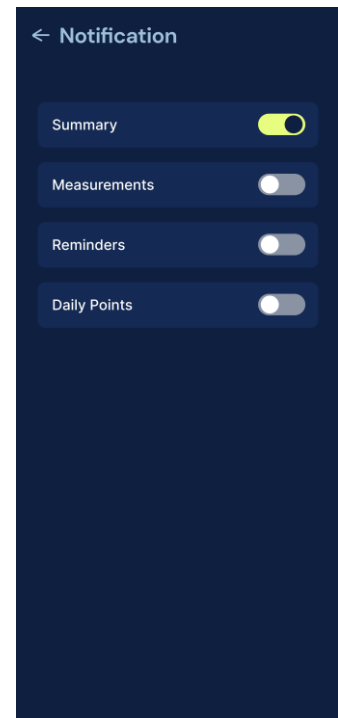


Abbildung 23 Push-Notifikationen Einstellungen

#### Konsistenz im Design

Um ein konsistentes Design zu gewährleisten, wurden stets die gleichen Farben für die gleichen Elemente verwendet und darauf geachtet, dass jeder Screen nach dem gleichen Prinzip gestaltet ist. Das Dark Design wurde beibehalten, da es bei technisch affinen Nutzern beliebter ist als ein Light Design. Es wurden die von Roam Network vorgegebenen Farben verwendet, wie zum Beispiel Gelb-Grün für Buttons und Tabs oder Dunkelblau für die Hintergrundfarbe der Screens. Für wichtigere Wörter wurde Weiss und für weniger wichtige Wörter Hellblau verwendet (Abbildung 24).

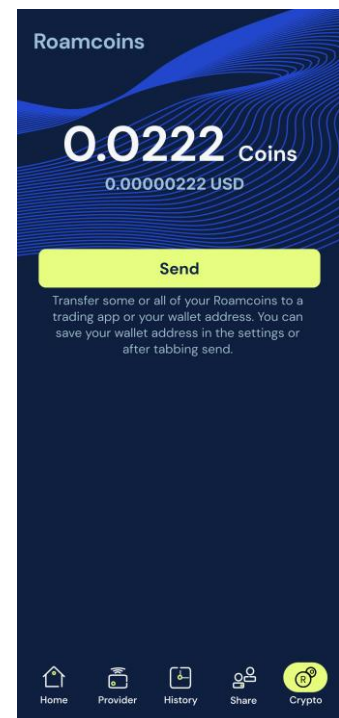


Abbildung 24 Design Roamcoins

#### Feedback und Bestätigung

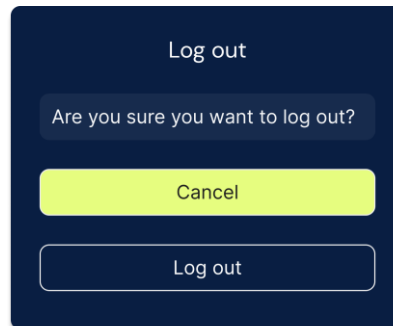


Abbildung 25 Log-out Bestätigung

Um das Vertrauen der Nutzer in die App zu stärken, wurden verschiedene Bestätigungen eingebaut, für das Ausloggen (Abbildung 25), das Löschen des Accounts und wenn der Nutzer keine Berechtigung für etwas, wie den Standortzugriff, erteilt. Das Feedback wurde so implementiert, dass der Nutzer auf eine Aktion eine sofortige Reaktion erhält. Beispielsweise wurde das manuelle Neuladen des Screens entfernt, damit der Nutzer gleich eine Veränderung sieht.

#### Nutzen

Durch die verschiedenen Features bieten sowohl die Provider- als auch die Roamgochi-Version den Nutzern jeweils spezifische Vorteile. Die Provider-Version ermöglicht es den Nutzern, verschiedene Vorschläge von Providern zu erhalten, während die Roamgochi-Version durch ihren spielerischen Ansatz besonders für jüngere Zielgruppen attraktiv ist. Neben dem Homescreen, der Informationen zur Signalstärke, Signalqualität und Geschwindigkeit anzeigt, bietet auch der Krypto Screen grossen Nutzen. Dies wurde durch Tests bestätigt.

#### Sozialer Aspekt

In den beiden Versionen ist der soziale Aspekt eher gering ausgeprägt. Nutzer können ihre Freunde über den Referral Screen (Abbildung 26) einladen und sich indirekt mit ihnen messen, indem sie vergleichen, wer mehr Punkte, Kryptowährung oder welches Roamgochi das höhere Level hat. Um den sozialen Aspekt zu stärken, wäre es sinnvoll, Ansätze aus dem Hi-Fi 1.1 Leaderboard, wie das Leaderboard und die Badges, zu integrieren. Die Nutzer könnten sich so besser mit Freunden messen. Die Badges, sollten wie das Leaderboard, für alle Freunde ersichtlich sein. Roam Network sollte sich intensiver mit Community Building auseinandersetzen, um diesen Aspekt zu erfüllen, weil sie so zu mehr Nutzern kommen könnten.

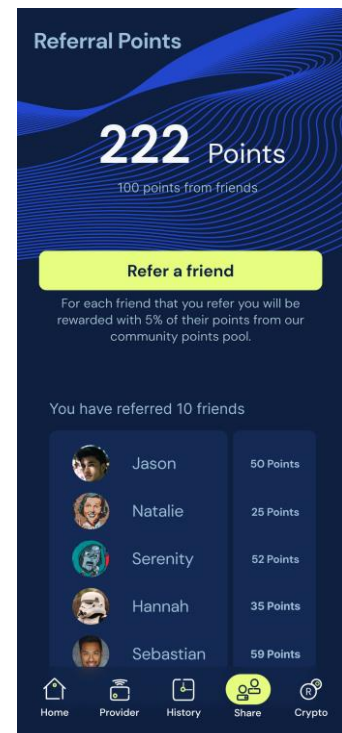


Abbildung 26 Referral Screen

### 3 Theoretische Umsetzung

#### Fazit

Somit kann die Fragestellung «Was sind die effektivsten UI-Anpassungen, welche die User Experience verbessern?» beantwortet werden. Alle aufgezählten UX-Konzepte wurden bei der Umsetzung der beiden Hi-Fi-Versionen berücksichtigt und durch die Usability-Tests verbessert. Es konnte bestätigt werden, dass die UX verbessert wird und die App dadurch motivierender und verständlicher ist, was schlussendlich die Nutzerbindung und -zufriedenheit erhöht.

#### 3.3 Mobilfunkanbietervergleich

Die Grundlage dieses Features besteht darin, die Möglichkeit zu bieten, Provider vergleichen zu können. Um dies zu erreichen, müssen alle Messungen bewertet werden können. Dies bestimmt die Platzierung der Mobilfunkanbieter in der Rangliste.

##### 3.3.1 Signalbewertung

Um die Signalbewertung zu berechnen, wurden die Gewichte pro Messgrösse definiert wie in der Tabelle 2 dargestellt.

Tabelle 2 Gewichtung der Messgrössen für die Berechnung der Signalstärkebewertung

RSRP	0.4
RSSI	0.2
RSRQ	0.2
SNR	0.2

Abhängig von der Anwendung können einige Metriken wichtiger sein als andere. Zum Beispiel kann in einer ländlichen Umgebung, in der die Signalstärke entscheidend ist, RSRP wichtiger sein. In städtischen Umgebungen mit viel Rauschen könnte SNR wichtiger sein.

Da RSRP die reine Signalstärke misst, wird es in dieser Bewertung höher gewichtet. Dies ist sinnvoll, wenn die reine Signalstärke ein wichtiger Indikator für die Netzabdeckung und -leistung ist. Weil wir die Qualität des Signals als genauso wichtig definieren wie die reine Signalstärke, wurde für eine gleichmässige Gewichtung von RSRQ und RSSI entschieden.

##### 3.3.2 Preisbewertung

Um eine Preisbewertung für jeden Mobilfunkanbieter zu erstellen, wurde versucht, Vergleichswebseiten zu scrapen wie beispielsweise [www.comparis.ch](http://www.comparis.ch). Diese Webseiten verteidigen sich jedoch gegen ein solches Vorgehen [24]. Für eine kommerzielle Nutzung der Vergleichsdaten ist dies nicht der optimale Ansatz, da diese Webseiten mit Anfragen belastet werden, welche keine Entschädigung darstellen.

Angebote mit den ungefähr gleichen Leistungen zu erfassen, stellt sich als sehr schwierig dar, da die Leistungen sich auf verschiedenen Ebenen unterscheiden. So ist es praktisch unmöglich, eine normalisierte Bewertung der Preise zu erstellen. Wir haben uns deshalb entschieden, einen Vergleich des schweizerischen Fernsehens zu heranzuziehen. In diesem Vergleich wurden die Angebote nach der Menge der Benutzung sortiert und verglichen [25].

### 3 Theoretische Umsetzung

Es wurde die Kategorie «intensive Nutzung Schweiz» [25] gewählt und darin nur die Angebote der Abonnemente verglichen, welche sich am ehesten gleichen ohne spezielle Vergünstigung. Es soll möglich sein, die Preise direkt einzulesen und daraus eine stufenweise Bewertung von 1-3 zu erstellen, welche in der Rangliste dementsprechend angezeigt werden können. Wenn nur Messungen eines Providers vorliegen, können keine Unterteilungen erstellt werden, da Vergleichsdaten fehlen.

### 4 Praktische Umsetzung

In diesem Kapitel wird beschrieben, wie die im Kapitel 3 erklärten Konzepte technisch in der Applikation umgesetzt wurden. Es wird zuerst auf grundsätzliche Umsetzungsmethoden und Änderungen eingegangen und danach auf die neu implementierten Features. Die Applikation wurde regelmäßig auf den Geräten Samsung Galaxy S10+ und Samsung Galaxy A52 getestet und während der gesamten Entwicklung genutzt.

#### 4.1 Grundlagen

Das Frontend der Applikation wurde auf Basis des bereits vorhandenen Codes implementiert. Die Benutzeroberfläche wurde mit Hilfe von XML-Dateien, die möglichst responsive gestaltet wurden, in Fragments und Activities eingebunden wurden über sogenannte *Bindings*. Zur Trennung von Benutzeroberfläche und Logik wurden ViewModels eingesetzt. Die ViewModels interagieren mit den Repositories, um Daten zu speichern und stellen mit *LiveData* den Fragments und Activities beobachtbare Daten zur Verfügung. Diese wiederum besitzen zuständige Observer, welche die UI-Elemente aktualisieren. Zur Speicherung kleinerer Datenmengen, wie beispielsweise, ob der Nutzer die App zum ersten Mal verwendet, wurde SharedPreferences verwendet. Für andere Daten, wie Providermessungen, wurde eine Datenbank verwendet.

