



Universität Ulm | 89069 Ulm | Germany

**Fakultät für
Ingenieurwissenschaften,
Informatik und
Psychologie**
Institut für Datenbanken
und Informationssysteme

Konzeption und Realisierung einer mobilen Anwendung zur Unterstützung der Insulintherapie für Diabetiker

Bachelorarbeit an der Universität Ulm

Vorgelegt von:

Leoni Holl
leoni.holl@uni-ulm.de

Gutachter:

Prof. Dr. Manfred Reichert

Betreuer:

Marc Schickler

2018

Fassung 12. April 2018

© 2018 Leoni Holl

This work is licensed under the Creative Commons Attribution-NonCommercial-ShareAlike 3.0 License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc-sa/3.0/de/> or send a letter to Creative Commons, 543 Howard Street, 5th Floor, San Francisco, California, 94105, USA.

Satz: PDF- \LaTeX 2 ϵ

Kurzfassung

Diabetes ist eine Krankheit, welche ein großes Forschungsthema bietet. Laufend werden neue Technologien und Anwendungen vorgestellt, welche Diabetiker im Alltag unterstützen sollen. Dabei liegt der Fokus auf der Selbstkontrolle des Blutzuckerwerts. Menschen, die an Diabetes Typ 1 erkrankt sind, werden zusätzlich mit Hilfe einer Insulintherapie behandelt. In diesem Fall müssen Kohlenhydrate gezählt werden und die Insulindosis, abhängig von weiteren Faktoren, berechnet werden, was mit viel Disziplin und Schwierigkeiten verbunden ist.

In dieser Bachelorarbeit wird eine mobile Anwendung konzipiert und entwickelt, die Diabetikern bei der Durchführung der Insulintherapie unterstützen soll. Die Applikation soll auf jeden Benutzer individuell angepasst sein, indem relevante Werte bezüglich der Therapie gespeichert werden. Außerdem wird dem Benutzer eine Auswahl an Mahlzeiten und Getränken geboten, durch die das Zählen der Kohlenhydrate von der App unterstützt wird.

Danksagung

An dieser Stelle möchte ich meinem Freund Simon danken, der jeder Zeit hinter mir stand, mir geholfen und mich unterstützt hat.

Auch möchte ich meinem Betreuer Marc danken, der sich für meine Ideen begeistert hat und über die ganze Zeit für Fragen und Feedback zur Verfügung stand.

Danke an meine Freunde und Familie, die sich regelmäßig erkundigten und sich am Ende die Zeit genommen haben alles durchzulesen.

Ein Danke geht auch an meine Diabetes Beraterin, die alle meine Fragen beantwortet hat und sich für den Fortschritt interessiert hat.

Inhaltsverzeichnis

1	Einleitung	1
1.1	Problemstellung	2
1.2	Zielsetzung	2
1.3	Struktur der Arbeit	3
2	Diabetes Hintergrundinformationen	5
2.1	Diabetes Typ 1	6
2.2	Insulintherapie	7
2.3	FreeStyle Libre	8
2.4	Mobile Anwendungen für Diabetiker	9
2.4.1	LibreLink	10
2.4.2	MySugr	12
2.4.3	Bolusrechner	13
2.4.4	Advanced Insulin Advisor Free	14
3	Anforderungsanalyse	17
3.1	Funktionale Anforderungen	17
3.2	Nichtfunktionale Anforderungen	19
4	Entwurf	21
4.1	Mockups	21
4.2	Datenmodell	23
4.3	Farbschema	25
5	Architektur	27
5.1	Komponenten	27
5.2	Dialogstruktur	28
5.3	Anwendungsfälle	29

6 Implementierung	31
6.1 Android	31
6.1.1 Plattform	31
6.1.2 Android Studio	33
6.2 Konzepte	33
6.2.1 NFC	33
6.2.2 Berechnung Insulindosis	38
6.2.3 Darstellung Mengenangabe	40
6.2.4 Erinnerung	42
6.3 Datenspeicherung	42
6.4 Frameworks	43
6.4.1 RangeSeekBar	43
6.4.2 VerticalSeekBar	44
6.4.3 CardView	44
6.4.4 RecyclerView	45
6.4.5 Material Design	46
6.5 Vorstellung der Applikation	54
7 Anforderungsabgleich	61
7.1 Abgleich funktionale Anforderungen	61
7.2 Abgleich nichtfunktionale Anforderungen	63
8 Zusammenfassung und Ausblick	65
8.1 Zusammenfassung	65
8.2 Ausblick	66
A Anhang	73
A.1 Mockups	73

1

Einleitung

Durch das Wachstum der Handyindustrie und dem Ausbau des Internets ist der Besitz eines Smartphones in unseren Regionen keine Seltenheit mehr. Im Allgemeinen steigt die Anzahl der Smartphonebesitzer auf der ganzen Welt Jahr für Jahr [1]. Die Benutzung von Smartphones ist heutzutage in kaum einem Gebiet mehr wegzudenken. Auch im Bereich Medizin gibt es eine Vielzahl an mobilen Anwendungen, die Menschen im Alltag unterstützen und begleiten [2].

Auch im Bereich Diabetes, in welchem eine regelmäßige Selbstkontrolle sehr wichtig ist, sind mobile Anwendungen sinnvoll. Durch deren Unterstützung wird der Benutzer zum Beispiel regelmäßig an die Blutzuckerkontrolle in Form von Benachrichtigungen erinnert. Auch Ernährungstipps oder das Protokollieren der Blutzuckerwerte zeigen eine deutliche Verbesserung der Selbstkontrolle bei Patienten, die solche Anwendungen benutzen. Die Anzahl der Benutzer für digitale Applikationen im Bereich Diabetes wird weiter steigen. Dafür ist nicht nur die steigende Rate der Smartphonebenutzer verantwortlich, sondern auch die Anzahl der Patienten [3]. Im Jahr 2017 gab es weltweit 425 Millionen Menschen, die an Diabetes erkrankt sind und laut Schätzungen steigt die Zahl der Erkrankten auf 628 Millionen im Jahr 2045 [4]. Hinzu kommt das steigende Angebot an mobilen Anwendungen zur Selbstkontrolle für Diabetiker [5].

Trotz der Vielzahl an Anwendungen ist es schwer eine benutzerfreundliche und vertrauenswürdige digitale Unterstützung für den Alltag als Diabetiker zu finden.

1.1 Problemstellung

Die meisten Applikationen, die im App-Store im Bereich Diabetes angeboten werden, beziehen sich auf die Protokollierung des Blutzuckerwerts und bieten ein Tagebuch an. Auch sind viele Anwendungen im Bereich Ernährungsberatung zu finden. Diabetiker mit Typ 1 Diabetes benötigen allerdings nicht nur Unterstützung für die Protokollierung des Blutzuckerwerts, sondern auch für das Spritzen des Insulins, nach der Einnahme von Kohlenhydraten. Um die richtige Anzahl an Kohlenhydraten zu bestimmen, wird viel Erfahrung benötigt. Trotz der Angabe auf den meisten Lebensmittelverpackungen müssen Diabetiker, vor allem in der Anfangsphase, viele Mahlzeiten abwägen und Berechnungen vornehmen, um die korrekte Menge zu ermitteln. Auch die Recherche nach der Anzahl an Kohlenhydraten für bestimmte Lebensmittel nimmt Zeit in Anspruch. Weitergehend spielen für die Berechnung der richtigen Insulindosis viele Faktoren eine Rolle.

1.2 Zielsetzung

Das Ziel dieser Arbeit ist die Konzeption und die Realisierung einer Anwendung, welche Typ 1 Diabetiker bezüglich der Insulintherapie unterstützt. Dabei soll das Bestimmen und Schätzen von Kohlenhydraten mit Hilfe einer visuellen Darstellung unterstützt werden. Der Benutzer soll alle benötigten Daten personalisieren können, sodass die Berechnung der Insulindosis auf die vom Arzt vorgegebene Einstellungen abgestimmt ist.

Die Anwendung soll Typ 1 Diabetikern in der Anfangsphase unterstützen und ihnen die Umstellung möglichst einfach gestalten. Für Diabetiker, die schon seit längerer Zeit davon betroffen sind, soll die App nicht nur einen gelegentlichen Rückhalt bieten, sondern als ständiger Begleiter dienen, der Restaurantbesuche möglichst angenehm gestaltet und den Fokus beim Essen auch wieder auf dieses lenkt. Durch die Benutzung der App sollen die Selbstkontrolle der Benutzer verbessert und starke Schwankungen des Blutzuckerwerts verhindert werden.

1.3 Struktur der Arbeit

Zu Beginn werden in Kapitel 2 grundlegende Informationen zum Thema Diabetes erklärt. Der Fokus liegt auf Diabetes Typ 1 und der Insulintherapie, die in der Anwendung unterstützt wird. Im Abschnitt 2.4 werden Applikationen vorgestellt, die bereits im App-Store erhältlich sind und für Diabetiker entwickelt wurden.

Kapitel 3 beschäftigt sich mit den Anforderungen an die Applikation. Es werden die Funktionalitäten beschrieben, welche die Anwendung beherrschen soll. Anschließend wird der Entwurf in Kapitel 4 vorgestellt. Dabei werden die ersten Designentscheidungen in Form von Mockups sowie das Datenmodell präsentiert. In Kapitel 5 wird die Architektur und das Zusammenspiel aller beteiligten Komponenten der Anwendung, in Form von Diagrammen erklärt. Kapitel 6 beschäftigt sich mit der Implementierung der Anwendung. Dabei werden wichtige Konzepte, die Datenspeicherung, verwendete Frameworks und die Applikation im Ganzen vorgestellt und erklärt. Nachfolgend werden in Kapitel 7 die Anforderungen aus Kapitel 3 mit den umgesetzten Anforderungen verglichen. Abschließend wird in Kapitel 8 die Arbeit zusammengefasst und ein Ausblick auf mögliche Verbesserungen und erweiterbare Funktionen gegeben.

2

Diabetes Hintergrundinformationen

In diesem Kapitel werden die Hintergrundinformationen über die Krankheit Diabetes behandelt. Die Anwendung spricht Patienten an, die mit der Insulintherapie behandelt werden. Diese Behandlung findet hauptsächlich bei dem Typ 1 Diabetes Anwendung, weshalb der Fokus auf der Insulintherapie und Diabetes Typ 1 liegt. Das Kapitel dient der Verständlichkeit bezüglich den Funktionalitäten und Werten der Applikation.

Diabetes mellitus ist eine chronische Zuckererkrankung. Dabei ist die Aufnahme und Verarbeitung von Glucose im Körper gestört, wodurch es zu einem erhöhten Blutzuckerwert kommt. Im Deutschen Gesundheitsbericht 2017 wurde veröffentlicht, dass allein in Deutschland 6,5 Mio. Menschen davon betroffen sind [6]. Es gibt verschiedene Formen von Diabetes. Die häufigsten sind der Typ 1 Diabetes und der Typ 2 Diabetes. 95% der betroffenen Menschen in Deutschland leiden am Typ 2 Diabetes, wovon hauptsächlich Menschen im höheren Alter betroffen sind [6]. Diabetes Typ 2 wird aus diesem Grund auch Altersdiabetes genannt. Diabetes Typ 1 tritt hingegen häufig bei Jugendlichen, Kindern und jungen Erwachsenen auf. Diese erkranken durchschnittlich in einem Alter von 20 Jahren. Eine weitere Form ist der Schwangerschaftsdiabetes, welcher bei Frauen in der Schwangerschaft auftritt und sich in den meisten Fällen nachfolgend wieder zurückbildet [7].

In Abbildung 2.1 ist die Verteilung von Diabetikern in verschiedenen Altersgruppen zu sehen. Es wird deutlich, wie die Anzahl an Diabetikern mit dem Alter steigt. Ein Grund dafür ist die eingeschränkte Bewegungsfähigkeit, die Diabetes Typ 2 auslösen kann [7].

2 Diabetes Hintergrundinformationen

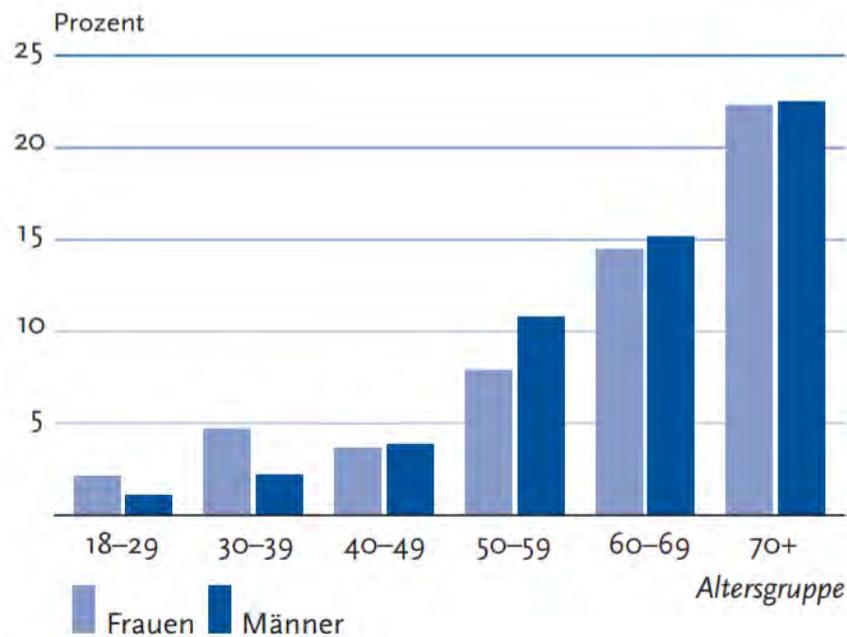


Abbildung 2.1: Anzahl Diabetiker in Deutschland [8]

2.1 Diabtetes Typ 1

Bei der Aufnahme von Glucose im Körper wird Insulin benötigt. Dieses wird in der Bauchspeicheldrüse produziert und hat die Aufgabe Körperzellen zu öffnen, damit die Glucose dort verarbeitet wird. Während beim Diabetes Typ 2 die Zellen eine Insulinresistenz aufweisen, kann die Bauchspeicheldrüse beim Typ 1 Diabetes kein Insulin mehr produzieren. Grund dafür ist, dass das Immunsystem die Betazellen in der Bauchspeicheldrüse zerstört, welche für die Produktion des Insulins verantwortlich sind [9] [7]. Die Krankheit wird diagnostiziert, wenn ca. 80 bis 90% der Zellen zerstört sind [7]. Die Folge ist ein Insulinmangel im Körper, wodurch die Glucose nicht mehr in den Körperzellen aufgenommen und verarbeitet werden kann und somit der Blutzuckerwert steigt. Von einem erhöhten Blutzuckerwert, auch genannt Hyperglykämie, wird gesprochen, wenn der Nüchternwert über 110 mg/dL liegt. Symptome dafür sind starker und plötzlicher

Durst, verstärkter Harndrang, Müdigkeit und Muskelkrämpfe. Je nach Ausmaß treten auch Erbrechen und starker Gewichtsverlust auf [7]. Die Ursachen für Diabetes Typ 1 sind weitestgehend unbekannt. Auslöser können virale Infektionen, Vererbung aber auch Stress und Überanstrengung sein [9].

Da die Funktion der Insulinproduktion bei einem Typ 1 Diabetiker nicht mehr vorhanden ist, muss dieser seinen Insulinhaushalt selbst aufrecht erhalten und es kommt eine Insulintherapie zum Einsatz [9].

2.2 Insulintherapie

Typ 1 und Typ 2 Diabetiker können mit der Insulintherapie behandelt werden. Dabei geht es darum, dass sich der Patient selbst Insulin spritzt, um den Blutzuckerwert auf einem gesunden Level zu halten.

Das Bolusinsulin wird bei der Aufnahme von Nahrungsmitteln oder Getränken, die Kohlenhydrate enthalten, gespritzt. Um zu berechnen, wie viele Einheiten Insulin gespritzt werden müssen, zählen Diabetiker KE. KE steht für Kohlenhydrateinheit und entspricht 10g Kohlenhydraten. Alternativ wird auch noch der BE-Faktor (Broteinheit) benutzt. Lebensmittel mit einem hohen Fettanteil (Fleisch), mit einem hohen Eiweißanteil (Käse) und mit einem hohen Wasseranteil (Gemüse) enthalten keine Kohlenhydrate und sind für Diabetiker unbedenklich [7].

Für die richtige Einstellung der Insulindosis muss außerdem die Tageszeit beachtet werden. Faktoren für morgens, mittags und abends bestimmen die Insulinmenge im Verhältnis zu den zu sich genommenen Kohlenhydraten. Auch die Höhe des Blutzuckerwerts spielt eine Rolle. Ist dieser über einem vorgegebenen Zielbereich, muss der Blutzuckerwert mit einer höheren Insulineinheit korrigiert werden.

Neben dem Bolusinsulin kann auch ein Basalinsulin (Langzeitinsulin) verabreicht werden. Nicht nur durch Mahlzeiten werden dem Körper Kohlenhydrate hinzugefügt. Auch die Leber schüttet in bestimmten Situationen Glucose aus, wodurch der Blutzuckerspiegel ebenfalls steigt. Das Basalinsulin liefert also die Grundlage über den Tag und über die Nacht.

