

**Konzeption und Implementierung einer  
mobilen Anwendung zur  
standortbasierten Gamifizierung der  
Nicht-Nutzung von Smartphones**

BACHELORARBEIT WiSE 2017/2018

ausgearbeitet von

Vu Phi Hai Dinh - Matrikelnr.: 11090042

zur Erlangung des akademischen Grades  
BACHELOR OF SCIENCE (B.Sc.)

vorgelegt an der

TECHNISCHEN HOCHSCHULE KÖLN  
CAMPUS GUMMERSBACH  
FAKULTÄT FÜR INFORMATIK UND  
INGENIEURWISSENSCHAFTEN

im Studiengang  
WIRTSCHAFTSINFORMATIK

Erstprüfer: Prof. Dr. Matthias Böhmer  
Technische Hochschule Köln

Zweitprüfer: Prof. Dr. Christian Kohls  
Technische Hochschule Köln

Gummersbach, der 20. Februar 2018

**Adressen:** Vu Phi Hai Dinh  
Hindenburgstraße 19  
51643 Gummersbach  
PhiHai.Dinh@gmx.de

Prof. Dr. Matthias Böhmer  
Technische Hochschule Köln  
Institut für Informatik  
Steinmüllerallee 1  
51643 Gummersbach  
matthias.boehmer@th-koeln.de

Prof. Dr. Christian Kohls  
Technische Hochschule Köln  
Institut für Informatik  
Steinmüllerallee 1  
51643 Gummersbach  
christian.kohls@th-koeln.de

# Kurzfassung

Die vorliegende Bachelorarbeit beschäftigt sich mit der *Konzeption und Implementierung einer mobilen Anwendung, die die Smartphone-Nutzung in Gaststätten mit Hilfe von Gamification einschränken soll*. Hierbei handelt es sich um einen Prototypen, der in Android geschrieben wird und sich noch in der Entwicklungsphase befindet. Die Grundsiedee dieser Arbeit basiert zum einen auf das Thema *Phubbing*<sup>1</sup>, welches in dem Modul Praxisprojekt<sup>2</sup> zusammen mit einem Kommilitonen mithilfe einer mobilen Anwendung bearbeitet wurde. Zum anderen wurde zusätzlich eine Umfrage an Gaststätten in Köln selbstständig durchgeführt, um Meinungen zu diesem Konzept zu erhalten bevor die Thematik innerhalb der vorliegenden Arbeit vertieft wurde. Darüber hinaus wurden mit zwei Testpersonen ein Usability-Test durchgeführt, um die Meinung über den Wi-reframe zu erhalten. Mit diesem Wissen wurden Anforderungen zur Erstellung des Prototypen erstellt und infogedessen sollen Lösungsansätze zur Einschränkung des Smartphones implementiert werden. Hierfür wurden Komponenten angewendet, die dazu verhelfen sollen, das Problem im Zusammenhang mit der Smartphone-Nutzung in Gaststätten zu lösen. Das Ziel dieser Bachelorarbeiter besteht darin mit der bereits genannten mobilen Anwendung die Smartphone-Nutzung im Kontext auf Gaststätten zu minimieren. Hierbei ist zu erwähnen, dass die Gaststätten in Bezug auf die Implementierung vorerst nicht berücksichtigt wurden.

---

<sup>1</sup>bezeichnet die Angewohnheit, sich mit dem Smartphone zu beschäftigen, während man die Menschen, mit denen man gerade gesellschaftlich verkehrt, vernachlässigt.

<sup>2</sup>Ein Pflichtmodul aus dem Studiengang Wirtschaftsinformatik

# **Eidesstattliche Erklärung**

Ich versichere, die von mir vorgelegte Arbeit selbstständig verfasst zu haben.

Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten oder nicht veröffentlichten Arbeiten anderer entnommen sind, habe ich als entnommen kenntlich gemacht. Sämtliche Quellen und Hilfsmittel, die ich für die Arbeit benutzt habe, sind angegeben.

Die Arbeit hat mit gleichem Inhalt bzw. in wesentlichen Teilen noch keiner anderen Prüfungsbehörde vorgelegen.

Gummersbach, 20. Februar 2018

Vu Phi Hai Dinh

# **Danksagung**

An dieser Stelle möchte ich mich bei all denjenigen bedanken, die mich während der Anfertigung dieser Bachelorarbeit unterstützt und motiviert haben.

Mein Dank gilt zunächst Herrn Prof. Dr. Matthias Böhmer, der mich seit dem Modul Praxisprojekt begleitet und diese Bachelorarbeit als Erstprüfer betreut und begutachtet hat. Für die hilfreichen Anregungen, tolle Unterstützung und die konstruktive Kritik bei der Erstellung dieser Arbeit möchte ich mich recht herzlich bedanken.

Ebenfalls möchte ich mich bei Herrn Prof. Dr. Christian Kohls für die Bereitschaft als Zweitprüfer und das Interesse an dieser Arbeit herzlich bedanken.

Ein weiterer Dank gilt an all die Gaststätten in Köln, die sich die Zeit für meine Umfrage genommen haben und mit ihren persönlichen Erfahrungen und Meinungen zu diesem Thema einen Grundbaustein dieser Arbeit gelegt haben.

Auch möchte ich mich bei meinen Korrekturlesern, die viel Zeit in die Korrektur meiner Arbeit investiert haben und an meine Probanden, die meinen Wireframe evaluiert haben, bedanken.

Abschließend möchte ich mich bei meinen Eltern herzlich bedanken, die mir mein Studium durch ihre tatkräftige Unterstützung ermöglicht haben und mich in jeglichen Situationen unterstützt haben.

# Inhaltsverzeichnis

<b>Kurzfassung</b>	<b>I</b>
<b>Danksagung</b>	<b>III</b>
<b>Glossar</b>	<b>IV</b>
<b>1. Einleitung</b>	<b>1</b>
1.1. Problematik der Smartphone-Nutzung in Gaststätten . . . . .	1
1.2. Motivation zur Unterbindung der Smartphone-Nutzung . . . . .	2
1.3. Zielsetzung zur Einschränkung der Smartphone-Nutzung in Gaststätten . . . . .	2
1.4. Aufbau der Arbeit . . . . .	3
<b>2. Grundlagen</b>	<b>4</b>
2.1. Gamification als Motivationssteigerung mittels Belohnungssystem . . . . .	4
2.1.1. Begriffserklärung Gamification . . . . .	4
2.1.2. Spielmechaniken in Gamification . . . . .	5
2.2. Beacon zur Lokalisierung des Standortes . . . . .	7
2.2.1. Begriffserklärung Beacon . . . . .	7
2.2.2. Beacon Ranging and Monitoring (Background Scanning) . . . . .	9
2.3. Firebase zur Anmeldung der mobilen Anwendung . . . . .	10
2.3.1. Begriffserklärung Firebase . . . . .	10
2.3.2. Firebase Authentication . . . . .	10
2.4. Usability Test mit Paper Prototyping . . . . .	11
2.4.1. Begriffserklärung Paper Prototyping . . . . .	11
2.4.2. Ablauf des Usability-Tests . . . . .	11
<b>3. Konzeption</b>	<b>12</b>
3.1. Anwendungsfall der mobilen Anwendung . . . . .	12
3.1.1. Sicht des Nutzers . . . . .	12
3.1.2. Sicht der Gaststätten . . . . .	13
3.2. Paper Prototyping . . . . .	13
3.2.1. Erstellung des Wireframes . . . . .	13
3.2.2. Evaluation mit zwei Testpersonen . . . . .	16
3.2.3. Feedback zur Evaluierung des Paper Prototyping . . . . .	17
3.3. Umfrage an Gaststätten . . . . .	18
3.3.1. Fragestellung an die Gaststätten . . . . .	18
3.3.2. Feedback der Umfrage . . . . .	18
3.3.3. Resultat der Umfrage . . . . .	19
3.4. Anforderungsermittlung der Anwendung . . . . .	20
3.4.1. Funktionale Anforderungen . . . . .	20

## *Inhaltsverzeichnis*

3.4.2. Nicht-funktionale Anforderungen . . . . .	22
3.5. Architekturentwurf der Anwendung . . . . .	23
<b>4. Implementierung</b>	<b>24</b>
4.1. Komponente zur Lösung der Problematik . . . . .	24
4.1.1. Anmeldung mit Facebook API über Firebase Authentication . .	24
4.1.2. Lokation mit Estimote Beacon . . . . .	25
4.1.3. Timer für die visuelle Darstellung der Zeit im Smartphones . .	26
4.1.4. Einschränkung des Smartphones . . . . .	28
4.1.5. Belohnungssystem von ShutPhone . . . . .	33
4.2. Lokales Speichern auf dem Smartphone . . . . .	35
4.3. Ablauf und Umsetzung des Prototypen . . . . .	36
<b>5. Fazit und Ausblick</b>	<b>40</b>
<b>Quellencodeverzeichnis</b>	<b>42</b>
<b>Abbildungsverzeichnis</b>	<b>43</b>
<b>Literaturverzeichnis</b>	<b>45</b>
<b>A. Anhang</b>	<b>46</b>
A.1. Vollständige Klasse: KioskService . . . . .	46
A.2. vollständige Methode onEnterdRegion() . . . . .	48
A.3. vollständige Methode onExitedRegion() . . . . .	49

# 1. Einleitung

Die vorliegende Bachelorarbeit beschäftigt sich mit dem Vorhaben mittels einer mobilen Anwendung die Smartphone Nutzung in Gaststätten zu senken. Der Ablauf ist chronologisch aufgebaut und beinhaltet die Vorgehensweise und Lösungsansätze zur Bewältigung der Problematik. Wichtige Abwägungen und Entscheidungen wurden dokumentiert.

## 1.1. Problematik der Smartphone-Nutzung in Gaststätten

Allgegenwärtig ist heutzutage das Smartphone, das dem Nutzer mit seiner weiten Auswahl an Anwendungen und Funktionen das Alltagsleben erleichtert. Der Anklang dieses Komforts spiegelt sich in der stetigen Interaktion zwischen Mensch und Smartphone in diversen Tätigkeiten wider; sei es das Surfen im Internet, die Benutzung von Spiele-Apps und sozialen Netzwerken, das Fotos schießen, die telefonische Erreichbarkeit oder die Beschäftigung beim Essen. Die Anwendungsbereiche des Smartphones sind vielfältig und werden in jeglichen Lebenslagen genutzt, wie beispielsweise bei Gaststättenbesuchen. Dabei ist es irrelevant, ob die Gäste alleine oder in Begleitung unterwegs sind. Das Smartphone wird überall und zu jeder Zeit benutzt und das birgt nicht nur Vorteile, sondern auch Nachteile.

In einer Umfrage von *OpenTable* auf der Webseite von *PREGAS GmbH*<sup>1</sup> wurden über 1000 Leute befragt, wie die Befragten die Smartphone-Nutzung des Gesprächspartners im Restaurant finden. Dabei stellte sich heraus, dass 97,5 Prozent der Befragten sich dabei gestört fühlen (PREGAS-Presseportal, 2016).

In diesem Fall liegt die Ursache darin, dass die Personen jederzeit auf ihr Smartphone zugreifen können und während eines Aufenthalts in einer Gaststätte ebenso die Möglichkeit besteht, Anrufe oder Nachrichten zu erhalten. Anstatt das Smartphone auszuschalten oder auf Flugmodus einzustellen, wird entweder der Anruf angenommen oder auf Nachrichten geantwortet. Das Bedürfnis darauf zu reagieren ist hoch, jedoch könnte die Lautstärke beim Telefonieren oder bei der Nutzung des Smartphones, die anderen Gäste stören. Infolgedessen kann das Essen beispielsweise durch ein Telefonat in der Zwischenzeit kalt werden, falls der Fokus dauerhaft auf dem Smartphone liegt. Das Führen einer Konversation mit Freunden während einer gemeinsamen Mahlzeit, kann durch das Smartphone kontinuierlich unterbrochen werden.

Aufgrund dieser Tatsache werden die geschilderten Problematiken im Laufe der Bachelorarbeit näher behandelt, um diesen entgegenzuwirken.

---

<sup>1</sup><http://pregas.de>

## 1.2. Motivation zur Unterbindung der Smartphone-Nutzung

Die Bachelorarbeit entstand zum einen aus einem vorherigen Praxisprojekt<sup>2</sup>, in dem das Thema Phubbing bearbeitet wurde. Mit einem Kommilitonen wurde eine mobile Anwendung in Android entwickelt, die die mobilen Anwendungen des gegenüberliegenden Smartphones blockieren kann, wenn sich mindestens ein Freund in der unmittelbaren Nähe befindet. Mit dieser Erfahrung entstand die Idee ein ähnliches Konzept mit dem Bezug auf Gaststätten zu entwickeln. Da in dem zuvor erwähnten Praxisprojekt bereits eine mobile Anwendung entwickelt wurde, bestand das Interesse weitere Methoden zur Einschränkung von Smartphones zu entdecken. Zum anderen wurde die Idee mithilfe einer Umfrage an verschiedenen Gaststätten in Köln untermauert. Demzufolge verstärkte sich die persönliche Motivation den Umstand, der zu hohen Smartphone-Nutzung in Gaststätten einzuschränken und die Kenntnisse in den Bereichen Android und Java zu erweitern.

## 1.3. Zielsetzung zur Einschränkung der Smartphone-Nutzung in Gaststätten

Die Lösung für die Problematik werden mit Zielsetzungen definiert, die mithilfe der angewandten Methoden erreicht werden können. Zu diesem Zweck wurde eine Zielhierarchie aufgestellt, mit der die Ziele mit den Verben "muss, soll und kann" priorisiert werden.

### Strategische Ziele:

- Die Nutzung des Smartphones muss in Gaststätten minimiert werden.

### Taktische Ziele:

- Es soll eine mobile Anwendung entwickelt werden, die die Funktionalität des Smartphones einschränkt.
- Der Nutzer muss für die Nicht-Nutzung des Smartphones honoriert werden.

### Operative Ziele:

- Der Bereich der Problematik muss detaillierter analysiert werden.
- Es muss ein Prototyp entwickelt werden, der die zu entwickelnde Anwendung repräsentiert.
- Es sollen Testläufe der Anwendung durchgeführt werden.
- Es sollen erste UI-Lösungen entwickelt werden, die in Form von Wireframes getestet werden.
- Es kann eine Umfrage durchgeführt werden, der die Meinungen der Gaststätten berücksichtigt.

---

<sup>2</sup>Ein Pflichtmodul Studiengang Wirtschaftsinformatik

## 1. Einleitung

- Es sollen Rabatte als Belohnungen für die Nutzern geben, wenn das Smartphone in der Gaststätte nicht genutzt wird.

### 1.4. Aufbau der Arbeit

Im weiteren Verlauf der Bachelorarbeit soll im zweiten Kapitel *Grundlagen* wichtige Bestandteile dieser Arbeit dargestellt werden, um ein besseres Verständnis der einzelnen Begriff zu erhalten. Hierin wird zunächst die Bedeutung von *Gamification* erläutert und die grundlegenden Spielmechaniken erwähnt. Danach werden die Begriffe *Beacon* und *Firebase* erklärt, die für die Implementierung der Anwendung benötigt werden. Am Ende wird das *Paper Prototyping* und der Ablauf des Usability-Tests erläutert, um die Vorgehensweise zu verstehen.

Das dritte Kapitel *Konzeption* stellt den theoretischen Aufbau der mobilen Anwendung dar. Dafür wird ein Anwendungsfall aus der Sicht des Nutzers, die die Anwendung benutzen und aus der Sicht der Gaststätten beschrieben. Darauf folgt sowohl die Umfrage an Gaststätten, als auch die Evaluation des Paper Prototypings mit Testpersonen um daraus in Verbindung mit der eigenen Idee, eine Anforderungsermittlung zu erstellen.

Im vierten Kapitel *Implementierung* werden durch die zuvor erwähnten Anforderungsermittlung die mobile Anwendung entwickelt. Dazu wurden Komponenten implementiert, die die Smartphone-Nutzung einschränken soll. Geschildert werden die Vorgehensweise und Lösungsansätze zur Implementierung der Anwendung und die aufgetretenen Probleme.

Abschließend wird im fünften Kapitel *Fazit und Ausblick* das Resultat im Zusammenhang mit der Zielsetzung der Bachelorarbeit wiedergegeben, die entstandenen Problemen bei der Umsetzung erläutert und potentielle Verbesserung und Features genannt.

## 2. Grundlagen

Im folgenden Kapitel werden Begrifflichkeiten erläutert, die im Laufe der Arbeit verwendet werden. Damit die Leser über die Kenntnisse der einzelnen Begriffe verfügen, wurden die Definitionen der Begriffe durch Literaturen, Internet Recherchen und eigenes Wissen wiedergeben.

### 2.1. Gamification als Motivationssteigerung mittels Belohnungssystem

#### 2.1.1. Begriffserklärung Gamification

Der Begriff *Gamification* umfasst eine Vielzahl von Definitionen, die von vielen Autoren unterschiedlich interpretiert wurden. Aus diesem Grund gibt es keine allgemeine Definition zu Gamification (Seaborn u. Fels, 2014, S. 14). Weswegen Deterding u. a. (2011) sich mit dem Thema beschäftigt und Gamification als „the use of game design elements in non-game contexts“ (Deterding u. a., 2011, S. 2) beschreibt. Eine fast identische Aussage gab Bendel (2013) und definiert Gamification wie folgt:

*„Gamification (auch «Gamifizierung» und «Spielifizierung» genannt) ist die Übertragung von spieltypischen Elementen und Vorgängen in spielfremde Zusammenhänge.“* (Bendel, 2013, S. 40)

Diese beiden Zitate verdeutlichen den Begriff *Gamification* im Kontext für diese Bachelorarbeit. Folglich werden die einzelnen Begriffe der Zitate kurz geschildert.