#### Beispiel CryptoFragment und CryptoViewModel (Roamcoins)

```
20     val _coins = MutableLiveData<Double>()
21     val coins: LiveData<Double> = _coins
22
23     private fun pointsToCoins(points: Int): Double {
24         return points.toBigDecimal().divide(dividerCoins, 4, RoundingMode.DOWN).toDouble()
25     }
```

Abbildung 27 CryptoViewModel MutableLiveData

Das *CryptoViewModel* (Abbildung 27) verwaltet die Daten und die Logik für die Kryptowährungsscreens. Es beinhaltet *MutableLiveData*, um UI bezogene veränderbare Daten zu speichern und bereitzustellen. Als Beispiel wird in den Zeilen 20 und 21 die Variabel *coins* erstellt und gesetzt und in der Methode *pointsToCoins* verwendet. In dieser Methode werden die Points, welche aus dem *PointsRepository* geladen werden, in Coins umgerechnet und als Wert für *coins* gesetzt.

## 4 Praktische Umsetzung

```
63     override fun onCreateView(view: View, savedInstanceState: Bundle?) {
64         super.onCreateView(view, savedInstanceState)
65         UiEventDispatcher.addUiEventListener(this)
66
67         cryptoViewModel.coins.observe(viewLifecycleOwner) { coins ->
68             binding.tvTotalCoins.text = coins.toString()
69         }
70
71         cryptoViewModel.currency.observe(viewLifecycleOwner) { currency ->
72             val formattedCurrency = String.format("%.4f CHF", currency) // 4 Decimals
73             binding.tvMoney.text = formattedCurrency
74         }
75     }
```

Abbildung 28 CryptoFragment Observer ViewModel

Mit den Observern in *CryptoFragment* werden die *LiveData* aus dem *ViewModel* beobachtet und aktualisiert (Abbildung 28). Als Beispiel dafür sind die Zeilen 55 bis 57, in denen die *coins* beobachtet werden.

### 4.2 Allgemeine Änderungen

Basierend auf den Rückmeldungen der Usability-Tests wurden Personalisierungsfunktionen implementiert, wie das Anzeigen des Profilbildes des Google Accounts sowie die Möglichkeit, sich aus dem Account auszuloggen und diesen auch zu löschen. Das Backend der Account-Löschfunktion funktionierte während der Umsetzung jedoch nicht fehlerfrei. Dies wurde an Roam Network gemeldet.

Zusätzlich wurde die Aktualisierung der Punkteanzahl im Home Screen so angepasst, dass sie nun mit *LiveData* arbeitet und sich dadurch automatisch und in Echtzeit aktualisiert. In der übernommenen Applikation erfolgte die Aktualisierung nur bei einem manuellen Refresh der Anzeige.

Der History Screen wurde dahingehend geändert, dass die Aktualisierung nun automatisch erfolgt, ohne dass sie manuell angestoßen werden muss. Dies verbessert die Benutzererfahrung (UX), da weniger manuelle Interaktionen erforderlich sind.

### 4.3 Erklärungsscreens

Um die Erklärungsscreens umzusetzen, wurden die von Roam Network bereitgestellten *Widget* Klassen und das *HomeFragment* angepasst und verwendet. Ursprünglich gab es im Home vier verschiedene Erklärungen, die durch Tippen auf ein Element, im Balkendiagramm, geöffnet werden konnten. Diese Klassen und Methoden wurden zusammengefasst und vereinfacht. Dadurch blieben nur noch zwei Klassen übrig: Das *ExplanationModel*, welches die Erklärungen mit Titel und Text bereitstellt und das *ExplanationDialogFragment*, welches die Erklärungen in einem Dialog anzeigt.

Das *ExplanationModel* nutzt das Factory Method Pattern, um verschiedene Erklärungen zu erzeugen. Die Methode *widgetExplanationModelFactory* erstellt Instanzen der konkreten Erklärungsmodelle basierend auf dem übergebenen *WidgetExplanationType*. Dadurch bleibt der Code flexibel und erweiterbar, da neue Erklärungstypen einfach hinzugefügt werden können.



## 4 Praktische Umsetzung

Der Vorteil vom *ExplanationDialogFragment* ist, dass es nur eine XML-Datei verwendet, die als Vorlage für alle Erklärungen dient. Die Factory Methode wird verwendet, um das entsprechende *ExplanationModel* basierend auf dem Aufruf des Fragments zu erstellen, der durch die Methoden *openWidget* ausgelöst wird. Die *openWidget* Methode wird vom Interface *UiEvent* über die Klasse *UiEventDispatcher* bereitgestellt und in allen Navigationsfragments aufgerufen, die von *UiEvent* erben. Diese Fragments verfügen auch über Listeners, die erkennen, wann der Screen über *onResume* geöffnet wurde, um dann mit dem entsprechendem *WidgetExplanationType* die Methoden *openWidget* auszuführen.

### Beispiel CryptoFragment

```
77     override fun onResume() {
78         super.onResume()
79         cryptoViewModel.init(pointsRepository.getLocalPoints())
80         UiEventDispatcher.dispatchExplanation(ExplanationModel.Companion.WidgetExplanationType.CRYPTO)
81     }
82
83     override fun openWidget(widgetExplanationModel: ExplanationModel.Companion.WidgetExplanationType) {
84         if (widgetExplanationModel == ExplanationModel.Companion.WidgetExplanationType.CRYPTO
85             && !sharedPrefs.getExplanationShown("Crypto"))
86         ) {
87             executeTransactionIfStateAtLeastResumed {
88                 ExplanationDialogFragment.getInstance(widgetExplanationModel)
89                     .show(childFragmentManager, "ExplanationDialogFragment")
90                 sharedPrefs.setExplanationShown("Crypto", true)
91             }
92         }
93     }
94 }
```

Abbildung 29 CryptoFragment Erklärungsanzeige

Damit beim ersten Öffnen der App die Erklärung angezeigt wird, mit einem Tippen auf das Crypto Navigationselement, erbt die Klasse *CryptoFragment* von *UiEvent*. In der *onResume* Methode welche in Abbildung 29 zu sehen ist, registriert sich der Listener vom *UiEventDispatcher*, der den Typ der Erklärung angibt, in diesem Fall der *WidgetExplanationType.CRYPTO* (Zeile 80). Anschliessend wird automatisch die *openWidget* Methode ausgeführt, die überprüft, ob der Typ *Crypto* ist und ob das Overlay bereits angezeigt wurde (Zeile 85). Danach wird dem *ExplanationDialogFragment* die Instanz des Typs übergeben, um damit das Overlay mit dem entsprechenden Titel und Text über die Factory Methode anzuzeigen.

### 4.4 Push-Benachrichtigungen

Es wurde ein Screen erstellt, mit dem Benutzer festlegen können, welche Push-Benachrichtigungen angezeigt werden sollen. Diese können alle in den Einstellungen ein- oder ausgeschaltet werden. Dies war ein Ergebnis aus den ersten Usabilitytests, in denen die Nutzer diesen Wunsch geäußert haben.

Folgende Push Benachrichtigungen wurden für das jeweilige Feature implementiert:

#### Roamgochi

- Ein Tag bevor es «verhungert» wird vor dem Verlust des Levels gewarnt.
- Bei einem Levelaufstieg.



## 4 Praktische Umsetzung

### Providervergleich

- Einmal im Monat erscheint eine Erinnerung, eine detaillierte Messung durchzuführen, um auf Netzveränderungen aufmerksam zu machen.

### Allgemein

- Eine morgendliche Erinnerung, um die erhaltenen Punkte einzusehen.
- Eine Mittags-Erinnerung, um die tägliche Belohnung abzuholen.
- Eine abendliche Zusammenfassung, die die gesammelten Punkte des Tages anzeigt.

### 4.5 Kryptowährungsintegration

Um die Kryptowährungsintegration in der RoamApp als Proof of Concept zu implementieren, ist der Punktestand, welcher in diesem Feature benutzt, vom Punktestand der eigentlichen Applikation getrennt worden. Die Anzahl verfügbaren Coins stellen die Punkte dar, welche mit einem Faktor von 10'000 dividiert wurden. Diese Coins werden basierend auf einem festgelegten Umrechnungsfaktor in Fiat-Währung (CHF) umgerechnet. Die Umrechnung erfolgt innerhalb der Klasse *CryptoViewModel*, in welcher die Punkte in Coins und anschliessend in die entsprechende Währung umgewandelt werden.

Nutzer können eine Ethereum-Adresse eingeben, die in der Funktion *validateAndProcessAddress* überprüft wird, um sicherzustellen, dass nur gültige Adressen für Transaktionen verwendet werden. Die Implementierung des Versands wird mit der Funktion *subtractPoints* auf Zeile 78 der Klasse *CryptoViewModel* simuliert. Sobald die Anzeige neu geladen wird, initialisiert sich der Kontostand der Coins wieder auf Basis der Punktestandes der eigentlichen Applikation.

Dieser Proof of Concept beschränkt sich derzeit auf die Benutzeroberfläche und dient als Grundlage für zukünftige Erweiterungen, um die vollständige Integration von Blockchain-Technologien zu ermöglichen.

### 4.6 Providervergleich

Dieses Kapitel beschreibt, wie das Providervergleich Feature umgesetzt wurde und geht auf spezifische Elemente ein.

#### 4.6.1 Messungen

Um die Messdaten für dieses Feature zu speichern, wurde ein Objekt erstellt, das in dieser Arbeit als "Providermessung" bezeichnet wird und im Code als *ProviderMeasurementEntity* implementiert ist. Wie diese Messungen erfasst werden, wird in diesem Kapitel erläutert.

Die Applikation beinhaltet die Klasse *ScanForegroundService*, welche während der Laufzeit der Applikation Messungen im Intervall von 10 Sekunden durchführt. In dieser Klasse werden auch die Providermessung erstellt, was in der Funktion *storeProviderMeasurement* auf Zeile 382 erfolgt. Dazu wird die Klasse *ProviderMeasurementMapper* benötigt, welche die ursprünglichen *MeasurementData* in eine Providermessung umwandelt, um diese in der Datenbank abzulegen wie Abbildung 30 zeigt.