2 Diabetes Hintergrundinformationen

Weitere Regeln, die ein Diabetiker bei seiner Insulintherapie beachten muss, treten bei sportlichen Aktivitäten auf. Durch die Bewegung benötigt der Körper und die Muskulatur mehr Energie, die unter anderem aus dem Glucosevorrat erworben wird. Das wirkt sich folgendermaßen aus: Die Körperzellen weisen eine verbesserte Insulinempfindlichkeit auf, öffnen sich und verbrauchen durch die Bewegung mehr Glucose. Normalerweise reduziert der Körper automatisch die Insulinproduktion, sodass der Insulinspiegel sinkt. Diabetiker müssen speziell darauf achten, um eine Unterzuckerung zu vermeiden. Maßnahmen dafür sind eine erhöhte Kontrolle des Blutzuckerwerts, Anpassung der Insulindosis und der Kohlenhydrataufnahme [7]. Zum Beispiel darf der Blutzuckerwert vor einer sportlichen Aktivität nicht unter 140 mg/dL liegen.

Auch bei der Nahrungsaufnahme nach Bewegungseinheiten muss die Bewegungsintensität beachtet und die Insulindosis gegebenenfalls reduziert werden. Je nach Ausmaß der Bewegung benötigen die Muskeln auch längere Zeit nach der Durchführung eine erhöhte Energiezufuhr.

Die richtige Einstellung ist für Diabetiker sehr wichtig. Durch hohe oder niedrige Blutzuckerwerte und starke Schwankungen können Folgeschäden auftreten. Darunter fällt ein höheres Risiko an Herzinfarkten oder Schlaganfällen, Schädigungen im Auge, Nervenkrankungen, Durchblutungsstörungen und weitere Komplikationen [10]. Umso wichtiger ist es für Diabetiker, die richtige Einstellung des Zuckerwerts zu finden.

Dafür ist vor allem in der ersten Zeit der Erkrankung viel Disziplin nötig. Das Lernen der KE- Werte erfordert viel Zeit. Jede Mahlzeit muss abgewogen und berechnet werden und der Blutzuckerwert erfordert regelmäßige Überprüfung.

2.3 FreeStyle Libre

In Kapitel 2.2 wurde bereits ausführlich erörtert, dass eine regelmäßige und zuverlässige Kontrolle des Blutzuckerspiegels gesundheitlich von Nöten ist.

Bisher war dies ausschließlich mittels eines klassischen Blutzuckermessgeräts messbar. Bei dieser Technik piekst sich der Patient in die seitlichen Fingerkuppen, sodass ein kleiner Blutropfen austritt. Dieser wird von einem Teststreifen, der in das Messgerät

geschoben wird, aufgenommen. Wurde der Wert erfolgreich gemessen, werden Teststreifen und Nadel ersetzt.

Im Jahr 2014 brachte *Abott Diabetes Care*¹ den FreeStyle Libre auf den Markt. Es handelt sich hierbei um einen Sensor, der auf der Hinterseite des Oberarms befestigt wird. Mit Hilfe eines Lesegeräts oder der dafür entwickelten App, kann mittels NFC der aktuelle Blutzuckerwert abgerufen werden. Des Weiteren liefert die Anwendung die Glucosedaten der letzten 8 Stunden sowie einen Trendpfeil, der eine Vorhersage des Werteverlaufs gibt, siehe Abbildung 2.3. Der Sensor wird vom Benutzer selbst angebracht und hält 14 Tage [11].



Abbildung 2.2: FreeStyle Libre Sensor mit Lesegerät [12]

2.4 Mobile Anwendungen für Diabetiker

Durch den Fortschritt der Digitalisierung werden auch mobile und webbasierte Anwendungen im medizinischen Bereich attraktiver. Das Smartphone findet bereits viel Verwendung, um Menschen bei gesundheitlichen Problemen zu unterstützen [3].

Um Folgeerkrankungen zu vermeiden, ist im Bereich Diabetes eine selbständige Überwachung des Blutzuckerwerts gesundheitlich sehr wichtig. Der einfachste Weg, Daten

¹<http://www.de.abbott/products/diabetes-care.html>

2 Diabetes Hintergrundinformationen

zu sammeln und immer abrufbereit zu haben, ist mit Hilfe eines mobilen Geräts. Wird im Google PlayStore² nach dem Begriff 'Diabetes' gesucht, so werden über 200 Apps vorgeschlagen. Viele mobile Anwendungen bieten die Möglichkeit, ein Tagebuch zu führen, in dem alle wichtigen Informationen gesammelt werden können [13].

Ein Vorteil davon ist, dass Benutzer die Möglichkeit haben, ihre Daten auf mehreren Geräten zu teilen und zu synchronisieren. Auch die Kommunikation zwischen Arzt und Patient wird dadurch deutlich verbessert. Daten können ohne großen Aufwand weitergegeben werden und stehen dem Arzt somit jeder Zeit zur Verfügung. Des Weiteren können Benachrichtigungen aktiviert werden, die an eine regelmäßige Kontrolle des Blutzuckerwerts erinnern [5].

Für Diabetiker ist es ebenfalls wichtig, eine gesunde Ernährung zu führen. Durch das richtige Zählen von Kohlenhydraten und das Wissen, welche Mahlzeit sich wie auf den Blutzuckerspiegel auswirkt, lassen sich ungewollte Werte vermeiden. Um sich dieses Wissen anzueignen und Informationen möglichst schnell verfügbar zu haben, stehen ebenfalls viele Anwendungen bereit [13].

2.4.1 LibreLink

LibreLink kann nur mit dem FreeStyle Libre Sensor verwendet werden. Die App bietet eine Übersicht der letzten 24 Stunden, siehe Abbildung 2.4, sowie ein Protokoll aller gemessenen Werte. Außerdem stehen dem Benutzer unter anderem Berichte des Tagesmusters, der Zeit im Zielbereich und des Glucose-Durchschnitts zur Verfügung. Abbildung 2.3 zeigt den gemessenen Glucosewert mit Trendpfeil, in diesem Fall liegt eine gleichbleibende Tendenz vor. Außerdem wird dem Benutzer der Verlauf über die letzten 8 Stunden dargestellt. Zu einem gemessenen Wert kann zusätzlich eine Notiz hinzugefügt werden, in der Lebensmittel, schnell wirkendes Insulin, lang wirkendes Insulin und sportliche Betätigungen eingetragen werden können. Eine solche Notiz kann auch unabhängig eines gemessenen Wertes protokolliert werden.

Die Applikation bietet die Möglichkeit, Glucosewerte über verschiedene Anwendungen zu

²<https://play.google.com/store>

2.4 Mobile Anwendungen für Diabetiker

teilen. Mit LibreLinkUp bekommen zum Beispiel Freunde und Familie Benachrichtigungen über gescannte Werte, die zusätzlich jeder Zeit abrufbar sind [14].

Auf der Startseite ist außerdem immer zu sehen, wie lange der Sensor noch gültig ist. Bereits bis zu drei Tage vor Ablauf der Haltbarkeitsdauer von 14 Tage schickt LibreLink Benachrichtigungen über den aktuellen Status der Lebensdauer. Ist die Frist vorüber, können keine Werte mehr gemessen werden.

FreeStyle LibreLink ist im Google PlayStore für Android und im App Store für iOS kostenlos erhältlich [11].



Abbildung 2.3: Glucosewert mit Trendpfeil [15]



Abbildung 2.4: Messwerte der letzten 24h [15]

FreeStyle Libre macht das Leben von Diabetikern einfacher. Der Sensor ersetzt das Pieksen in den Finger und die App ersetzt das Lesegerät, das sonst immer zur Hand sein muss. LibreLink bietet eine große Übersicht an Statistiken, womit der Benutzer einen besseren Überblick über seine Therapie bekommt und somit in der Selbstkontrolle unterstützt wird.

2.4.2 MySugr

Mehr als 1 Million Benutzer haben die Applikation MySugr [16]. Diese bietet das Führen eines Diabetes-Tagebuchs auf einfache und spielerische Weise. Einträge können jederzeit eingesehen und verglichen werden, siehe Abbildung 2.5 und 2.6. Außerdem werden die Daten durch graphische Darstellungen lesbar und übersichtlich aufbereitet [17]. Der Benutzer hat die Möglichkeit, personalisierte Daten wie Ernährung, Medikamente, Mahlzeiten, Blutzucker, Insulin und Weiteres zu dokumentieren, siehe Abbildung 2.7. Auch ein Bolusrechner steht den Anwendern bereit. Dieser berechnet mit Hilfe des Blutzuckerwerts und der eingegebenen Kohlenhydraten den Boluswert. Dabei werden zusätzlich vergangene Einträge der letzten zwei Stunden berücksichtigt.



Abbildung 2.5: Fertiger Eintrag [17]



Abbildung 2.6: Blutzucker Report [17]

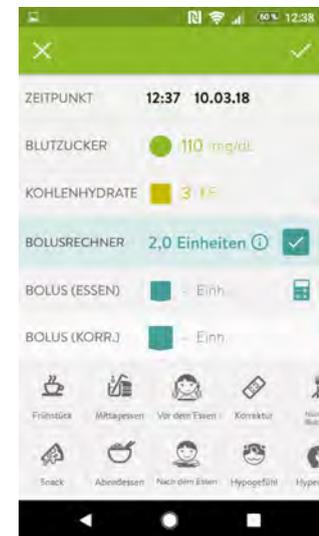


Abbildung 2.7: Neuer Eintrag [17]

Die Applikation wurde von Diabetikern für Diabetiker entwickelt. Zu Beginn müssen sehr viele Einstellungen festgelegt werden, doch wurde dies erledigt, ist die Benutzung der Anwendungen einfach. Dem Benutzer stehen sehr viele Felder zum Eintragen von Informationen zur Verfügung. Diese können personalisiert werden, indem sie geschlossen, hinzugefügt oder verschoben werden. Auch bei der Eingabe selbst werden bereits viele vordefinierte Werte vorgeschlagen, sodass es der Nutzer möglichst einfach hat und

Einträge schnell gespeichert werden können.

Auch der Bolusrechner ersetzt das Kopfrechnen, beziehungsweise die Eingabe in einen Taschenrechner. Allerdings müssen Kohlenhydrate trotz allem selbst gezählt oder geschätzt werden.

Durch Challenges wird der Benutzer motiviert seine Selbstkontrolle zu verbessern und eine Regelmäßigkeit zu erlangen. Zur Belohnung gibt es Punkte, Organisationen werden durch Spenden unterstützt oder die Pro-Version wird für einen bestimmten Zeitraum freigeschaltet.

2.4.3 Bolusrechner

Der Bolusrechner ist eine einfache Anwendung, um den Bolus, d.h. die zu spritzende Insulinsubstanz, zu berechnen. Der Benutzer muss lediglich seinen Zielwert des Blutzuckers und die Korrektoreinheit für eine Einheit Insulin in den Einstellungen hinterlegen, siehe Abbildung 2.9. Zur Berechnung des Bolus sind der Blutzuckerwert und die Proteineinheiten einzutragen. Zusätzlich besteht die Möglichkeit einen BE-Faktor mit anzugeben. Das Ergebnis liefert einen Vorschlag über die zu spritzende Insulinmenge [18].



Abbildung 2.8: Dateneingabe [18]

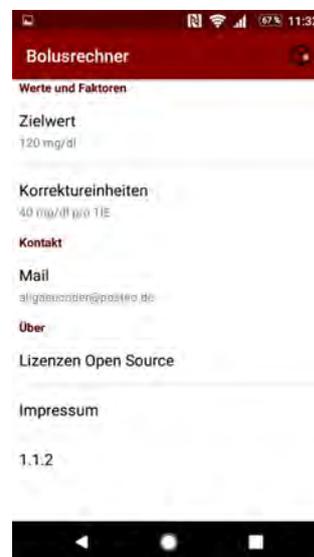


Abbildung 2.9: Einstellungen [18]

2 Diabetes Hintergrundinformationen

Die Applikation wurde sehr einfach gehalten. Für Benutzer, die keine anderen Anwendungen mit integriertem Bolusrechner benutzen, ist die App von Nutzen. Der BE- Faktor muss bei jeder Rechnung von neuem eingegeben werden, obwohl Änderungen selten sind und es benutzerfreundlicher wäre, wenn der Faktor in den Einstellungen vermerkt würde.

Die schwierigste Aufgabe, das Zählen von Kohlenhydraten, wird auch in dieser Anwendung nicht unterstützt.

2.4.4 Advanced Insulin Advisor Free

Auch die Applikation Advanced Insulin Advisor beschäftigt sich mit der Berechnung des Insulinbolus. Die Berechnung basiert auf dem aktuellen Blutzuckerwert, der eingenommenen Kohlenhydrate, der vorhergehenden Menge an gespritztem Insulin und weiteren Parametern, die in Abbildung 2.10 zu sehen sind. Die Kohlenhydratmenge kann manuell eingegeben oder über ein Menü mit einer hinterlegten Datenbank aus hunderten Gerichten ausgewählt werden, siehe Abbildung 2.11.



Abbildung 2.10: Dateneingabe [19]



Abbildung 2.11: Auswahl Essen [19]

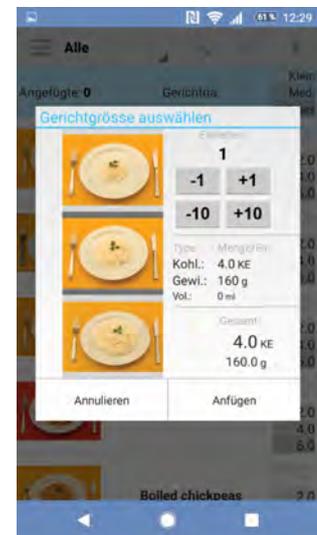


Abbildung 2.12: Mengenauswahl [19]

2.4 Mobile Anwendungen für Diabetiker

Wurde ein Gericht ausgewählt, kann die Größe anhand von maximal drei Bildern entschieden werden, vergleiche Abbildung 2.12 [19].

Advanced Insulin Advisor bietet dem Benutzer eine sehr große Datenbank an Mahlzeiten. Zu jeder Mahlzeit werden außerdem ausreichend Informationen präsentiert, dazu zählen Bilder, Kohlenhydrate, Gewicht und Volumen. Die Applikation erleichtert dem Benutzer das Zählen von Kohlenhydraten, indem zwischen drei Größen der Mahlzeit gewählt werden kann.

Das Design ist sehr schlicht gehalten. Die Startseite bietet viele Möglichkeiten, Einstellungen vorzunehmen, weshalb sie auf den ersten Blick unübersichtlich wirkt. Es ist unklar, hinter welchen Buttons sich welche Funktion verbirgt, zum Beispiel wann der Benutzer auf eine neue Seite geleitet wird und wann eine Auswahl getätigt wird.

3

Anforderungsanalyse

In diesem Kapitel werden die funktionalen und nichtfunktionalen Anforderungen an die Anwendung vorgestellt. Diese werden in der Anfangsphase einer Entwicklung zusammengestellt. Sie bieten einen Überblick über alle Funktionalitäten, welche die Applikation aufweisen soll. Während der Entwicklung können Anforderungen angepasst und erweitert werden. Nach der Entwicklungsphase werden die Funktionalitäten der Anwendung mit den geforderten Anforderungen verglichen.

3.1 Funktionale Anforderungen

Die funktionalen Anforderungen beschreiben die Funktionalitäten, welche die Anwendung aufweisen soll. Im Folgenden werden 19 funktionale Anforderungen aufgelistet.

Bezeichnung	Erklärung
FA#01	Die Anwendung ist auf einen Nutzer personalisiert.
FA#02	In den Einstellungen sind die Insulinfaktoren, der Korrekturwert, der Zielbereich und die Erinnerung jederzeit änderbar.
FA#03	Die Insulinfaktoren werden für morgens, mittags und abends angegeben.
FA#04	Der Korrekturwert besteht aus dem Wert, auf welchen korrigiert werden soll, sowie dem Wert, der pro Einheit gesenkt wird.
FA#05	Der Zielbereich besteht aus einem minimalen und maximalen Zielwert.

3 Anforderungsanalyse

FA#06	Die Benachrichtigung, die den Benutzer beim Verlassen des Hauses an Traubenzucker erinnert, ist aktivierbar und deaktivierbar.
FA#07	Für die Benachrichtigung kann der WLAN-Name für das Heimnetzwerk gespeichert werden.
FA#08	Ist die Erinnerung aktiviert, bekommt der Benutzer beim Verlassen des Heimnetzwerks eine Benachrichtigung.
FA#09	Die Einstellungen können beim ersten Starten der App eingestellt werden. Wird dies übersprungen, werden zunächst Standard-Werte eingetragen.
FA#10	Auf der Startseite kann der aktuelle Blutzuckerwert entweder manuell eingegeben oder über einen FreeStyle Libre Sensor gescannt werden.
FA#11	Auf der Startseite kann gewählt werden, ob etwas gegessen oder getrunken wird.
FA#12	Lebensmittel können nach Kategorie ausgewählt und gesucht werden.
FA#13	Ausgewählte Lebensmittel können entfernt werden.
FA#14	Die Menge des Lebensmittels kann zwischen 10 Stufen gewählt werden.
FA#15	Auf der Startseite ist optional wählbar, in welchem Ausmaß Sport betrieben wurde.
FA#16	Auf der Startseite ist optional ein Getränk und dessen Menge in 3 Stufen wählbar.
FA#17	Nach Eingabe aller erforderlichen Informationen wird die zu spritzende Insulineinheit ausgegeben.
FA#18	Die Eingaben auf der Hauptseite sind nach der Bestätigung jederzeit änderbar.
FA#19	Die Eingaben auf der Hauptseite können alle zurückgesetzt werden.