Zu den spieltypischen Elementen gehören z. B. Regeln, Ranglisten oder Auszeichnungen (Bendel, 2013, S. 40). Im Unternehmen kann mittels Gamification z. B. die Effizienz und Motivation gesteigert werden, indem die Mitarbeiter für die erledigten Aufgaben, Belohnungen oder Erfolge erhalten. Ein Beispiel wären die Prämien, die die Arbeitnehmer im Vertrieb erhalten, sobald sie die Firmenprodukte oder Dienstleistungen verkauft haben. Ein anderes Beispiel wurde von Sailer (2016) erwähnt, worin die Treuepunkte in Supermärkten mit Gamification in Verbindung gebracht werden:

*„Im Supermarkt können beispielsweise Treuepunkte gesammelt und später gegen Prämien oder Gewinne eingetauscht werden. Hierbei wird ein simples Punktesystem in einem Einkaufsprozess implementiert mit dem Ziel, die Kundenbindung an einen bestimmten Anbieter zu erhöhen.“* (Sailer, 2016, S. 5)

Somit besteht ebenfalls die Möglichkeit, Gamification in multiplen Bereichen einzusetzen. Darüber hinaus stammt der Begriff Gamification ursprünglich aus dem Unterhaltungs-, Marketing- und Werbebranche (Bendel, 2013, S. 40). In der heutigen Zeit werden

## 2. Grundlagen

auch mobile Anwendungen gamifiziert. Dabei belohnt die mobile Anwendung für die Bewältigung der gestellten Aufgaben, die durch individuelle oder kollaborative Leistungen erbracht wurden, die Anwender/Innen.

Aus der Sicht von Sailer u. a. (2017) ist das Ziel von Gamification, durch die Spiel-Design-Elemente, die in einem realen, nicht spielerischen Kontext eingefügt werden, die Motivation und Leistung eines Menschen zu fördern (Sailer u. a., 2017, S. 371).

### 2.1.2. Spielmechaniken in Gamification

Die grundlegenden Spielmechaniken sind Punkte, Level, Aufgaben und Belohnungen (Pohlmann, 2012). Mit diesen Spielmechaniken wird der Anreiz bestimmte Aufgaben zu bewältigen gefördert. Diese umfassen eine Vielzahl von Optionen, welche in diesem Abschnitt anhand eines Beispiels kurz erläutert werden.

Hierfür wird Pokemon Go<sup>1</sup> als Beispiel genommen. Pokemon GO ist ein Spiel von der Firma *Niantic*, das für Android und iOS entwickelt wurde. Diese basiert auf der *GPS*-Technologie und motiviert die Spieler/Innen dazu, sich in der realen Welt zu bewegen. Zu diesem Zweck wird eine virtuelle Figur im Spiel auf einer Karte abgebildet, die sich zeitgleich mit den Spielern bewegt. Das Ziel dieses Spiels ist es, virtuelle Wesen namens *Pokemon* zu fangen, trainieren, entwickeln und gegen andere Mitspieler/Innen in einer virtuellen Welt kämpfen zu lassen (Eadicicco, 2016). Dafür erhalten die Spieler/Innen Erfahrungspunkte, Belohnungen und Errungenschaften für ihren erstellten Charakter.

#### Aufgaben

Die Aufgaben werden durch bestimmte Interaktionen gelöst und sollen sowohl facettenreich als auch unterschiedliche Schwierigkeitsgrade bieten. Dabei ist es wichtig, den Anwendern und Anwenderinnen eine Herausforderung zu vermitteln, damit die Aufgaben nicht zu langweilig wirken. Zudem sollten die gestellten Aufgaben auch passend zum Kontext sein (Pohlmann, 2012). In *Pokemon GO* wäre eine passende Aufgabe, alle vorhandenen *Pokemons* zu fangen.

#### Punkte

Die Punkte stellen eine Möglichkeit dar, für eine bestimmte Tätigkeit, etwas zu erhalten bzw. zu sammeln. Dabei kann je nach Gestaltung diverse Definitionen von Punkten gemeint sein. Zum einen können die Punkte zum Einlösen für bestimmte Belohnungen sein und zum anderen können die Erfahrungspunkte, die durch die Aktivitäten gesammelt werden, gemeint sein. Trotz allem gilt für Pohlmann (2012) ein Resultat:

*“Punkte bilden den Grundstein eines jeden Gamification-Projektes.”* (Pohlmann, 2012).

---

<sup>1</sup><https://www.pokemongo.com/de-de/>

## 2. Grundlagen

### Level

Das Level soll den Fortschritt im Prozess darstellen. Mit den gesammelten Erfahrungspunkten werden den Anwendern und Anwenderinnen z. B. durch die Fertigstellung von Aufgaben, einen Rangauftstieg gewährleistet.



Abbildung 2.1.: Gamification: Profil vom Verfasser

Zur Verdeutlichung zeigt das Beispiel in Abbildung 2.1 ein Profil vom Verfasser aus dem Spiel Pokemon GO. Darauf ist das Level und die gesammelten Erfahrungspunkte des Profils ersichtlich. Um einen Stufenaufstieg zu erhalten, muss der Nutzer in diesem Spiel Erfahrungspunkte sammeln. Dies wird erreicht indem Pokemons gefangen werden und bildet damit eine von vielen Aufgaben dieses Spiels.

### Belohnungen und Errungenschaften

Den Anwendern und Anwenderinnen werden nach der Fertigstellung der Aufgabe ihre Belohnungen und Errungenschaften gegeben. Die Motivation und Bereitschaft zur Bewältigung der Aufgaben soll dadurch gesteigert werden. Diese beiden Aspekte können vielfältig betrachtet werden, weswegen im Folgenden ein Beispiel gezeigt wird.

## 2. Grundlagen



Abbildung 2.2.: Gamification: Belohnungen und Errungenschaften nach Bewältigung der Aufgabe

Zu sehen ist in der Abbildung 2.2 ein Belohnungssystem von Pokemon GO. Hierbei wird nach der erfolgreichen Beendung der Aufgabe, den Spielern sowohl Gegenstände als auch Erfahrungspunkte als Belohnung gegeben. Mit diesen Belohnungen hat der Spieler und die Spielerin die Option gewisse Interaktionen durchführen, wie z. B. einen Gegenstand zum Fangen eines Pokemons benutzen. Als Errungenschaft ist das Fangen des Pokemons gemeint. Mit dem Ziel, dass die Spieler/Innen regelmäßig das Spiel spielen, gibt es so genannte Tagesaufgaben, um weitere Belohnungen zu erhalten und in diesem Fall höherer Erfahrungspunkte einbringt.

## 2.2. Beacon zur Lokalisierung des Standortes

### 2.2.1. Begriffserklärung Beacon

Der Beacon ist ein Sender oder Empfänger, der auf Basis einer Bluetooth Low Energy oder Bluetooth Smart Technologie fungiert. Mit dem Beacon können kleinere Mengen an Daten per Bluetooth auf das Smartphone gesendet werden. Dazu kommuniziert der Beacon mit einem Smartphone über eine mobile Anwendung, sobald dieser sich in Reichweite befindet und kann darauffolgend z. B. Push-Benachrichtigungen versenden. Die Reichweite der Verbindung ist je nach Entwicklungsstand und Hersteller unterschiedlich, kann aber bis zu 50 Meter betragen. Dabei ist zu erwähnen, dass bei der Datenübertragung zwischen den Geräten weniger Energie verbraucht wird als bei der normalen Bluetooth-Technologie. Daraus resultiert ein geringer Batterieverbrauch beim Smartphone. (OnlineMarketing.de). Die Beacons ermöglichen nicht nur einen Datenübertragung, sondern können auch durch mehrere Sender den Standort eines Smartphones mittels UUID<sup>2</sup>, also die ID des Gerätes genauer bestimmen (Böhme, 2015). Je mehr Beacons sich in einem Raum befinden, desto genauer können diese den Standort ermitteln. Mithilfe eines Beispiels soll die Vorgehensweise des Beacons

---

<sup>2</sup>Universally Unique Identifier

## 2. Grundlagen

genauer geschildert werden.

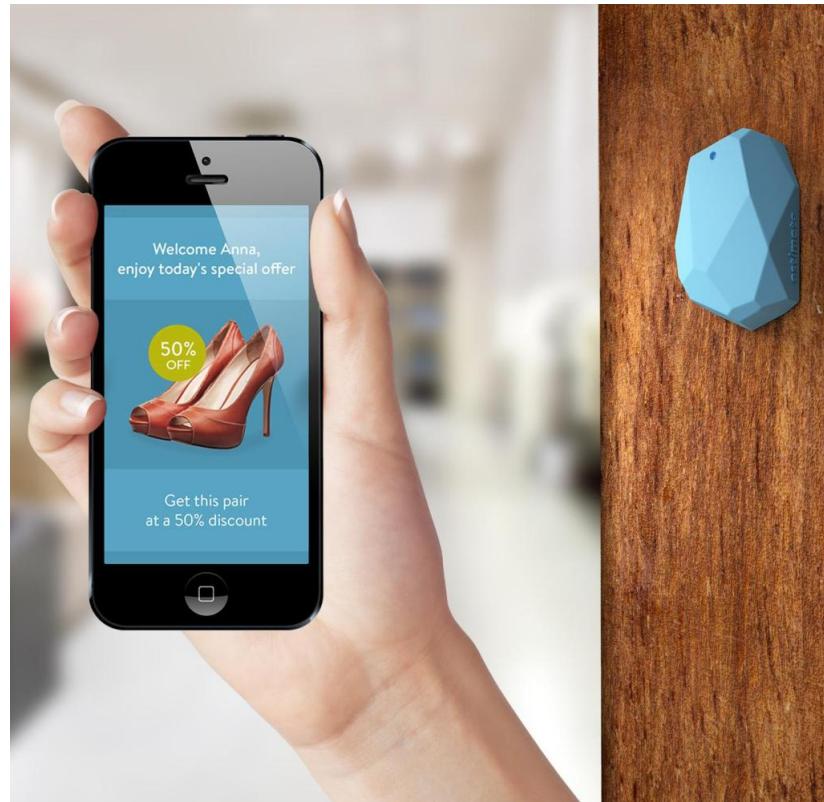


Abbildung 2.3.: Beacon Beispiel: *Supermarkt-Schnäppchen funkeln aufs Handydisplay*  
Quelle: Denker (2014)

In dieser Abbildung 2.3 ist ein Beacon zu erkennen, der Produktangebote auf dem Smartphone anzeigt. Hierbei handelt es sich um eine Situation in einem Einzelhandel. Dieser verfügt über eine mobile Anwendung, welche der Nutzer auf dem Smartphone installiert hat und den Beacon unterstützt. Damit das Smartphone die Informationen über bestimmte Produktangebote erhalten kann, muss die Bluetooth-Funktion des Smartphones aktiviert sein. Dadurch wird ein Ereignis in der Anwendung ausgelöst, wenn sich der Nutzer in der Reichweite des Beacons befindet. Die Produktangebote des Einzelhandels werden per Push-Benachrichtigung auf dem Smartphone verschickt.

### 2.2.2. Beacon Ranging and Monitoring (Background Scanning)

Die Beacons können auf zwei verschiedene Arten, mit Beacon Ranging und Beacon Monitoring, das Smartphone bzw. die mobile Anwendung lokalisieren. Diese beiden Methoden werden kurz erläutert, mit dem Ziel den Vorgang beim Erkennen des Smartphones zu verdeutlichen.

#### Monitoring

Beacon Monitoring bietet die Möglichkeit eine definierte Reichweite um den Beacon zu erschaffen, die bei der Erkennung des Smartphones über eine mobile Anwendung ein Ereignis auslöst. Dadurch erkennt das Smartphone, ab welchem Zeitpunkt es sich im Radius des Beacons befindet und wieder verlässt. Dies wird in der Abbildung 2.4 verdeutlicht. Dabei spielt es keine Rolle, ob die Anwendung geschlossen oder im Hintergrund läuft.

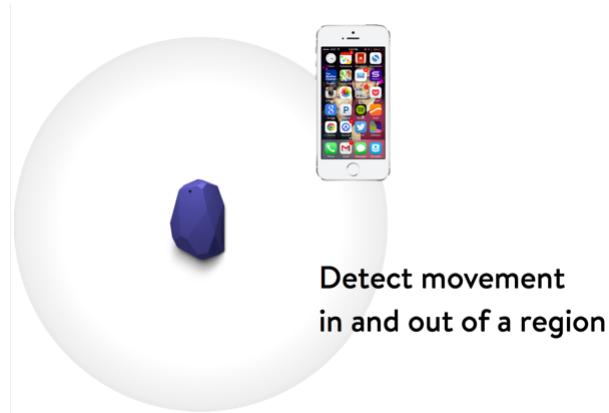


Abbildung 2.4.: Beacon: Beispiel zum Monitoring  
Quelle: Malek (2017)

#### Ranging

Beim Ranging erkennt der Beacon im Gegensatz zu Monitoring nicht, wer sich innerhalb oder außerhalb der Reichweite befindet. Anhand der Abbildung 2.5 wird stattdessen die Distanz zwischen dem Beacon und der dazugehörigen mobilen Anwendung gemessen. Durch ein Beispiel wird der beschriebene Begriff verdeutlicht. Falls mehrere Beacons mit verschiedenen Interaktionen in der Nähe sind und sich ein Smartphone näher an dem Beacon befindet, als an einem anderen Beacon, wird die Interaktion vom näheren Beacon ausgeführt. Dazu muss die Anwendung im Vordergrund laufen oder mit dem Monitoring zusammen arbeiten, um im Hintergrund laufen zu können (Malek, 2017).

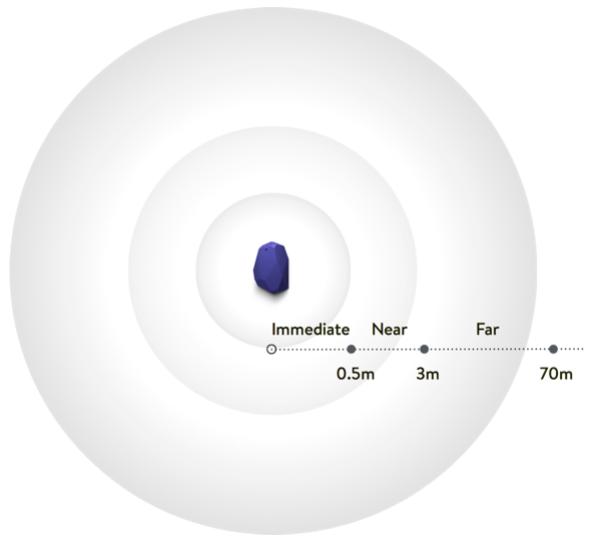


Abbildung 2.5.: Beacon: Beispiel zum Ranging  
Quelle: Malek (2017)

## 2.3. Firebase zur Anmeldung der mobilen Anwendung

### 2.3.1. Begriffserklärung Firebase

Firebase<sup>3</sup> ist eine Plattform von Google für Entwickler, die dazu verhelfen soll mit verschiedenen Tools eine Anwendung einfacher zu entwickeln. Hierbei stellt Firebase eine große Auswahl an Tools zur Verfügung und bietet zusätzlich ein SDK<sup>4</sup> an, die über verschiedene Bibliotheksklassen verfügen. Diese Plattform ist systemunabhängig und unterstützt Android, iOS und Webanwendungen (Schuster, 2017). Einer der Tools „Authentication“ wird in dieser Arbeit verwendet.

### 2.3.2. Firebase Authentication

Firebase Authentication ist eines von vielen Tools, welche Firebase anbietet. Unzählige Anwendungen möchten die Identität bzw. die persönlichen Daten des Nutzers in Erfahrung bringen und benutzen dafür verschiedene Anmeldemethoden. Dafür verwaltet dieses Tool die Nutzer über diverse Drittanbieter wie z. B. Facebook, Google, Twitter, GitHub oder per E-Mail und Telefonnummer. Mittels des Firebase SDK soll die Implementierung zur Authentifizierung von Benutzern vereinfacht werden.

---

<sup>3</sup><https://firebase.google.com/>

<sup>4</sup>Software Development Kit

## 2.4. Usability Test mit Paper Prototyping

### 2.4.1. Begriffserklärung Paper Prototyping

Paper Prototyping ist eine Methode zur visuellen Darstellung des Prototyps. Mithilfe von Utensilien wie Papier, Schere, Stift usw., kann eine Benutzeroberfläche und die zugehörigen Interaktionen abgebildet werden. Diese Methode wird in der Software- oder Webentwicklung eingesetzt, um in einem frühen Stadium der Entwicklung einen Eindruck für Testpersonen zu erhalten, ohne etwas programmiert zu haben. Zudem ist die Methode schnell, kostengünstig und benötigt keine Tools zur Erstellung des Prototypen.

Ziel des Paper Prototypings ist es, ein Feedback der Testpersonen bezüglich der optischen Darstellung, seiner Wirkungsweise und der Verständlichkeit der Anwendung zu erhalten, um auf diese Weise potenzielle weitere Anforderungen zu ermitteln.

### 2.4.2. Ablauf des Usability-Tests

Zunächst muss eine Benutzeroberfläche erstellt werden, die die Vorstellung des Konzeptes widerspiegelt. Diese soll nicht zu komplex dargestellt werden und trotzdem jegliche Informationen beinhalten, die für die Anwendung relevant sind. Mithilfe von Aufgabenstellungen, die die Testpersonen bewältigen müssen, sollen die Nutzerfreundlichkeit und die Funktionalität geprüft und evaluiert werden.

Die Aufgabenstellungen müssen sich auf den Kontext der Anwendung beziehen. Dabei wird auf das Verhalten der Testperson geachtet, wie diese die Aufgaben löst und ob Missverständnisse beim Durchführen entstehen. Die Testperson soll beim Vorgang laut denken, damit ein Protokollant alle Aktivitäten notieren kann. Bei der Ausführung der Aufgaben soll die Testperson mit der Benutzeroberfläche interagieren. Der Moderator, der den Test leitet, muss den Prozess anstatt der Anwendung, ausführen. Je nachdem, was die Testperson bei der Aufgabe tut, wird das dazugehörige Blatt hinzugefügt oder entfernt. Am Ende resultiert ein Feedback, welche möglichen Problematiken, Vorschläge oder positive Erfahrungen der Testperson wiedergibt.

# **3. Konzeption**

Damit die mobile Anwendung realisiert werden kann, muss zunächst ein Konzept festgelegt werden, um die Idee zu repräsentieren. Darauf folgt eine Umfrage an Gaststätten, um die Meinungen der Mitarbeiter zu dem Konzept zu erhalten. Außerdem wurden auch zwei Probanden gefragt, den Prototyp auf Papier zu evaluieren. Die daraus abgeleitete Resultate wurden dokumentiert und als Grundlage für die Implementation entnommen. Im Verlauf der Bachelorarbeit wird der Name „ShutPhone“ erwähnt, der die mobile Anwendung repräsentiert.