## 4 Praktische Umsetzung

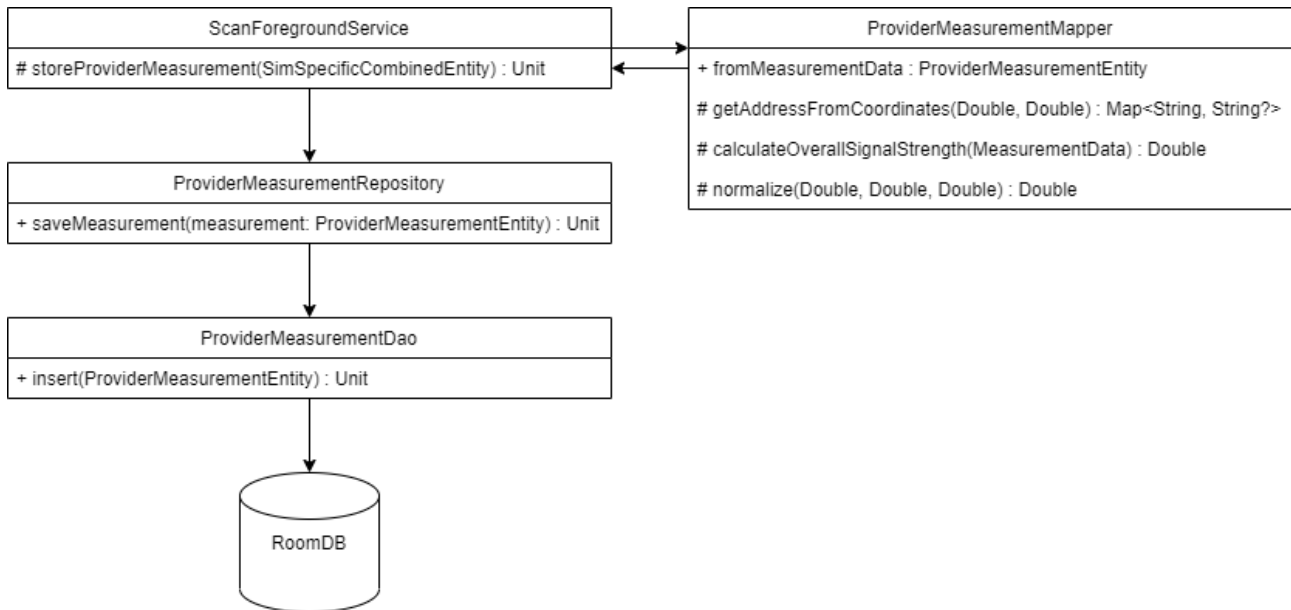


Abbildung 30 Klassendiagramm Erfassen von Providermessungen

Dabei wird der Standort, der ursprünglich nur als GPS-Daten vorliegt, in eine vollständige Adresse umgewandelt, die bis zum Strassennamen reicht. Dies erleichtert das gezielte Abfragen und Filtern der Messdaten aus der Datenbank. Die Umwandlung erfolgt durch eine Abfrage der Google Maps API, implementiert in der Funktion `getAddressFromCoordinates(lat: Double, lon: Double)` in Zeile 42 der Klasse `ProviderMeasurementMapper`.

Zusätzlich wird die Signalbewertung direkt vor dem Speichern für jede Providermessung berechnet. Dies geschieht in Zeile 75 der Klasse `ProviderMeasurementMapper` mit der Funktion `calculateOverallSignalStrength(data: MeasurementData)`. Falls eine Metrik null ist, wird ihr Gewicht auf 0 gesetzt, sodass sie nicht in die Berechnung einfließt. Diese Metriken werden gewichtet und normalisiert, um eine einheitliche Bewertung der Signalqualität zu ermöglichen. Die Normalisierung wird durch die Funktion `normalize` erreicht, indem der Bereich des jeweiligen Messwertes berücksichtigt wird.

Die normalisierten Werte werden mit ihren jeweiligen Gewichten multipliziert und summiert wie in Abbildung 31 zu sehen ist.

## 4 Praktische Umsetzung

```
93     private fun calculateOverallSignalStrength(data: MeasurementData): Double {
94         val weights = mutableMapOf(
95             "rsrp" to 0.4, // Importance of pure signal strength
96             "rsrq" to 0.2, // Importance of signal quality in relation to overall strength
97             "rssi" to 0.2, // Total received signal strength
98             "snr" to 0.2 // Ratio of useful signal to noise
99         )
100
101         // Set the weight to 0 if value is null to exclude it from the calculation
102         if (data.rsrp == null) weights["rsrp"] = 0.0
103         if (data.rsrq == null) weights["rsrq"] = 0.0
104         if (data.rssi == null) weights["rssi"] = 0.0
105         if (data.snr == null) weights["snr"] = 0.0
106
107         // Normalize the measured values
108         val rsrp = normalize(value: data.rsrp ?: 0.0, min: -140.0, max: -44.0)
109         val rsrq = normalize(value: data.rsrq ?: 0.0, min: -19.5, max: -3.0)
110         val rssi = normalize(value: data.rssi?.toDouble() ?: 0.0, min: -90.0, max: -30.0)
111         val snr = normalize(value: data.snr ?: 0.0, min: 0.0, max: 30.0)
112
113         val weightedSum =
114             rsrp * weights["rsrp"]!! +
115             rsrq * weights["rsrq"]!! +
116             rssi * weights["rssi"]!! +
117             snr * weights["snr"]!!
118
119         val totalWeight = weights.values.sum()
120
121         return weightedSum / totalWeight
122     }
```

Abbildung 31 Funktion für die Berechnung der Signalbewertung

Der Signalbewertung wird berechnet, indem die gewichtete Summe durch die Gesamtgewichtung dividiert wird. Dies führt zu einer Gesamtbewertung der Signalstärke der übergebenen Messung. Das Ergebnis der Funktion *calculateOverallSignalStrength* befindet sich im Bereich von 0 bis 1. Ein Wert näher bei 1 bedeutet eine bessere Signalqualität, während ein Wert näher bei 0 eine schlechtere Signalqualität anzeigt.

Bei der Angabe von Detailinformationen werden der neuesten Providermessung die entsprechenden Details hinzugefügt welche im Code als *ProviderContext* bezeichnet wird. Somit enthalten einzelne Providermessungen Detailinformationen zum Kontext, in dem die Messung gemacht wurde. Eine detaillierte Messung muss mindestens einmal durchgeführt werden, um dieses Feature zu benutzen. Dies soll dazu führen, dass möglichst viele Messungen mit Kontext in der Datenbank vorhanden sind. Ohne diese manuell ausgefüllten Angaben, werden nur die normalen Provider Messungen bei der Erstellung der Rangliste betrachtet. Zusätzlich werden pro detaillierte Messung zehn Punkte dem Endbenutzer hinzugefügt, um die Motivation für solche Messungen zu steigern.

## 4 Praktische Umsetzung

### 4.6.2 Datenbank

Um die *ProviderMeasurements*, die in Kapitel 3.3 beschrieben wurden, zu speichern, wurde eine Datenbank benötigt. Um Änderungen an der Hauptdatenbank von Roam Network zu vermeiden, wurde entschieden, eine Datenbank direkt auf dem Smartphone zu verwenden. Die Room DB war bereits in den Abhängigkeiten der ursprünglichen Applikation enthalten und wurde genutzt, um Messwerte abzulegen. Diese Datenbank läuft lokal und wird daher nicht mit Messdaten von realen Benutzern aktualisiert. Sollte das Feature bereitgestellt werden, kann lediglich die Dao-Klasse ersetzt werden. Da die für RoomDB verwendete Sprache dieselbe ist wie die für PostgreSQL [26], kann das Ersetzen der Datenbank einfach erfolgen. Die vorhandene Struktur wurde beibehalten, um das Zusammenführen der Änderungen zu erleichtern.

Um die Datenbank mit Daten zu befüllen, wurde eine eigene Klasse erstellt, die *ProviderMeasurements* einfügt. Da die Daten in einem spezifischen Bereich liegen müssen, konnte nicht auf eine existierende Bibliothek zurückgegriffen werden. Beispielsweise mussten die Messwerte der Signalstärke innerhalb eines definierten Bereichs liegen, um der Praxis zu entsprechen. Insgesamt wurden 20 Benutzer simuliert, die jeweils in 10 Gemeinden 5 Messungen vorgenommen haben. Somit sind insgesamt 1000 Messungen vorhanden, von denen 250 Details zum Kontext beinhalten. Dies findet in der Klasse *ProviderCompareDbSeeder* statt.

### 4.6.3 Preisbewertung

Die erfassten Preise pro Mobilfunkanbieter, welche in Kapitel 3.3.2 beschrieben wurden, sollen auf einer Skala von 1-3 unterteilt werden, um die Unterschiede der Preise in der Rangliste besser darstellen zu können. Die Preise welche aus dem Vergleich [25] entnommen wurden, werden als JSON-Datei im Ordner *assets* abgelegt. Die Klasse *ProviderPriceFinder* ist zuständig für das Erfassen der Preise und der Normalisierung. Die Funktion *normalizePrice(price: Double, minPrice: Double, maxPrice: Double)* auf Zeile 58 normalisiert den Preis innerhalb des Bereichs von *minPrice* bis *maxPrice* was den tiefsten und den höchsten Preis des gesamten Vergleichs darstellt. Somit ist ein klarerer Unterschied der Preise in der Rangliste zu erkennen.

### 4.6.4 Rangliste

Um die Rangliste zu erstellen, wird der Durchschnitt aller verfügbaren Messwerte pro Mobilfunkanbieter berechnet. Da der normalisierte Wert für die Signalbewertung bereits in jeder Messung vorhanden ist, ist dies eine einfache Aufgabe. Dieser Durchschnittswert bestimmt die Rangfolge der Mobilfunkanbieter. Es werden dabei nur Messdaten verwendet, welche maximal ein Jahr alt sind, um die Verwendung veralteter Daten zu verhindern. Dieser Vorgang findet in der *ProviderViewModel* Klasse statt innerhalb der Funktion *generateSummary* auf Zeile 354. Innerhalb dieser Funktion werden auch die Messungen, welche einen übereinstimmenden Kontext besitzen, 50-fach einberechnet. Um die Anzeige für den Preis zu bestimmen, wird auf die *ProviderPriceFinder* Klasse zurückgegriffen, welche im vorherigen Kapitel erwähnt wurde. Damit diese Arbeit asynchron stattfinden kann, wird die Rangliste als *LiveData* definiert.