Tabelle 3.1: Funktionale Anforderungen

3.2 Nichtfunktionale Anforderungen

Die nichtfunktionalen Anforderungen beschreiben die zu erbringende Leistung der Applikation. Im Folgenden werden sieben nichtfunktionale Anforderungen aufgelistet.

Bezeichnung	Erklärung
NFA#01	Das Design richtet sich nach den Gestaltungsrichtlinien des Material Designs für Android.
NFA#02	Die Anwendung ist robust gegenüber vielen und unerwarteten Ereignissen.
NFA#03	Die Anwendung hat einen möglichst geringen Speicherverbrauch.
NFA#04	Die Anwendung hat einen möglichst geringen Akkuverbrauch.
NFA#05	Die Bedienung ist einfach und intuitiv nutzbar.
NFA#06	Die Anwendung ist offline nutzbar.
NFA#07	Auf Nutzereingaben wird möglichst schnell reagiert.

Tabelle 3.2: Nichtfunktionale Anforderungen

4

Entwurf

Das Kapitel beschäftigt sich mit dem Entwurf der Applikation. Dieser dient dafür, erste Entscheidungen bezüglich Design, Umsetzung der funktionalen Anforderungen und des Datenbankaufbaus zu treffen. Diese Entscheidungen werden evaluiert, wodurch Fehler frühzeitig erkannt und behoben werden können.

Im Folgenden werden die MockUps und der Datenbankentwurf vorgestellt.

4.1 Mockups

Die Mockups wurden mit dem Tool *NinjaMock*¹ erstellt. NinjaMock ist eine Plattform, mit der auf einfache Weise Prototypen für verschiedene Betriebssysteme erstellt werden können. Das Design wurde an Bleistiftzeichnungen angelehnt, wodurch die verschiedenen Elemente nicht realitätsgetreu dargestellt sind. Das Verlinken von Objekten liefert eine Simulation der Funktionalitäten. Außerdem können Projekte geteilt werden, sodass mehrere Personen zur selben Zeit daran arbeiten können [20].

Die Mockups dienen einer ersten Evaluation der Bedienbarkeit. Das Design der finalen Fassung stützt sich auf die erarbeiteten Ideen und Entscheidungen.

Nachfolgend werden ausgewählte Mockups vorgestellt und kurz beschrieben. Eine detaillierte Erklärung der Funktionalitäten und der Oberfläche ist in Abschnitt 6.5 zu finden.

¹<https://ninjamock.com/>

4 Entwurf

Auf den Abbildungen 4.1, 4.2 und 4.3 wird das Design der Hauptseite vorgestellt. Abbildung 4.1 zeigt die Seite unbearbeitet. Der Benutzer hat die Möglichkeit den Blutzuckerwert, die Menge, das Essen, das Getränk und den Sport einzutragen. Durch einen Klick auf das jeweilige Element kann dieses bearbeitet werden. In Abbildung 4.2 und 4.3 wird die Seite im Bearbeiteten-Modus für verschiedene Eingaben dargestellt.



Abbildung 4.1: Hauptseite



Abbildung 4.2: Hauptseite mit bearbeiteten Kacheln



Abbildung 4.3: Hauptseite mit bearbeiteten Kacheln

Das Hinzufügen eines Essens auf der Hauptseite wird in Abbildung 4.4 angezeigt. Hier besteht die Möglichkeit in verschiedenen Kategorien ein gewünschtes Element zu suchen und auszuwählen. Die gespeicherte Eingabe wird anschließend auf der Hauptseite übernommen.

In den Einstellungen, siehe Abbildung 4.5, kann der Benutzer seine aktuellen Faktoren bezüglich Diabetes, die in Abschnitt 2.2 behandelt wurden, eintragen. Dazu gehören die Insulinfaktoren für morgens, mittags und abends, der Korrekturwert, der Zielbereich und die Einstellungen für die Erinnerungen.



Abbildung 4.4: Auswahl Essen



Abbildung 4.5: Einstellungen

4.2 Datenmodell

In diesem Abschnitt werden alle Elemente und Variablen präsentiert, die in der Applikation gespeichert werden müssen. Nach der Präsentation werden die Attribute kurz beschrieben.

Für die Einstellungen müssen die persönlichen Daten des Benutzers, welche für die individuelle Insulindosis eine Rolle spielen, gespeichert werden. Die Werte werden von einem Arzt festgelegt und können sich im Laufe der Behandlung verändern. Für das Essen und die Getränke müssen Daten bezüglich des Gewichts und der Kohlenhydrate gespeichert werden.

Für die Einstellungen müssen folgende Werte gesetzt werden. Diese richten sich nach den in Abschnitt 2.2 erläuterten Werten.

Einstellungen

Insulinfaktoren:	Faktor Morgen, Faktor Mittag, Faktor Abend
Zielbereich:	Zielbereich Minimum, Zielbereich Maximum
Korrekturwert:	Korrektur auf, Korrekturereinheit
Erinnerung:	Status, WLAN-Name

Im Folgenden werden alle genannten Werte erklärt.

Faktor Morgen:	Wert, mit welchem die KE ² morgens multipliziert werden, um die Insulineinheit zu bestimmen.
Faktor Mittag:	Wert, mit welchem die KE mittags multipliziert werden.
Faktor Abend:	Wert mit welchem die KE abends multipliziert werden.
Zielbereich Minimum:	Die untere Grenze des Blutzuckerzielbereichs.
Zielbereich Maximum:	Die obere Grenze des Blutzuckerzielbereichs.
Korrektur auf:	Der Blutzuckerwert, auf welchen korrigiert werden soll, wenn er die obere Grenze des Zielbereichs überschritten hat.
Korrekturereinheit:	Eine Einheit, um die der Blutzuckerwert pro gespritzter Insulineinheit sinkt.
Status:	Erinnerung aktiviert oder deaktiviert.
WLAN-Name:	Der Name des Heimnetzwerks.

Welche Werte für das Essen gespeichert werden müssen, werden nachfolgend dargestellt.

Essen

Information	Name, Kategorie, Abrechnung
Stückabrechnung	Max. Stücke, Einheit/Stück
Normale Abrechnung	Max. Gewicht, Einheit/100g

Im Folgenden werden alle genannten Werte erklärt.

²Kohlenhydrateinheiten

Name:	Der Name des Essens.
Kategorie:	Die Kategorie zu welcher das Essen gehört.
Abrechnung:	Stückabrechnung oder normale Abrechnung. Stückabrechnung für Mahlzeiten, die in Stücke geteilt werden, zum Beispiel Pizza. Normale Abrechnung für alles andere, zum Beispiel Nudeln.
Max. Stücke:	Für die Stückabrechnung. Maximale Anzahl an Stücken, die das Essen auf einem Teller haben kann.
Einheit/Stück:	Die KE für ein Stück.
Max. Gewicht:	Für die normale Abrechnung. Maximales Gewicht, das das Essen auf einem Teller haben kann.
Einheit/100g:	Die KE für 100g des Essens.

Die Daten, welche für die Getränke gespeichert werden müssen, werden nachfolgend präsentiert.

Getränk: Name, Einheit/10ml

Im Folgenden werden alle genannten Werte erklärt.

Name:	Der Name des Getränks.
Einheit/10ml:	Die KE für 10ml des Getränks.

4.3 Farbschema

Die Farben wurden mit colors.co³ erstellt. In diesem Tool werden fünf zufällige Farben generiert, die gut miteinander harmonieren. Der Benutzer des Tools kann selbst entscheiden, welche Farben sein Farbschema beinhalten soll, indem einzelne Farben gesichert werden können, sodass die restlichen Farben anhand dieser neu generiert werden.

³<https://colors.co/>

4 Entwurf

In der Abbildung 4.6 ist das ausgewählte Farbschema für die Applikation zu sehen. Die Farben bilden die Kacheln auf der Hauptseite, siehe Abbildung 6.29. Es wurden satte Farben gewählt, da die restliche Anwendung sehr schlicht gehalten ist, sobald sich die Kacheln im Bearbeiteten-Modus befinden.

Die Farbe #77BA99 wurde zusätzlich als Akzentfarbe festgelegt.



Abbildung 4.6: Farbschema [21]

5

Architektur

In diesem Kapitel wird die Architektur der Applikation vorgestellt. Zunächst wird mit Hilfe des Komponentendiagramms verdeutlicht wie der Benutzer, der FreeStyle Libre Sensor und das Smartphone zusammenarbeiten. Die Dialogstruktur stellt die Navigation der Applikation dar, das Anwendungsfalldiagramm verdeutlicht mögliche Aktionen des Benutzers.

5.1 Komponenten

Dieses Kapitel stellt drei Komponenten und deren Verbindung mit Hilfe eines Komponentendiagramms, Abbildung 5.1, vor. Der FreeStyle Libre Sensor bezieht einmal in der Minute Daten bezüglich des Blutzuckerwerts des Benutzers. Der Benutzer selbst kann Eingaben in der Applikation tätigen, darunter auch die Eingabe des Blutzuckerwerts. Um diesen zu erhalten, kann die Applikation mittels NFC die Daten des Sensors beziehen. Hat der Benutzer alle nötigen Eingaben getätigt, berechnet die Applikation eine Ausgabe, siehe Kapitel 6.2.2.

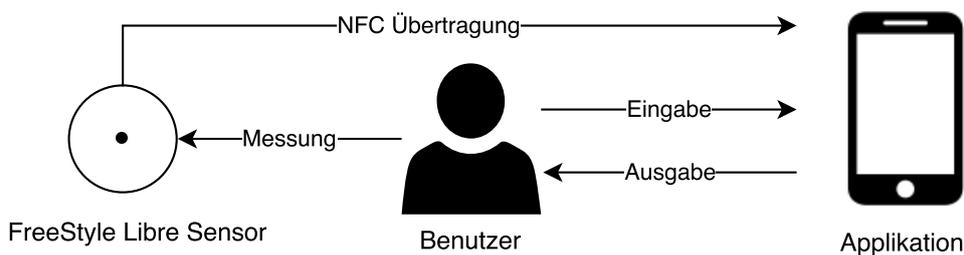


Abbildung 5.1: Komponentendiagramm

5.2 Dialogstruktur

Die Dialogstruktur in Abbildung 5.2 repräsentiert den Aufbau der Anwendung. Jeder Dialog stellt eine Activity oder ein Fragment der Projektstruktur dar. Mit Hilfe des Diagramms kann verfolgt werden, welcher Dialog über welchen Weg aufgerufen werden kann.

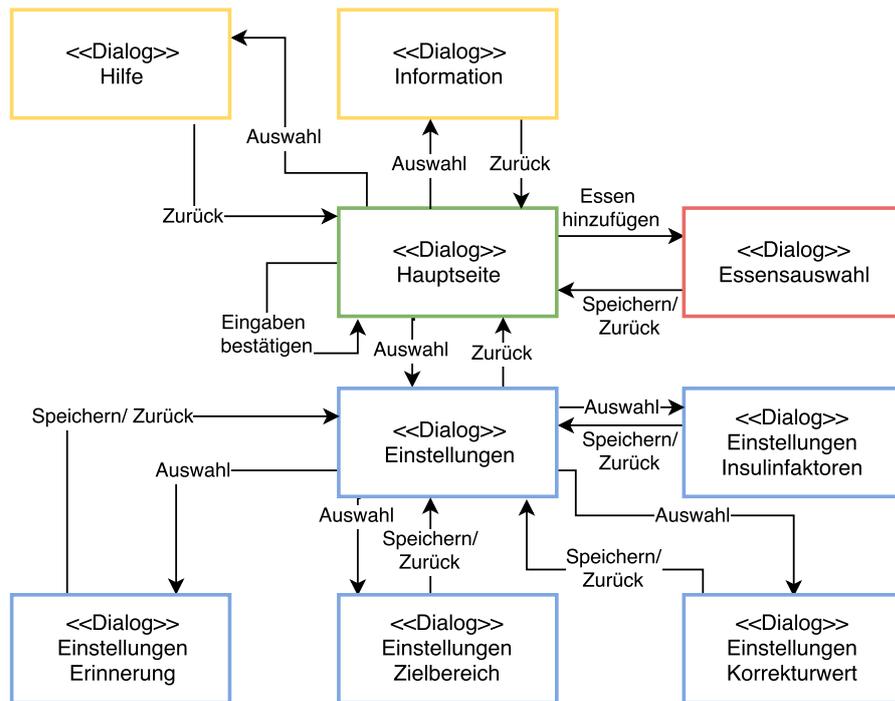


Abbildung 5.2: Dialogstruktur

Öffnet der Benutzer die Applikation, befindet er sich auf der Hauptseite. Dort hat er die Möglichkeit Daten einzugeben und diese zu bestätigen, um eine Ausgabe zu erhalten. Für die Eingabe des Essens, muss der Benutzer den Dialog der Essensauswahl aufrufen. Ausgewählte Daten können gespeichert und somit an die Hauptseite übergeben werden. Über das Menü kann der Anwender die Hilfe, die Informationen und die Einstellungen aufrufen. Unter Einstellungen können vier weitere Dialoge geöffnet werden, dazu gehören die Insulinfaktoren, der Zielbereich, der Korrekturwert und die Erinnerung. Auf

jeder Seite können Daten eingegeben, verändert und gespeichert werden. Durch den Zurück-Button navigiert der Benutzer wieder zurück zur Hauptseite.

5.3 Anwendungsfälle

Mit Hilfe des Anwendungsfalldiagramms, Abbildung 5.3, werden alle Aktionen dargestellt, die der Benutzer auf der Hauptseite ausführen kann.

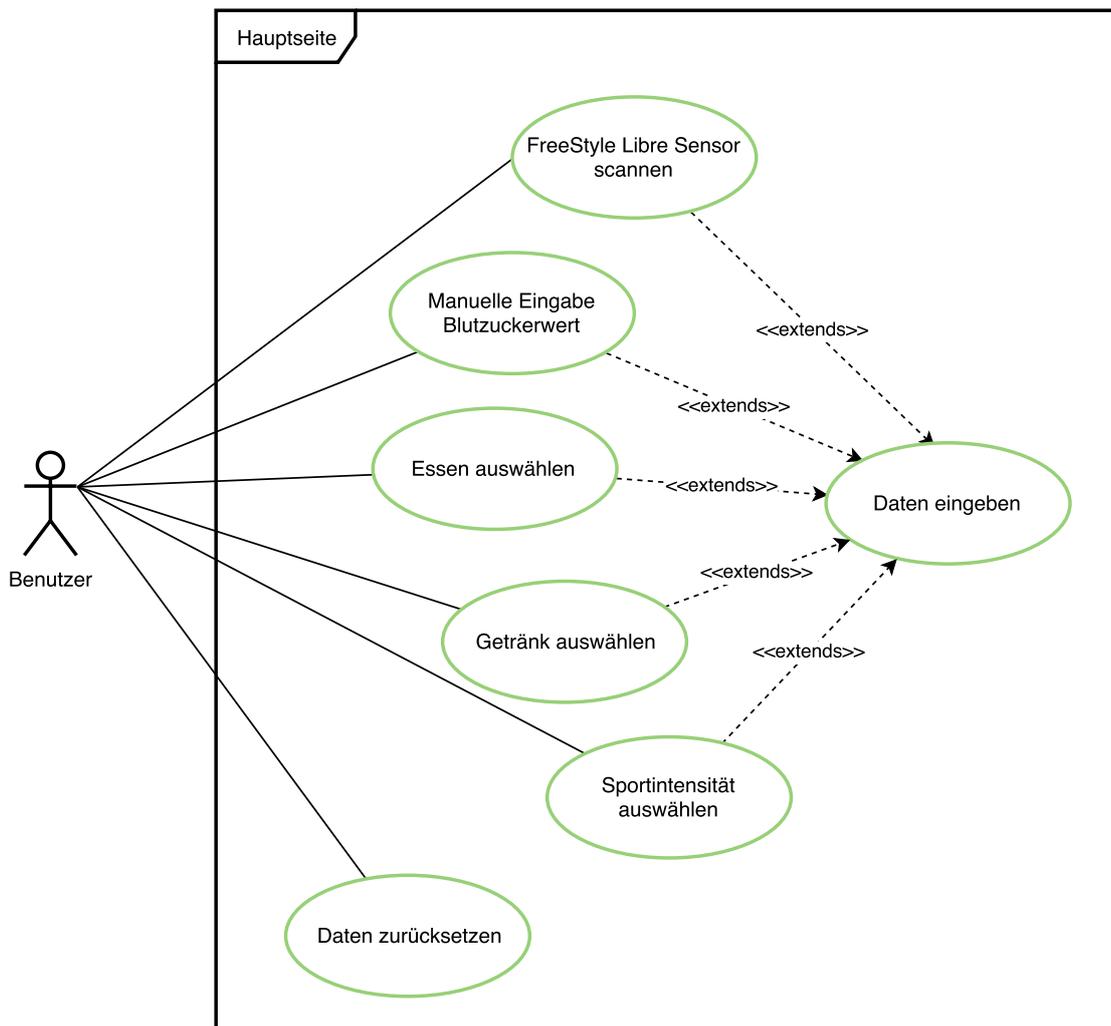


Abbildung 5.3: Anwendungsfalldiagramm

5 Architektur

Die Aktionen bestehen zum größten Teil aus der Eingabe von Daten. Diese ist unterteilt in die Eingabe des Blutzuckerwerts, der manuell eingetragen oder über den FreeStyle Libre Sensor gemessen werden kann, die Auswahl und Bearbeitung des Essens und des Getränks sowie die Auswahl der Sportintensität. Anschließend kann der Benutzer alle eingegebenen Daten zurücksetzen.