## **3.1. Anwendungsfall der mobilen Anwendung**

### **3.1.1. Sicht des Nutzers**

Zunächst stellt sich die Frage, welchen Nutzen die Anwendung haben soll. Aus diesem Grund wird an dieser die Idee des Konzeptes erläutert. Die praktische Verwendung der mobilen Anwendung wird aus zwei Sichtweisen betrachtet. In diesem Fall vom Nutzer und von den Gaststätten. Im Folgenden wird vorerst aus der Sichtweise des Nutzers erläutert.

Der Nutzer soll die Möglichkeit haben, ungestört alleine oder mit seinem Mitmenschen in Gaststätten zu kommunizieren und das Essen zu genießen. Damit der Anreiz für die Nutzung der mobilen Anwendung besteht, soll zunächst ein Timer für die optische Darstellung angezeigt werden. Mit dem Betätigen des Start-Button wird gleichzeitig der Timer aktiviert, der anschließend die Zeit hoch zählt und das Smartphone blockiert. Das bedeutet, dass die Nutzung des Smartphones vorübergehend eingeschränkt wird. Der Start-Button kann nur betätigt werden, falls der Nutzer sich in einer Gaststätte befindet, die die mobile Anwendung mit einem Beacon unterstützt. Wenn eine bestimmte Zeit wie beispielsweise eine Stunde erreicht wurde, erhält der Anwender eine gewisse Punktzahl und kann dies gegen Angebote der Gaststätte wie z. B. kostenlose Getränke, vergünstigte Gerichte oder Desserts einlösen. Die Angebote sind von Gaststätten zu Gaststätten unterschiedlich. Es bestehen zwei Möglichkeiten, um die Einschränkung des Smartphones zu stoppen. Die erste Möglichkeit wäre über den implementierten Stop-Button die Zeit anzuhalten. Zwischenzeitlich kann der Nutzer die Funktionalitäten des Gerätes wieder nutzen, aber verliert unterdessen die Zeit, die der Nutzer bei der Nicht-Nutzung des Smartphones haben könnte. Solange der Nutzer sich in der Gaststätte befindet, kann er den Timer beliebig anhalten und wieder starten. Die zweite Möglichkeit wäre aus der Reichweite des Beacons beziehungsweise der Gaststätte zu gehen. Somit wäre der Zugang zum Smartphone wieder verfügbar. Die Punkte, die der Nutzer erhält, werden gespeichert und können entweder sofort oder beim nächsten Besuch eingelöst werden. Dieser Sachverhalt wird als Gamification

### 3. Konzeption

bezeichnet und wurde im Abschnitt Grundlagen erklärt. Hierbei motiviert ShutPhone den Nutzern, das eigene Smartphone zu blockieren, damit der Nutzer Belohnungen für seine Handlung erhält.

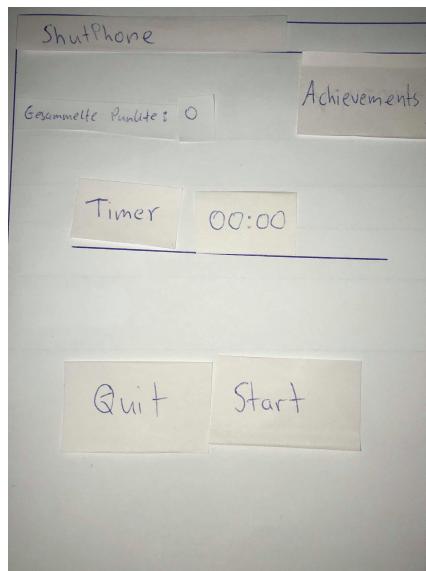
#### 3.1.2. Sicht der Gaststätten

Da heutzutage ein Großteil der Bevölkerung ein Smartphone besitzt, würde sich die Möglichkeit für die Gaststätte ergeben, über den Beacon eine Push-Benachrichtigung in Form von Mittagsmenüs, Tagesangebote, usw., an ein Smartphone zu versenden. Auf diese Weise könnte sich das Image und der Bekanntheitsgrad mittels *ShutPhone* mit der Anpreisung von lukrativen Angeboten verbessern. Des Weiteren würde sich durch ein Belohnungssystem in dem einfache Aufgaben zu erfüllen sind, der Kundenstamm stetig festigen. Somit könnte es die Motivation des Kunden sein, des Öfteren die Gaststätten zu besuchen, um die Belohnungen zu erhalten. Aus der Sicht der Gaststätten dient *ShutPhone* zur Kundengewinnung und der gezielten Vermarktung seiner Produkte.

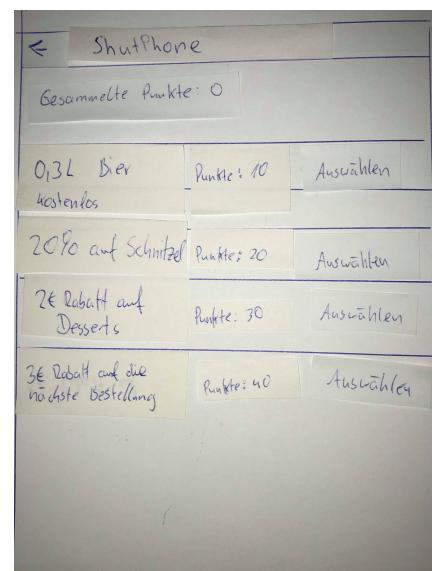
## 3.2. Paper Prototyping

#### 3.2.1. Erstellung des Wireframes

In diesem Punkt wird mit Hilfe eines Wireframes die visuelle Darstellung der mobilen Anwendung erstellt. Die Anwendung soll schlicht und einfach aufgebaut sein. Hierbei handelt es sich nur um die Bedienbarkeit der Anwendung, die vereinfacht dargestellt wird. In den folgenden Abbildungen werden die erstellten Wireframes kurz erklärt.



(a) Paper Prototyping - Start der Anwendung



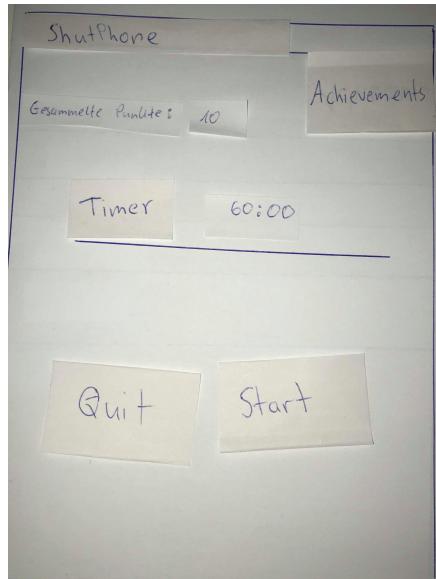
(b) Paper Prototyping - Belohnungsliste

Abbildung 3.1.: Paper Prototyping - Anfangszustand der Anwendung

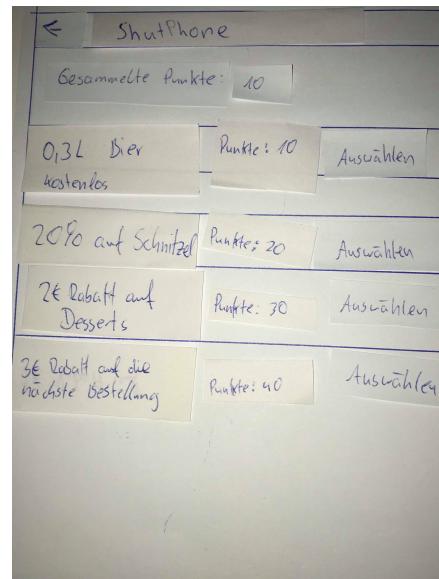
### 3. Konzeption

In Abbildung 3.1(a) ist der Anfangszustand vom ShutPhone zusehen. Es sind insgesamt drei Buttons, der Start-Button, der Quit-Button und der Achievement-Button, die betätigt werden können. Mit der Betätigung des Start-Buttons beginnt der Timer die Zeit zu zählen. Der Quit-Button hingegen hält die Zeit an und der Achievement-Button navigiert die Anwendung in ein neues Fenster, in dem die Belohnungen aufgelistet sind. Oben links stehen die gesammelten Punkte, die der Nutzer in der Zeit gesammelt hat.

Die darauf folgende Abbildung 3.1(b) zeigt den Achievement Bereich. Es ist eine Liste von möglichen Belohnungen, die der Nutzer in einer Gaststätte erhalten kann. Diese können gegen die gesammelten Punkte eingetauscht werden. Mit dem Auswählen-Button löst der Nutzer einen bestimmten Achievement ein, falls die zu erreichenden Punkte vorhanden sind. Der aktuelle Punktstand wird hierbei ebenso für das Verständnis des Nutzers abgebildet.



(a) Paper Prototyping - Bestimmte Zeit erreicht

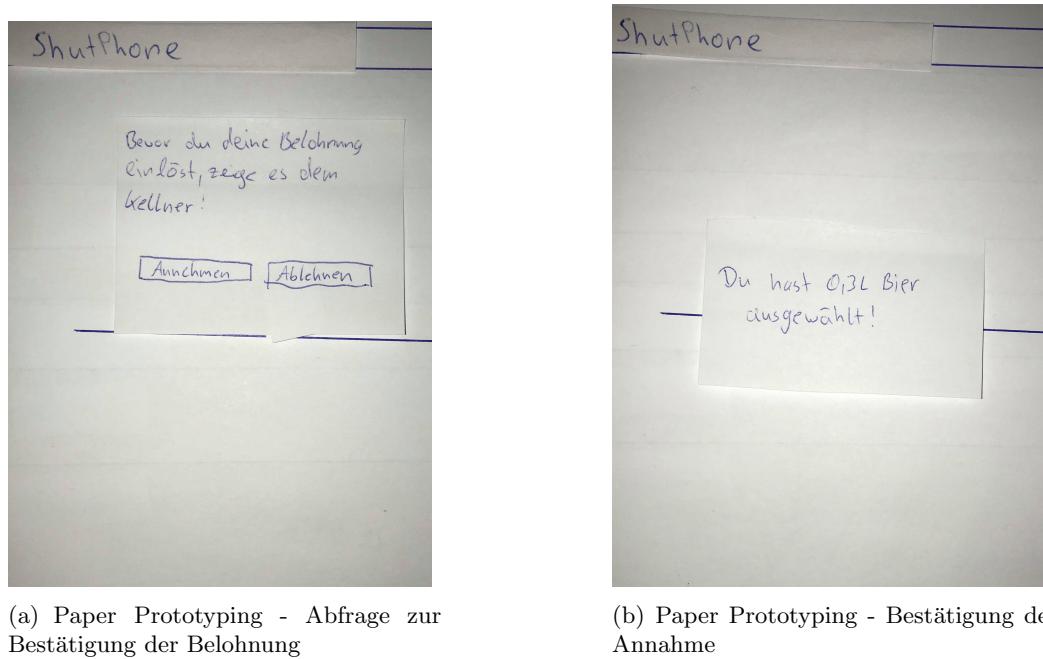


(b) Paper Prototyping - Punkte gesammelt

Abbildung 3.2.: Paper Prototyping - Ablauf der Anwendung

In diesem Beispiel 3.2(a) wurde der Start-Button betätigt. Der Nutzer hat derweil keine Möglichkeit die Funktionalitäten des Smartphones zu benutzen. Ab der 60. Minute erhält der Nutzer, für das Erreichen der Zeit, zehn Punkte. Wenn der Nutzer die Zeit über den Quit-Button stoppt, besteht die Möglichkeit den Achievement-Button zu drücken, um in den Achievement Bereich zu kommen und die Punkte zu verwerten. Hier folgt in Abbildung 3.2(b) der Achievement Bereich. Die erreichten zehn Punkte können in diesem Bereich benutzt werden. In diesem Fall hat der Nutzer die Möglichkeit mit den zehn Punkten ein 0.3L Bier gratis zu bekommen, wenn er es haben möchte. Dazu muss der Auswählen-Button betätigt werden.

### 3. Konzeption



(a) Paper Prototyping - Abfrage zur Bestätigung der Belohnung

(b) Paper Prototyping - Bestätigung der Annahme

Abbildung 3.3.: Paper Prototyping - Belohnung einlösen

In Abbildung 3.3(a) soll ein PopUp-Fenster angezeigt werden, worauf die Anfrage zur Bestätigung des Achievements erfolgt. Die Mitarbeiter müssen diese Anfrage sehen und die Bestätigung dafür geben. Ohne diese Abfrage besteht kein Anspruch auf die Belohnung, weil kein Beweis vorgelegt werden kann.

In der letzten Abbildung 3.3(b) wurde die Anfrage aufgenommen. Damit kann der Mitarbeiter die Belohnung in das System der Gaststätte eintragen und der Nutzer erhält ein kostenloses Bier. Die Punkte werden beim *Annehmen* verrechnet und der Nutzer kann weiterhin das Smartphone blockieren und Punkte sammeln, indem zurück auf die ShutPhone-Startseite navigiert wird.

### *3. Konzeption*

#### **3.2.2. Evaluation mit zwei Testpersonen**

Es wurden zwei Probanden gefragt, ob die Bereitschaft und die Zeit vorhanden wäre an einer Usability-Test teilzunehmen. Beide bejahten die Anfrage. Beide Probanden wurden einzeln beauftragt sich mit dem Wireframe und die dazu erstellten Aufgabenstellungen zu befassen. Zuvor haben die Probanden eine Erläuterung der Idee bzw. des Konzeptes erhalten. Die Aufgabenstellungen dienen zur Analyse des Wireframes und die darauf zu implementierte Benutzeroberfläche der Anwendung. Außerdem sollen mithilfe der Probanden, weitere Anforderungen zur Verbesserung des ShutPhones ermittelt werden.

##### Aufgabenstellungen

1. Wie finden Sie die UI auf dem ersten Blick?
2. Starten Sie die Zeit.
3. Erkennen Sie eine Veränderung des UI's?
4. Gehen Sie zu den Achievements.
5. Welchen Achievement könnten Sie auswählen?
6. Wählen Sie das Achievement.
7. Bestätigen Sie das Achievement.

##### Proband 1

1. Proband 1 findet das UI übersichtlich und einfach gehalten und versteht die einzelnen Darstellungen.
2. Der Start-Button wurde vom Probanden betätigt.
3. Dem Proband fällt auf, dass die Punktzahl sich um zehn erhöht hat.
4. Proband 1 hat den Achievement-Button betätigt. Hat aber nicht erkannt, dass zuvor die Zeit gestoppt werden muss.
5. Proband 1 sieht sich alle Belohnungen an und hat das 0,3L Bier ausgewählt.
6. Nach der Bestätigung bemerkt Proband 1, dass sich die Punktzahl auf null zurückgesetzt hat und hat daraufhin versucht die anderen Belohnungen auszuwählen.

### 3. Konzeption

#### Proband 2

1. Proband 2 findet UI simpel und übersichtlich.
2. Der Proband hat die Zeit über den Start-Button gestartet.
3. Ja, die Zeit läuft und hat die sechzigste Minute erreicht. Die Punkte haben sich um 10 erhöht.
4. Der Achievement-Button wurde betätigt, aber Proband 2 hat nicht erkannt, dass zuvor die Zeit gestoppt werden muss.
5. Proband 2 sieht sich zunächst alle Achievements an und erkennt, dass das Achievement 0,3L Bier die gleichen Punkte benötigt, die der Proband bereits erreicht hat und wählt es aus diesem Grund aus.
6. Der Proband hat 0,3L Bier über den Auswahl-Button eingelöst.
7. Bei der Bestätigung sieht der Proband ein PopUp-Fenster und hat die Frage gestellt, ob zuerst der Mitarbeiter das Fenster sehen muss oder nachdem man auf Annehmen gedrückt hat. Sicherheitshalber hat der Proband sich entschieden, erst den Mitarbeiter zu holen und dann den Bestätigungsbutton betätigt. Daraufhin kommt die Benachrichtigung, dass das Bier ausgewählt wurde und die Punkte abgezogen wurden.

#### **3.2.3. Feedback zur Evaluierung des Paper Prototyping**

Nach der Evaluation äußerten die Probanden Anregungen, die die Anwendung potenziell verbessern bzw. erweitern könnten.

Proband 1 wünscht sich eine Übersicht über die gesparten Kosten durch die Belohnungen. Zudem soll eine Benachrichtigung kommen, wenn eine erzielte Zeit erreicht wurde. Die Gesamtzeit bei der Nicht-Nutzung des Smartphones wäre für die Übersicht wünschenswert. Die gesammelten Punkten sollen auch in anderen Gaststätten einsetzbar sein.

Proband 2 gab als Anregung an, die Punkte, die gesammelt wurden auch in anderen Gaststätten anwenden zu wollen. Zwar weiß der Proband, dass es für die Gaststätte nicht geeignet ist, aber für den Nutzer wäre es gut. Zudem wünscht sich der Proband, dass die Zeit gespeichert wird, weil es Kunden gibt, die aus zeitlichen Gründen nicht allzu lange in einer Gaststätte bleiben können.

Im Großen und Ganzen finden die Probanden die Anwendung schlicht und einfach gehalten. Die Idee fanden beide Probanden sehr gut, da diese Problematik auch in ihren Freundeskreisen besteht. Eine Aufgabe haben beide Probanden nicht vollständig gelöst. In der vierten Aufgabe hatten beide nicht in Betracht gezogen zuerst den *Stop*-Button zu betätigen, bevor sie zu den *Achievements* gehen wollten. Dies muss besser verdeutlicht werden. Die Anregungen der Probanden zur Verbesserung der mobilen Anwendung werden berücksichtigt.

### 3.3. Umfrage an Gaststätten

#### 3.3.1. Fragestellung an die Gaststätten

Für die Umfrage an den Gaststätten in Köln wurden fünf Fragen chronologisch aufgestellt, dessen Thematik und Erläuterung des Konzeptes beinhalten. Diese Fragen sollen die Denkweise der Gaststätten zum Vorschein bringen und zugleich in Erfahrung bringen, ob das Interesse an der erwähnten Anwendung besteht. Hinzukommend können persönliche Erfahrungen der Mitarbeiter von Nutzen sein. Hauptsächlich fand die Umfrage mit Mitarbeitern statt, da die Manager zu diesem Zeitpunkt verhindert waren. Die Umfrage war eine spontane Handlung, weswegen die Gaststätten nicht darauf vorbereitet waren. Unter den befragten Gaststätten befinden sich: Restaurant Em Krützche (Köln), Aloha Bar (Köln), Wirtshaus Schwejk (Köln), Steakhaus Angusto (Köln) und Joe Champs - American Sportsbar and Restaurant (Köln).