Es wurden insgesamt drei verschiedene Ranglisten implementiert:

1. Rangliste welche Orte einbezieht, in denen man sich in einer auswählbaren Zeitperiode befand.
2. Rangliste für eine auswählbare Adresse, welche Details miteinbezieht.

## 4 Praktische Umsetzung

### 3. Rangliste für alle Orte an denen eine manuelle Detailangabe gemacht wurde.

In Abbildung 32 ist der Filter für die Details Rangliste zu sehen. Der genaue Standort kann bis auf Strassenebene manuell gewählt werden. Zusätzlich können die Details ausgewählt werden. Wenn Messungen mit den angegebenen Detailinformationen vorhanden sind, werden diese Daten mit einer Gewichtung von 50 in die Berechnungen einbezogen. Zur Vereinfachung wird der aktuelle Ort automatisch in die Textfelder geschrieben. Die Genauigkeit ist durch die Angabe der Adresse konfigurierbar. Wird ein Textfeld leer gelassen, wird die Genauigkeit auf die nächsthöhere Stufe erweitert.

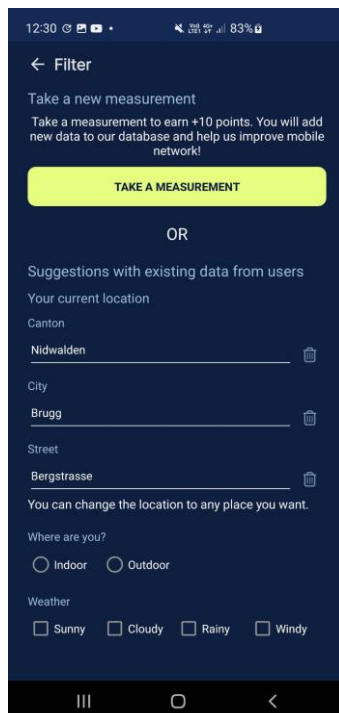


Abbildung 32 Screen Detailed Filter

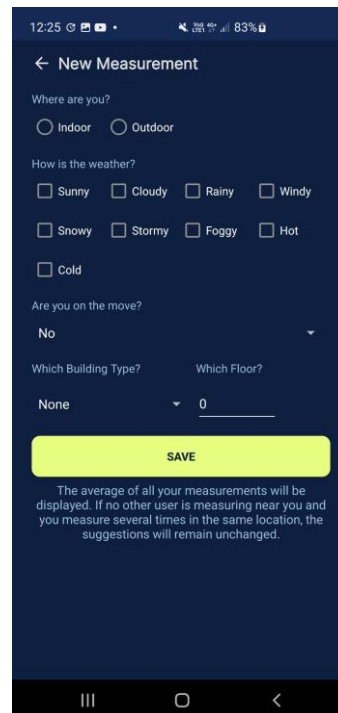


Abbildung 33 Erfassen von Detailangaben

In dieser Anzeige ist es auch möglich, eine Detailangabe für den aktuellen Standort vorzunehmen (Abbildung 33). Bei einer Detailangabe wird automatisch die Rangliste angezeigt, die alle bisherigen Standorte aus früheren Detailangaben einbezieht.

### 4.7 Roamgochi Game

Die Anzeige des Spiels wurde mit einem einzelnen Fragment umgesetzt. Dies verhindert, dass neue Anzeigen gerendert werden müssen und dadurch Wartezeiten entstehen. Stattdessen werden Elemente ein- und ausgeblendet, die in einem State definiert sind.

#### 4.7.1 State Pattern

Die Implementierung eines State Patterns ermöglicht eine präzise Steuerung der verschiedenen Zustände des Roamgochi und vereinfacht die Definition ihrer Übergänge. Das Spiel verfügt über vier Zustände, wie in Abbildung 34 dargestellt. Dieses Design Pattern gewährleistet zudem eine hohe Erweiterbarkeit, sodass in Zukunft neue Zustände problemlos hinzugefügt werden können. In

## 4 Praktische Umsetzung

der Klasse *RoamgochiViewModel*, welche als Kontext des State Pattern dient, wird der aktuelle Zustand verwaltet und die Übergänge gesteuert.

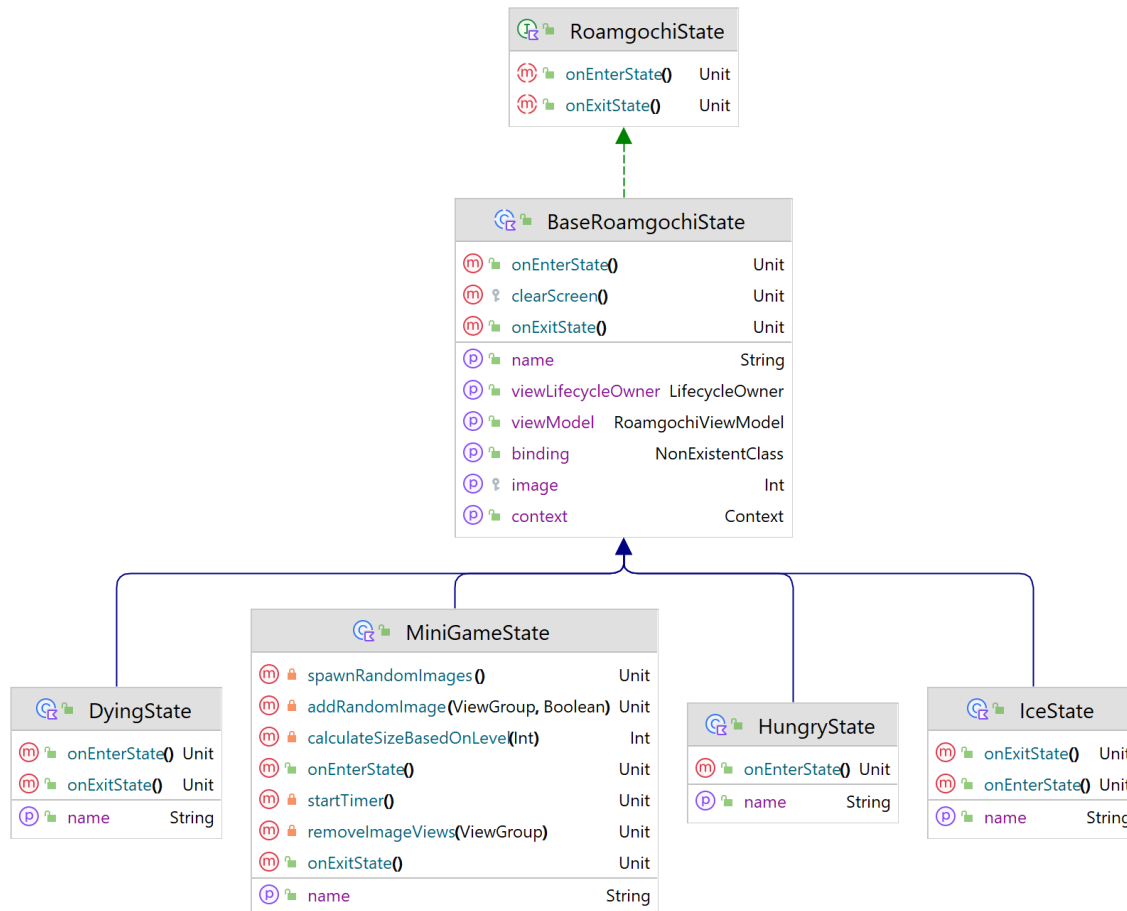


Abbildung 34 Klassendiagramm State Pattern Roamgochi

### 4.7.2 Levelsystem

Das Ziel des Spiels ist es, ein möglichst hohes Level zu erreichen. Um das nächste Level zu erreichen, muss der Avatar mit Messungen gefüttert werden. Eine erfolgreiche Messung mit der RoamApp trägt zum Fortschritt bei. Der Fortschritt pro Messung verringert sich jedoch, je höher das aktuelle Level ist. Dies findet auf Zeile 31 in der Klasse *RoamgochiRepository* statt. Sobald der Fortschrittsbalken gefüllt ist, steigt das Roamgochi ein Level auf.

Zusätzlich kann durch ein Mini-Spiel weiterer Fortschritt erspielt werden. Dabei handelt es sich um ein Reaktionsspiel, bei dem die korrekten Symbole, die auf dem Bildschirm erscheinen, angewählt werden müssen. Auch hier erhöht sich der Schwierigkeitsgrad proportional zum aktuellen Level, indem die erscheinenden Symbole verkleinert werden. Es ist jedoch auch möglich ein Level abzuweichen, wenn das Roamgochi Game länger als drei Tage nicht geöffnet wird. Falls dies der Fall ist, wird das Spiel im *DyingState* gestartet und symbolisiert das Verhungern des Avatars welches danach wiederbelebt werden kann. Dies führt zusätzlich zu einem Verlust des Fortschritts.

### 5 Testings

Im Verlauf des Projekts wurden drei Testings durchgeführt. Usability-Testings anhand von Interviews und A/B Testings, welche aus einer Umfrage und Interviews bestand. Die Testings wurden in der 21. bis 23. Kalenderwoche, in der 27. bis 29. Kalenderwoche und in der 33. Kalenderwoche durchgeführt.

Für die Interviews wurden 5 Personen befragt, da in diversen Design Modulen und im IP5 gelehrt wurde, wenn alle 5 Personen der Zielgruppe entsprechen, man einen grossen Teil der Meinungen der Zielgruppe erhält.

Die für alle Testings verwendete 6-Punkte-Likert-Skala wird im Abschnitt 2.4 Methoden und Materialien erläutert.

#### 5.1 Usability-Testings Figma

In diesen Usability-Testings wurde die Hi-Fi 1.0 Version des Figma Prototypen getestet. Dieser Prototyp hat ein neues UI-Design, bei welchem die Elemente besser zusammenpassen und eine verbesserte UX bietet, sodass der Nutzer mehr Verständnis für die Screens hat und die App leichter zu Nutzen ist. Es wurden neue Screens wie Leaderboard, Referral, Your Points, und Profil hinzugefügt und weitere Details wie die Searchbar in den Einstellungen oder die Erklärungen für die Screens. Generell wurde alles so verbessert, dass der Nutzer länger auf der App verweilt.