6

Implementierung

In diesem Kapitel werden die Konzepte und Komponenten erklärt, welche für die Umsetzung der Anforderungen verwendet wurden. Zu Beginn wird Android in einem kurzem Überblick vorgestellt, anschließend werden verwendete Konzepte, die im Hintergrund ablaufen sowie die verwendeten Methoden für die Datenspeicherung erklärt. Das Kapitel Frameworks gibt eine Übersicht an Komponenten, die für die Oberflächengestaltung eingesetzt wurden. Zum Abschluss wird die fertige Applikation anhand von Screenshots beschrieben.

6.1 Android

In diesem Kapitel wird das Betriebssystem Android vorgestellt, sowie dessen Entwicklungsumgebung Android Studio.

6.1.1 Plattform

Android ist ein Linux-basiertes Betriebssystem, das mittlerweile in Smartphones, Tablets, Smartwatches, Fernseher und Autos zum Einsatz kommt. Es bietet seinen Nutzern über zwei Millionen Apps, welche im Play Store verfügbar sind [22].

Die verschiedenen Versionen des Betriebssystems sind alle nach einer Süßigkeit benannt. Angefangen bei der Version 1.6 mit dem Namen Donut, ist seit dem Jahr 2017 die Version 8.0 mit dem Namen Oreo auf dem Markt. Oreo bietet viele neue Funktionen im Vergleich zu seinen Vorgängern. Die Version ist intelligenter und schneller, bietet eine automatische Formularausfüllung, Bild-in-Bild Anwendungen, Verbesserungen im

6 Implementierung

Bereich Benachrichtigungen und mehr Sicherheit [23]. Der Quellcode des Betriebssystems ist für jeden Entwickler und Interessenten frei verfügbar. Dieser kann nicht nur gelesen, sondern auch beliebig geändert und weitergegeben werden. Android will damit erreichen, dass möglichst viele Menschen Zugriff auf die heutigen Mobiltechnologie haben und selbst Vorteile daraus ziehen können. Außerdem soll die Entwicklung von Apps vorangetrieben und Entwicklern die Chance gegeben werden, eine breite Zielgruppe anzusprechen. Um eine App im Play Store zu veröffentlichen, müssen Entwickler eine einmalige Gebühr von 25\$ bezahlen. Die App wird einigen Prüfungen unterzogen, in welchen auf Sicherheit, Vertrauenswürdigkeit und Richtlinien getestet wird [24]. Eine Android App besteht aus Activities. Diese können verschiedene Zustände annehmen, wenn der Benutzer durch eine Applikation navigiert. Wurde ein Zustand geändert, so werden entsprechende Callbacks der Activity aktiviert, wodurch der Entwickler das Verhalten der App steuern kann. Wann welcher Callback zum Einsatz kommt, wird in Abbildung 6.1 deutlich [25].

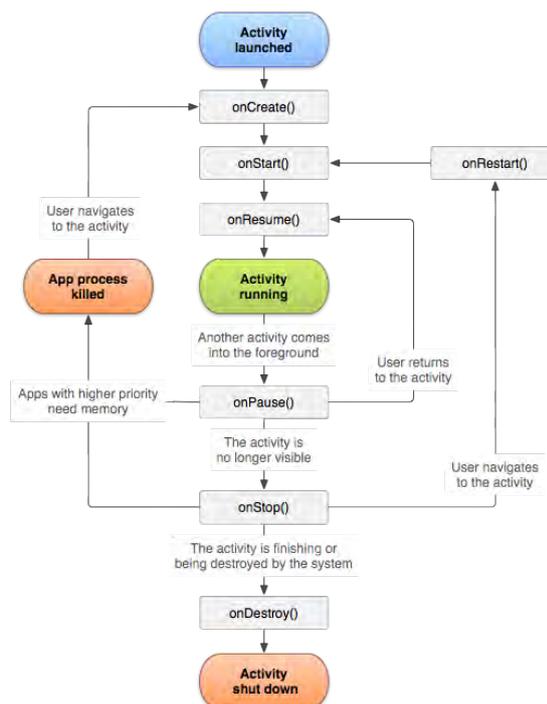


Abbildung 6.1: Activity Lifecycle [25]

6.1.2 Android Studio

Android Studio ist die offizielle Entwicklungsumgebung, um Android Anwendungen zu implementieren. Diese können für Android Smartphones, Tablets, Smartwatches, Android TV und Android Auto entwickelt werden. Dem Entwickler stehen viele Features zur Verfügung. Der Instant Run ermöglicht es, neuen Code direkt auf dem eigenen Gerät oder auf Emulatoren laufen zu lassen. Um Code noch schneller zu schreiben, verfügt Android Studio über einen intelligenten Code-Editor. Dieser bietet eine Autovervollständigung, einfache Überarbeitung und Analyse des Codes. Außerdem wird das Einfügen von Bibliotheken und somit das Konfigurieren des Build-Prozesses unterstützt [26].

Wie bereits erwähnt, kann Code direkt auf Emulatoren getestet werden. Diese stehen dem Entwickler in allen nötigen Formen und Größen zur Verfügung. Ein Emulator repräsentiert die meisten Funktionalitäten, die auch ein physisches Gerät implementiert hat. Dazu gehört zum Beispiel die Verwendung der Hardware-Sensoren. Bluetooth und NFC wird nicht unterstützt [27].

Der Layout-Editor ermöglicht es, Oberflächen einfach und schnell zu gestalten. Struktur und Design basiert auf einer XML-Datei, die der Layout-Editor durch Drag-and-Drop-Funktionalitäten modifiziert [28].

6.2 Konzepte

In diesem Kapitel werden Konzepte präsentiert, die zur Realisierung bestimmter Komponenten der Applikation entwickelt wurden. Dazu gehört die Implementierung für die Messung mit dem FreeStyle Libre Sensor, die Berechnung der Insulindosis und die Umsetzung der Erinnerung.

6.2.1 NFC

NFC ist die Abkürzung für Near Field Communication, auf Deutsch Nahfeldkommunikation. Das heißt, es wird eine Verbindung in einer kurzen Distanz, circa 10 bis 20 Zentimeter, zwischen zwei Geräten aufgebaut. Die Verbindung erfolgt kontaktlos und

6 Implementierung

ermöglicht das Senden und Empfangen von Daten. Die Kommunikation wird zum Beispiel eingesetzt, um Daten zwischen zwei Geräten mittels Android Beam auszutauschen. Dazu werden die Geräte mit der Rückseite aneinander gehalten, um möglichst schnell eine Verbindung aufzubauen. Diese Technik wird zum Beispiel auch für bargeldloses Bezahlen mit dem Smartphone oder einer Kreditkarte verwendet. Für Beträge bis 25 Euro wird die PIN-Eingabe nicht benötigt [29].

Dieses Kapitel erklärt den Aufbau einer NFC-Verbindung und wie Daten daraus gelesen werden. Anschließend wird die Datenausgabe des FreeStyle Libre Sensors vorgestellt.

Grundlagen

Ein entsperartes Gerät, das NFC unterstützt und dieses auch aktiviert hat, kann eine Verbindung aufbauen, sobald es einen NFC-Tag liest. Ist dies der Fall wird ein Intent, eine Beschreibung einer auszuführenden Operation [30], an eine entsprechende Activity gesendet, die diesen verarbeitet. Besitzt das Gerät nur eine Applikation, die auf ein solches Intent wartet, erfolgt der Ablauf ohne weitere Nutzereingaben. Für den Fall, dass ein Gerät mehrere Applikationen besitzt, die verschiedene Daten via NFC erwarten, muss der Benutzer bei jedem Scan den Vorgang abrechnen und die gewünschte Applikation wählen. Um dies zu verhindern, kann das Android Tag Dispatch System eingesetzt werden. Dieses analysiert das eingescannte NFC-Tag, verarbeitet und liest NDEF¹-Daten und gibt diese an die entsprechende Applikation weiter. Um das NFC-Tag zu verarbeiten, müssen drei Schritte durchlaufen werden. Im ersten Schritt werden die Daten aus dem Tag gefiltert, im zweiten Schritt wird das Intent aufgebaut und im dritten Schritt wird basierend auf dem Intent eine Activity gestartet. Im Folgenden wird der letzte Schritt näher erklärt.

Das Tag Dispatch System unterscheidet zwischen drei Intents, die in Schritt Zwei gebildet werden. Das Intent ACTION_NDEF_DISCOVERED wird verwendet, wenn das NFC-Tag NDEF-Daten beinhaltet, die sofort verarbeitet werden können. Dieses Intent hat die höchste Priorität und wird, wenn möglich, benutzt. Kann das Tag Dispatch System keine Activity starten, kommt das Intent ACTION_TECH_DISCOVERED zum Einsatz. Dies

¹NFC Forum Data Format [31]

ist zum Beispiel der Fall, wenn das Tag keine NDEF-Daten beinhaltet, die Technologie aber bekannt ist [32]. Hierbei muss die Activity mit den Rohwerten in Byte-Format arbeiten [33]. Wartet auch auf dieses Intent keine Applikation, steht an letzter Stelle das ACTION_TAG_DISCOVERED Intent. In Abbildung 6.2 wird der Ablauf des Tag Dispatch Systems noch einmal verdeutlicht [32].

Nach welchem Intent in der Applikation gefiltert werden soll, kann im Android Manifest festgelegt werden [32].

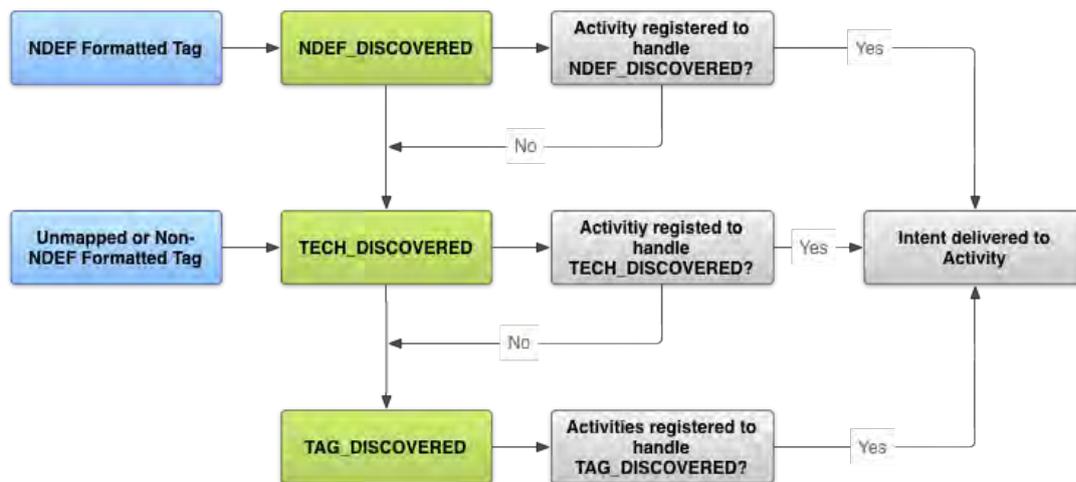


Abbildung 6.2: Tag Dispatch System [32]

Sensordaten auslesen

Um Informationen über das NFC-Tag des FreeStyle Libre Sensors zu bekommen, wurde die App NFC TagInfo verwendet [34]. In Abbildung 6.3 und 6.4 sind die Ergebnisse zu sehen. Das NFC-Tag des Sensors arbeitet mit der Technologie NfcV, was bedeutet, dass das Tag Dispatch System ein ACTION_TECH_DISCOVERED Intent an die Applikation übergibt. Des Weiteren kann NFC TagInfo die Information liefern, dass die Daten in 244 Blöcke mit der Größe von jeweils acht Byte aufgeteilt sind.

6 Implementierung



Abbildung 6.3: NFC TagInfo Tag information

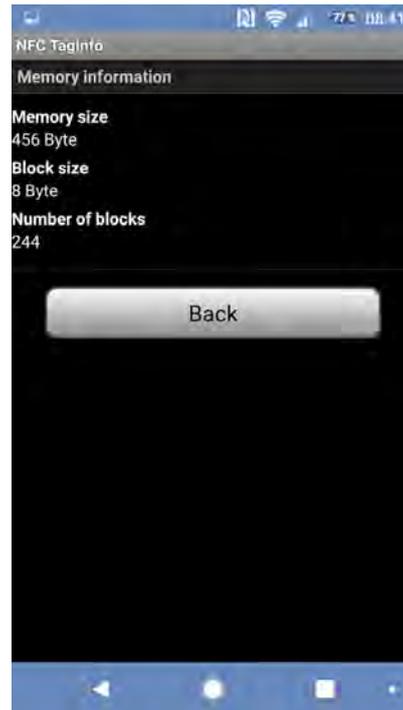


Abbildung 6.4: NFC TagInfo Memory information

Von den 244 Blöcken sind für die Messung des Blutzuckerwerts lediglich die ersten 40 Blöcke von Bedeutung. Außerdem besteht ein Datenpaket aus sechs Bytes. In Abbildung 6.5 ist eine Aufteilung der ersten 40 Blöcke dargestellt. Welche Information in welchem Abschnitt gespeichert wird, wird nachfolgend erklärt.

Im ersten Datenpaket wird die Seriennummer des Sensors gespeichert, welche für die Messung allerdings nicht von Bedeutung ist. In den Bytes 28-123 wird eine Liste gespeichert, welche 16 Werte beinhaltet. Pro Minute nimmt der Sensor eine Messung vor, die in die genannte Liste gespeichert wird [35] [36]. Das heißt, die Liste beinhaltet die Messwerte der letzten 16 Minuten und der neuste Wert ist maximal 59 Sekunden alt. Des Weiteren fungiert die Liste als Ringpuffer. Wurden also bereits 16 Werte gespeichert, wird der älteste Wert überschrieben. In Byte 26 wird der Zeiger gespeichert, der auf das Datenpaket zeigt, in welches der nächste Wert geschrieben wird [36]. In den darauffolgenden Bytes 124-315 werden 32 Messwerte gespeichert. Dies sind die Werte

der letzten 8 Stunden mit einem Abstand von 15 Minuten. Der Zeiger für den ältesten Wert, wird in Byte 27 gespeichert. Zuletzt wird in den Bytes 316 und 317 ein Timer gespeichert, der das Alter des Sensors zählt [35].

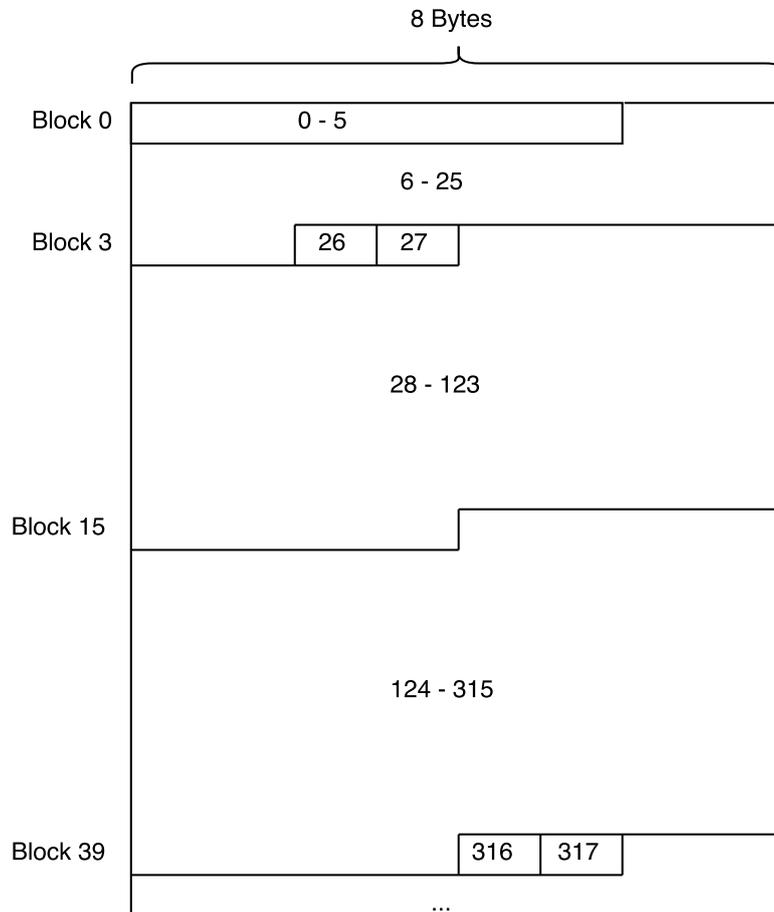


Abbildung 6.5: Übersicht Sensordaten [35]

Um den neusten Messwert zu erhalten müssen also die Bytes 28-123, sowie das Byte 26 analysiert werden. Der Glucosewert ist in den ersten zwei Bytes eines Datenpakets gespeichert [37], siehe Abbildung 6.6, und kann mit Hinzufügen eines Offsets und eines Faktors ausgerechnet werden [38]. Dazu werden die Byte-Werte in einen Hexadezimalwert und anschließend in einen Integerwert umgerechnet. In der Applikation wurde der Offset auf -20 und der Faktor auf 0,14 gesetzt.