1. Sagt Ihnen Phubbing etwas?
2. Schauen Sie manchmal auf die Gäste, ob sie ihren Smartphone benutzen? Sowohl alleine, als auch in Begleitung?
3. Konzept erklären.
4. Was halten Sie von dem Konzept?
5. Welche Anforderungen fehlen Ihnen in diesem Konzept?
6. Würden Sie die mobile Anwendung zur Einschränkung des Smartphone-Nutzens in ihrer Gaststätte einführen?

#### 3.3.2. Feedback der Umfrage

Auf die erste Frage, ob das Wort *Phubbing* bekannt ist, haben alle Gaststätten kein Vorwissen über den Begriff vorweisen können. Diese wurde anschließend vor Ort erläutert, damit die befragten Mitarbeitern einen Grundverständnis über das Thema haben.

Bei der zweiten Frage bejahten alle Mitarbeiter diese Frage. Die Mitarbeitern im Restaurant *Em Krützche* sagten: „Ja, besonders die jüngeren Gäste sind häufig am Smartphone.“ Der Betreiber von *Aloha Bar* meinte: „Ja. Die Gäste waren z. B. so sehr mit dem Smartphone beschäftigt, dass das Essen länger auf dem Tisch stand und dadurch kalt wurde.“ Im *Wirtshaus Schwejk* erzählte der Mitarbeiter: „Ja. Ich habe Situationen erlebt, wo ein Tisch ruhig war, weil die alle auf dem Smartphone gestarrt haben.“ *Steakhaus Angusto* sah ab und zu, dass das Essen nach dem Servieren unberührt blieb, weil die Kunden Fotos vom Essen machen wollten und *Joe Champs- American Sportsbar and Restaurant* erzählte: „Die Gäste sind ständig am Smartphone, um irgendetwas zu machen.“

Daraufhin wurde das Konzept dieser Arbeit erläutert, um in Erfahrung zu bringen, wie die Mitarbeitern der befragten Gaststätten darüber denken. *Em Krützche* sagte: „Ich finde die Idee gut und kenne das Problem in meinem eigenen Umfeld sehr gut.“ *Aloha Bar* meinte: „Tolle Idee. Ich kenne die Situation mit meinen Freunden selbst und finde es schade, dass die digitale Welt die Konversation stört.“ Der Mitarbeiter

### 3. Konzeption

von *Wirtshaus Schwejk* erklärte kurz und knapp: "Ich finde die Idee sehr gut.", worin der Mitarbeiter von *Steakhaus Angusto* eine ähnliche Aussage gegeben hat: "Ich finde die Idee gut, um wieder aus der virtuellen Welt raus zukommen." Zu guter Letzt sagte die Mitarbeiterin von *Joe Champs - American Sportsbar and Restaurant*: "Die Idee des Konzepts ist super."

Zu der dritten Frage hatten nur *Em Krützche* und *Aloha Bar* Anforderungen zu diesem Konzept gegeben. *Em Krützche* sagte: "Eine Übersicht über die Nutzung der Anwendung für die Mitarbeiter wäre vom Vorteil." Worauf der Betreiber von *Aloha Bar* sagte: "Es soll eine Möglichkeit geben, den Akkuverbrauch der Anwendung zu senken und ein Echtzeit Update beim Einlösen der Angebote bzw. Aktualisierung der Angebote wäre von Vorteil. Außerdem wäre eine grafische Übersicht der Nutzung für den Nutzer auch gut."

Auf die letzte Frage waren die Gaststätten geteilter Meinung. *Aloha Bar* und *Steakhaus Angusto* würden diese Anwendung benutzen. Woraufhin *Em Krützche* zu dieser Frage neutral blieb. Die Mitarbeitern meinten: "Die Idee ist gut, nur wäre die Anwendung in dieser Gaststätte unangebracht, da die Zielgruppe überwiegend ältere Kunden betrifft und die Preise im Vergleich zu anderen Gaststätten teurer sind." Wiederrum verneinten *Wirtshaus Schwejk* und *Joe Champs - American Sportsbar and Restaurant* die Nutzung dieser mobilen Anwendung. Der Mitarbeiter von *Wirtshaus Schwejk* sagte: "Ich habe meine Bedenken, ob die Mentalität der jüngeren Menschen sich ändern kann. Außerdem sind hier hauptsächlich nur ältere Kunden." Die Mitarbeiterin von *Joe Champs - American Sportsbar and Restaurant* meinte: "Die WLAN-Nutzung ist umsonst und viele Touristen kommen hierher, weil in der Nähe Hostels bzw. Hotels sind. Des Weiteren möchte der Geschäftsführer, dass die Kunden schnell ihre Mahlzeiten verzehren und den Platz für neue Kunden frei machen. Der Andrang ist besonders am Wochenende sehr groß."

#### 3.3.3. Resultat der Umfrage

Das Resultat zeigt eine eindeutige Meinung bezüglich der Theorie des Konzeptes. Alle Befragten fanden die Idee gut. Mit ihren eigenen Erfahrungen zum Thema Smartphone-Nutzung in Gaststätten befestigen die befragten Mitarbeiter diese Theorie. Hingegen gibt es unterschiedliche Meinungen bei der praktischen Anwendung des Konzeptes. Die Zielgruppe in den Gaststätten darf nicht zu alt sein. Außerdem hängt die mobile Anwendung von der Bereitschaft der Kunden ab. Dazu müssen gewisse Anreize vorhanden sein. Die mobile Anwendung muss zudem in Gaststätten anwendbar sein, die die Smartphone-Nutzung nicht mittels WLAN-Zugang fördert und den Kunden die Zeit gibt in Gaststätten zu verbringen. Die Anwendung kann nur auf gezielte Gaststätten angewendet werden, die die zuvor erwähnten Punkte berücksichtigen.

### 3.4. Anforderungsermittlung der Anwendung

Auf Grundlage der eigene Idee, der Umfrage und der Evaluation mit den Testpersonen wurden Anforderungen abgeleitet und ermittelt. Die folgenden Anforderungen dienen für die Implementation der einzelnen Funktionen der Anwendung. Hierbei unterscheiden sich die Anforderungsermittlung zwischen funktionalen und nicht-funktionalen Anforderungen. Die Formulierung der Anforderungen orientiert sich an der Anforderungsschablone aus der Literatur von Rupp u. Joppich (2014). Mit Hilfe der Verben "muss, soll und kann" werden die Anforderungen nach der Wichtigkeit priorisiert.

#### 3.4.1. Funktionale Anforderungen

Die funktionale Anforderungen beschreiben, was ein System tun soll und was das System leisten muss (Böhm, 2002, S. 139). Dazu werden im Folgenden die funktionalen Anforderungen der Anwendung "ShutPhone" aufgelistet:

- F10: Die Anwendung muss fähig sein, die Funktionalität des Smartphones einzuschränken.
  - F11: Die Anwendung muss fähig sein, den Zugang zu anderen Anwendungen zu blockieren.
  - F12: Die Anwendung muss fähig sein, den Zugang zu anderen Anwendungen zu entblocken.
  - F13: Die Anwendung muss fähig sein, das Smartphone nur zu blockieren, wenn der Nutzer sich in einer Gaststätte befindet, die die Anwendung unterstützt.
- F20: Die Anwendung muss fähig sein, das Smartphone in Gaststätten zu identifizieren.
  - F21: Die Anwendung soll fähig sein, beim Eintritt der Gaststätte eine Benachrichtigung an das Smartphone zu versenden.
  - F22: Die Anwendung soll fähig sein, beim Verlassen der Gaststätte eine Benachrichtigung an das Smartphone zu versenden.
- F30: Die Anwendung muss fähig sein, einen Timer für die verbrachte Zeit in der Gaststätte zu erstellen.
  - F31: Die Anwendung soll dem Nutzer die Möglichkeit bieten, den Timer in der Gaststätte zu starten.
  - F32: Die Anwendung soll dem Nutzer die Möglichkeit bieten, den Timer in der Gaststätte zu stoppen.
  - F33: Die Anwendung kann dem Nutzer die Möglichkeit bieten, die gestoppte Zeit zu speichern.
  - F34: Die Anwendung kann dem Nutzer die Möglichkeit bieten, beim nächsten Besuch den Timer bei der gestoppten Zeit zu starten.

### *3. Konzeption*

- F40: Die Anwendung muss fähig sein, die Angebote der Gaststätte auf dem Smartphone anzuzeigen.
- F50: Die Anwendung muss fähig sein, den Nutzer eine Belohnung zu geben.
  - F51: Die Anwendung muss dem Nutzer die Möglichkeit bieten, die Belohnungen zu erhalten.
  - F52: Die Anwendung muss dem Nutzer die Möglichkeit bieten, die Belohnungen zu verwerten.
  - F53: Die Anwendung kann dem Nutzer die Möglichkeit bieten, die angewendeten Belohnungen zu speichern.
- F60: Die Anwendung muss fähig sein, den Nutzer für das Erreichen einer Zeit, Punkte zu geben.
  - F61: Die Anwendung muss dem Nutzer die Möglichkeit bieten, die Punkte zu erhalten.
  - F62: Die Anwendung muss dem Nutzer die Möglichkeit bieten, die Punkte zu verwerten.
  - F63: Die Anwendung muss dem Nutzer die Möglichkeit bieten, die erhaltenen Punkten zu speichern.
  - F64: Die Anwendung kann dem Nutzer die Möglichkeit bieten, die erhaltenen Punkte in anderen Gaststätten zu verwerten.
- F70: Die Anwendung muss dem Nutzer die Möglichkeit bieten, einen Benutzerkonto in der Anwendung zu erstellen.
  - F71: Die Anwendung muss fähig sein, die persönlichen Daten des Nutzers zu speichern.
  - F72: Die Anwendung muss dem Nutzer die Möglichkeit bieten, sich zu authentifizieren.
  - F73: Die Anwendung soll dem Nutzer die Möglichkeit bieten, sich über soziale Netzwerke anzumelden.
  - F74: Die Anwendung soll dem Nutzer die Möglichkeit bieten, sich per Telefonnummer anzumelden.
- F80: Die Anwendung soll fähig sein, die Gaststätten, die die Anwendung nutzen, zu differenzieren.
- F90: Die Anwendung soll dem Nutzer die Möglichkeit bieten, eine Gesamtübersicht des Nutzers anzuzeigen.
  - F91: Die Anwendung kann dem Nutzer die Möglichkeit bieten, eine Übersicht über die ersparten Kosten in den Gaststätten anzuzeigen.
  - F92: Die Anwendung kann dem Nutzer die Möglichkeit bieten, eine Übersicht über die Nicht-Nutzung des Smartphones in den Gaststätten anzuzeigen.

### *3. Konzeption*

#### **3.4.2. Nicht-funktionale Anforderungen**

Die nicht-funktionalen Anforderungen beschreibt Böhm (2002), wie gut das System die Leistung erbringen muss (Böhm, 2002, S. 139). Wiederum beschreibt Eigner u. a. (2012) eine nicht-funktionale Anforderung als: “eine Randbedingung oder eine Qualitätsmerkmal, dem das System zu genügen hat“ (Eigner u. a., 2012, S. 52). Folglich werden die nicht-funktionalen Anforderungen der Anwendung aufgelistet:

- Die Anwendung muss auf dem Smartphone mit dem Android Betriebssystem laufen können.
- Das Smartphone, dass die Anwendung benutzt, muss mit dem Internet verbunden sein.
- Das Smartphone, dass die Anwendung benutzt, muss den Standort des Nutzers lokalisieren können.
- Das Smartphone, dass die Anwendung benutzt, muss die Bluetooth-Funktion an haben.
- Die Anwendung soll in Java entwickelt werden.
- Die Benutzeroberfläche der Anwendung muss übersichtlich und einfach dargestellt sein.
- Die Anwendung muss auf dem Smartphone einwandfrei laufen können.
- Die Sicherheit der persönlichen Daten des Nutzers muss gewährleistet sein.

### 3.5. Architekturentwurf der Anwendung

Bevor die Anforderungen implementiert werden, wird zunächst ein Architekturentwurf entwickelt. Die soll zeigen, aus welchen Komponenten die Anwendung besteht. Das Fundament zur Erstellung des Architekturentwurfs ist das Vorwissen aus anderen Modulen des Studiums. Hierbei handelt es sich um eine Client-Server-Architektur, welche in der Abbildung 3.4 zu sehen ist. Dabei ist der Client die mobile Anwendung, die in Android entwickelt wird. Hinzu kommen die Sensoren, welche in dieser Anwendung der Bluetooth und das GPS sind und dienen zur Kommunikation mit dem Estimote Beacon, damit der Standort des Smartphones lokalisiert werden kann. Im Background-Services finden lang andauernde Vorgänge im Hintergrund der Anwendung statt, die nicht direkt mit dem Nutzer interagieren wie bspw. der Ablauf des Block-Services. Da in dieser Anwendung keine Datenbank benutzt wird, werden die Punkte, die der Nutzer sammelt, lokal auf dem Client gespeichert. Die dazugehörige Anwendungslogik besteht aus dem Timer, der die Zeit hoch zählt, die Zugriffseinschränkung des Smartphones und das Belohnungssystem mit den Punkten, welches auf dem Client abläuft. Die Facebook API soll den Client mit Nutzerdaten personalisieren und in der Anwendung mit einem Sign-In Button zur Anmeldung des Nutzers vervollständigen. Dazu kommuniziert der Client mit dem Firebase-Service, der die Firebase Authentication zur Verfügung stellt. Mit Hilfe des Google Play Games Services sollen die erstellten Errungenschaften auf dem Server von Google gespeichert und wieder auf dem Client aufgerufen werden.

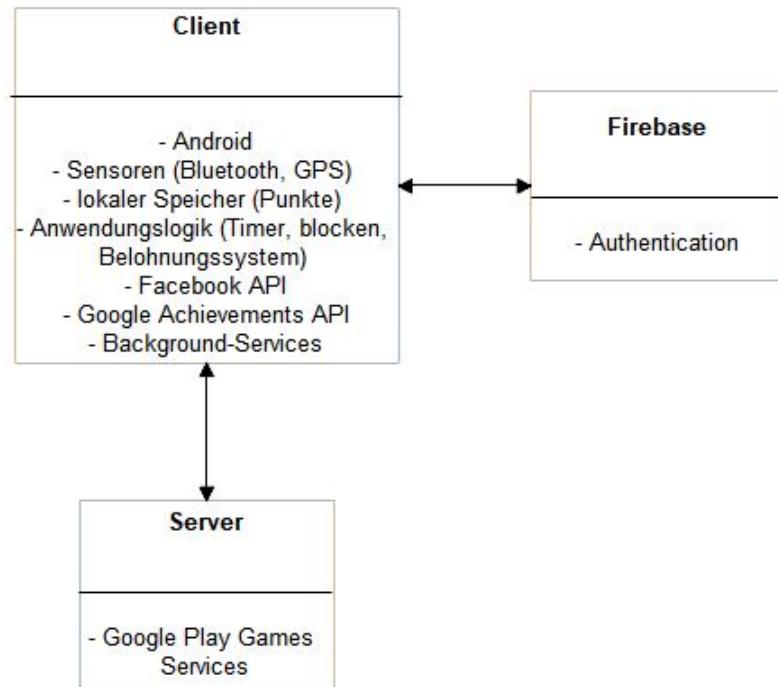


Abbildung 3.4.: Client-Server-Architektur

# 4. Implementierung

Im letzten Kapitel der Bachelorarbeit wird die Vorgehensweise der Implementierung erläutert. Mittels verschiedener Lösungsansätzen soll die Problematik unter Berücksichtigung der aufgelisteten Anforderungsermittlung gelöst werden. In Anbetracht der Tatsache, dass nicht jede Methode direkt zur Lösung des Problems führt, werden die verworfenen Lösungsansätze erklärt und begründet. Außerdem wird bei der Implementierung von ShutPhone aus zeitlichen Gründen nur aus der Sicht der Nutzer, die im Kapitel 3.1.1 auf der Seite 12 erwähnt wurden, umgesetzt. Die mobile Anwendung wurde für Android geschrieben, da im Gegensatz zu iOS keine Einschränkungen bei der Implementierung existieren.

## 4.1. Komponente zur Lösung der Problematik

### 4.1.1. Anmeldung mit Facebook API über Firebase Authentication

Folglich soll die Implementierung es dem Nutzer ermöglichen, ShutPhone zu personalisieren. Hierfür benötigt ShutPhone eine Verknüpfung mit Facebook<sup>1</sup>. Die Voraussetzung für die Anmeldung ist ein bestehender Facebook Account vom Nutzer selbst. Dazu benötigt ShutPhone über die Facebook API den Zugriff auf die Benutzerdaten von Facebook. Facebook API bietet außerdem ein Login UI an, indem der Nutzer sich über einen Facebook-Button anmelden kann. Damit die Verwendung des Facebook Logins möglich ist, stellt Firebase Authentication die Verbindung zu Facebook zur Verfügung. Diese soll die Anbindung der Anmeldemethode im ShutPhone vereinfachen. Sowohl Facebook, als auch Firebase müssen über die jeweiligen Entwicklerplattformen eine Anbindung zum Android Studio bekommen, um somit die SDK benutzen zu können. Nach der Integration ist es möglich auf die Bibliotheksklassen von Facebook und Firebase zuzugreifen. Die Vorgehensweise wurde auf den Entwicklerplattformen von Facebook<sup>2</sup> und Firebase<sup>3</sup> dokumentiert und übernommen.

---

<sup>1</sup><https://www.facebook.com/>

<sup>2</sup><https://developers.facebook.com/docs/facebook-login/android>

<sup>3</sup><https://firebase.google.com/docs/auth/android/facebook-login?authuser=0>

#### 4.1.2. Lokation mit Estimote Beacon

Im weiteren Fortgang kommt der Beacon zum Einsatz. Hierbei benutzt *ShutPhone* den Estimote Beacon, der von der TH Köln zur Verfügung gestellt wurde, damit der Standort genauer ermittelt werden kann. Außerdem ermöglicht der Estimote Beacon über eine mobile Anwendung, Informationen zu publizieren, was in diesem Fall bspw. ein Tagesmenü der Gaststätte sein kann. Diese Eigenschaft soll den Gaststätten helfen, die Produkte zu repräsentieren. Dies wurde bei der Implementierung vorerst nicht berücksichtigt. Die Vorgehensweise zur Implementierung des Estimote Beacons in *ShutPhone* wurde auf der Webseite von Estimote<sup>4</sup> in der Entwickler Dokumentation<sup>5</sup> erläutert.