#### Durchführung

Ein Excel-Dokument wurde erstellt, das Fragen umfasst, die den Figma-Prototyp abdecken und Klarheit über die Fragestellung «Was sind die effektivsten UI-Anpassungen, welche die User Experience verbessern?» schaffen soll. Die Tests wurden mit fünf Testpersonen durchgeführt, die unterschiedliche technische Kenntnisse und Erfahrungen im Bereich Netzwerkdaten und Krypto-Apps hatten. Die Altersgruppe der Teilnehmer lag zwischen 19 und 28 Jahren und bestand aus vier Frauen und einem Mann, alle Schweizer.

Für jede Testperson wurde ein Interview durchgeführt, in dem verschiedene Aufgaben gestellt wurden. Zusätzlich wurde die Wichtigkeit jedes Features von den Testpersonen anhand der Punkteskala bewertet. Jedes Interview dauerte etwa 45 Minuten und lieferte wertvolle Einblicke in die Nutzererfahrungen.

#### Resultat

Die fünf Resultate wurden miteinander verglichen und zusammengefasst (Tabelle 3). Das am häufigsten erwähnte und relevanteste Feedback wurde in dem entsprechenden Hi-Fi 1.1 Leaderboard Figma-Prototyp umgesetzt. Somit kann die Fragestellung «Was sind die effektivsten UI-Anpassungen, welche die User Experience verbessern?», beantwortet werden mit dem Feedback vom Usability-Testing und der Erläuterung im Abschnitt 3.2.4.

## 5 Testings

Tabelle 3 Usability-Testing Resultate

Aufgaben	Durchschnitt der Bewertungen (Punkte)	Verbesserungen & Feedback
Onboarding kommentieren und bewerten	Onboarding: 4.4	Texte waren klar. Ein paar Worte wurden verändert.
Leaderboardscreen kommentieren und bewerten	Leaderboard: 4.2	Klar. Farben wurden geändert auf Gold, Silber und Bronze und eigener Rang immer Anzeigen lassen.
Referralscreen kommentieren und bewerten	Referralscreen: 4.4	Klar. Erklärung wurde klarer geschrieben. «Referrals» mit «Friends» getauscht, für mehr Community Gefühl
Punktescreen kommentieren und bewerten (Glücksrad, Badges, Missionen)	Glücksrad: 4.4	Grundsätzlich alles motivierend, um die App mehr zu gebrauchen. Glücksrad nur einmal am Tag drehbar. Erklärungstext für Badges.
	Badges: 3.8	
	Missionen: 4.6	
Profilscreen kommentieren und bewerten	Profil: 4.6	Wichtig, dass man sich ausloggen kann oder den Account löschen kann.
Einstellungsscreen (Searchbar) kommentieren und bewerten	Einstellungen: 5	Wichtig wenn es zu viele Informationstexte hat.
Push-Notifications kommentieren und bewerten	Push-Benachrichtigungen: 4	Gut, muss man aber ausschalten können. Nicht zu viele Benachrichtigungen. Wichtige und Call-to-Action Benachrichtigungen.
UI bewerten	UI: 4.4	Diverse Verbesserungsvorschläge
UX bewerten	UX: 4.6	Sehr gut, fast keine Bemerkungen

### 5.2 Usability-Testing Provider Prototyp

Es wurde ein Usability-Testing in Form eines Interviews durchgeführt mit einer Person, um die Verständlichkeit des ausprogrammierten Provider Features zu testen. Das Testing hat 30 Minuten gedauert. Die Resultate wurden direkt im Word erfasst.



**Resultat**

<b>Element</b>	<b>Verbesserungen &amp; Feedback</b>
Layout (Test auf kleinerem Gerät)	Manche Elemente sind abgeschnitten. Wurde behoben, nur «Take a Measurement» Text ist noch abgeschnitten.
Erklärung Login 1	Klar
Erklärung Time Period Tab	Klar. Die Wörter «Time Period» und «Detailed» wurden grossgeschrieben, damit es klarer ist. Es wurde ein Satz hinzugefügt, um klarzustellen, dass nur Daten berücksichtigt werden, vom Download Tag bis heute.
Time Period Kalender	Ist klar. Man sollte jedoch nur vom Tag, an dem man die App heruntergeladen hat, bis zum Heutigen Tag auswählen können. Den Bleistift (Edit) rausnehmen, zwei verschiedene Varianten, um das Datum einzugeben nicht nötig. Könnte auch verwirrend sein.
Time Period Tab	Klar. Es wurde einen Satz geschrieben, als Hinweis welche Daten (Messungen) genommen werden für das Resultat. Erklärung von den Daten, wenn man auf eine Zeile klick kommt, unerwartet.
Tabelle Time Period Tab	Die 5 Preisangaben sind unklar. Es wurde angepasst auf nur noch 3 Preisangaben: «Avg», «High», «Low»
Erklärung Detailed Tab	Klar. Einen Umbruch hinzufügen. Erklären warum man zuerst eine Messung machen muss.
Tabelle Detailed Tab	Man sollte anstatt nur den Standort noch die ausgewählten Details anzeigen. Vielleicht als Icon. Signal zeigt immer nur Stärke 3 an.
Detailed Tab	Button «Please select Details» unklar nach der ersten Messung. Anderer Satz wie «Filter and Take a Measurement» oder nur «Filter Details».
Details Screen	Name «Filter» unklar wurde zu «Details» geändert. «Take Measurement» wurde verbessert, führt nun mit einem tippen auf den Zurück Button in den Details Screen.

Tabelle 4 Provider Feature Usability-Testing

## 5 Testings

### 5.3 A/B-Testings

Beim A/B-Testing wurden die Figma Prototypen Hi-Fi 1.2.1 Provider, Hi-Fi 1.2.2 und die RoamApp (Version: 1.0.4) getestet. Das Ziel war es Gewissheit über die Frage «Hat Gamifizierung oder der praktische Nutzen einen grösseren Einfluss auf die Motivation der Nutzer, eine App zu nutzen?» und «Was sind die effektivsten Strategien, um die Anzahl der aktiven Nutzer der aktuellen RoamApp zu steigern?» zu bringen. Der Nutzer soll die App langfristig und regelmässig verwenden, idealerweise mehrmals täglich oder mindestens mehrmals wöchentlich.

#### 5.3.1 Umfrage

Um ein möglichst quantitatives Resultat zu erhalten, wurde eine Umfrage erstellt. Diese soll anhand von Single Choice Fragen aufzeigen, welches Feature am motiviertesten ist für die Nutzer, ausserhalb der definierten Zielgruppen. Zu den Fragen, wurde auch immer gefragt, wie motivierend der Nutzer, das Feature fand mit der Punkteskala.

Die Umfrage war in 5 Abschnitten gegliedert:

1. Angaben zur Person
2. Fragen über die Feature der RoamApp
3. Fragen über das Roamgochi Feature
4. Fragen über das Provider Feature
5. Gegenüberstellung der Features

#### Durchführung

Um möglichst viele Personen zu erreichen und da es keine Berechtigung gab die Umfrage an andere Hochschulen zu versenden, wurde die Umfrage an alle der Hochschule für Technik versendet. Des Weiteren wurde die Umfrage an Freunde und Bekannte weitergeleitet. Der Aufwand die Umfrage auszufüllen, dauerte etwa 5 Minuten und wurde bewusst so gewählt, um die Motivation der Tester nicht zu verlieren während dem Ausfüllen der Umfrage.

#### Resultat

Diese Umfrage wurde von 52 Personen aus verschiedenen Alterskategorien von 13 bis über 60 Jahren ausgefüllt, hauptsächlich von Angehörigen der Hochschule für Technik. Die Mehrheit (50%) gehörte zur Alterskategorie von 20 bis 30 Jahren. In den restlichen Alterskategorien waren mindestens drei Personen vertreten. Die Nationalitäten der Teilnehmer waren unterschiedlich, jedoch überwiegend Schweizer. Auch die Bildungsrichtungen variierten, wobei der Grossteil im technischen Bereich war. Bezüglich des Geschlechts waren mehr Männer als Frauen vertreten. Die Mehrheit der Befragten verfügte über mittelmässige bis gute technische Kenntnisse, mit durchschnittlich 3,96 Punkten.

Die Features der ursprünglichen RoamApp haben gezeigt, dass detaillierte Messdaten nicht motivierend sind. Ebenso wurden Signalstärke und -Qualität nur knapp von der Mehrheit der Tester als motivierend empfunden wie in Abbildung 35 zu sehen ist. Im Gegenteil dazu steht das Punkte sammeln, welches klar von der Mehrheit als motivierend angesehen wurde. Insgesamt wurden die Features der ursprünglichen RoamApp nur mit durchschnittlich 2.9 Punkten bewertet.

## 5 Testings

Was hältst du insgesamt von der Roam App?

52 Antworten

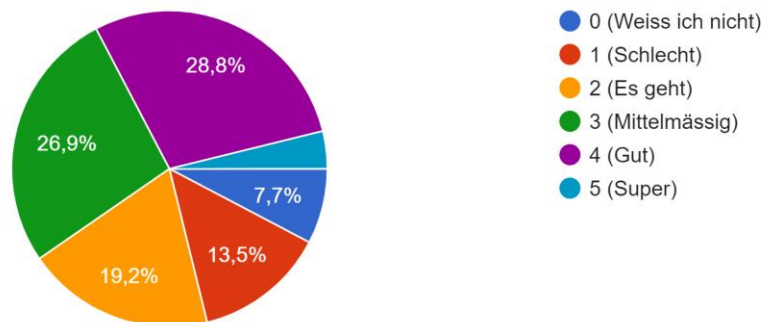


Abbildung 35 Umfrage RoamApp Bewertung

Das Roamgochi Feature hat gezeigt, dass sie für alle Altersgruppen ausser Teenager eher demotivierend wirkt. Insgesamt erhielten sie eine Bewertung (Abbildung 36) von durchschnittlich 2.47 Punkten, was die schlechteste Bewertung unter den getesteten Features darstellt. Das passive Füttern des Roamgochi im Hintergrund ist beliebt und zeigt, dass Hintergrundaktivitäten wie auch das Punkte sammeln sehr erwünscht sind.