6 Implementierung

Der Glucosewert ergibt sich also aus folgender Gleichung [38].

$$\text{Glucosewert} = \text{Rohdaten} \cdot \text{Faktor} + \text{Offset}$$

In den restlichen Bytes des Datenpakets werden Informationen über die Sensortemperatur und andere Daten gespeichert [38], welche für den Glucosewert nicht von Bedeutung sind.

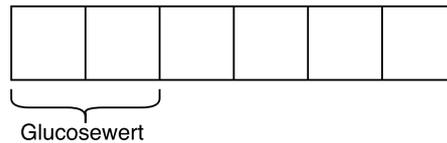


Abbildung 6.6: 6-Byte Datenpaket

Um nun das Datenpaket mit dem neusten Messwert zu erhalten, wird der Zeiger aus Byte 26 um einen Wert dekrementiert.

6.2.2 Berechnung Insulindosis

Die Insulindosis berechnet sich aus mehreren Faktoren. In diesem Abschnitt wird erklärt, wie sich diese Berechnung zusammenstellt.

Zunächst müssen die Kohlenhydrate, welche von der Art und des Gewichts einer Mahlzeit und eines Getränks abhängig sind, bestimmt werden. Die nachfolgende Erklärung richtet sich nach der Berechnung einer Portion Nudeln. Im Speicher hinterlegt sind in diesem Fall das maximale Gewicht und die Kohlenhydrate für 100g, welche in der Gleichung als KE100g beschrieben sind. Für die Berechnung wird ebenfalls der maximale Wert des Mengensliders, welcher bei 10 liegt, und die aktuelle Sliderposition benötigt. Daraus ergibt sich:

$$\text{aktuellesGewicht} = \text{maximalesGewicht} / 10 \cdot \text{Sliderposition}$$

$$\text{Insulin1} = \text{KE} = \text{aktuellesGewicht} / 100 \cdot \text{KE100g}$$

Der erste Insulinwert entspricht der Menge der KE und muss nun personalisiert wer-

den. Je nach Tageszeit wird der entsprechende Insulinfaktor für morgens, mittags oder abends verrechnet.

$$Insulin2 = Insulin1 \cdot Insulinfaktor$$

Anschließend wird die Insulindosis abhängig vom Blutzuckerwert berechnet. Hier können mehrere Fälle auftreten. Im ersten Fall liegt der Blutzuckerwert innerhalb des Zielbereichs und an der Insulindosis muss nichts verändert werden. Im zweiten Fall liegt der Blutzuckerwert oberhalb des Zielbereichs und zur Insulindosis muss ein Korrekturwert hinzugerechnet werden. Dieser ergibt sich aus den Werten KorrekturAuf und Korrekturereinheit, welche in den Einstellungen festgelegt werden.

$$Korrekturwert = \lfloor \text{Blutzuckerwert} - \text{KorrekturAuf} \rfloor / \text{Korrekturereinheit}$$

$$Insulin3 = Insulin2 + Korrekturwert$$

Im dritten Fall liegt der Blutzuckerwert unterhalb des Zielbereichs und die Insulindosis muss entsprechend verringert werden. Dies wird mit den Werten Korrekturereinheit und Zielbereich Minimum berechnet.

$$Korrekturwert = \lfloor \text{ZielbereichMinimum} - \text{Blutzuckerwert} \rfloor / \text{Korrekturereinheit}$$

$$Insulin3 = Insulin2 - Korrekturwert$$

Hat der Benutzer eine Sportintensität ausgewählt, muss diese ebenfalls bei der Insulinberechnung berücksichtigt werden. Wurde *bisschen Sport* ausgewählt, wird die aktuelle Insulindosis um ein Drittel verringert. Wurde *anstrengender Sport* ausgewählt, wird die Insulindosis um die Hälfte vermindert.

$$Insulin4 = Insulin3 \cdot \text{Sportintensität}$$

Zum Abschluss wird der berechnete Insulinwert auf eine ganze Zahl gerundet. Liegt der Wert genau auf x,5 wird überprüft, ob der Blutzuckerwert näher am oberen oder unteren Zielbereich liegt und dementsprechend auf- oder abgerundet.

$$Insulin5 \approx Insulin4$$

Alle Werte und Berechnungen wurden mit ärztlichen Fachkräften im Bereich Diabetes abgesprochen.

6.2.3 Darstellung Mengenangabe

Im Rahmen der Bachelorarbeit beschränkt sich die Anzahl der verfügbaren Mahlzeiten auf Nudeln, Nudeln mit Soße und Pizza. Die Darstellung der Auswahlmöglichkeiten wird jeweils mit Hilfe eines anderen Konzepts implementiert. Diese werden nachfolgend vorgestellt.

Nudeln

Die Nudeln werden auf einem Teller dargestellt, auf welchen der Betrachter schräg von oben blickt, siehe Abbildung 6.7.



Abbildung 6.7: Teller [39]

Die Darstellung der Nudeln wurde mit Hilfe von Paint 3D² gezeichnet. Die Farbe der Nudeln ist Gold mit unterschiedlicher Deckkraft, um Tiefe zu visualisieren. Es befinden sich circa 10 einzelne Nudeln auf einem Bild. Um die unterschiedliche Menge auf dem Teller darzustellen, liegt über dem mittleren Bereich des Tellers ein leeres RelativeLayout. Wird der zugehörige Slider bewegt, werden pro Stufe drei ImageViews erstellt, die das Bild, das in Abbildung 6.8 zu sehen ist, präsentieren. Die Position der ImageView wird durch Zufallswerte berechnet. Wird der Slider in die entgegengesetzte Richtung bewegt, werden die zuletzt hinzugefügten ImageViews wieder gelöscht und die Menge der Nudeln verringert sich. Der Maximalwert des Sliders ist 10, das heißt es befinden sich maximal 30 ImageViews und somit 300 Nudeln auf einem Teller, zu sehen in Abbildung 6.9.

²<https://www.microsoft.com/de-de/store/p/paint-3d/9nblggh5fv99>

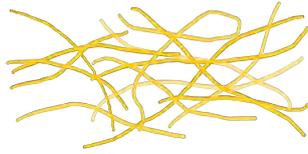


Abbildung 6.8: Anzahl Nudeln einer ImageView



Abbildung 6.9: Maximale Anzahl Nudeln

Nudeln mit Soße

Das Konzept für Nudeln mit Soße ist ähnlich dem Abschnitt 6.2.3. Der Teller wird in diesem Fall von zwei leeren RelativeLayouts überlagert. Das linke Layout beinhaltet die Nudeln und das rechte die Soße. Um die Menge der Soße zu vergrößern, wird die ImageView in Höhe und Breite angepasst. Um zwischen Nudeln und Soße zu wählen, muss auf der linken Seite das jeweils gewünschte Element ausgewählt werden, siehe Abbildung 6.10. Der Slider passt sich dem ausgewählten Element bezüglich der maximalen Menge an.

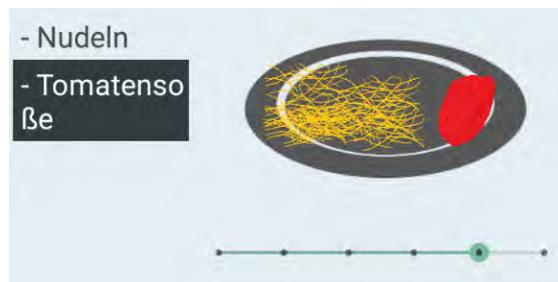


Abbildung 6.10: Nudeln mit Soße

Pizza

Die Pizza wird auf einem Teller dargestellt, auf welchen der Betrachter von oben blickt, siehe Abbildung 6.11. Die Pizza wird aus einem Stück, das mehrfach verwendet wird, zusammengestellt, vergleiche Abbildung 6.12.

6 Implementierung



Abbildung 6.11: Teller [40]



Abbildung 6.12: Pizzastück [41]

Da Pizza in Stücken abgerechnet wird, wird jede Menge mit Hilfe eines neuen Bildes repräsentiert, das die gewünschte Menge an Pizzastücken beinhaltet. In Abbildung 6.13 ist die Pizza in sechs verschiedenen Größen zu sehen.

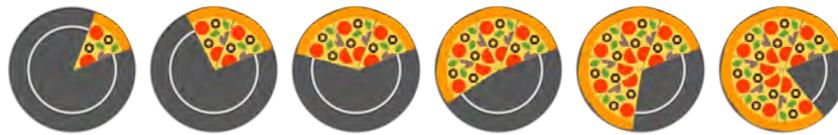


Abbildung 6.13: Menge Pizza

6.2.4 Erinnerung

Die Erinnerung kann in den Einstellungen aktiviert oder deaktiviert werden. Ist sie aktiviert, bekommt der Benutzer eine Benachrichtigung, wenn er sein Zuhause verlässt, um an Traubenzucker und Insulin erinnert zu werden. Dies wird mit Hilfe eines Broadcast-Receiver realisiert, welcher im Hintergrund läuft. Dieser wird informiert, wenn sich der Status des WLAN-Netzes ändert. Wird das WLAN deaktiviert, so wird der Name des davor verbundenen Netzes mit dem gespeicherten Namen des Heimnetzwerks verglichen. Stimmen diese überein, wird eine Benachrichtigung angezeigt [42].

6.3 Datenspeicherung

In Kapitel 4.2 wurde bereits der Aufbau des Datenmodells beschrieben. Für die Datenspeicherung der Einstellungen wird die SharedPreferences API für kleine Datensätze verwendet. In einer SharedPreferences-Datei werden Schlüssel-Wert-Paare verwaltet,

welche gelesen und geschrieben werden können. Für eine Anwendung können mehrere SharedPreferences-Dateien angelegt werden [43].

Für die Daten der Getränke und der Mahlzeiten wird eine externe JSON-Datei verwendet. Mit Hilfe des JsonReaders werden Datensätze aus der Datei gelesen und in JsonObject und JsonArrays geparkt [44]. Das Schreiben in die Datei wird im Bezug auf die Applikation nicht benötigt.

6.4 Frameworks

In diesem Kapitel werden die Frameworks vorgestellt, die in diesem Projekt verwendet wurden. Diese werden im App Gradle unter Dependencies vermerkt und können anschließend verwendet werden. Jedes Framework beinhaltet eine Komponente, welche die Gestaltung der Oberfläche unterstützt.

6.4.1 RangeSeekBar

Die RangeSeekBar funktioniert ähnlich der Seekbar von Android. Der Unterschied liegt darin, dass die RangeSeekBar zwei Regler implementiert hat, um einen Wertebereich festzulegen. Unterstützt wird ein Bereich zwischen ganzen Zahlen und Gleitkommazahlen. Außerdem kann eingestellt werden, ob über den Reglern der aktuell ausgewählte Wert angezeigt wird oder nicht [45]. In Abbildung 6.14 ist die RangeSeekBar aus der Anwendung dargestellt.

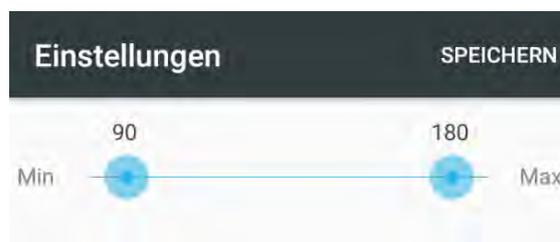


Abbildung 6.14: RangeSeekBar

6 Implementierung

In der Applikation wird die RangeSeekBar in den Einstellungen zur Festlegung des Zielbereich verwendet, wofür die Anzeige der Werte über den Reglern aktiviert wurde. Der linke Regler gibt den minimalen Zielbereich und der rechte Regler den maximalen Zielbereich des Blutzuckerwerts an.

6.4.2 VerticalSeekBar

Die VerticalSeekBar funktioniert ebenfalls ähnlich der Seekbar von Android. Im Gegensatz zur Android Seekbar ist die VerticalSeekBar vertikal ausgerichtet. Unter anderem kann das Maximum festgelegt und die Animation des Schiebers implementiert werden. Das Framework wird von Android 2.3 bis 7.x unterstützt [46].

In der Applikation wird die vertikale Seekbar für die Getränke verwendet, welche die Menge des Inhalts in Bezug auf die Höhe des Glases angibt, siehe Abbildung 6.15. Das Maximum der vertikalen Seekbar ist in diesem Fall drei.

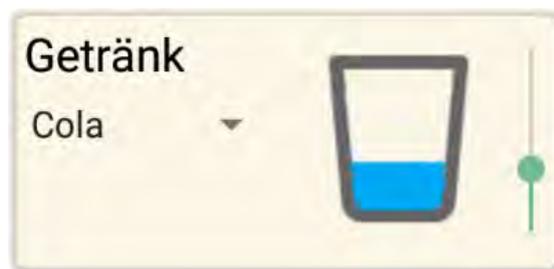


Abbildung 6.15: VerticalSeekBar

6.4.3 CardView

Die Komponente CardView wird dafür verwendet, Informationen hervorzuheben. Diese können als Foto oder Text repräsentiert werden und zusätzlich kann eine Aktion implementiert werden. Die CardView findet zum Beispiel Anwendung, wenn eine Information aus mehreren zusammengehörigen Komponenten oder interaktivem Inhalt, wie Buttons oder Kommentaren, besteht. Wichtige Informationen werden auf einer CardView oben

platziert und, wenn gewünscht, durch eine Überschrift hervorgehoben. Sekundäre Informationen werden anschließend präsentiert. Der Hintergrund von CardViews kann ebenfalls personalisiert werden, Farbe oder Hintergrundbilder sind möglich.

Innerhalb einer CardView kann vertikal gescrollt werden, wenn der Inhalt zu lang ist und die maximale Höhe der Karte nicht ausreicht. Swipe-Gesten nach rechts oder links sollen die gesamte Karte entfernen. Die CardView wird vom Hintergrund hervorgehoben, indem Schatten an den Rändern eine Erhebung visualisieren. Die Ecken sind abgerundet [47]. In Abbildung 6.16 ist die Anwendung der CardView in der Applikation zu sehen. Eine Karte repräsentiert eine Eingabe für den Benutzer. Der Hintergrund ist farbig und von einem Icon überlagert, das nicht die volle Deckkraft besitzt.



Abbildung 6.16: CardView

6.4.4 RecyclerView

Die RecyclerView findet Anwendung, wenn eine Menge an Informationen, welche sich zur Laufzeit durch Benutzereingaben oder anderen Ereignissen ändern können, als Liste oder Aufzählung präsentiert werden soll. Die Komponente ist an einen Adapter gebunden, welcher die Daten verarbeitet und an die Oberfläche weiter gibt. Das Layout kann horizontal oder vertikal mit Hilfe eines LayoutManagers ausgerichtet werden. Der

6 Implementierung

Ablauf und die Zusammenhänge werden in Abbildung 6.17 deutlich. Der Entwickler hat zusätzlich die Möglichkeit das Hinzufügen oder Entfernen von einzelnen Daten zu animieren, indem ein `ItemAnimator` implementiert wird [48].

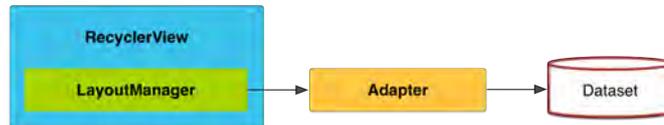


Abbildung 6.17: RecyclerView Workflow [48]

Die RecyclerView wird in der Anwendung mehrmals benutzt. Zum einen auf der Hauptseite, dort werden die ausgewählten Mahlzeiten als vertikale Aufzählung dargestellt, siehe Abbildung 6.32. Zum anderen wird die RecyclerView bei der Essensauswahl benutzt, um die Mahlzeiten in den Kategorien vertikal und um ausgewählte Mahlzeiten horizontal im oberen Abschnitt anzuzeigen, vergleiche Abbildung 6.33.

6.4.5 Material Design

Für die Oberflächengestaltung wurde mit dem Material Design von Android gearbeitet. Dieses bietet Komponenten, Richtlinien und Empfehlungen bezüglich der Gestaltung von Android-Anwendungen. Das Material Design wird ständig optimiert und erweitert [49].