Für die mobile Anwendung soll das Monitoring genutzt werden. Dieses soll *ShutPhone* in einem bestimmten Radius finden und eine Verbindung zum Beacon erstellen. Für den Prototyp braucht *ShutPhone* vorerst nur einen Beacon. Zu diesem Zweck wird die UUID, Major und Minor über die Estimote Anwendung ermittelt.

Im Listing 4.1 wird in der Methode *beaconManager.connect()*, die zur Verbindung des Beacons dient, die Methode *ServiceReadyCallback()* erzeugt. Diese benutzt eine Rückgabetyp-Methode *onServiceReady()*, worin die Methode *beaconManager.startMonitoring()* aufgerufen wird, um nach dem bestimmten Beacon per UUID, Major und Minor zu suchen.

Listing 4.1: Verbindung zum Beacon erstellen

```

1 beaconManager.connect(new BeaconManager.ServiceReadyCallback() {
2     @Override
3     public void onServiceReady() {
4         beaconManager.startMonitoring(new BeaconRegion("monitored
5             region",
6             UUID.fromString("B9407F30-F5F8-466E-AFF9-25556B57FE6D"),
7             38862,26732));
8         Log.d(TAG, "beacon connected!");
9     }
10 });

```

---

<sup>4</sup><https://estimote.com/>

<sup>5</sup><https://developer.estimote.com/android/tutorial/part-1-setting-up/>

#### 4. Implementierung

Als Beispiel zeigt Listing 4.2 die Methode `beaconManager.setMonitoringListener()`, womit `ShutPhone` in den Methoden `onEnteredRegion()` und `onExitedRegion()` definieren kann, was beim Betreten und Verlassen des Beacons passieren soll. Die von Estimote darlegende Methode `showNotification()`, wird in diesen beiden Methoden hinzugefügt, damit der Nutzer mithilfe der Benachrichtigung weiß, ob der Nutzer sich entweder im Radius befindet oder diesen Bereich verlassen hat. Die vollständigen Methoden sind im Anhang A.2 und A.3 auf den Seiten 48 und 49 ersichtlich.

Listing 4.2: Betreten und Verlassen aus der Beacon Reichweite

```
1 beaconManager.setMonitoringListener(new
2     BeaconManager.BeaconMonitoringListener() {
3         @Override
4         public void onEnteredRegion(BeaconRegion region,
5             List<Beacon> beacons) {
6             .
7             .
8             .
9             showNotification(
10                 "Welcome to ShutPhone!",
11                 "You are currently in Beacon Range.");
12         @Override
13         public void onExitedRegion(BeaconRegion region) {
14             showNotification(
15                 "Scanning stopped!",
16                 "You are out of range.");
17         }
18     });
19 }
```

##### 4.1.3. Timer für die visuelle Darstellung der Zeit im Smartphones

Für die Einschränkung bzw. für die Unterbindung des Smartphone-Nutzens wurde wie in Kapitel 3.1.1 auf der Seite 12 erwähnt, ein Timer implementiert. Im Listing 4.3 ist die Methode `startTime()` abgebildet, die den Prozess des Timers darstellt. Bei der Vorgehensweise wird ein neues Objekt über die Bibliotheksklasse `CountdownTimer` erzeugt. Dieser hat zwei Parameter, die einmal die gesamte Laufzeit in Millisekunden (720000 = zwei Stunden) und das Intervall in Millisekunden (1000 = 1 Sekunde) des Zählens darstellt. Daraufhin erzeugt die Klasse `CountDownTimer` die Methoden `onTick()` und `onFinish()`. In der Methode `onTick()` beginnt die laufende Zeit. Hierdurch wird die Variable `time` hoch gezählt und in Minuten und Sekunden mit Hilfe von `String.format()` konvertiert. Ebenso dient die Bibliotheksklasse `Chronometer` mit `chronometer.setText()` zur Darstellung der Zeit auf der Anwendung, die in Minuten und Sekunden angezeigt werden. Die Methode `onFinished()` wird ausgelöst, wenn die Gesamtlaufzeit von zwei Stunden erreicht wurden und gibt eine kurze Meldung über einen `Toast` von der Anwendung aus.

#### 4. Implementierung

Listing 4.3: Timer Methode: startTime()

```
1  public void startTime(){
2      Timer = new CountDownTimer(7200000, 1000) {
3          public void onTick(long millisUntilFinished) {
4              time++;
5              int seconds = time % 60;
6              int minutes = time / 60;
7              String stringTime = String.format("%02d:%02d",
8                  minutes, seconds);
9              chronometer.setText(stringTime);
10         }
11     public void onFinish() {
12         Toast.makeText(getApplicationContext(), "You reached
13             2 hours!", Toast.LENGTH_LONG).show();
14     }.start();
15 }
```

Zu einem Start gehört auch ein Stopp. Aus diesem Grund wird neben der Methode *startTime()* auch eine *stopTime()* Methode implementiert. Durch die *if*-Abfrage wird gefragt, ob der *Timer*, der aus der Methode *CountDownTimer* im obigen Listing 4.3 kommt, ungleich null ist. Wenn dies der Fall sein sollte, soll die Methode *Timer.cancel()* ausgeführt werden und die Zeit anhalten.

Listing 4.4: Timer Methode: stopTime()

```
1  public void stopTime() {
2      if (null != Timer) {
3          Timer.cancel();
4          Timer = null;
5      }
```

#### 4.1.4. Einschränkung des Smartphones

In diesem Abschnitt liegt der Fokus auf die Einschränkung des Smartphones. In dem Artikel von Schrade (2015) sind Methoden zur Implementierung des Kiosk Modes aufgelistet. Diese verhindern, dass außer der eigenen implementierten Anwendung, andere mobile Anwendungen im Vordergrund laufen können (Schrade, 2015). Kiosk Mode ist ein Modus, der eine Anwendung ermöglicht, die Kontrolle über das System zu erhalten (SDG Systems, 2013). Die Methoden unterstützen ebenfalls die Unterbindung der Android-Funktionen am Smartphone, womit das Verlassen der Anwendung selbst verhindert werden kann. Die betroffenen Funktionen sind die Hardware Buttons wie *Back-Button*, *Home-Button*, *Power-Button* und die *Volume-Buttons* (Schrade, 2015). Die Einschränkung des Smartphones soll in ShutPhone nur möglich sein, wenn das Smartphone im Radius des Beacons befindet, wodurch der Start-Button einerseits den Timer und andererseits den Kiosk Mode starten kann. Zur Veranschaulichung wird im Vorfeld die Abbildung 4.1 dargestellt, welche benötigte Methoden zur Hilfe des Blokkens aufweist. Die Erklärung der einzelnen Methoden werden im folgenden Ablauf erläutert.

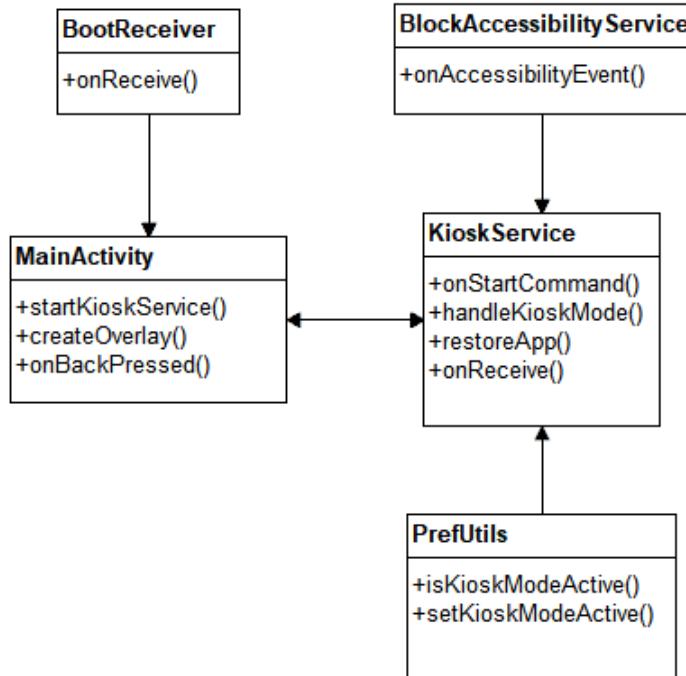


Abbildung 4.1.: Methoden des Kiosk Modes

### **Klasse: BootReceiver zum Starten der Anwendung**

Bevor die Methoden ihre Wirkung zeigen können, benötigt ShutPhone die Berechtigung, den Kiosk Mode automatisch nach dem Hochfahren des Smartphones zu starten. Danach wird die Klasse *BootReceiver* erstellt, welche *BroadcastReceiver* nutzt, um damit die mobile Anwendung zu starten. In der Methode *onReceive()* wird eine Nachricht an die Klasse *MainActivity* mittels des *Intent*-Objekts gesendet, damit beim Start der Anwendung die *Activity*<sup>6</sup> aufgerufen wird, welche im Listing 4.5 in Zeile 6 zu sehen ist.

Listing 4.5: Klasse BootReceiver zum starten der Anwendung

```

1 public class BootReceiver extends BroadcastReceiver {
2     @Override
3     public void onReceive(Context context, Intent intent) {
4         Intent myIntent = new Intent(context,
5             MainActivity.class);
6         myIntent.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
7         context.startActivity(myIntent);
8     }
9 }
```

### **Klasse: KioskService für die Abfrage der Methode handleKioskMode()**

Des Weiteren wird die Klasse *KioskService*, die mit dem *Service* erweitert wird, erstellt. Der *Service* ist eine Anwendungskomponente in Android, der lang andauernde Vorgänge im Hintergrund ausführt wie beispielsweise die Wiedergabe des Musik-Players, die im Hintergrund weiter läuft.

In dieser Klasse findet die Unterbindung des *Home-Buttons* statt. Diese Vorgehensweise unterbindet indirekt den *Home-Button*, da seit Android 4 (Ice Cream Sandwich) nicht mehr möglich ist, den Home-Button zu deaktivieren (Schrade, 2015). Deswegen kreierte Schrade (2015) die Idee, alle neu geöffneten Anwendungen zu erkennen und mit diesen Methoden die implementierte Anwendung neu zu starten, sobald eine andere Anwendung versucht in den Vordergrund zu gelangen.

Die Methode *onStartCommand()* im Listing 4.6 soll verhindern, dass andere mobile Anwendungen beim Start des Kiosk Mode im Vordergrund laufen. Dazu wird ein *Thread*-Objekt erzeugt, der eine separate Aufgabe ausführen soll. Diese Aufgabe soll mithilfe der Methode *run()*, die aus dem Interface *Runnable* kommt, erledigt werden. In diesem Zusammenhang soll die *do*-Anweisung die Methode *handlerKioskMode()* ausführen, welche im Anhang A.1 in der Zeile 48 zu sehen ist. Dabei wird abgefragt, ob der Kiosk Mode aktiv ist und ob die laufende Anwendung “ShutPhone“ im Vordergrund läuft. Falls die zweite Abfrage nicht zutreffen sollte, wird die Methode *restoreApp()* aufgerufen, die die Anwendung neu startet und in den Vordergrund holt. Solange *running* über den Wert *true* verfügt, wiederholt sich dieser Vorgang. Die komplette Klasse ist im Anhang A.1 auf der Seite 46 abgebildet.

---

<sup>6</sup>Ein Activity ist die Darstellung einer Bildschirmseite der mobilen Anwendung

#### 4. Implementierung

Listing 4.6: Kiosk Mode: Abfrage, ob ShutPhone im Vordergrund läuft und Kiosk Mode aktiv ist

```
1 @Override
2 public int onStartCommand(Intent intent, int flags, int
3     startId) {
4     Log.i(TAG, "Starting service 'KioskService'");
5     running = true;
6     ctx = this;
7
8     // start a thread that periodically checks if your app is in
9     // the foreground
10    t = new Thread(new Runnable() {
11        @Override
12            public void run() {
13                do {
14                    handleKioskMode();
15                    try {
16                        Thread.sleep(INTERVAL);
17                    } catch (InterruptedException e) {
18                        Log.i(TAG, "Thread interrupted:
19                            'KioskService'");
20                    }
21                } while(running);
22                stopSelf();
23            }
24        });
25    }
26
27    t.start();
28    return Service.START_STICKY;
29 }
```

#### Methode: onAccessibilityEvent() zum Vergleichen der Paketnamen

Die Methode im Listing 4.7, welche aus der abstrakten Klasse *AccessibilityService* kommt, wurde im Praxisprojekt benutzt, um den Paketname der neu geöffneten Anwendungen zu bestimmen.

Dies ermöglicht das Accessibility-Event `TYPE_WINDOW_STATE_CHANGED`. Dieses Event wird ausgelöst, sobald sich etwas innerhalb des Smartphones ereignet wie bspw. die Öffnung eines Anwendungsfensters. Daraufhin wird der Paketname der geöffneten Anwendungsfenster in `currentApp` gespeichert. Anschließend wird ein *Intent*-Objekt mit dem String `PACKAGE_NAME` erzeugt. Darauf folgt die Methode `intent.putExtra()`, worin in dem String `packageName`, der Paketname aus `currentApp` übermittelt wird. Mit `sendBroadcast()` wird eine Nachricht an die Klasse *KioskService* gesendet. Im Anhang A.1 auf der Seite 46 in der Zeile 24-26 wird ein neues Objekt aus der innere Klasse *PackageReceiver* erzeugt. Außerdem wird ein *IntentFilter*-Objekt erzeugt, um den `PACKAGE_NAME` mit dem aus der Methode `onAccessibilityEvent()` zu vergleichen. Wenn beide Strings identisch sind, wird der Paketname an die Klasse übersendet. Diese wird in der Methode `registerReceiver()` registriert, um den Paketname in der Klasse *KioskService* zu erhalten. Somit wird in der inneren Klasse *PackageReceiver* die `onRe-`

#### 4. Implementierung

*ceive()* Methode aufgerufen, der den Paketname empfängt. Dort findet eine *Boolean*-Abfrage statt, ob der empfangende Paketname gleich der Paketname von ShutPhone ist, welche im Anhang A.1 auf der Seite 46 in der Zeile 83 ersichtlich wird. Wenn der Wert *true* ist, wird daraufhin die Methode *restoreApp()* ausgeführt.

Dieser Vorgang soll anhand eines Beispiels nochmal kurz verdeutlicht werden. Falls der Nutzer versucht ShutPhone zu verlassen, erkennt die Methode *onAccessibilityEvent()* eine Veränderung in der Anwendung. Nachfolgend speichert dies den Paketname wie beispielsweise Facebook und übersendet diese Informationen an die innere Klasse *PackageReceiver*. Dort vergleicht die Methode *onReceive()* den Paketname von Facebook mit dem von ShutPhone. Wenn die Paketnamen nicht identisch sind, wird der *Boolean*-Wert *true* ausgegeben und führt die Methode *restoreApp()* aus, um ShutPhone wieder aufzurufen.

Listing 4.7: Methode: *onAccessibilityEvent()*

```
1 public void onAccessibilityEvent(AccessibilityEvent
2     accessibilityEvent) {
3     if(accessibilityEvent.getEventType() ==
4         AccessibilityEvent.TYPE_WINDOW_STATE_CHANGED) {
5         if(accessibilityEvent.getPackageName() != null) {
6             String currentApp =
7                 accessibilityEvent.getPackageName().toString();
8             Log.d(TAG, currentApp);
9             Intent intent = new Intent("PACKAGE_NAME");
10            intent.putExtra("packageName", currentApp);
11            sendBroadcast(intent);
12        }
13    }
14}
```

#### Klasse: PrefUtils zur Abfrage des Kiosk Mode

Hinzu kommt die Klasse *PrefUtils*, die im Listing 4.8 mit den zwei Methoden *isKioskModeActive()* und *setKioskModeActive()* zu erkennen ist. Die Methode *isKioskModeActive()* soll feststellen, ob der Kiosk Mode aktiv ist oder nicht und die *setKioskModeActive()* soll den Kiosk entweder aktivieren oder deaktivieren.

Listing 4.8: Klasse *PrefUtils*

```
1 public class PrefUtils {
2     private static final String PREF_KIOSK_MODE =
3         "pref_kiosk_mode";
4
5     public static boolean isKioskModeActive(final Context
6         context) {
7         SharedPreferences sp =
8             PreferenceManager.getDefaultSharedPreferences(context);
9         if (sp.getBoolean(PREF_KIOSK_MODE, false)) return true;
10        else return false;
11    }
12}
```

#### 4. Implementierung

```
11  public static void setKioskModeActive(final boolean active,
12      final Context context) {
13      SharedPreferences sp =
14          PreferenceManager.getDefaultSharedPreferences(context);
15      sp.edit().putBoolean(PREF_KIOSK_MODE, active).commit();
16  }
```

#### Methode: startKioskService() zum Starten des Kiosk Modes

Die letzte Methode die benötigt wird, ist *startKioskService()* zum Starten des Kiosk Mode Service, welche im Listing 4.9 zu sehen ist. Diese wird in *MainActivity.class* implementiert und bekommt von der Klasse *KioskService.class* die nötigen Komponenten übermittelt. Der Aufruf der Methode findet beim Starten des Timers über den Start-Button statt, welches im Anhang A.2 in der Zeile 8 zusehen ist.

Die Methode *createOverlay()* soll beim Starten des Kiosk Modes aufgerufen werden, im Zuge dessen wird die Navigationsbar des Smartphone blockiert. Das Ziel hierbei ist, ein unsichtbares Fenster zu erzeugen, welches die Navigationsbar überdeckt und den möglichen Zugriff darauf verhindert. Beim Erkennen des Beacons und demnach die Aktivierung des Timers besteht die Möglichkeit, die Bluetooth-Funktion auszuschalten, bevor der Timer läuft. Infolgedessen erhält der Nutzer die Punkte, jedoch muss dieser sich nicht Zwangsweise in der Gaststätte aufzuhalten. Diese Methode dient vorübergehend als eine Alternativlösung zum Blockieren der Navigationsbar. Bei der Implementierung der Methode entstand eine Komplikation mit dem Entfernen des erzeugten Fensters, weswegen dieser Lösungsansatz noch nicht vollendet ist.