Das Roamgochi wurde jedoch eher für Teenager konzipiert. Betrachtet man nur diese Bewertungen dieser Zielgruppe, schneidet das Roamgochi sehr gut ab, mit einer Bewertung von durchschnittlich 4.67 Punkten. Das Roamgochi würde dementsprechend die Teenager motivieren die App herunterzuladen und täglich zu nutzen. Auch das Minispiel wurde bevorzugt über die Hintergrund Funktion. Dies zeigt, dass Jugendliche eher aktivere Nutzer sind, und Gamifizierung bevorzugen über einen praktischen Nutzen.

Wie findest du insgesamt die Roamgochi-Version?

52 Antworten

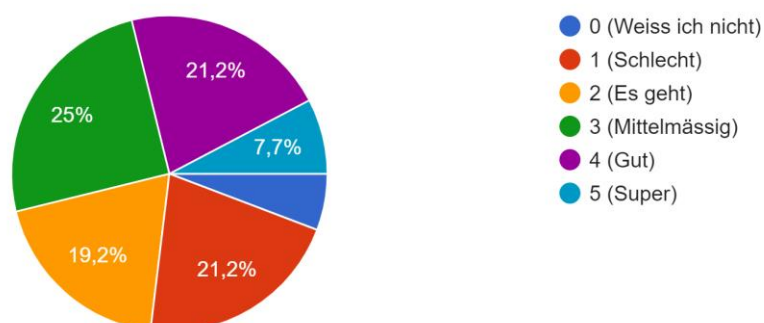


Abbildung 36 Umfrage Roamgochi Bewertung

## 5 Testings

Das Provider Feature hat gezeigt, dass es ausreichend motivierend ist, um die App herunterzuladen, jedoch nicht für eine langfristige Nutzung. Insgesamt wurde das Provider Feature mit durchschnittlich 3.76 Punkten bewertet (Abbildung 37).

Wie findest du insgesamt die Provider-Version?

52 Antworten

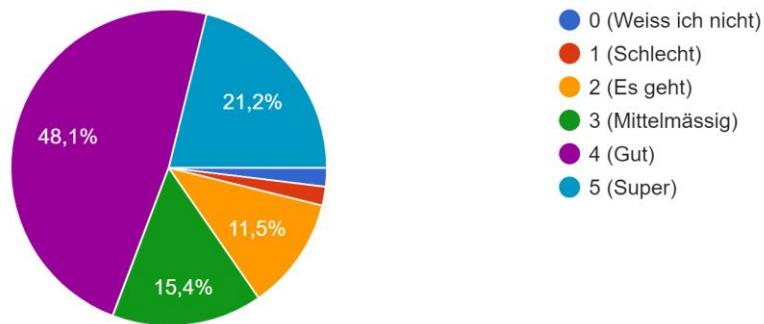


Abbildung 37 Umfrage Provider Bewertung

Das Feature, welches insgesamt am besten abgeschnitten hat, ist das Provider Feature. Das Roamgochi Feature und die Features der ursprünglichen RoamApp wurden ähnlich, als das Provider Feature bewertet (Abbildung 38).

Welche der vorgestellten Funktionen gefällt dir am besten?

52 Antworten

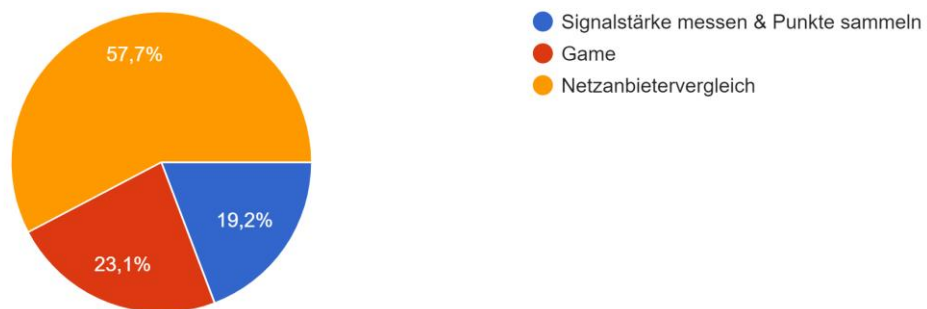


Abbildung 38 Umfrage Bewertung Provider, Roamgochi, RoamApp

Insgesamt wurde von 15 Testern kommentiert, dass die Provider Version jedoch nur kurzfristig interessant wäre. Des Weiteren wurde der fehlende Datenschutz und das Laufen der App im Hintergrund kritisiert. Eine Person hat bemerkt, dass mehr Informationen, über die von Roam Network gesammelten Daten die Transparenz verbessern könnten. Die meisten Tester würden die App nicht herunterladen. Somit kann man sagen, dass die App für die aktuelle Zielgruppe der RoamApp nicht geeignet ist, aufgrund des fehlenden Nutzens und des mangelnden Datenschutzes. Wenn die

## 5 Testings

RoamApp jedoch auf eine jüngere Zielgruppe und/oder weniger technisch versierte Zielgruppe abzielt, könnten die Ergebnisse besser ausfallen.

### 5.3.2 Interviews

Um ein möglichst genaues Resultat zu erhalten, wurden neben der Umfrage noch Interview Fragen geschrieben. In diesen Fragen wurde, wie auch in der Umfrage die Motivation der Tester untersucht. Im Gegensatz zur Umfrage jedoch, wurden Fragen für die gesamte Provider Version, Roamgochi Version und RoamApp geschrieben. Neben den Hauptfeatures Providervorschläge und das Roamgochi Spiel, wurde auch andere Elemente wie der Kryptowährungsscreen, die Erklärungsscreen und die Hauptscreens der App (Home, Points History, Referral) erwähnt.

#### **Durchführung**

Die Tests wurden mit den gleichen fünf Testpersonen, wie beim Usability-Testing durchgeführt. Es wurde jeweils ein Interview durchgeführt mit hauptsächlich verschiedenen Fragen zur Motivation der Tester. Zusätzlich wurde für jedes Feature, die Wichtigkeit für den Tester mit der Punkteskala abgefragt. Ein Interview hat etwa 30 Minuten gedauert und wichtige Einblicke gebracht.

#### **Resultat**

Die fünf Resultate wurden miteinander verglichen und zusammengefasst (Tabelle 5).

## 5 Testings

Motivation	Langfristiger Nutzen	Allgemeine Anmerkungen	Insgesamt	Durchschnitt der Bewertungen
<b>Provider</b>	<p>Provider Vorschläge würden nur motivieren, um es ein zwei Mal auszuprobieren, danach wenn man ein gutes Abo hat, nicht mehr. Jedoch könnte man die Nutzer motivieren die App einige Wochen zu brauchen für bessere Vorschläge.</p> <p>Daily Points sind motivierend, wenn sie, in Geld umgewandelt werden können.</p>	Beide Tabs Location und Time Period sind motivierend, je nach Nutzer.	Das Provider Feature hat nur ein kurzfristiges Nutzen und befriedigt nur eine sehr technische Zielgruppe. Hingegen Geld auszahlen zu können bringt viel mehr Motivation für alle Nutzer.	4
<b>Roamgochi</b>	<p>Das Kümmern hat mittelmässige bis gute Bewertungen bekommen und wirkt grundsätzlich motivierend. Auch für nicht Gamer, da man indirekt Punkte sammeln kann.</p> <p>Das Level aufsteigen hat nur eine mittelmässige Bewertung erhalten. Nur Skins als Belohnung, wirkt nur wenig motivierend. Mehr Punkte hingegen mehr.</p>	Das Minispiel und die App im Hintergrund offen zu lassen wird beides als motivierend empfunden.	Meinungen sind unterschiedlich die einen finden es gut die anderen vermissen die Daten, vom Provider oder generell Daten oder sind einfach keine Gamer. Nur Gamifizierung ist motivierender für jüngere. Ältere bräuchten einen praktischen Nutzen, der aber auch langfristig etwas bringt.	3.7
<b>RoamApp</b>	Die Messdaten motivieren nicht. Auch nicht mit Erklärungen. Ausser die Zielgruppe wären Personen, die sich sehr gut mit Messdaten auskennen, dann würde es motivieren.	Die App hat keinen Mehrwert für die meisten Nutzer.	Die RoamApp ist demotivierend. Mit einer Kryptowährung wäre sie auch nur knapp befriedigend.	2.4

## 5 Testings

	Die Punkte motivieren nicht, weil man sie noch nicht nutzen kann und weil die App keinen Nutzen hat für die Nutzer.			
<b>Kryptowährung</b>	Ist für alle drei Versionen motivierend. Für die RoamApp weniger, weil die App sonst keinen Nutzen hat. Vor allem wichtig, dass die Coins viel Geld wert sind.	Ist motivierend für Trader. Für die meisten ist das Wichtigste, dass man die Kryptowährung als Geld auszahlen lassen kann.	Muss eingebaut werden in die RoamApp. Am besten, wenn man an sein Konto über seine IBAN einen Betrag senden kann.	
<b>Am meisten motivierend</b>	Die Provider Version jedoch gemischt mit Roamochi.	Interessanter wäre auf dem Homescreen eine Angabe zur Internetgeschwindigkeit und/oder wie gut die Anrufverbindung ist.	Die RoamApp schneidet sehr schlecht ab. Der nicht vorhandene Datenschutz und Nutzen sind demotivierend.	
<b>Push-Notifikationen</b>	Sind motivierend.	Eins bis Zwei pro Tag sonst wieder demotivierend.	Motivierend als Erinnerung oder Information.	