Komponenten

Im Folgendem wird eine Auswahl an Komponenten vorgestellt, welche in der Anwendung verwendet wurden.

Buttons lösen eine Aktion aus, wenn sie geklickt werden. Es gibt drei Typen: Den Floating Action-Button, den Flat-Button und den Raised-Button. Die zwei zuletzt genannten Typen werden in der Applikation verwendet.

Der Flat-Button, siehe Abbildung 6.18 links, hat keine Hintergrundfarbe und wirkt daher flach. Diese werden in Toolbars, Dialogen und innerhalb von Inhalten verwendet. Der Raised-Button, auf der rechten Seite in Abbildung 6.18 zu sehen, wirkt durch Farbe und Schatten von der Oberfläche abgehoben. Er wird in flachen Layouts verwendet, um wichtige Funktionen zur Geltung zu bringen [49].



Abbildung 6.18: Flat Button und Raised Button

Chips repräsentieren eine Entität in einem kleinen, markierten Abschnitt. Die Entitäten können verschiedene Typen annehmen, zum Beispiel Texte oder Kontakte. Auch Icons können hinzugefügt werden. Wenn Chips wieder entfernt werden können, befindet sich am rechten Rand ein Icon zum löschen, vergleiche Abbildung 6.19. Höhe und Breite des Chips sowie Abstände des Texts und des Icons werden im Material Design standardisiert [49].

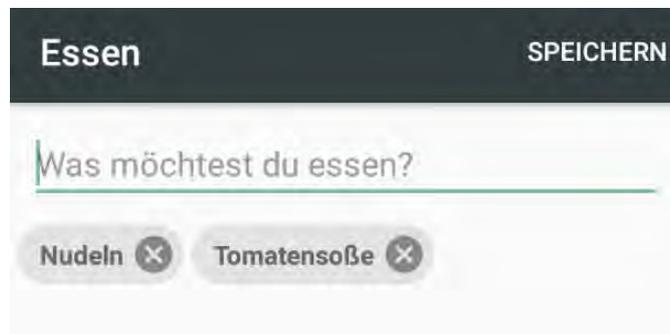


Abbildung 6.19: Chips

Dialoge dienen als Hinweis für den Benutzer. Dabei können harmlose oder kritische Informationen dem Benutzer näher gebracht oder Entscheidungen gefordert werden.

6 Implementierung

Dialoge unterbrechen die Benutzung der Applikation und werden solange in den Fokus gestellt, bis der Nutzer eine Aktion ausführt. Inhalte können Überschriften, Icons, Texte und Buttons, siehe Abschnitt 6.4.5, zur weiteren Navigation sein [49].

Ein Alert-Dialog, wie in Abbildung 6.20 zu sehen, informiert den Benutzer zum Beispiel über eine bestimmte Änderung. Er besteht aus einem Satz oder einer Frage, welcher die Situation kurz zusammenfasst. In Abbildung 6.20 muss der Benutzer eine Entscheidung bezüglich des Löschens eines Elements treffen. Er hat die Möglichkeit, die Änderung abbrechen oder diese durchzuführen. Dabei befindet sich der Button, der eine Aktion in Gang setzt und den weiteren Prozess fortführt, auf der rechten Seite. Der Button, welcher eine Aktion abbricht und den Benutzer zum Ausgangszustand der Anwendung zurück navigiert, wird auf der linken Seite des erst genannten Buttons platziert [49].

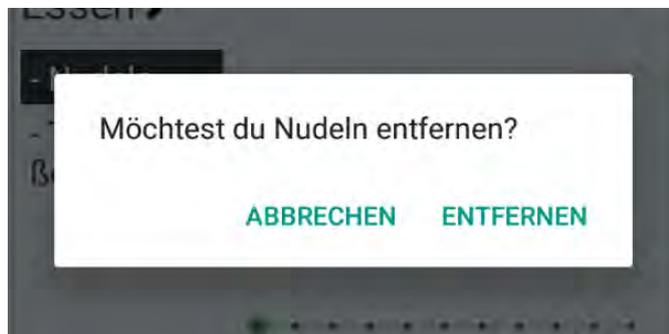


Abbildung 6.20: Alert Dialog

Menüs überlagern durch bestimmte Aktionen die eigentlichen Elemente. Dem Benutzer werden mehrere Auswahlmöglichkeiten in Form einer vertikalen Liste geboten. Eine solche Liste muss aus mindestens zwei Elementen bestehen, welche jeweils eine eigenständige Aktion ausführen. Ein Element besteht aus einer Zeile des Menüs. Des Weiteren können diese deaktiviert sein oder weitere Untermenüs aufrufen. Werden zu viele Elemente in einem Menü untergebracht, kann innerhalb diesem gescrollt werden [49].

In Abbildung 6.21 bietet das Menü dem Benutzer die Möglichkeit, zu den Einstellungen, den Informationen und der Hilfe zu gelangen.

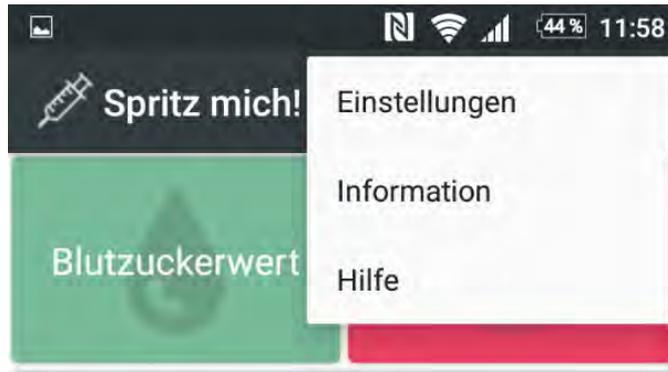


Abbildung 6.21: Menü

Radio Buttons sind eine Art der Auswahlkomponenten. Sie bieten dem Nutzer eine Menge an Optionen, aus welchem eine Option gewählt werden soll. Radio Buttons werden verwendet, wenn alle Auswahlmöglichkeiten für den Nutzer sichtbar sein sollen. Weitere Auswahlkomponenten sind die Checkbox, in welcher mehrere Optionen einer Menge ausgewählt werden können, und Switches, welche eine Option aktivieren oder deaktivieren können [49].

In der Anwendung werden Radio Buttons für die Auswahl der Sportintensität verwendet. Der Nutzer hat die Möglichkeit zu wählen, ob er keinen, ein bisschen oder anstrengenden Sport gemacht hat, siehe Abbildung 6.22.



Abbildung 6.22: Radio Buttons

6 Implementierung

Sliders geben dem Benutzer die Möglichkeit, einen Wert aus einem festgelegten Bereich, durch das Verschieben von Reglern, zu wählen. Die Komponente findet im Android Betriebssystem zum Beispiel Anwendung für das Einstellen der Helligkeit, der Lautstärke oder der Farbsättigung. Der minimale und der maximale Wert kann jeweils auf der linken und auf der rechten Seite des Sliders vermerkt werden. Es wird unterschieden zwischen konstanten und diskreten Slidern. Der konstante Slider benötigt keinen konkreten Wert, um Anpassungen in den zugehörigen Einstellungen vorzunehmen. Der diskrete Slider hingegen, liest für jede Einstellung des Sliders einen genauen Wert, der im vorgegebenen Wertebereich liegt [49]. In Abbildung 6.23 ist zu sehen, in welchen Abständen Werte gewählt werden können und wie viele dem Benutzer zur Auswahl stehen. Mit Hilfe des Sliders kann in diesem Fall die Menge des Essens auf dem Teller gewählt werden.

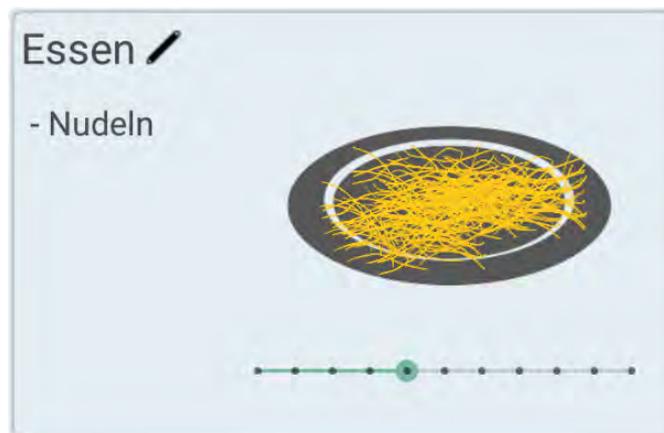


Abbildung 6.23: Slider

Die Snackbar wird am unteren Rand des Bildschirms in Abhängigkeit einer bestimmten Aktion angezeigt. Die Animation erfolgt von unten nach oben und von oben nach unten, wenn die Komponente wieder deaktiviert oder eine neue Snackbar aktiviert wird. Befinden sich weitere Elemente am unteren Rand des Bildschirms, werden diese von der Snackbar überlagert. Die Information wird auf der Snackbar in Form von Text mit einer optionalen Aktion präsentiert. Diese Aktion kann ein Button mit der Funktion 'Schließen'

oder 'Rückgängig' sein, welche die Snackbar wieder deaktiviert. Das Swipen nach links oder rechts bewirkt das gleiche Ergebnis sowie ein Timer, nach dessen Ablauf die Snackbar automatisch geschlossen wird. Diese Funktionalitäten können von Entwicklern modifiziert und deaktiviert werden [49].

In der Applikation zeigt die Snackbar das Ergebnis der berechneten Insulindosis an, siehe Abbildung 6.24. Es ist kein Timer hinterlegt, sodass diese nicht nach einer bestimmten Zeit deaktiviert wird. Außerdem wurde die Swipe-Geste, welche für das Schließen zuständig ist, deaktiviert. Die Aktion 'Erledigt' schließt die Snackbar und setzt alle eingegebenen Daten zurück. Möchte der Benutzer seine Eingabe bearbeiten, ist die angezeigte Information nicht mehr gültig, weshalb die Snackbar deaktiviert wird bis der Benutzer die neuen Eingaben bestätigt.

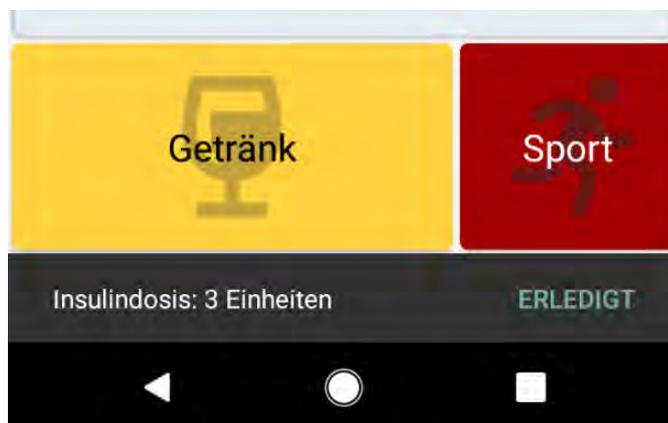


Abbildung 6.24: Snackbar

Tabs erleichtern das Wechseln zwischen mehreren Ansichten einer Applikation. Sie werden über dem wechselnden Inhalt in einer Zeile angezeigt. Dort wird für jedes Tab ein Label definiert, das dessen Inhalt mit Hilfe von Text und Icons beschreibt. Um zwischen den Tabs zu wechseln, kann der Benutzer von Swipe-Gesten Gebrauch machen oder das gewünschte Tab direkt auswählen [49]. Abbildung 6.25 zeigt die Verwendung der Komponente in der Applikation. Jedes Tab repräsentiert eine Kategorie des Essens, sodass der Benutzer Mahlzeiten schnell und einfach zusammenstellen kann.

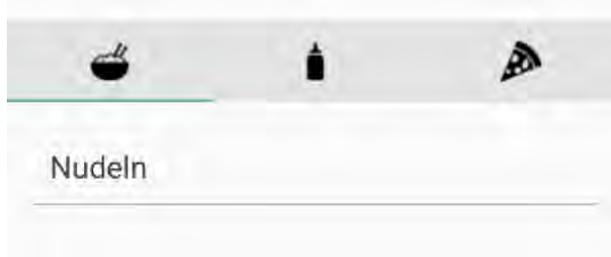


Abbildung 6.25: Tabs

Patterns

Im Folgendem wird eine Auswahl an Patterns bezüglich des Designs vorgestellt, welche in der Anwendung verwendet wurden.

Empty States treten auf, wenn zum Beispiel Listen ohne Inhalt angezeigt werden oder eine Suche keine Vorschläge gespeichert hat. Diese Fälle kommen selten vor, müssen aber trotz allem beachtet werden. Android schlägt zum Beispiel vor, ein neutrales Bild mit einem kurzen, informativen Text als Platzhalter zu benutzen. Alternativ kann dem Benutzer beispielhafter Inhalt oder eine Erklärung des weiteren Vorgangs präsentiert werden [49]. In der Applikation wird der Benutzer durch einen kurzen Text darauf hingewiesen, dass noch kein Essen gewählt wurde, dies aber notwendig ist, vergleiche Abbildung 6.26.

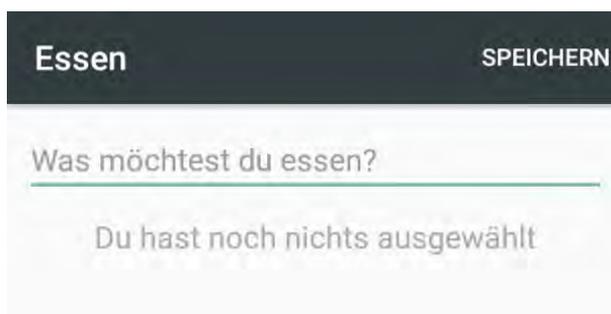


Abbildung 6.26: Empty State

Errors können in jeder Situation der Anwendung auftreten, wenn eine Aktion nicht vollständig ausgeführt werden kann. Dabei können Nutzereingaben, aber auch Implementierungsfehler der Auslöser sein. Tritt ein solcher Fehler auf, soll der Benutzer über den Status und den Grund informiert werden. Liegt die Ursache bei einer Benutzereingabe, ist es hilfreich, den Fehler genau zu beschreiben und dem Benutzer zu einer richtigen Eingabe zu verhelfen [49].

In Abbildung 6.27 wird der Benutzer mit Hilfe eines Toasts, einer kurzlebigen Information am unteren Bildschirmrand, auf eine ungültige Benutzereingabe hingewiesen.



Abbildung 6.27: Fehler in der Benutzereingabe

Gesten werden unterteilt in Touch-Mechanismen und Touch-Aktivitäten. Zu dem Touch-Mechanismus fallen alle Aktivitäten, die mit den Fingern aktiviert werden. Beispiele dafür sind ein einfacher Touch, ein doppelter Touch, eine Swipe-Bewegung oder ein langer Touch. Eine Touch-Aktivität wird durch die Kombination mehrerer Touch-Mechanismen ausgelöst [49]. In der Applikation hat der Benutzer zum Beispiel die Möglichkeit durch ein langes Drücken auf ein ausgewähltes Essen, dies zu löschen.

Der Launch Screen dient in der Zeit, in der eine Applikation startet, als Platzhalter. Es ist die erste Ansicht, die der Benutzer zu sehen bekommt. Für Apps mit einer kurzen Ladezeit wird eine Placeholder UI verwendet, welche leeren Inhalt zusammen mit den Strukturelementen wie die Status-Bar oder Navigationsleiste anzeigt. Der Branded Launch Screen wird eingesetzt, wenn das Logo und der Name der App präsentiert werden sollen. Dabei werden neben dem Launch Screen keine Strukturelemente angezeigt.

6 Implementierung

Sobald die App geladen ist, wird der Launch Screen mit dem eigentlichen Inhalt ersetzt [49].

Eine Benachrichtigung wird eingesetzt, wenn eine Applikation nicht aktiv verwendet wird, aber der Nutzer über ausgeführte Aktionen informiert werden soll. Erscheint eine Benachrichtigung, hat der Benutzer verschiedene Möglichkeiten, mit dieser zu kommunizieren. Durch einen Klick auf diese wird die App geöffnet und eine Ansicht präsentiert, die mit dem Inhalt der Benachrichtigung zusammenhängt. Ist das Smartphone im Sperrmodus, gelangt der Benutzer durch einen Doppelklick zur Passwort- oder PIN-Eingabe und anschließend direkt in die App. Außerdem kann die Möglichkeit bestehen, eine Benachrichtigung durch eine Swipe-Bewegung zu vergrößern, um weitere Informationen anzuzeigen. Eine Benachrichtigung kann ebenfalls durch horizontale Swipe-Bewegungen geschlossen werden [49].

In Abbildung 6.28 ist eine einfache Benachrichtigung mit einem Icon, einer Überschrift und einem kurzen Inhaltstext zu sehen.