Listing 4.9: Start des Kiosk Mode Services mit der Methode startKioskService()

```
1  private void startKioskService() { // ... and this method
2      startService(new Intent(this, KioskService.class));
3      createOverlay();
4  }
```

Idealerweise sollte ShutPhone auch in der Lage sein den Kiosk Mode stoppen zu können, falls der Nutzer sich nicht mehr im Radius des Beacons befindet oder den Stop-Button betätigt, um die Funktionalität des Smartphones wiederherzustellen. Ohne weiteres Ereignis lässt sich dies allerdings nicht bewerkstelligen. Deshalb werden die Methoden *onEnteredRegion()* und *onExitedRegion()* im Listing 4.10 eingefügt, worüber der Stop-Button oder das Verlassen der Gaststätte den Modus deaktiviert.

Listing 4.10: Stoppen des Kiosk Modes

```
1 setKioskModeActive(false, getApplicationContext());
```

#### 4.1.5. Belohnungssystem von ShutPhone

Damit der Nutzer einen Anreiz bekommt die Anwendung zu nutzen, soll ein Belohnungssystem implementiert werden. Im Kapitel 3.1.1 auf der Seite 12 wurde der Einfall anhand von Punkten und Belohnungen erwähnt und umgesetzt. Zu diesem Zweck wurden drei Lösungsansätze in Betracht gezogen. Der Erste ist mithilfe von dem *Achievements API* aus dem Google Play Games Service SDK<sup>7</sup>. Mit dieser API ist es möglich, auf dem Server von Google eigene Achievements zu definieren, zu speichern und in ShutPhone aufzurufen. Aus programmietechnischen Gründen war es nicht realisierbar die Achievements über die mobile Anwendung zu entfernen, sobald der Nutzer ein Achievement erhalten hat. Da dieses Tool mit dem Google Play Account des Nutzers verbunden ist und durch das Erreichen des Ziels, Erfahrungspunkte für seinen Account erhält, wird ShutPhone dieses Tool behalten.

Der zweite Lösungsansatz ist die Verwendung der *Events API* von Google Play Games Services. Bei der Implementierung entstanden Komplikationen, die nicht rechtzeitig behoben werden konnten. Die Punkte, die der Nutzer erhält, wurden nicht auf dem Google Server gespeichert. Aufgrund dessen wurde dieser Lösungsansatz aus zeitlichen Gründen nicht weiter vertieft. Nach anfänglichen Schwierigkeiten ist die letzte Alternative ein eigenes Belohnungssystem einzubetten, welches nachfolgend erörtert wird.

Bei der Implementierung wird als erstes eine Variable deklariert, die *points* heißt. Mit einer Switch-Case Anweisung soll der Nutzer nach einer bestimmten Zeitspanne Punkte erhalten. Diese soll in der Methode *startTime()* im obigen Listing 4.3 erweitert werden. Als Beispiel wird ein Ausschnitt vom Code im Listing 4.11 dargestellt. Für die Vorführung wurden 60 Sekunden genommen. Wenn der Nutzer den Timer startet und 60 Sekunden erreicht, wird die Punktzahl um zehn erhöht und in ShutPhone angezeigt.

Listing 4.11: Switch-Case-Anweisung für die Punkte

```

1      .
2      .
3      .
4      switch (time) {
5          case 60:
6              points += 10;
7              pointsView.setText(String.valueOf(points));
8              pointsView.setText(" " +points);
9              break;
10             .
11             .
12             .

```

---

<sup>7</sup><https://developers.google.com/games/services/>

#### 4. Implementierung

Daraufhin besteht die Aussicht die erhaltenen Punkten zu verwerten. Mit Hilfe der Methode `onItemClick()` kann der Nutzer die Belohnungen auf dem Smartphone anwählen. Die Belohnungen wurden in Form einer Liste implementiert, welches im Unterkapitel 4.3 ab der Seite 36 ersichtlich wird. In diesem Listing Beispiel 4.12 wird eine Abfrage ausgeführt. Zum einen wird abgefragt, welche Position in der Liste ausgewählt wurde und zum anderen, ob die benötigte Punktzahl (hier: 10) für das Item vorhanden ist. Ist dies der Fall, so werden entsprechende Punkte abgezogen und ein *Toast* von der Anwendung ausgegeben, welcher über ein entsprechendes Ereignis informiert. In diesem Fall: "You got Bier!".

Listing 4.12: Belohnungen erhalten

```
1 public void onItemClick(AdapterView<?> parent, View view, int
2     position, long id) {
3     Object viewTag = view.getTag();
4     Integer tagi = (Integer) viewTag;
5     if(tagi == 1 && points >= 10){
6         points -=10;
7         pointsView2.setText(" " +points);
8         Toast.makeText(getApplicationContext(), "You got Bier!",
9             Toast.LENGTH_LONG).show();
10    .
11    .
12    .
13    .
14    .
15    .
16    .
17    .
18    .
19    .
20    .
21    .
22    .
23    .
24    .
25    .
26    .
27    .
28    .
29    .
30    .
31    .
32    .
33    .
34    .
35    .
36    .
37    .
38    .
39    .
40    .
41    .
42    .
43    .
44    .
45    .
46    .
47    .
48    .
49    .
50    .
51    .
52    .
53    .
54    .
55    .
56    .
57    .
58    .
59    .
60    .
61    .
62    .
63    .
64    .
65    .
66    .
67    .
68    .
69    .
70    .
71    .
72    .
73    .
74    .
75    .
76    .
77    .
78    .
79    .
80    .
81    .
82    .
83    .
84    .
85    .
86    .
87    .
88    .
89    .
90    .
91    .
92    .
93    .
94    .
95    .
96    .
97    .
98    .
99    .
100   .
101   .
102   .
103   .
104   .
105   .
106   .
107   .
108   .
109   .
110   .
111   .
112   .
113   .
114   .
115   .
116   .
117   .
118   .
119   .
120   .
121   .
122   .
123   .
124   .
125   .
126   .
127   .
128   .
129   .
130   .
131   .
132   .
133   .
134   .
135   .
136   .
137   .
138   .
139   .
140   .
141   .
142   .
143   .
144   .
145   .
146   .
147   .
148   .
149   .
150   .
151   .
152   .
153   .
154   .
155   .
156   .
157   .
158   .
159   .
160   .
161   .
162   .
163   .
164   .
165   .
166   .
167   .
168   .
169   .
170   .
171   .
172   .
173   .
174   .
175   .
176   .
177   .
178   .
179   .
180   .
181   .
182   .
183   .
184   .
185   .
186   .
187   .
188   .
189   .
190   .
191   .
192   .
193   .
194   .
195   .
196   .
197   .
198   .
199   .
200   .
201   .
202   .
203   .
204   .
205   .
206   .
207   .
208   .
209   .
210   .
211   .
212   .
213   .
214   .
215   .
216   .
217   .
218   .
219   .
220   .
221   .
222   .
223   .
224   .
225   .
226   .
227   .
228   .
229   .
230   .
231   .
232   .
233   .
234   .
235   .
236   .
237   .
238   .
239   .
240   .
241   .
242   .
243   .
244   .
245   .
246   .
247   .
248   .
249   .
250   .
251   .
252   .
253   .
254   .
255   .
256   .
257   .
258   .
259   .
260   .
261   .
262   .
263   .
264   .
265   .
266   .
267   .
268   .
269   .
270   .
271   .
272   .
273   .
274   .
275   .
276   .
277   .
278   .
279   .
280   .
281   .
282   .
283   .
284   .
285   .
286   .
287   .
288   .
289   .
290   .
291   .
292   .
293   .
294   .
295   .
296   .
297   .
298   .
299   .
300   .
301   .
302   .
303   .
304   .
305   .
306   .
307   .
308   .
309   .
310   .
311   .
312   .
313   .
314   .
315   .
316   .
317   .
318   .
319   .
320   .
321   .
322   .
323   .
324   .
325   .
326   .
327   .
328   .
329   .
330   .
331   .
332   .
333   .
334   .
335   .
336   .
337   .
338   .
339   .
340   .
341   .
342   .
343   .
344   .
345   .
346   .
347   .
348   .
349   .
350   .
351   .
352   .
353   .
354   .
355   .
356   .
357   .
358   .
359   .
360   .
361   .
362   .
363   .
364   .
365   .
366   .
367   .
368   .
369   .
370   .
371   .
372   .
373   .
374   .
375   .
376   .
377   .
378   .
379   .
380   .
381   .
382   .
383   .
384   .
385   .
386   .
387   .
388   .
389   .
390   .
391   .
392   .
393   .
394   .
395   .
396   .
397   .
398   .
399   .
400   .
401   .
402   .
403   .
404   .
405   .
406   .
407   .
408   .
409   .
410   .
411   .
412   .
413   .
414   .
415   .
416   .
417   .
418   .
419   .
420   .
421   .
422   .
423   .
424   .
425   .
426   .
427   .
428   .
429   .
430   .
431   .
432   .
433   .
434   .
435   .
436   .
437   .
438   .
439   .
440   .
441   .
442   .
443   .
444   .
445   .
446   .
447   .
448   .
449   .
450   .
451   .
452   .
453   .
454   .
455   .
456   .
457   .
458   .
459   .
460   .
461   .
462   .
463   .
464   .
465   .
466   .
467   .
468   .
469   .
470   .
471   .
472   .
473   .
474   .
475   .
476   .
477   .
478   .
479   .
480   .
481   .
482   .
483   .
484   .
485   .
486   .
487   .
488   .
489   .
490   .
491   .
492   .
493   .
494   .
495   .
496   .
497   .
498   .
499   .
500   .
501   .
502   .
503   .
504   .
505   .
506   .
507   .
508   .
509   .
510   .
511   .
512   .
513   .
514   .
515   .
516   .
517   .
518   .
519   .
520   .
521   .
522   .
523   .
524   .
525   .
526   .
527   .
528   .
529   .
530   .
531   .
532   .
533   .
534   .
535   .
536   .
537   .
538   .
539   .
540   .
541   .
542   .
543   .
544   .
545   .
546   .
547   .
548   .
549   .
550   .
551   .
552   .
553   .
554   .
555   .
556   .
557   .
558   .
559   .
550   .
551   .
552   .
553   .
554   .
555   .
556   .
557   .
558   .
559   .
560   .
561   .
562   .
563   .
564   .
565   .
566   .
567   .
568   .
569   .
570   .
571   .
572   .
573   .
574   .
575   .
576   .
577   .
578   .
579   .
580   .
581   .
582   .
583   .
584   .
585   .
586   .
587   .
588   .
589   .
580   .
581   .
582   .
583   .
584   .
585   .
586   .
587   .
588   .
589   .
590   .
591   .
592   .
593   .
594   .
595   .
596   .
597   .
598   .
599   .
590   .
591   .
592   .
593   .
594   .
595   .
596   .
597   .
598   .
599   .
600   .
601   .
602   .
603   .
604   .
605   .
606   .
607   .
608   .
609   .
610   .
611   .
612   .
613   .
614   .
615   .
616   .
617   .
618   .
619   .
620   .
621   .
622   .
623   .
624   .
625   .
626   .
627   .
628   .
629   .
630   .
631   .
632   .
633   .
634   .
635   .
636   .
637   .
638   .
639   .
640   .
641   .
642   .
643   .
644   .
645   .
646   .
647   .
648   .
649   .
650   .
651   .
652   .
653   .
654   .
655   .
656   .
657   .
658   .
659   .
660   .
661   .
662   .
663   .
664   .
665   .
666   .
667   .
668   .
669   .
670   .
671   .
672   .
673   .
674   .
675   .
676   .
677   .
678   .
679   .
680   .
681   .
682   .
683   .
684   .
685   .
686   .
687   .
688   .
689   .
690   .
691   .
692   .
693   .
694   .
695   .
696   .
697   .
698   .
699   .
690   .
691   .
692   .
693   .
694   .
695   .
696   .
697   .
698   .
699   .
700   .
701   .
702   .
703   .
704   .
705   .
706   .
707   .
708   .
709   .
710   .
711   .
712   .
713   .
714   .
715   .
716   .
717   .
718   .
719   .
720   .
721   .
722   .
723   .
724   .
725   .
726   .
727   .
728   .
729   .
720   .
721   .
722   .
723   .
724   .
725   .
726   .
727   .
728   .
729   .
730   .
731   .
732   .
733   .
734   .
735   .
736   .
737   .
738   .
739   .
740   .
741   .
742   .
743   .
744   .
745   .
746   .
747   .
748   .
749   .
740   .
741   .
742   .
743   .
744   .
745   .
746   .
747   .
748   .
749   .
750   .
751   .
752   .
753   .
754   .
755   .
756   .
757   .
758   .
759   .
750   .
751   .
752   .
753   .
754   .
755   .
756   .
757   .
758   .
759   .
760   .
761   .
762   .
763   .
764   .
765   .
766   .
767   .
768   .
769   .
760   .
761   .
762   .
763   .
764   .
765   .
766   .
767   .
768   .
769   .
770   .
771   .
772   .
773   .
774   .
775   .
776   .
777   .
778   .
779   .
770   .
771   .
772   .
773   .
774   .
775   .
776   .
777   .
778   .
779   .
780   .
781   .
782   .
783   .
784   .
785   .
786   .
787   .
788   .
789   .
780   .
781   .
782   .
783   .
784   .
785   .
786   .
787   .
788   .
789   .
790   .
791   .
792   .
793   .
794   .
795   .
796   .
797   .
798   .
799   .
790   .
791   .
792   .
793   .
794   .
795   .
796   .
797   .
798   .
799   .
800   .
801   .
802   .
803   .
804   .
805   .
806   .
807   .
808   .
809   .
800   .
801   .
802   .
803   .
804   .
805   .
806   .
807   .
808   .
809   .
810   .
811   .
812   .
813   .
814   .
815   .
816   .
817   .
818   .
819   .
810   .
811   .
812   .
813   .
814   .
815   .
816   .
817   .
818   .
819   .
820   .
821   .
822   .
823   .
824   .
825   .
826   .
827   .
828   .
829   .
820   .
821   .
822   .
823   .
824   .
825   .
826   .
827   .
828   .
829   .
830   .
831   .
832   .
833   .
834   .
835   .
836   .
837   .
838   .
839   .
830   .
831   .
832   .
833   .
834   .
835   .
836   .
837   .
838   .
839   .
840   .
841   .
842   .
843   .
844   .
845   .
846   .
847   .
848   .
849   .
840   .
841   .
842   .
843   .
844   .
845   .
846   .
847   .
848   .
849   .
850   .
851   .
852   .
853   .
854   .
855   .
856   .
857   .
858   .
859   .
850   .
851   .
852   .
853   .
854   .
855   .
856   .
857   .
858   .
859   .
860   .
861   .
862   .
863   .
864   .
865   .
866   .
867   .
868   .
869   .
860   .
861   .
862   .
863   .
864   .
865   .
866   .
867   .
868   .
869   .
870   .
871   .
872   .
873   .
874   .
875   .
876   .
877   .
878   .
879   .
870   .
871   .
872   .
873   .
874   .
875   .
876   .
877   .
878   .
879   .
880   .
881   .
882   .
883   .
884   .
885   .
886   .
887   .
888   .
889   .
880   .
881   .
882   .
883   .
884   .
885   .
886   .
887   .
888   .
889   .
890   .
891   .
892   .
893   .
894   .
895   .
896   .
897   .
898   .
899   .
890   .
891   .
892   .
893   .
894   .
895   .
896   .
897   .
898   .
899   .
900   .
901   .
902   .
903   .
904   .
905   .
906   .
907   .
908   .
909   .
900   .
901   .
902   .
903   .
904   .
905   .
906   .
907   .
908   .
909   .
910   .
911   .
912   .
913   .
914   .
915   .
916   .
917   .
918   .
919   .
920   .
921   .
922   .
923   .
924   .
925   .
926   .
927   .
928   .
929   .
930   .
931   .
932   .
933   .
934   .
935   .
936   .
937   .
938   .
939   .
940   .
941   .
942   .
943   .
944   .
945   .
946   .
947   .
948   .
949   .
950   .
951   .
952   .
953   .
954   .
955   .
956   .
957   .
958   .
959   .
960   .
961   .
962   .
963   .
964   .
965   .
966   .
967   .
968   .
969   .
970   .
971   .
972   .
973   .
974   .
975   .
976   .
977   .
978   .
979   .
980   .
981   .
982   .
983   .
984   .
985   .
986   .
987   .
988   .
989   .
990   .
991   .
992   .
993   .
994   .
995   .
996   .
997   .
998   .
999   .
990   .
991   .
992   .
993   .
994   .
995   .
996   .
997   .
998   .
999   .
1000  .
1001  .
1002  .
1003  .
1004  .
1005  .
1006  .
1007  .
1008  .
1009  .
1010  .
1011  .
1012  .
1013  .
1014  .
1015  .
1016  .
1017  .
1018  .
1019  .
1020  .
1021  .
1022  .
1023  .
1024  .
1025  .
1026  .
1027  .
1028  .
1029  .
1030  .
1031  .
1032  .
1033  .
1034  .
1035  .
1036  .
1037  .
1038  .
1039  .
1040  .
1041  .
1042  .
1043  .
1044  .
1045  .
1046  .
1047  .
1048  .
1049  .
1050  .
1051  .
1052  .
1053  .
1054  .
1055  .
1056  .
1057  .
1058  .
1059  .
1060  .
1061  .
1062  .
1063  .
1064  .
1065  .
1066  .
1067  .
1068  .
1069  .
1070  .
1071  .
1072  .
1073  .
1074  .
1075  .
1076  .
1077  .
1078  .
1079  .
1080  .
1081  .
1082  .
1083  .
1084  .
1085  .
1086  .
1087  .
1088  .
1089  .
1090  .
1091  .
1092  .
1093  .
1094  .
1095  .
1096  .
1097  .
1098  .
1099  .
1100  .
1101  .
1102  .
1103  .
1104  .
1105  .
1106  .
1107  .
1108  .
1109  .
1110  .
1111  .
1112  .
1113  .
1114  .
1115  .
1116  .
1117  .
1118  .
1119  .
1120  .
1121  .
1122  .
1123  .
1124  .
1125  .
1126  .
1127  .
1128  .
1129  .
1130  .
1131  .
1132  .
1133  .
1134  .
1135  .
1136  .
1137  .
1138  .
1139  .
1140  .
1141  .
1142  .
1143  .
1144  .
1145  .
1146  .
1147  .
1148  .
1149  .
1150  .
1151  .
1152  .
1153  .
1154  .
1155  .
1156  .
1157  .
1158  .
1159  .
1160  .
1161  .
1162  .
1163  .
1164  .
1165  .
1166  .
1167  .
1168  .
1169  .
1170  .
1171  .
1172  .
1173  .
1174  .
1175  .
1176  .
1177  .
1178  .
1179  .
1180  .
1181  .
1182  .
1183  .
1184  .
1185  .
1186  .
1187  .
1188  .
1189  .
1190  .
1191  .
1192  .
1193  .
1194  .
1195  .
1196  .
1197  .
1198  .
1199  .
1200  .
1201  .
1202  .
1203  .
1204  .
1205  .
1206  .
1207  .
1208  .
1209  .
1210  .
1211  .
1212  .
1213  .
1214  .
1215  .
1216  .
1217  .
1218  .
1219  .
1220  .
1221  .
1222  .
1223  .
1224  .
1225  .
1226  .
1227  .
1228  .
1229  .
1220  .
1221  .
1222  .
1223  .
1224  .
1225  .
1226  .
1227  .
1228  .
1229  .
1230  .
1231  .
1232  .
1233  .
1234  .
1235  .
1236  .
1237  .
1238  .
1239  .
1230  .
1231  .
1232  .
1233  .
1234  .
1235  .
1236  .
1237  .
1238  .
1239  .
1240  .
1241  .
1242  .
1243  .
1244  .
1245  .
1246  .
1247  .
1248  .
1249  .
1240  .
1241  .
1242  .
1243  .
1244  .
1245  .
1246  .
1247  .
1248  .
1249  .
1250  .
1251  .
1252  .
1253  .
1254  .
1255  .
1256  .
1257  .
1258  .
1259  .
1250  .
1251  .
1252  .
1253  .
1254  .
1255  .
1256  .
1257  .
1258  .
1259  .
1260  .
1261  .
1262  .
1263  .
1264  .
1265  .
1266  .
1267  .
1268  .
1269  .
1260  .
1261  .
1262  .
1263  .
1264  .
1265  .
1266  .
1267  .
1268  .
1269  .
1270  .
1271  .
1272  .
1273  .
1274  .
1275  .
1276  .
1277  .
1278  .
1279  .
1270  .
1271  .
1272  .
1273  .
1274  .
1275  .
1276  .
1277  .
1278  .
1279  .
1280  .
1281  .
1282  .
1283  .
1284  .
1285  .
1286  .
1287  .
1288  .
1289  .
1280  .
1281  .
1282  .
1283  .
1284  .
1285  .
1286  .
1287  .
1288  .
1289  .
1290  .
1291  .
1292  .
1293  .
1294  .
1295  .
1296  .
1297  .
1298  .
1299  .
1290  .
1291  .
1292  .
1293  .
1294  .
1295  .
1296  .
1297  .
1298  .
1299  .
1300  .
1301  .
1302  .
1303  .
1304  .
1305  .
1306  .
1307  .
1308  .
1309  .
1300  .
1301  .
1302  .
1303  .
1304  .
1305  .
1306  .
1307  .
1308  .
1309  .
1310  .
1311  .
1312  .
1313  .
1314  .
1315  .
1316  .
1317  .
1318  .
1319  .
1310  .
1311  .
1312  .
1313  .
1314  .
1315  .
1316  .
1317  .
1318  .
1319  .
1320  .
1321  .
1322  .
1323  .
1324  .
1325  .
1326  .
1327  .
1328  .
1329  .
1320  .
1321  .
1322  .
1323  .
1324  .
1325  .
1326  .
1327  .
1328  .
1329  .
1330  .
1331  .
1332  .
1333  .
1334  .
1335  .
1336  .
1337  .
1338  .
1339  .
1330  .
1331  .
1332  .
1333  .
1334  .
1335  .
1336  .
1337  .
1338  .
1339  .
1340  .
1341  .
1342  .
1343  .
1344  .
1345  .
1346  .
1347  .
1348  .
1349  .
1340  .
1341  .
1342  .
1343  .
1344  .
1345  .
1346  .
1347  .
1348  .
1349  .
1350  .
1351  .
1352  .
1353  .
1354  .
1355  .
1356  .
1357  .
1358  .
1359  .
1350  .
1351  .
1352  .
1353  .
1354  .
1355  .
1356  .
1357  .
1358  .
1359  .
1360  .
1361  .
1362  .
1363  .
1364  .
1365  .
1366  .
1367  .
1368  .
1369  .
1360  .
1361  .
1362  .
1363  .
1364  .
1365  .
1366  .
1367  .
1368  .
1369  .
1370  .
1371  .
1372  .
1373  .
1374  .
1375  .
1376  .
1377  .
1378  .
1379  .
1370  .
1371  .
1372  .
1373  .
1374  .
1375  .
1376  .
1377  .
1378  .
1379  .
1380  .
1381  .
1382  .
1383  .
1384  .
1385  .
1386  .
1387  .
1388  .
1389  .
1380  .
1381  .
1382  .
1383  .
1384  .
1385  .
1386  .
1387  .
1388  .
1389  .
1390  .
1391  .
1392  .
1393  .
1394  .
1395  .
1396  .
1397  .
1398  .
1399  .
1390  .
1391  .
1392  .
1393  .
1394  .
1395  .
1396  .
1397  .
1398  .
1399  .
1400  .
1401  .
1402  .
1403  .
1404  .
1405  .
1406  .
1407  .
1408  .
1409  .
1400  .
1401  .
1402  .
1403  .
1404  .
1405  .
1406  .
1407  .
1408  .
1409  .
1410  .
1411  .
1412  .
1413  .
1414  .
1415  .
1416  .
1417  .
1418  .
1419  .
1410  .
1411  .
1412  .
1413  .
1414  .
1415  .
1416  .
1417  .
1418  .
1419  .
1420  .
1421  .
1422  .
1423  .
1424  .
1425  .
1426  .
1427  .
1428  .
1429  .
1420  .
1421  .
1422  .
1423  .
1424  .
1425  .
1426  .
1427  .
1428  .
1429  .
1430  .
1431  .
1432  .
1433  .
1434  .
1435  .
1436  .
1437  .
1438  .
1439  .
1430  .
1431  .
1432  .
1433  .
1434  .
1435  .
1436  .
1437  .
1438  .
1439  .
1440  .
1441  .
1442  .
1443  .
1444  .
1445  .
1446  .
1447  .
1448  .
1449  .
1440  .
1441  .
1442  .
1443  .
1444  .
1445  .
1446  .
1447  .
1448  .
1449  .
1450  .
1451  .
1452  .
1453  .
1454  .
1455  .
1456  .
1457  .
1458  .
1459  .
1450  .
1451  .
1452  .
1453  .
1454  .
1455  .
1456  .
1457  .
1458  .
1459  .
1460  .
1461  .
1462  .
1463  .
1464  .
1465  .
1466  .
1467  .
1468  .
1469  .
1460  .
1461  .
1462  .
1463  .
1464  .
1465  .
1466  .
1467  .
1468  .
1469  .
1470  .
1471  .
1472  .
1473  .
1474  .
1475  .
1476  .
1477  .
1478  .
1479  .
1470  .
1471  .
1472  .
1473  .
1474  .
1475  .
1476  .
1477  .
1478  .
1479  .
1480  .
1481  .
1482  .
1483  .
1484  .
1485  .
1486  .
1487  .
1488  .
1489  .
1480  .
1481  .
1482  .
1483  .
1484  .
1485  .
1486  .
1487  .
1488  .
1489  .
1490  .
1491  .
1492  .
1493  .
1494  .
1495  .
1496  .
1497  .
1498  .
1499  .
1490  .
1491  .
1492  .
1493  .
1494  .
1495  .
1496  .
1497  .
1498  .
1499  .
1500  .
1501  .
1502  .
1503  .
1504  .
1505  .
1506  .
1507  .
150
```