Tabelle 5 A/B Testing Interviews

### Fazit

Mit allen Tests können die Fragestellungen «Was sind die effektivsten Strategien, um die Anzahl der aktiven Nutzer der aktuellen RoamApp zu steigern?» und «Hat Gamifizierung oder der praktische Nutzen einen grösseren Einfluss auf die Motivation der Nutzer, eine App zu nutzen?» beantwortet werden. Für die Zielgruppe von Roam Network (siehe Abschnitt 3.1) ist eine Applikation mit praktischem Nutzen die aussichtsreichste Variante. Die Provider-Version mit dem Kryptowährungsscreen erweist sich als die erfolgreichste. Die Providervorschläge bieten den Nutzern interessante Dateneinblicke, ähnlich wie der Home Screen, welcher jedoch eher auf Internetgeschwindigkeit und Anrufverbindung angepasst werden sollte. Die Points könnten komplett weggelassen und durch Roamcoins ersetzt werden. Der wichtigste Faktor ist, dass die Roamcoins einen hohen Wert haben und ausgezahlt werden können. Dies würde viele Nutzer anlocken und dazu führen, dass die App langfristig und sogar mehrmals in der Woche bis täglich genutzt wird. Jedoch sollte Roam Network eine Möglichkeit finden, wie sie entweder nur die nötigsten Daten erfassen können oder dem Nutzer in der App und im Play Store erklären welche Daten für welchen Zweck verwendet werden und wieso das ein positives Resultat mit sich bringt für den Nutzer.

## 6 Diskussion

### 6.1 Ergebnisse

Die folgenden Fragestellungen wurden in dieser Arbeit untersucht und beantwortet:

RQ1. Was sind die effektivsten Strategien, um die Anzahl der aktiven Nutzer der aktuellen RoamApp zu steigern?

**Beantwortung:** siehe 5.2 A/B-Testings, Fazit

RQ2. Was sind die effektivsten UI-Anpassungen, welche die User Experience verbessern?

**Beantwortung:** siehe 3.2.4 UX und UI-Verbesserungen und 5.1 Usability-Testings, Resultat

RQ3. Ist eine iOS-Version mit den vorhandenen Einschränkungen ausreichend benutzerfreundlich, um die Nutzeranzahl zu steigern?

**Beantwortung:** siehe 2.2.3 Literatur und Websuche, iOS-Portation

Die übergeordnete Fragestellung: Hat Gamifizierung oder der praktische Nutzen einen grösseren Einfluss auf die Motivation der Nutzer, eine App zu nutzen?

**Beantwortung:** siehe 5.2 A/B-Testings, Fazit

Die Ergebnisse dieser Arbeit wurden durch Studien, Web-Recherchen, Vergleiche zwischen diversen Applikationen, Usability-Testings, A/B-Testings und einer Umfrage gewonnen. Es wurde aufgezeigt, wie Roam Network ihre aktiven Nutzer steigern, langfristig motivieren und die Nutzerbindung stärken kann mit einer Variante der App, die den Nutzern praktische Nutzen anbieten kann. Dazu gehört zusammengefasst, die Applikation mit einem einheitlichen UI zu gestalten, eine verständliche UX zu gewährleisten, das Provider Feature zu integrieren, die Points zu ersetzen durch die Roamcoins und diese auszahlfähig zu machen und die Internetgeschwindigkeit und die Anrufverbindung im Home-Balkendiagramm anzuzeigen.

Die Arbeit hat gezeigt, dass eine gamifizierte Variante für die Zielgruppen von Roam Network weniger erfolgsversprechend wäre. Zudem konnte durch eine gründliche Recherche festgestellt werden, dass eine Umsetzung der RoamApp auf iOS nicht möglich ist, da nur Apple Zugriff auf die für Roam Network benötigten Daten hat. In dieser Arbeit wurde für das iOS-Problem nur der Field Test Mode untersucht, da Roam Network bereits erfolglos versucht hat, die Applikation auf iOS zum Laufen zu bringen und sie den Field Test Mode empfohlen haben.

Zu beachten wäre auch das Marketing. Um die Anzahl der aktiven Nutzer zu steigern, müsste Roam Network mehr in ihr Marketing investieren. Ebenso würde eine App mit Community-Funktionen, wie die in Figma geplanten Features Leaderboards und Badges, dazu beitragen, ihr Ziel zu erreichen. Apps, die auf Community-Building setzen, sind generell am erfolgreichsten [12]. Diese beiden Strategien wurden jedoch bewusst nicht in dieser Arbeit berücksichtigt, da andere Strategien im Verlauf des Projekts geplant wurden.

Das Hauptproblem von Roam Network besteht darin, die Nutzer der Zielgruppen davon zu überzeugen, ihre Daten preiszugeben. Datenschutz ist, wie sowohl laut der Umfrage als auch in den Interviews, für die Nutzer von grosser Bedeutung. Der von Roam Network vorgeschlagene Ansatz,



mehr Informationen über die gesammelten Daten bereitzustellen, würde das Problem nicht lösen. Ein vielversprechenderer Ansatz wäre, Werbung dafür zu machen, dass Nutzer mit der RoamApp passiv Geld verdienen können. Diese Strategie könnte dazu beitragen, dass mehr Nutzer trotz Bedenken hinsichtlich des Datenschutzes bereit sind, die App herunterzuladen.

### 6.2 Herausforderungen

Die folgenden Herausforderungen sind während der Arbeit aufgetreten:

#### **Wissenschaftliche Fragestellungen**

Ein grosses Hindernis war der Beginn des Projektes, als bekannt wurde, dass die RoamApp bereits veröffentlicht war und kein klares Ziel für diese Arbeit vorlag. Daher wurden verschiedene Meetings abgehalten, um eine entsprechende Aufgabe zu finden. Jedoch kreuzten sich mehrmals die Ziele des Kunden mit denen des Projekts. Des Weiteren fehlten dem Prototyp eine gewisse Komplexität. Aufgrund dessen wurde im Juni die Hi-Fi Leaderboard Version nicht weiterentwickelt, sondern durch die Provider und Roamgochi Version ersetzt. Dies führte dazu, dass der Zeitplan nicht vollständig eingehalten werden konnte.

#### **Erlernen des verwendeten Technologie-Stacks**

Die Technologie hinter der RoamApp zu verstehen, stellte eine grosse Herausforderung dar. Zuerst musste das Wissen erlernt werden, wie man mit Fragments, Activities, ViewModels, XMLs usw. umgeht. Dies nahm viele Stunden in Anspruch, da aufgrund fehlenden Wissens immer wieder Probleme auftraten.

#### **API und Datenbank**

Während der Entwicklung standen wir vor der Herausforderung, Umsetzungen vorzunehmen, ohne die bestehende API und Datenbank zu verändern. Deshalb wurde entschieden Providermessungen in einer separaten Datenbank abzulegen. Um dies zu ermöglichen, wurde eine lokale Datenbank direkt auf dem Smartphone erstellt.

Der Nachteil dieser Lösung ist, dass eine lokale Datenbank nicht die gleiche Leistungsfähigkeit und Skalierbarkeit wie eine Remote-Datenbank bietet. Aus diesem Grund war es notwendig, die Anzahl der in der lokalen Datenbank gespeicherten Fake-Daten möglichst gering zu halten, um die Performance und Stabilität der Anwendung zu gewährleisten.

#### **UX**

Die Gestaltung der UX des Provider-Features so, dass sie für verschiedene Personen verständlich ist, war schwierig. Die meisten Menschen haben kein grosses Wissen oder Interesse an Providern. Dies machte es zu einer Herausforderung, allen Projektbeteiligten gerecht zu werden.

### 7 Schlussbemerkungen

Diese Bachelorarbeit beschäftigt sich mit dem Thema, wie man Nutzer langfristig an die Applikation binden und sie motivieren kann, die App regelmässig zu verwenden. Diese Ziele konnte teilweise mit der Provider-Version erreicht werden. In dieser Version wird der Nutzer durch Providervorschläge und Roamcoins motiviert, die App langfristig zu nutzen. Dies konnte durch die durchgeführten Testings gezeigt werden.

#### **Ausblick**

Um die RoamApp zukünftig weiter zu verbessern, sollten verschiedene Elemente ergänzt werden. Derzeit ist die App nur auf Englisch verfügbar, jedoch wäre es sinnvoll, mehrere Sprachoptionen anzubieten, da die RoamApp weltweit erhältlich ist.

Die Auswahl des Zeitraums im Kalender könnte so eingeschränkt werden, dass nur Daten ab der ersten eigenen Messung wählbar sind.

Die Providervorschläge könnten ein Pattern identifiziert werden, das dem Nutzer zeigt, in welchen Strassen er sich häufiger aufhält. Eine weitere interessante Funktion wäre eine Karte, die dem Nutzer anzeigt, wo überall Messungen durchgeführt wurden. In der Tabelle könnten die ausgewählten Details in Form von Wörtern oder Icons dargestellt werden.

Eine zukünftige Optimierung der Signalbewertung könnte durch die Berücksichtigung der Umgebung erfolgen. So könnte beispielsweise eine Liste von Orten mit hohem Rauschpegel integriert werden, bei denen die Gewichtung von RSRP und SNR angepasst wird. An solchen Orten könnte die Gewichtung des SNR erhöht und die des RSRP entsprechend verringert werden.

Die Benutzeroberfläche könnte weiter optimiert werden, um die Nutzerzahlen noch stärker zu steigern. Strategien zur Förderung des Community-Bindings wären hierbei besonders wertvoll und sollten in die App integriert werden. Darüber hinaus könnte die Oberfläche mit zusätzlichen Erklärungen ausgestattet werden, die sich durch Antippen eines „i“-Icons öffnen lassen. Eine weitere Idee wäre die Einführung eines Light Modes, um den Nutzern eine alternative Darstellung anzubieten.

Ein Problem, das bei der RoamApp aufgefallen ist, besteht darin, dass sie ältere Geräte nicht vollständig unterstützt. Einige Funktionen erfordern ein zu hohes API-Level, weshalb an bestimmten Stellen bereits als veraltet geltender Code verwendet werden musste, um Abstürze der App zu vermeiden. Dieses Problem sollte von Roam Network behoben werden. In diesem Projekt wurden bereits vereinzelt Lösungsansätze dafür ausprobiert.

Beim Roamgochi-Spiel sollte eine Art Verzögerung eingebaut werden, damit die Fortschrittsanzeige bei 100% einige Sekunden lang stehen bleibt und der Text den Nutzer darauf hinweist, dass das Roamgochi ein Level aufgestiegen ist.