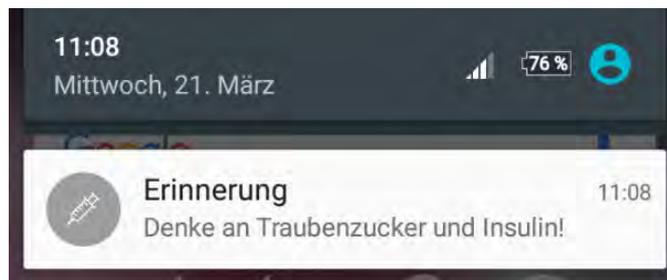


Abbildung 6.28: Benachrichtigung

6.5 Vorstellung der Applikation

In diesem Abschnitt wird die Applikation als Ganzes vorgestellt. Anhand von Screenshots, welche möglichst viele Zustände der Applikation präsentieren, wird die Funktionsweise und Navigation erklärt.

Öffnet der Benutzer die Anwendung, wird direkt die Hauptseite angezeigt, siehe Abbildung 6.29. Auf dieser Ansicht, können direkt die wichtigen Daten zur Berechnung der Insulindosis eingegeben werden. Dazu gehören der Blutzuckerwert, die Auswahl, das Essen, das Getränk und Sport. Um einen Wert zu bearbeiten, muss der entsprechende Abschnitt ausgewählt werden, damit die Eingabefelder sichtbar werden. Oben rechts hat der Benutzer die Möglichkeit, ein Menü zu öffnen, mit welchem er zu den Einstellungen, der Hilfe und der Information navigieren kann.



Abbildung 6.29: Hauptseite

Im Bereich Blutzuckerwert befindet sich ein einfaches Textfeld, in welches der Wert manuell eingegeben werden kann. Die Information teilt dem Benutzer mit, dass er auch die Möglichkeit hat, den Blutzuckerwert mit einem FreeStyle Libre Sensor zu messen,

6 Implementierung

siehe Abbildung 6.31. Der gemessene Wert kann anschließend auch bearbeitet oder gelöscht werden. Im Bereich Auswahl kann der Benutzer wählen, ob er ein Essen, ein Getränk oder beides bearbeiten möchte. Wird das entsprechende Symbol aktiviert, werden die Eingabefelder des zugehörigen Bereichs sichtbar. Die Eingabe ist auch ohne die Benutzung des Auswahlfeldes möglich. Im Bereich Essen, kann die Menge des bereits gewählten Essens angepasst werden.



Abbildung 6.30: Eingabe Pizza

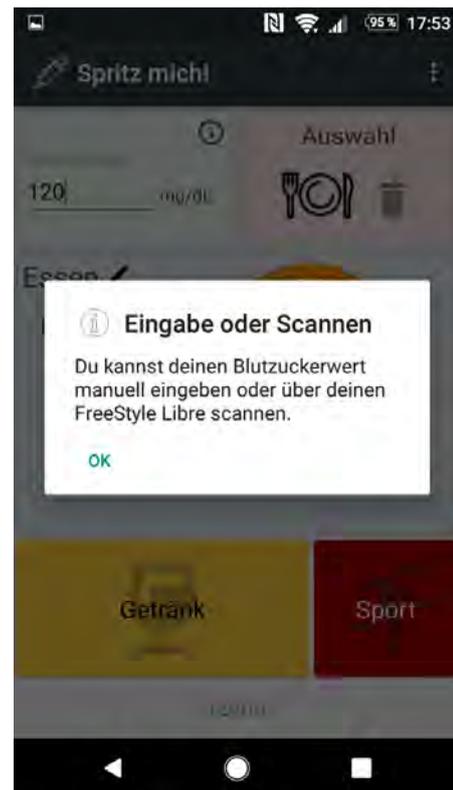


Abbildung 6.31: Eingabe Blutzuckerwert Information

Essen kann über das Bearbeiten-Symbol hinzugefügt oder bearbeitet werden. Dafür wird eine neue Seite geöffnet, welche in Abbildung 6.33 zu sehen ist. Hier kann der Benutzer über die Suche sein gewünschtes Essen eingeben, was ihm durch die Autovervollständigung erleichtert wird. Alternativ zur manuellen Suche, sind alle Mahlzeiten in Kategorien unterteilt und können so im unteren Bereich ausgewählt werden. Die ausgewählten

6.5 Vorstellung der Applikation

Elemente werden unter der Suche angezeigt und können dort auch wieder entfernt werden. Auf der Hauptseite kann der Benutzer durch langes Drücken auf ein Element dieses ebenfalls löschen.

Im Rahmen der Bachelorarbeit ist die Anzahl der Lebensmittel auf einem Teller auf zwei beschränkt. Auch bestimmte Kombinationen, wie Nudeln mit Pizza, sind nicht erlaubt. Das Getränk kann direkt auf der Hauptseite ausgewählt werden und die Menge wird durch die Füllhöhe des Glases angezeigt. Die Eingabe des Sports ist ebenfalls optional. Der Benutzer kann wählen zwischen keinem Sport, ein bisschen Sport und anstrengendem Sport. Wird die Eingabe des Bereichs nicht aktiviert, entspricht das der Auswahl von keinem Sport.



Abbildung 6.32: Eingabe Nudeln

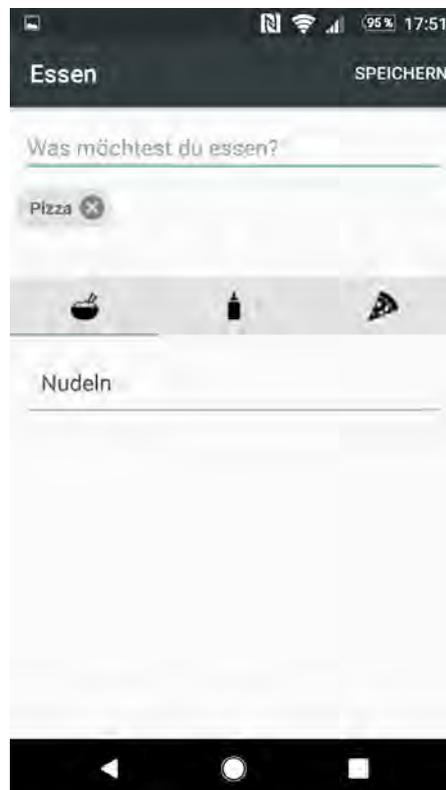


Abbildung 6.33: Essensauswahl

Wurde für den Blutzucker ein gültiger Wert im Bereich 40 bis 600 eingegeben und mindestens für das Essen oder das Getränk etwas ausgewählt, kann die Insulindosis berechnet

6 Implementierung

werden. Diese erscheint am unteren Rand des Bildschirms. Die Eingaben können anschließend bearbeitet werden, indem auf eine beliebige Stelle des Bildschirms geklickt wird. Ist der Benutzer fertig, kann der Aktionsbutton der Snackbar aktiviert werden und alle Eingaben werden zurückgesetzt.

In Abbildung 6.34 ist die Übersicht der Einstellungen zu sehen. Dabei werden jeweils unter dem Namen der Einstellung die aktuell gespeicherten Daten angezeigt, sodass der Benutzer alle personalisierten Werte auf einen Blick hat. Eingestellt werden können die Insulinfaktoren, der Korrekturwert, der Zielbereich und die Erinnerung. Über den Zurück-Button wird die Hauptseite wieder erreicht. Die Werte werden bereits in der jeweiligen Einstellung durch einen Speichern-Button aktualisiert.



Abbildung 6.34: Übersicht Einstellungen

Für die Insulinfaktoren werden drei Werte gespeichert, der Faktor für morgens, für mittags und für abends, vergleiche Abbildung 6.35. Der Benutzer kann halbe und ganze

6.5 Vorstellung der Applikation

Werte in einem Bereich von null bis fünf wählen. Für den Zielbereich werden zwei Werte gespeichert, der minimale und der maximale Zielwert, siehe Abbildung 6.36. Der größte Zielbereich, der gewählt werden kann, liegt zwischen 80 und 190 mg/dL. Auf den Reglern des Sliders wird dem Benutzer die aktuell ausgewählten Werte angezeigt.



Abbildung 6.35: Einstellungen
Insulinfaktoren

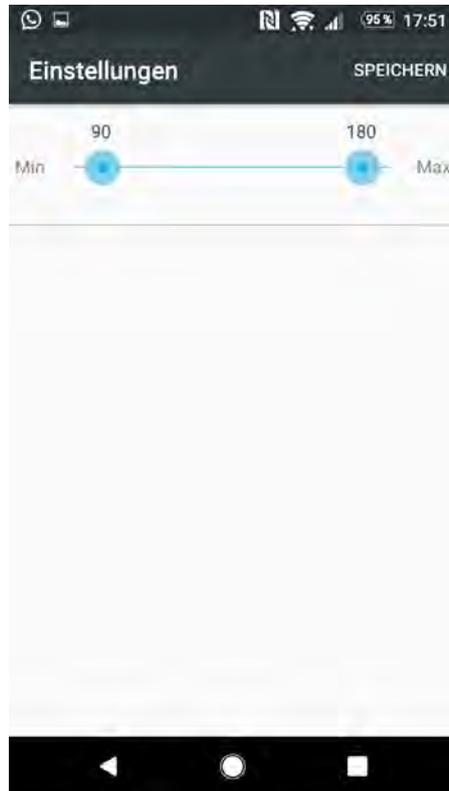


Abbildung 6.36: Einstellungen
Zielbereich

In der Einstellung Korrekturwert, siehe Abbildung 6.37, wird der Blutzuckerwert gespeichert, auf welchen korrigiert werden soll. Zusätzlich wird gespeichert, um wie viel der Blutzuckerwert pro Einheit Insulin sinkt. Dies wird zur Berechnung des Korrekturwerts benötigt, wenn der Blutzuckerwert oberhalb des Zielbereichs liegt.

Die Erinnerung kann aktiviert oder deaktiviert werden. Ist sie aktiviert, bekommt der Benutzer eine Erinnerung in Form einer Benachrichtigung, wenn er sein Zuhause verlässt. Dafür wird der Name des Heimnetzes gespeichert, zu sehen in Abbildung 6.38.

6 Implementierung

Das bedeutet, dass der Benutzer im Falle einer aktivierten Erinnerung Zugriff auf das angegebene WLAN-Netzwerk benötigt, damit diese Funktion ausgeführt werden kann.

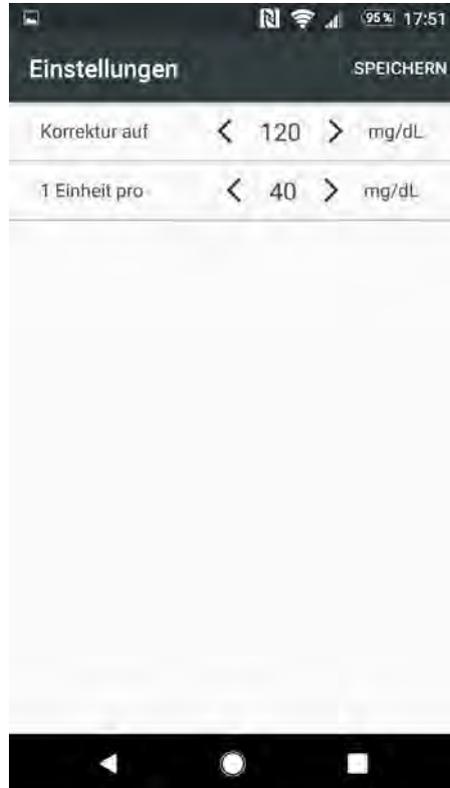


Abbildung 6.37: Einstellungen Korrekturwert



Abbildung 6.38: Einstellungen Erinnerung aktiviert

7

Anforderungsabgleich

In diesem Kapitel wird der Abgleich der Anforderungen aus Kapitel 3 diskutiert. Dabei wird überprüft, ob alle Funktionalitäten, welche zu Beginn festgelegt worden sind, umgesetzt wurden. Außerdem wird deutlich, welche Anforderungen angepasst worden sind, da eventuell die Realisierung auf diese Weise nicht möglich war und nicht alle Faktoren beachtet wurden. Für nachfolgende Projekte kann auf diesen Abgleich zurückgegriffen werden, um Anforderungen von Beginn an realistisch zu gestalten.

7.1 Abgleich funktionale Anforderungen

Im Bereich funktionale Anforderungen wurde alles erfüllt.

Bezeichnung	Erfüllt	Erklärung
FA#01	Ja	Es können nur die Einstellungen eines Nutzers gespeichert werden. Somit kann die App nur von einem Nutzer verwendet werden.
FA#02	Ja	Der Benutzer hat jederzeit Zugriff auf die Einstellungen und kann Änderungen vornehmen.
FA#03	Ja	Es werden drei Werte unter den Insulinfaktoren gespeichert.
FA#04	Ja	In den Einstellungen können zwei Werte im Bereich Korrekturwert geändert werden.
FA#05	Ja	Der Zielbereich hat eine untere und eine obere Grenze, die in den Einstellungen festgelegt wird.

7 Anforderungsabgleich

FA#06	Ja	Die Erinnerung ist aktivierbar und deaktivierbar.
FA#07	Ja	In der Einstellung Erinnerung kann nach Aktivierung der WLAN-Name eingegeben werden.
FA#08	Ja	Ist die Erinnerung aktiviert, bekommt der Benutzer beim Verlassen des Heimnetzwerks eine Benachrichtigung.
FA#09	Ja	Wird die App das erste mal gestartet, wird der Benutzer darauf hingewiesen, die Daten in den Einstellungen zu personalisieren. Wird dies übersprungen, werden Standardwerte eingetragen.
FA#10	Ja	Auf der Startseite kann der aktuelle Blutzuckerwert manuell oder über einen FreeStyle Libre Sensor eingegeben werden.
FA#11	Ja	Auf der Startseite kann gewählt werden, ob etwas gegessen oder getrunken wird.
FA#12	Ja	Lebensmittel können nach Kategorie ausgewählt und gesucht werden.
FA#13	Ja	Ausgewählte Lebensmittel können entfernt werden.
FA#14	Ja	Die Menge des Lebensmittels kann zwischen 10 Stufen gewählt werden. Bei manchen Lebensmitteln ist die maximale Menge von den Stücken abhängig.
FA#15	Ja	Der Benutzer kann die Sportintensität zwischen kein Sport, ein bisschen Sport und anstrengendem Sport wählen.
FA#16	Ja	Das Getränk muss nicht ausgewählt werden. Die Menge des Getränks kann zwischen drei Stufen gewählt werden.
FA#17	Ja	Nach Eingabe des Blutzuckerwerts und eines Essens oder Getränks wird die zu spritzende Insulineinheit ausgegeben.
FA#18	Ja	Der Benutzer kann die Eingaben wieder ändern, nachdem er 'Fertig' gedrückt hat.
FA#19	Ja	Die Eingaben können alle zurückgesetzt werden, wenn der Benutzer 'Erledigt' drückt.

Tabelle 7.1: Abgleich funktionale Anforderungen

7.2 Abgleich nichtfunktionale Anforderungen

Die meisten nichtfunktionalen Anforderungen wurden erfüllt. Da es sich bei der Anwendung um einen Prototypen handelt, kann nicht über alle Anforderungen eine Aussage gemacht werden.

Bezeichnung	Erfüllt	Erklärung
NFA#01	Ja	Für das Design wurden Komponenten und Patterns des Material Design gewählt.
NFA#02	Ja	Der Benutzer wird auf möglichst alle fehlerhaften Eingaben hingewiesen.
NFA#03	-	Über den Speicherverbrauch kann keine Aussage gemacht werden.
NFA#04	-	Über den Akkuverbrauch kann keine Aussage gemacht werden.
NFA#05	Ja	Die Bedienung wurde so einfach wie möglich gehalten.
NFA#06	Ja	Die Anwendung benötigt keine Internetverbindung.
NFA#07	Ja	Auf Nutzereingaben wird möglichst schnell reagiert.

Tabelle 7.2: Abgleich nichtfunktionale Anforderungen

8

Zusammenfassung und Ausblick

In diesem Kapitel wird eine kurze Zusammenfassung über die Arbeit sowie ein Ausblick über weitere Funktionen und Verbesserungen gegeben, welche in die mobile Anwendung eingebaut werden können.

8.1 Zusammenfassung

In dieser Arbeit wurde eine Anwendung konzipiert und entwickelt, die Diabetiker mit Typ 1 Diabetes unterstützen soll. Viele Apps im Bereich Diabetes bieten den Patienten ein Tagebuch, in welchem sie Blutzuckerwerte dokumentieren können, um Ursachen für Schwankungen und Abweichungen zu finden. Bei Diabetikern, welche sich Insulin spritzen, können die Ursachen allerdings in der Bestimmung der Kohlenhydrate und der Berechnung der Insulindosis liegen. Nur wenige Anwendungen bieten Unterstützung in diesem Bereich und noch weniger Anwendungen bieten Hilfe im Thema Kohlenhydrate von Mahlzeiten schätzen. In dieser Arbeit wurde eine App entwickelt, mit welcher Diabetiker Mahlzeiten auswählen und durch eine visuelle Darstellung die Menge dieser festlegen können. Das Schätzen der Kohlenhydrate, sowie die Berechnung der Insulindosis, welche abhängig von personalisierten Daten ist, übernimmt die Applikation. Der Benutzer stellt zu Beginn einmalig seine persönlichen und mit dem Arzt abgesprochenen Werte ein, welche nachfolgend jeder Zeit geändert werden können. Um eine Insulindosis zu berechnen, muss der aktuelle Blutzuckerwert und eine Mahlzeit mit angepasster Mengenangabe eingetragen werden. Diabetiker, welche den FreeStyle Libre Sensor für die Messung des Blutzuckerwerts benutzen, können diesen ebenfalls in der Anwendung verwenden. Die Arbeit beschäftigt sich mit den genannten Konzepten.

8 Zusammenfassung und Ausblick

Im Kapitel 2 werden alle relevanten Informationen zum Thema Diabetes erläutert, die zum Verständnis der Anwendung dienen.

8.2 Ausblick

Im Rahmen der Bachelorarbeit beschränkt sich die Wahl des Essens und der Getränke nur auf wenige Elemente. Diese sollten zunächst ergänzt werden, um dem Benutzer eine realistische Auswahl zu bieten. Dabei sollten auch die Kategorien erweitert werden, damit die Suche möglichst intuitiv verläuft. Des Weiteren sollen bei einer größeren Auswahl möglichst viele Kombinationen an Lebensmitteln möglich sein, sodass sich der Benutzer möglichst alle Mahlzeiten zusammenstellen kann.

In Bezug auf die Darstellung soll die Farbe der jeweiligen Getränke angepasst werden. Außerdem soll der Benutzer im Bereich Auswahl zwischen verschiedenen Tellergrößen und -formen auswählen können. Zusätzlich zur Mengenanzeige des Essens, sollen dem Benutzer Daten bezüglich des Gewichts angezeigt und somit falsche Schätzungen vermieden werden.

Um die Insulindosis möglichst genau zu berechnen, sollen die bisher berechneten Ergebnisse mit allen wichtigen Daten gespeichert werden. Auf der einen Seite hat der Benutzer dadurch den Vorteil, die Zusammenstellung bereits berechneter Insulindosen nachzuschauen. Auf der anderen Seite kann bereits gespritztes Insulin, dessen Wirkung noch aktiv ist, in den Berechnungen berücksichtigt werden.

Hinzufügend soll dem Nutzer eine ausführliche und verständliche Information bezüglich der Funktionalitäten und Tipps zur Behandlung von Diabetes zur Verfügung stehen. Auch eine Hilfe soll die Benutzung der App möglichst unterstützen. Die Applikation soll außerdem für alle Smartphonebesitzer, unabhängig des Betriebssystems, verfügbar sein.

Die Anwendung bietet eine gute Grundlage, Verbesserungen vorzunehmen und weitere Funktionalitäten verfügbar zu machen.

Literaturverzeichnis

- [1] Poushter, J.: Smartphone Ownership and Internet Usage Continues to Climb in Emerging Economies. Website (2016) Online erhältlich unter <http://www.pewglobal.org/2016/02/22/smartphone-ownership-and-internet-usage-continues-to-climb-in-emerging-economies/> aufgerufen am 25. März 2018.
- [2] Schickler, M., Reichert, M., Pryss, R., Schobel, J., Schlee, W., Langguth, B.: Entwicklung mobiler Apps: Konzepte, Anwendungsbausteine und Werkzeuge im Business und E-Health. Springer-Verlag (2015)
- [3] Holtz, B., Lauckner, C.: Diabetes management via mobile phones: A systematic review. TELEMEDICINE and e-HEALTH (2012)
- [4] Statista GmbH: Prävalenz und Fallzahlen von Diabetes und Glukosetoleranzstörungen (IGT) weltweit in den Jahren 2017 und 2045. Website (2017) Online erhältlich unter <https://de.statista.com/statistik/daten/studie/256852/umfrage/praevalenz-und-fallzahlen-von-diabetes-und-glukosetoleranzstoerungen-weltweit/> aufgerufen am 25. März 2018.
- [5] El-Gayar, O., Timsina, P., Nawar, N., Eid, W.: Mobile applications for diabetes self-management: Status and potential. Journal of Diabetes Science and Technology (2013)
- [6] Jacobs, E., Tamayo, T., Rathmann, W.: Epidemiologie des diabetes in deutschland. Deutscher Gesundheitsbericht Diabetes 2017 (2017)
- [7] beurer medical: Ratgeber Diabetes. (german engineering)
- [8] Robert Koch Institut: Diabetes mellitus in Deutschland. Website (2018) Online erhältlich unter https://www.rki.de/DE/Content/Gesundheitsmonitoring/Gesundheitsberichterstattung/GBEDownloadsK/2011_3_diabetes.pdf?__blob=publicationFile aufgerufen am 28. März 2018.

Literaturverzeichnis

- [9] World Health Organization: Part 1, diagnosis and classification of diabetes mellitus. In: Definition, Diagnosis and Classification of Diabetes Mellitus and its Complications. (1999)
- [10] Heller, T., Blum, M., Spraul, M., Wolf, G., Müller, U. A.: Folgeerkrankungen des Diabetes mellitus: Prävalenzen in der Bundesrepublik Deutschland. Georg Thieme Verlag (2014)
- [11] Abott Diabetes Care: FreeStyle Libre. Website (2014) Online erhältlich unter www.freestylelibre.de/ aufgerufen am 07. März 2018.
- [12] Abott Diabetes Care: Sensor. Website (2014) Online erhältlich unter <http://www.freestylelibre.de/freestyle-libre-sensor-kit-de-fr-it-at.html> aufgerufen am 07. März 2018.
- [13] Shah, V. N., Garg, S. K.: Managing diabetes in the digital age. Clinical Diabetes and Endocrinology (2015)
- [14] Newyu, Inc.: LibreLinkUp. Website (2018) Online erhältlich unter <https://play.google.com/store/apps/details?id=org.nativescript.LibreLinkUp&hl=de> aufgerufen am 10. März 2018.
- [15] Abbott Diabetes Care Inc.: FreeStyle LibreLink. Website (2018) Online erhältlich unter <https://play.google.com/store/apps/details?id=com.freestylelibre.app.de> aufgerufen am 10. März 2018.
- [16] mySugr GmbH: mySugr. Website (2018) Online erhältlich unter <https://mysugr.com/de/> aufgerufen am 10. März 2018.
- [17] mySugr GmbH: mySugr: dein intelligentes Diabetes Tagebuch. Website (2018) Online erhältlich unter <https://play.google.com/store/apps/details?id=com.mysugr.android.companion> aufgerufen am 10. März 2018.
- [18] Allgaeu Coder: Bolusrechner. Website (2018) Online erhältlich unter <https://play.google.com/store/apps/details?id=com.codeallgaeu.elmo.bolusrechner> aufgerufen am 10. März 2018.

- [19] SCHApps: Advanced Insulin Advisor Free. Website (2018) Online erhältlich unter <https://play.google.com/store/apps/details?id=insulin.calculator> aufgerufen am 10. März 2018.
- [20] NinjaMock.com: NinjaMock. Website (2018) Online erhältlich unter <https://ninjamock.com/> aufgerufen am 08. März 2018.
- [21] Bianchi, F.: Colors. Website (2016) Online erhältlich unter <https://colors.co/2274a5-6a041d-ee4266-ffd23f-77ba99> aufgerufen am 08. März 2018.
- [22] Google: Android. Website (2018) Online erhältlich unter <https://www.android.com/> aufgerufen am 14. März 2018.
- [23] Google: Android - 8.0 Oreo. Website (2018) Online erhältlich unter <https://www.android.com/versions/oreo-8-0/> aufgerufen am 14. März 2018.
- [24] Google: Android schafft Möglichkeiten. Website (2018) Online erhältlich unter <https://www.android.com/everyone/enabling-opportunity/> aufgerufen am 14. März 2018.
- [25] Google: Activity Lifecycle. Website (2018) Online erhältlich unter <https://developer.android.com/guide/components/activities/activity-lifecycle.html> aufgerufen am 15. März 2018.
- [26] Google: Android Studio. Website (2018) Online erhältlich unter <https://developer.android.com/studio/index.html> aufgerufen am 16. März 2018.
- [27] Google: Android Emulator. Website (2018) Online erhältlich unter <https://developer.android.com/studio/run/emulator.html> aufgerufen am 16. März 2018.
- [28] Google: Android Layout Editor. Website (2018) Online erhältlich unter <https://developer.android.com/studio/features.html> aufgerufen am 16. März 2018.

Literaturverzeichnis

- [29] Netzwelt: NFC erklärt: Was ist das? Und wofür lässt es sich alles nutzen? Website (2018) Online erhältlich unter <https://www.netzwelt.de/nfc/index.html> aufgerufen am 23. März 2018.
- [30] Google: Intent. Website (2018) Online erhältlich unter <https://developer.android.com/reference/android/content/Intent.html> aufgerufen am 23. März 2018.
- [31] Google: Ndef. Website (2018) Online erhältlich unter <https://developer.android.com/reference/android/nfc/tech/Ndef.html> aufgerufen am 23. März 2018.
- [32] Google: NFC Basics. Website (2018) Online erhältlich unter <https://developer.android.com/guide/topics/connectivity/nfc/nfc.html> aufgerufen am 23. März 2018.
- [33] Google: Advanced NFC. Website (2018) Online erhältlich unter <https://developer.android.com/guide/topics/connectivity/nfc/advanced-nfc.html> aufgerufen am 23. März 2018.
- [34] NFC Research Lab: NFC TagInfo. Website (2018) Online erhältlich unter <https://play.google.com/store/apps/details?id=at.mroland.android.apps.nfctaginfo&hl=de> aufgerufen am 23. März 2018.
- [35] Klug, M.: FreeStyle Libre ein offenes Buch. Website (2018) Online erhältlich unter <http://www.marcelklug.de/2015/04/15/freestyle-libre-ein-offenes-buch/> aufgerufen am 23. März 2018.
- [36] Vicktor: FreeStyleLibre-NFC-Reader. Website (2018) Online erhältlich unter <https://github.com/vicktor/FreeStyleLibre-NFC-Reader/wiki/Progress-&-ToDo> aufgerufen am 23. März 2018.
- [37] Klug, M.: FreeStyle Libre die Android-App. Website (2018) Online erhältlich unter <http://www.marcelklug.de/2015/05/28/freestyle-libre-die-android-app/> aufgerufen am 23. März 2018.

- [38] UPetersen: LibreMonitor. Website (2018) Online erhältlich unter <https://github.com/UPetersen/LibreMonitor> aufgerufen am 23. März 2018.
- [39] Noun Project, Jager, T.: Teller Icon. Website (2018) Online erhältlich unter <https://thenounproject.com/term/plate/14770/> aufgerufen am 22. März 2018.
- [40] Noun Project, Afacan, E.: Teller Icon. Website (2018) Online erhältlich unter <https://thenounproject.com/term/plate/342270/> aufgerufen am 22. März 2018.
- [41] Icons8 LLC: Icon8. Website (2018) Online erhältlich unter <https://icons8.de/> aufgerufen am 22. März 2018.
- [42] Google: BroadcastReceiver. Website (2018) Online erhältlich unter <https://developer.android.com/reference/android/content/BroadcastReceiver.html> aufgerufen am 22. März 2018.
- [43] Google: Shared Preferences. Website (2018) Online erhältlich unter <https://developer.android.com/training/data-storage/shared-preferences.html> abgerufen am 16. März 2018.
- [44] Google: JsonReader. Website (2018) Online erhältlich unter <https://developer.android.com/reference/android/util/JsonReader.html> aufgerufen am 16. März 2018.
- [45] Anothem: RangeSeekBar. Website (2017) Online erhältlich unter <https://github.com/anothem/android-range-seek-bar> aufgerufen am 16. März 2018.
- [46] h6ah4i: VerticalSeekBar. Website (2017) Online erhältlich unter <https://github.com/h6ah4i/android-verticalseekbar> aufgerufen am 16. März 2018.
- [47] Google: CardView. Website (2018) Online erhältlich unter <https://material.io/guidelines/components/cards.html> aufgerufen am 17. März 2018.

Literaturverzeichnis

- [48] Google: RecyclerView. Website (2018) Online erhältlich unter https://github.com/codepath/android_guides/wiki/Using-the-RecyclerView aufgerufen am 18. März 2018.
- [49] Google: Material Design Guidelines. Website (2018) Online erhältlich unter <https://material.io/guidelines/> aufgerufen am 18. März 2018.

A

Anhang

A.1 Mockups



Abbildung A.1: Hauptseite



Abbildung A.2: Hauptseite mit Eingabe



Abbildung A.3: Hauptseite mit Eingabe

A Anhang



Abbildung A.4: Hauptseite mit allen Eingaben



Abbildung A.5: Hauptseite mit Eingabe Nudeln



Abbildung A.6: Hauptseite mit Nudeln und Getränk



Abbildung A.7: Hauptseite mit Eingabe Pizza



Abbildung A.8: Hauptseite mit Eingabe Pizza



Abbildung A.9: Hauptseite mit Nudeln und Ausgabe



Abbildung A.10: Abfrage Essen löschen



Abbildung A.11: Einstellungen



Abbildung A.12: Erinnerung Aus



Abbildung A.13: Erinnerung An



Abbildung A.14: Benachrichtigung



Abbildung A.15: Insulinfaktoren

A Anhang



Abbildung A.16: Korrekturwert



Abbildung A.17: Zielbereich



Abbildung A.18: Auswahl Essen



Abbildung A.19: Auswahl Beilagen



Abbildung A.20: Auswahl Soßen



Abbildung A.21: Essen ausgewählt

Abbildungsverzeichnis

2.1	Anzahl Diabetiker in Deutschland [8]	6
2.2	FreeStyle Libre Sensor mit Lesegerät [12]	9
2.3	Glucosewert mit Trendpfeil [15]	11
2.4	Messwerte der letzten 24h [15]	11
2.5	Fertiger Eintrag [17]	12
2.6	Blutzucker Report [17]	12
2.7	Neuer Eintrag [17]	12
2.8	Dateneingabe [18]	13
2.9	Einstellungen [18]	13
2.10	Dateneingabe [19]	14
2.11	Auswahl Essen [19]	14
2.12	Mengenauswahl [19]	14
4.1	Hauptseite	22
4.2	Hauptseite mit bearbeiteten Kacheln	22
4.3	Hauptseite mit bearbeiteten Kacheln	22
4.4	Auswahl Essen	23
4.5	Einstellungen	23
4.6	Farbschema [21]	26
5.1	Komponentendiagramm	27
5.2	Dialogstruktur	28
5.3	Anwendungsfalldiagramm	29
6.1	Activity Lifecycle [25]	32
6.2	Tag Dispatch System [32]	35
6.3	NFC TagInfo Tag information	36
6.4	NFC TagInfo Memory information	36
6.5	Übersicht Sensordaten [35]	37

Abbildungsverzeichnis

6.6	6-Byte Datenpaket	38
6.7	Teller [39]	40
6.8	Anzahl Nudeln einer ImageView	41
6.9	Maximale Anzahl Nudeln	41
6.10	Nudeln mit Soße	41
6.11	Teller [40]	42
6.12	Pizzastück [41]	42
6.13	Menge Pizza	42
6.14	RangeSeekbar	43
6.15	VerticalSeekbar	44
6.16	CardView	45
6.17	RecyclerView Workflow [48]	46
6.18	Flat Button und Raised Button	47
6.19	Chips	47
6.20	Alert Dialog	48
6.21	Menü	49
6.22	Radio Buttons	49
6.23	Slider	50
6.24	Snackbar	51
6.25	Tabs	52
6.26	Empty State	52
6.27	Fehler in der Benutzereingabe	53
6.28	Benachrichtigung	54
6.29	Hauptseite	55
6.30	Eingabe Pizza	56
6.31	Eingabe Blutzuckerwert Information	56
6.32	Eingabe Nudeln	57
6.33	Essensauswahl	57

6.34 Übersicht Einstellungen	58
6.35 Einstellungen Insulinfaktoren	59
6.36 Einstellungen Zielbereich	59
6.37 Einstellungen Korrekturwert	60
6.38 Einstellungen Erinnerung aktiviert	60
A.1 Hauptseite	73
A.2 Hauptseite mit Eingabe	73
A.3 Hauptseite mit Eingabe	73
A.4 Hauptseite mit allen Eingaben	74
A.5 Hauptseite mit Eingabe Nudeln	74
A.6 Hauptseite mit Nudeln und Getränk	74
A.7 Hauptseite mit Eingabe Pizza	74
A.8 Hauptseite mit Eingabe Pizza	74
A.9 Hauptseite mit Nudeln und Ausgabe	74
A.10 Abfrage Essen löschen	75
A.11 Einstellungen	75
A.12 Erinnerung Aus	75
A.13 Erinnerung An	75
A.14 Benachrichtigung	75
A.15 Insulinfaktoren	75
A.16 Korrekturwert	76
A.17 Zielbereich	76
A.18 Auswahl Essen	76
A.19 Auswahl Beilagen	76
A.20 Auswahl Soßen	76
A.21 Essen ausgewählt	76

Tabellenverzeichnis

3.1 Funktionale Anforderungen	18
3.2 Nichtfunktionale Anforderungen	19
7.1 Abgleich funktionale Anforderungen	62
7.2 Abgleich nichtfunktionale Anforderungen	63

Name: Leoni Holl

Matrikelnummer: 866729

Erklärung

Ich erkläre, dass ich die Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

Ulm, den

Leoni Holl