## 4.2. Lokales Speichern auf dem Smartphone

Abschließend sollen die gesammelten Punkte des Nutzers nach dem Schließen der Anwendung nicht entfallen. Deshalb benötigt ShutPhone einen Speicherort, damit der Nutzer beim nächsten Besuch der Gaststätte auf seine Punkte zurückgreifen kann. Geplant war eine Datenbank anzulegen, der sowohl das Benutzerkonto des Nutzers, als auch die dazugehörigen Punkte speichert. Hierfür sollte mit Hilfe der Datenbank “MongoDB”<sup>8</sup> dieser Gedanke umgesetzt werden. Da die Einarbeitung in die Thematik den zeitlichen Rahmen überschreiten würde, musste eine Alternative für dieses Problem gefunden werden.

Dazu stellt Android einen Framework zur Verfügung, welches die Klasse *SharedPreferences* darstellt. Darin können über einen persistenten *key-value*-Wert, primitive Datentypen wie *Int*, *Booleans*, *Longs* usw. auf dem Smartphone gespeichert und abgerufen werden. Hierfür benötigt ShutPhone die abstrakte Klasse *Context*, um Informationen über die Umgebung der Anwendung zu erhalten. Mit *context.getSharedPreferences()* wird die Datei “ShutPhone” erzeugt, womit jede Klasse in der Anwendung den Zugriff darauf bekommt und es abrufen kann. Damit nur ShutPhone den Zugriff auf die Datei bekommt, wird mit dem Event *MODE\_PRIVATE* die Datei definiert. Die Punkte werden in der Variable *points* gespeichert.

Listing 4.13: Erstellung des SharedPreferences zum Speichern und Laden der Punkte

```
1     Context context = this;
2     sharedpreferences = context.getSharedPreferences(
3             "ShutPhone", Context.MODE_PRIVATE);
4     points = sharedpreferences.getInt("points", 0);
```

Um die Punkte in der mobilen Anwendung zu verändern, wird aus dem *SharedPreferences* der *Editor* gerufen, damit der Wert im *SharedPreferences* verändert werden kann. Im Listing 4.14 wird als Beispiel die gespeicherte Variable *points* verändert, wenn der Nutzer durch das Erreichen einer bestimmten Zeit Punkte erhält. Diese Punkte werden mit der Methode *commit()* in die Datei gespeichert.

Listing 4.14: SharedPreferences zum Speichern und Laden der Punkte

```
1 SharedPreferences.Editor editor = sharedPreferences.edit();
2     editor.putInt("points", points+=10);
3     editor.commit();
```

---

<sup>8</sup>MongoDB ist eine NoSQL-Datenbank, die in der Programmiersprache C++ geschrieben wird.

### 4.3. Ablauf und Umsetzung des Prototypen

Im letzten Abschnitt dieses Kapitels soll der Ablauf und die Umsetzung der mobilen Anwendung anhand von Screenshots erläutert werden. Diese dienen zur Veranschaulichung des Prototypen.

Zu Beginn ist zu erwähnen, dass beim Starten der mobilen Anwendung die Bluetooth-Funktion automatisch aktiviert wird, um den Beacon zu erkennen. Des Weiteren muss der Nutzer sich mit seinem Facebook Account in ShutPhone anmelden, indem er über den Facebook-Button in Abbildung 4.2, seine Nutzerdaten einträgt oder falls die Facebook Anwendung auf dem Smartphone installiert und angemeldet ist, den Account anwählt.



Abbildung 4.2.: ShutPhone - Anmeldung über Facebook

#### 4. Implementierung

Daraufhin erscheint ein Fenster, das in der Abbildung 4.3 dargestellt wird, worin der Nutzer die Aktivierung des Accessibility Services bestätigen muss. Nach der Bestätigung leitet die Anwendung weiter zur Eingabehilfe, welche sich in der Einstellung des Smartphones befindet, um den Service zu aktivieren. Hierbei muss die Aktivierung des Accessibility Service erfolgen, damit das Blockieren des Smartphones freigeschaltet wird. Danach folgt die Abfrage zur Aktivierung des Standortes. Dies dient zur Suche des Beacons, sobald ShutPhone im Hintergrund des Smartphones läuft.

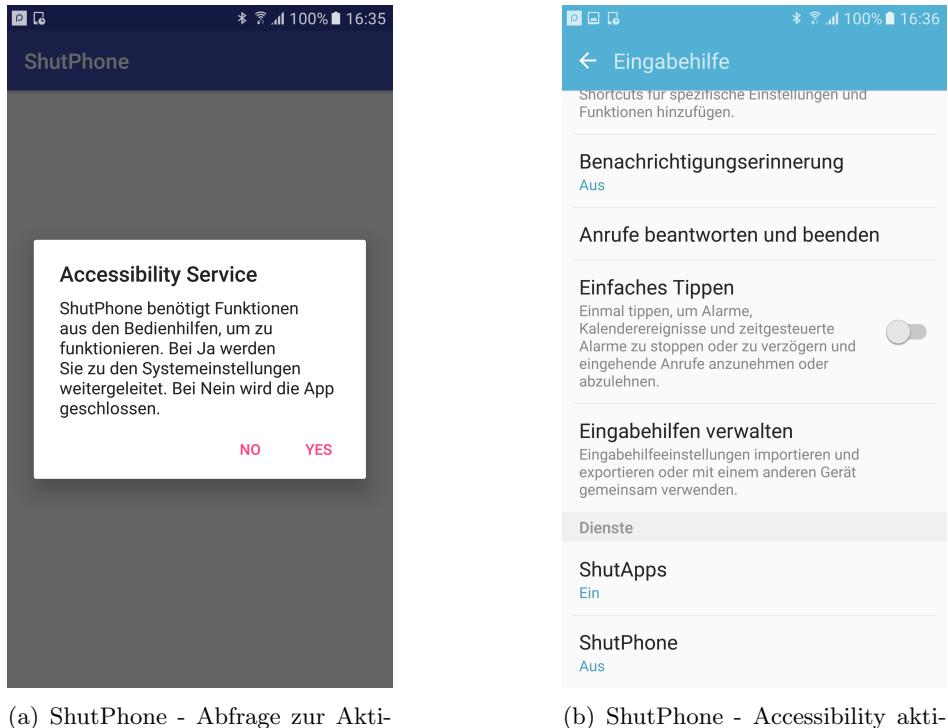


Abbildung 4.3.: ShutPhone - Aktivierung des Accessibility Services

#### 4. Implementierung

Nachdem die Einstellungen ausgeführt wurden, navigiert ShutPhone auf die Hauptseite der Anwendung. In Abbildung 4.4 ist der Start-Button, Quit-Button, Achievement-Button, die gesammelten Punkte und oben rechts die Menütaste zu erkennen. Mit dem Start-Button beginnt die Zeitzählung und zugleich wird das Smartphone blockiert. Der Quit-Button wiederum stoppt die Zeit und stellt die Funktionalitäten des Smartphones wieder her. Bei dem Achievement-Button handelt es sich um die Errungenschaften von Google. Diese können beim Erreichen einer bestimmten Zeit erhalten werden und auf den Google Play vom Nutzer gespeichert. In der Menütaste befinden sich vier Optionen, die zu den jeweiligen Seiten navigiert.

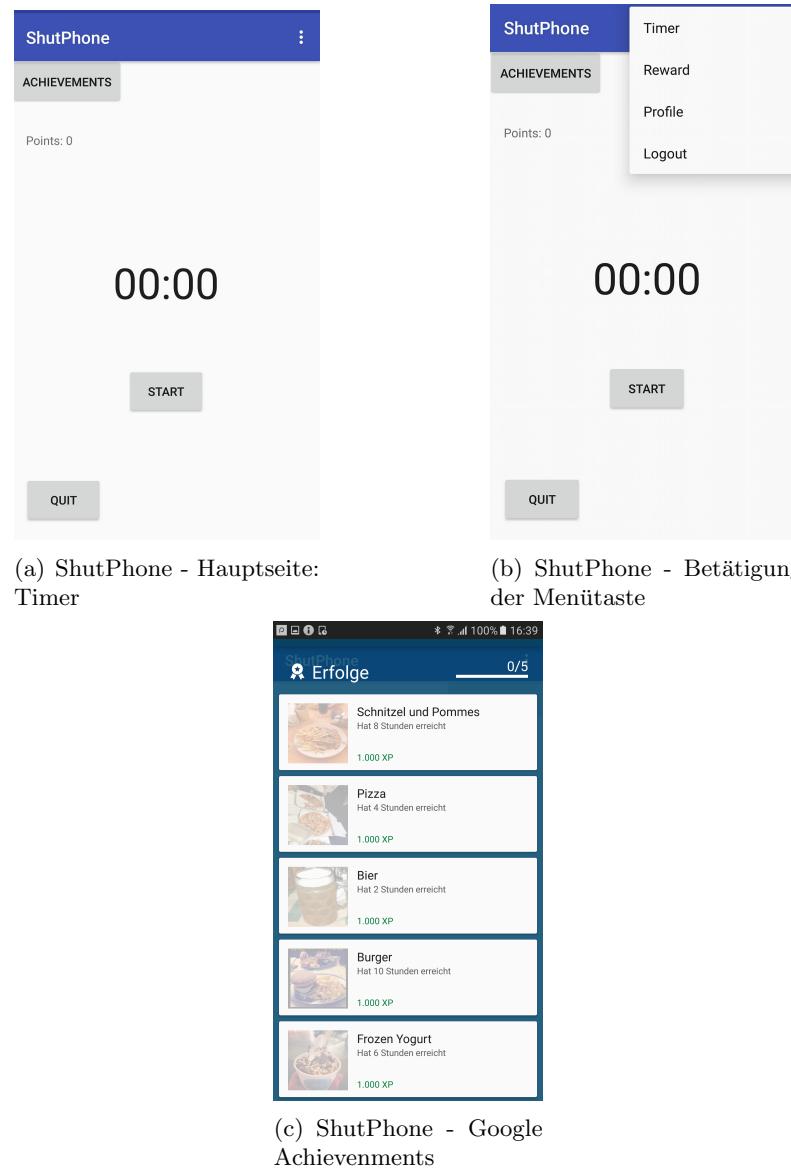


Abbildung 4.4.: ShutPhone - Hauptseite der Anwendung

#### 4. Implementierung

Die Option *Reward* führt zu den Belohnungen, die der Nutzer erhalten kann. In der Abbildung 4.5 sind potenzielle Belohnungen aufgelistet, die zum Test der Anwendung dienen. Je nach Gaststätte können andere Belohnungen aufgeführt sein. Wenn der Nutzer die abgebildeten Punktzahlen erreicht, kann er diese anwählen und verwerten. Dafür muss der Nutzer den Timer zuvor gestartet und bestimmte Zeiten erreicht haben.

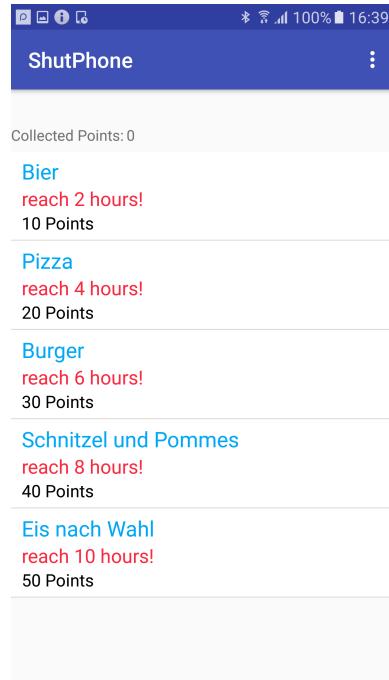


Abbildung 4.5.: ShutPhone - Belohnungsliste der Anwendung

## 5. Fazit und Ausblick

### Fazit

Das Ziel der vorliegenden Arbeit war es eine mobile Anwendung in Bezug auf die Nicht-Nutzung des Smartphones in Gaststätten mit Hilfe eines Belohnungssystems zu konzipieren und implementieren. Hierbei wurden lediglich die Anforderungen der Nutzer an die Anwendung berücksichtigt.

In Bezug auf die Zielsetzung der Bachelorarbeit, die im Kapitel 1.3 auf der Seite 2 erwähnt wurde, war es zum jetzigen Zeitpunkt noch nicht möglich das strategische Ziel zu erreichen. Die mobile Anwendung "ShutPhone" muss durch Testpersonen im Alltag getestet werden, um eine Resonanz zu bekommen, ob durch ShutPhone die Smartphone-Nutzung in Gaststätten minimiert werden kann. Dazu muss ShutPhone einen gewissen Fortschritt in der Entwicklungsphase erreicht haben, damit die Anwendung ohne jegliche Probleme veröffentlicht werden kann.

Jedoch wurden sowohl die operativen Ziele, als auch die taktischen Ziele umgesetzt. Der Prototyp in Android ermöglicht die Einschränkung des Smartphones in einer Gaststätte bzw. an einem Standort. Durch die Realisierung der Hauptfunktionen, wie das Erkennen des Smartphones in einer bestimmten Reichweite eines Beacons, die Einschränkung des Smartphones und die Darstellung des Belohnungssystems, besteht die Möglichkeit mit Hilfe von Gamification der Smartphone-Nutzung entgegenwirken.

Durch die Umfrage der Gaststätte, das Feedback des Usability-Tests und die eigene Idee des Konzeptes entstanden viele Anforderungen, welche zur Verbesserung der Anwendung beitragen. Die genannten Anforderungen, die im Kapitel 3.4 auf der Seite 20 aufgelistet sind, konnten aus zeitlichen Gründen nicht alle umgesetzt werden. Zudem konnte der Lösungsansatz mit dem Speichern der Punkte und der Benutzerdaten in eine Datenbank ebenfalls aus zeitlichen Gründen nicht realisiert werden. Des Weiteren entstanden bei der Implementierung Komplikationen, die nicht behoben werden konnten (wie z. B. die Implementierung von Google Event). Dafür wurde die Alternativlösung, ein eigenes Belohnungssystem zu implementieren, um diese Komplikationen zu umgehen, verwendet.

## 5. Fazit und Ausblick

### Ausblick

Im Ausblick sollen zukünftige Implementierungen vorgenommen werden, die aus zeitlichen Gründen nicht mehr realisierbar waren. Einerseits müssen die restlichen Anforderungen erfüllt werden, damit ShutPhone das vorangegangene Konzept vollständig umsetzt. Die Optimierung der Lösungsansätze, die als alternative Lösung verwendet wurde, könnte möglicherweise durch andere Vorgehensweisen weiterentwickelt werden. Andererseits können mögliche Features ShutPhone erweitern die im Folgenden erläutert werden:

Ein mögliches Feature wäre die Darstellung einer Freundesliste. Damit besteht für den Nutzer die Möglichkeit eine Liste von seinen Freunden angezeigt zu bekommen, die Anwendung nutzen.

Darauf basierend könnte eine Multiplayer-Funktion implementiert werden, das als weiteres Feature gilt. Dies könnte befreundeten Nutzern, welche sich in der selben Gaststätte befinden, ermöglichen gemeinsam bei der Nicht-Nutzung des Smartphones höhere Punkte zu erlangen, um in kürzerer Zeit ihre Belohnungen zu erhalten. Dieses Feature könnte eine weitere Motivation sein, die Freunde dazu einlädt ShutPhone zu nutzen und entgegen der Smartphone-Nutzung zu wirken.

Überdies wäre ein weiteres Feature, zusätzliche Anmeldemethoden für die Nutzer zu ermöglichen. Da nicht jeder im Besitz eines Facebook Accounts ist, könnte somit für die Anmeldung innerhalb der Anwendung die Telefonnummer verwendet werden. Weiterhin kann eine Implementierung vorgenommen werden, welche es dem Nutzer verbietet sich auf mehreren Geräten gleichzeitig mit einem Account anzumelden. So lässt sich verhindern, dass dieser in erhöhter Rate Punkte generiert.

Damit die Gaststätten die Vorteile des ShutPhones nutzen können, wäre die Publikation von Produkten durch Push-Benachrichtigungen, durch den Estimote Beacon auf Smartphones, denkbar.

Folglich wäre ein potentielles Feature, ein Gesamtüberblick über die Nicht-Nutzung des Smartphones in den Gaststätten. Diese können in einem Diagramm dargestellt werden. Hinzu könnten die Nutzer beispielsweise einen Überblick darüber erhalten, wie lange das Smartphone nicht genutzt wurde. Darüber hinaus wäre auch eine Übersicht über die verwendeten Belohnungen möglich.

Abschließend lässt sich sagen, dass ShutPhone durch die Einbettung der Features das Potenzial hat in der Zukunft zu wachsen und das Ziel der Einschränkung von Smartphone Nutzungen in Gaststätten effizienter durchzusetzen.

# Quellencodeverzeichnis

4.1.	Verbindung zum Beacon erstellen . . . . .	25
4.2.	Betreten und Verlassen aus der Beacon Reichweite . . . . .	26
4.3.	Timer Methode: startTime() . . . . .	27
4.4.	Timer Methode: stopTime() . . . . .	27
4.5.	Klasse BootReceiver zum starten der Anwendung . . . . .	29
4.6.	Kiosk Mode: Abfrage, ob ShutPhone im Vordergrund läuft und Kiosk Mode aktiv ist . . . . .	30
4.7.	Methode: onAccessibilityEvent() . . . . .	31
4.8.	Klasse PrefUtils . . . . .	31
4.9.	Start des Kiosk Mode Services mit der Methode startKioskService() . .	32
4.10.	Stoppen des Kiosk Modes . . . . .	32
4.11.	Switch-Case-Anweisung für die Punkte . . . . .	33
4.12.	Belohnungen erhalten . . . . .	34
4.13.	Erstellung des SharedPreferences zum Speichern und Laden der Punkte	35
4.14.	SharedPreferences zum Speichern und Laden der Punkte . . . . .	35
A.1.	Klasse KioskService . . . . .	46
A.2.	Vollständige Methode onEnteredRegion() . . . . .	48
A.3.	Vollständige Methode onExitedRegion() . . . . .	49

# **Abbildungsverzeichnis**

2.1. Gamification: Profil vom Verfasser . . . . .	6
2.2. Gamification: Belohnungen und Errungenschaften nach Bewältigung der Aufgabe . . . . .	7
2.3. Beacon Beispiel: Supermarkt-Schnäppchen funkeln aufs Handydisplay . . . . .	8
2.4. Beacon: Beispiel zum Monitoring . . . . .	9
2.5. Beacon: Beispiel zum Ranging . . . . .	10
3.1. Paper Prototyping - Anfangszustand der Anwendung . . . . .	13
3.2. Paper Prototyping - Ablauf der Anwendung . . . . .	14
3.3. Paper Prototyping - Belohnung einlösen . . . . .	15
3.4. Client-Server-Architektur . . . . .	23
4.1. Methoden des Kiosk Modes . . . . .	28
4.2. ShutPhone - Anmeldung über Facebook . . . . .	36
4.3. ShutPhone - Aktivierung des Accessibility Services . . . . .	37
4.4. ShutPhone - Hauptseite der Anwendung . . . . .	38
4.5. ShutPhone - Belohnungsliste der Anwendung . . . . .	39

# Literaturverzeichnis

- [Bendel 2013] BENDEL, Oliver: *Gamification - Schluss mit lustig.* [http://oliverbendel.net/publikationen/Gamification\\_Netzwoche\\_Published\\_Version.pdf](http://oliverbendel.net/publikationen/Gamification_Netzwoche_Published_Version.pdf). Version: 2013. – [zuletzt zugegriffen: 21.01.2018]
- [Böhm 2002] BÖHM, Rolf: *System-Entwicklung in der Wirtschaftsinformatik*. Zürich : 5. Auflage Vdf Hochschulverlag AG an der ETH-Zürich, 2002
- [Böhme 2015] BÖHME, Florian: *Beacon: Was ist es und was bringt es?* <http://www.wirsindwoar.de/blog/beacon-was-ist-es-und-was-bringt-es>. Version: 2015. – [zuletzt zugegriffen: 24.01.2018]
- [Denker 2014] DENKER, Helge: *Supermarkt-Schnäppchen funkeln aufs Handydisplay.* <https://www.welt.de/wirtschaft/webwelt/article130041491/Supermarkt-Schnaepchen-funkeln-aufs-Handydisplay.html>. Version: 2014. – [zuletzt zugegriffen: 24.01.2018]
- [Deterding u.a. 2011] DETERDING, Sebastian ; DIXON, Dan ; KHALED, Rilla ; NACKE, Lennard E.: *Gamification: Toward a Definition.* <http://hci.usask.ca/uploads/219-02-Deterding,-Khaled,-Nacke,-Dixon.pdf>. Version: 2011. – [zuletzt zugegriffen: 21.01.2018]
- [Eadicicco 2016] EADICICCO, Lisa: *Here's How to Play Pokémon Go.* <http://time.com/4401084/pokemon-go-how-play-guide-tips/>. Version: 2016. – [zuletzt zugegriffen: 03.02.2018]
- [Eigner u.a. 2012] EIGNER, Martin ; GERHARDT, Florian ; GILZ, Torsten ; MOGO NEM, Fabrice: *Informationstechnologien für Ingenieure*. Berlin : 1. Auflage Springer-Verlag Berlin Heidelberg, 2012
- [Malek 2017] MALEK, Piotr: *What are region Monitoring and Ranging?* <https://community.estimote.com/hc/en-us/articles/203356607-What-are-region-Monitoring-and-Ranging-/>. Version: 2017. – [zuletzt zugegriffen: 24.01.2018]
- [OnlineMarketing.de ] ONLINEMARKETING.DE, GmbH: *Beacon.* <https://onlinemarketing.de/lexikon/definition-beacon>. – [zuletzt zugegriffen: 24.01.2018]
- [Pohlmann 2012] POHLMANN, Jan: *Gamification (2/4) – Spielmechaniken: Die Nutzer durch die richtigen Instrumente motivieren und belohnen.* <https://www.usabilityblog.de/>

## Literaturverzeichnis

- gamification-24-spielmechaniken-die-nutzer-durch-die-richtigen-instrumente-motivieren-und-belohnen. Version: 2012. – [zuletzt zugegriffen: 22.01.2018]
- [PREGAS-Presseportal 2016] PREGAS-PRESSEPORTAL: *Das Handy auf dem Tisch – Top oder Flop im Restaurant?* <http://pregas.de/allgemein/item/das-handy-auf-dem-tisch-top-oder-flop-im-restaurant>. Version: 2016. – [zuletzt zugegriffen: 18.01.2018]
- [Rupp u. Joppich 2014] RUPP, Chris ; JOPPICH, Rainer: *Requirements-Engineering und -Management*. München : 6. Auflage Carl Hanser Verlag, 2014
- [Sailer 2016] SAILER, Michael: *Die Wirkung von Gamification auf Motivation und Leistung*. München : 1. Auflage Springer Verlag, 2016
- [Sailer u. a. 2017] SAILER, Michael ; HENSE, Jan U. ; MAYR, Sarah K. ; MANDL, Heinz: How gamification motivates: An experimental study of the effects of specific game design elements on psychological need satisfaction. In: *Computers in Human Behavior* 69 (2017), Nr. 66–77, S. 371–380
- [Schrade 2015] SCHRADE, Andreas: *How-To Create a Working Kiosk Mode in Android.* <http://www.andreas-schrade.de/2015/02/16/android-tutorial-how-to-create-a-kiosk-mode-in-android>. Version: 2015. – [zuletzt zugegriffen: 30.01.2018]
- [Schuster 2017] SCHUSTER, Philipp: *Was ist Google Firebase?* <https://phip1611.de/2017/01/web-und-technik/was-ist-google-firebase>. Version: 2017. – [zuletzt zugegriffen: 24.01.2018]
- [SDG Systems 2013] SDG SYSTEMS, LLC: *Implementing Kiosk Mode in Android - Part 1.* <https://sdgsystems.com/blog/implementing-kiosk-mode-android-part-1>. Version: 2013. – [zuletzt zugegriffen: 30.01.2018]
- [Seaborn u. Fels 2014] SEABORN, Katie ; FELS, Deborah I.: Gamification in theory and action: A survey / University of Toronto, Toronto ,Canada. 2014. – Forschungsbericht. – International Journal of Human-Computer Studies, 74, 14-31

# A. Anhang

## A.1. Vollständige Klasse: KioskService

Listing A.1: Klasse KioskService

```
1 public class KioskService extends Service {
2     private static final long INTERVAL =
3         TimeUnit.SECONDS.toMillis(2); // periodic interval to
4         check in seconds -> 2 seconds
5     private static final String TAG =
6         KioskService.class.getSimpleName();
7
8     private Thread t = null;
9     private Context ctx = null;
10    private boolean running = false;
11    private boolean isInBackground = false;
12
13    @Override
14    public void onDestroy() {
15        Log.i(TAG, "Stopping service 'KioskService'");
16        running = false;
17        super.onDestroy();
18    }
19
20    @Override
21    public int onStartCommand(Intent intent, int flags, int
22        startId) {
23        Log.i(TAG, "Starting service 'KioskService'");
24        running = true;
25        ctx = this;
26
27        PackageReceiver packageReceiver = new PackageReceiver();
28        IntentFilter intentFilter = new
29            IntentFilter("PACKAGE_NAME");
30        registerReceiver(packageReceiver, intentFilter);
31
32        // start a thread that periodically checks if your app
33        // is in the foreground
34        t = new Thread(new Runnable() {
35            @Override
36            public void run() {
37                do {
38                    handleKioskMode();
39                    try {
40                        Thread.sleep(INTERVAL);
41                    } catch (InterruptedException e) {
42                        Log.e(TAG, "Thread interrupted while waiting");
43                    }
44                } while (running);
45            }
46        });
47    }
48
49    private void handleKioskMode() {
50        // Implement logic to handle kiosk mode here
51    }
52}
```

## A. Anhang

```
35             Thread.sleep(INTERVAL);
36         } catch (InterruptedException e) {
37             Log.i(TAG, "Thread interrupted:
38                 'KioskService');");
39         }
40         stopSelf();
41     }
42 }
43
44 t.start();
45 return Service.START_STICKY;
46 }
47
48 private void handleKioskMode() {
49     // is Kiosk Mode active?
50     if(PrefUtils.isKioskModeActive(this)) {
51         // is App in background?
52         if(isInBackground) {
53             restoreApp(); // restore!
54             isInBackground = false;
55         }
56     }
57 }
58
59 // Restart activity
60 private void restoreApp() {
61     Intent i = new Intent(ctx, MainActivity.class);
62     i.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
63     PendingIntent pendingIntent =
64         PendingIntent.getActivity(ctx, 0, i, 0);
65
66     try {
67         pendingIntent.send();
68         ctx.startActivity(i);
69     } catch (PendingIntent.CanceledException e) {
70         e.printStackTrace();
71     }
72 }
73
74 @Override
75 public IBinder onBind(Intent intent) {
76
77     return null;
78 }
79
80 public class PackageReceiver extends BroadcastReceiver {
81     @Override
82     public void onReceive(Context context, Intent intent) {
83         if(intent.getStringExtra("packageName") != null) {
84             if(!ctx.getApplicationContext().getPackageName().equals(
85                 intent.getStringExtra("packageName"))) {
```

```
85                     isInBackground = true;
86                 }
87             }
88         }
89     }
90 }
```

## A.2. vollständige Methode onEnteredRegion()

Listing A.2: Vollständige Methode onEnteredRegion()

```
1 @Override
2 public void onEnteredRegion(BeaconRegion region, List<Beacon>
3     beacons) {btnStart.setOnClickListener(new
4     View.OnClickListener() {
5
6     @Override
7     public void onClick(View view) {
8         btnStart.setEnabled(false);
9         btnQuit.setEnabled(true);
10        if(btnStart.isClickable()) {
11            startKioskService();
12            startTime();
13            setKioskModeActive(true, getApplicationContext());
14            Toast.makeText(getApplicationContext(), "Time is
15                running!", Toast.LENGTH_SHORT).show();
16        }
17        btnQuit.setOnClickListener(new View.OnClickListener() {
18            @Override
19            public void onClick(View v) {
20                if(btnQuit.isEnabled()){
21                    stopTime();
22                    sendData();
23                    setKioskModeActive(false, getApplicationContext());
24                    Toast.makeText(getApplicationContext(), "You can leave the
25                        app now!", Toast.LENGTH_SHORT).show();
26                }
27            }
28        });
29        showNotification(
30            "Welcome to ShutPhone!",
31            "You are currently in Beacon Range.");
32    }
33 }
```

### A.3. vollständige Methode onExitedRegion()

Listing A.3: Vollständige Methode onExitedRegion()

```
1 @Override
2 public void onExitedRegion(BeaconRegion beaconRegion) {
3     stopTime();
4     sendData();
5     btnStart.setEnabled(false);
6     btnQuit.setEnabled(false);
7     setKioskModeActive(false, getApplicationContext());
8     showNotification(
9         "Scanning stopped!",
10        "You are out of range.");
11 }
12});
```