### Quellenverzeichnis

- [1] „Dependency injection with Hilt“, Android Developers. Zugegriffen: 13. August 2024. [Online]. Verfügbar unter: <https://developer.android.com/training/dependency-injection/hilt-android>
- [2] Velmurugan, „Creating android app using MVVM + Coroutines + Flow + Hilt“, Howtodoandroid.com. Zugegriffen: 13. August 2024. [Online]. Verfügbar unter: <https://www.howtodoandroid.com/android-app-using-mvvm-coroutines-flow-hilt/>
- [3] „SharedPreferences“, Android Developers. Zugegriffen: 13. August 2024. [Online]. Verfügbar unter: <https://developer.android.com/reference/android/content/SharedPreferences>
- [4] „Home | Silencio“, Silencio.network. Zugegriffen: 13. August 2024. [Online]. Verfügbar unter: <https://www.silencio.network>
- [5] „Sweatcoin: Healthier planet. Healthier, wealthier you“, Sweatcoin. Zugegriffen: 13. August 2024. [Online]. Verfügbar unter: <https://sweatco.in/>
- [6] „22 mobile app KPI metrics you should be tracking“, Sendbird. Zugegriffen: 5. August 2024. [Online]. Verfügbar unter: <https://sendbird.com/blog/essential-mobile-app-kpis-engagement-metrics>
- [7] M. Froehlich, C. Kobiella, A. Schmidt, und F. Alt, „Is it Better With Onboarding? Improving First-Time Cryptocurrency App Experiences“, in *Proceedings of the 2021 ACM Designing Interactive Systems Conference*, in DIS '21. New York, NY, USA: Association for Computing Machinery, Juni 2021, S. 78–89. doi: 10.1145/3461778.3462047.
- [8] B. Strahm, C. M. Gray, und M. Vorvoreanu, „Generating Mobile Application Onboarding Insights Through Minimalist Instruction“, in *Proceedings of the 2018 Designing Interactive Systems Conference*, in DIS '18. New York, NY, USA: Association for Computing Machinery, Juni 2018, S. 361–372. doi: 10.1145/3196709.3196727.
- [9] J. Hamari, J. Koivisto, und H. Sarsa, „Does Gamification Work? – A Literature Review of Empirical Studies on Gamification“, in *2014 47th Hawaii International Conference on System Sciences*, Jan. 2014, S. 3025–3034. doi: 10.1109/HICSS.2014.377.
- [10] P. Bitrián, I. Buil, und S. Catalán, „Enhancing user engagement: The role of gamification in mobile apps“, *J. Bus. Res.*, Bd. 132, S. 170–185, Aug. 2021, doi: 10.1016/j.jbusres.2021.04.028.
- [11] M. A. T. Pratama und A. T. Cahyadi, „Effect of User Interface and User Experience on Application Sales“, *IOP Conf. Ser. Mater. Sci. Eng.*, Bd. 879, Nr. 1, S. 012133, Juli 2020, doi: 10.1088/1757-899X/879/1/012133.
- [12] O. Al-Shamaileh und A. Sutcliffe, „Why people choose Apps: An evaluation of the ecology and user experience of mobile applications“, *Int. J. Hum.-Comput. Stud.*, Bd. 170, S. 102965, Feb. 2023, doi: 10.1016/j.ijhcs.2022.102965.
- [13] „LTE RSSI, RSRP and RSRQ Measurement“, CableFree. Zugegriffen: 5. August 2024. [Online]. Verfügbar unter: <https://www.cablefree.net/wirelesstechnology/4glte/rsrp-rsrq-measurement-lte/>
- [14] „What is a good RSSI Signal Strength and Can it be Affected by Interference? - Ruijie Networks“. Zugegriffen: 5. August 2024. [Online]. Verfügbar unter: <https://www.ruijienetworks.com/support/faq/what-is-a-good-rssi-signal-strength>
- [15] „Field Test Mode For iPhones and Android - Waveform“. Zugegriffen: 6. Mai 2024. [Online]. Verfügbar unter: <https://www.waveform.com/a/b/guides/field-test-guide>
- [16] S. Sweet, „iOS 16 Breaks field test mode AGAIN? Really?“, The Solid Signal Blog. Zugegriffen: 6. Mai 2024. [Online]. Verfügbar unter: <https://blog.solidsignal.com/tutorials/ios-16-breaks-field-test-mode-again-really/>
- [17] S. Sweet, „iOS 17 breaks field test mode, AGAIN“, The Solid Signal Blog. Zugegriffen: 6. Mai 2024. [Online]. Verfügbar unter: <https://blog.solidsignal.com/tutorials/ios-17-breaks-field-test-mode-again/>
- [18] B. S, „Answer to ‚Copying data from Field Test Mode‘“, Stack Overflow. Zugegriffen: 6. Mai 2024. [Online]. Verfügbar unter: <https://stackoverflow.com/a/1944001/24126551>

## Quellenverzeichnis

- [19] Cool, „Is it possible to get the data from field test mode into an app“, Stack Overflow. Zugegriffen: 6. Mai 2024. [Online]. Verfügbar unter: <https://stackoverflow.com/q/78294642/24126551>
- [20] „Mobile Network Experience Metrics | Opensignal“. Zugegriffen: 16. August 2024. [Online]. Verfügbar unter: <https://www.opensignal.com/our-approach/mobile-metrics>
- [21] „App | weBoost“. Zugegriffen: 16. August 2024. [Online]. Verfügbar unter: <https://www.weboost.com/app>
- [22] T. Nemoto und D. Beglar, „Developing Likert-Scale Questionnaires“.
- [23] S. S. Andersen, „The Tamagotchi Was Tiny, but Its Impact Was Huge“, *Wired*. Zugegriffen: 5. August 2024. [Online]. Verfügbar unter: <https://www.wired.com/story/tamagotchi-25-year-anniversary-impact/>
- [24] „Comparis Automates Scraping Protection & Saves 1 Day/Week“, DataDome. Zugegriffen: 6. August 2024. [Online]. Verfügbar unter: <https://datadome.co/customers-stories/comparis-automates-scraping-protection-saves-a-day-per-week-with-datadome/>
- [25] „Angebotsdschungel Mobilfunk - So finden Sie den richtigen Handy-Tarif“, Schweizer Radio und Fernsehen (SRF). Zugegriffen: 6. August 2024. [Online]. Verfügbar unter: <https://www.srf.ch/sendungen/kassensturz-espresso/espresso/angebotsdschungel-mobilfunk-so-finden-sie-den-richtigen-handy-tarif>
- [26] „Android Room vs PostgreSQL Modeler | What are the differences?“, StackShare. Zugegriffen: 12. August 2024. [Online]. Verfügbar unter: <https://stackshare.io/stackups/android-room-vs-postgresql-modeler>
- [27] D. Gavilan und G. Martinez-Navarro, „Exploring user’s experience of push notifications: a grounded theory approach“, *Qual. Mark. Res. Int. J.*, Bd. 25, Nr. 2, Art. Nr. 2, Feb. 2022, doi: 10.1108/QMR-05-2021-0061

## Glossar

Begriff	Seite	Erläuterung
Provider	ii	Englischer Begriff für Anbieter. Damit ist in diesem Dokument ein Mobilfunkanbieter gemeint.
Roamcoins / Coins	ii	Im Kryptowährungsumfeld bezeichnet ein Coin eine digitale Währungseinheit, die auf einer eigenen Blockchain basiert. Ein Coin fungiert oft als Tauschmittel, ähnlich wie traditionelle Währungen, und kann verwendet werden, um Werte zu transferieren.
Roam Network	ii	Name des Kunden
RoamApp	1	Name der Applikation im Play Store
Coin Economy	5	In Bezug auf Kryptowährungen bezieht sich der Begriff Coin Economy auf das gesamte wirtschaftliche System, das die Schaffung, Verteilung, Verwendung und den Austausch einer bestimmten Kryptowährung umfasst.
Mobilfunkzelle	5	Eine Mobilfunkzelle ist ein geografisches Gebiet, das von einem Mobilfunknetz abgedeckt wird. Es ist die grundlegende Einheit in einem Mobilfunknetz.
Roam	26	Name auf der Webseite / Name unterhalb der heruntergeladenen App
Responsive	31	Der XML-Code wurde möglichst so implementiert, dass er sich automatisch an verschiedene Bildschirmgrößen anpassen kann.
Fiat Währung	36	Fiat-Währungen sind das gängige Zahlungsmittel in den meisten Ländern der Welt, wie zum Beispiel der US-Dollar (USD)

## Eigenständigkeitserklärung

### Eigenständigkeitserklärung

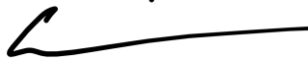
Ich (wir) erkläre(n) hiermit, dass ich (wir) den vorliegenden Leistungsnachweis selbst und selbständig verfasst habe(n),

- dass ich (wir) sämtliche nicht von mir (uns) selber stammenden Textstellen und anderen Quellen wie Bilder etc. gemäss gängigen wissenschaftlichen Zitierregeln [z.B. APA oder IEEE] korrekt zitiert und die verwendeten Quellen klar sichtbar ausgewiesen habe(n);
- dass ich in einer Fussnote oder einem Hilfsmittelverzeichnis alle verwendeten Hilfsmittel (KI-Assistenzsysteme wie Chatbots [z.B. ChatGPT], Übersetzungs- [z.B. DeepL] Paraphrasier- [z.B. Quillbot]) oder Programmierapplikationen [z.B. Github Copilot] deklariert und ihre Verwendung bei den entsprechenden Textstellen angegeben habe;
- dass ich sämtliche immateriellen Rechte an von mir allfällig verwendeten Materialien wie Bilder oder Grafiken erworben habe oder dass diese Materialien von mir selbst erstellt wurden;
- dass das Thema, die Arbeit oder Teile davon nicht bei einem Leistungsnachweis eines anderen Moduls verwendet wurden, sofern dies nicht ausdrücklich mit der Dozentin oder dem Dozenten im Voraus vereinbart wurde und in der Arbeit ausgewiesen wird;
- dass ich mir bewusst bin, dass meine Arbeit auf Plagiate und auf Drittautorenschaft menschlichen oder technischen Ursprungs (Künstliche Intelligenz) überprüft werden kann;
- dass ich mir bewusst bin, dass die Hochschule für Technik FHNW einen Verstoß gegen diese Eigenständigkeitserklärung bzw. die ihr zugrundeliegenden Studierendenpflichten der Studien- und Prüfungsordnung der Hochschule für Technik verfolgt und dass daraus disziplinarische (Verweis oder Ausschluss aus dem Studiengang) Folgen resultieren können.

Windisch, 16.08.2024

Name: Elena Algaria

Unterschrift:



Name: Yannick Hohler

Unterschrift:

