

BOB

Ein Projekt von

Dennis Timmermann, 424575

Fabian Schucht, 629071

Mathis Krüper, 553380

Veranstaltungen

Interaction Design 2

Webtechnologien 2

Dozenten

Prof. Dipl.-Designer Johannes Nehls

Dipl.-Inf. (FH) Björn Plutka

Inhalt

<i>Allgemeines</i>	4
<i>Ideenfindung</i>	4
<i>Konzept</i>	5
<i>Zielgruppe</i>	6
<i>Einsatzfelder</i>	7
Gestaltung	8
<i>Bob</i>	8
<i>App</i>	11
<i>Gürtelschnalle</i>	18
Programmierung	34
<i>Framework 7</i>	35
<i>Aufbau</i>	36
<i>Probleme</i>	38
<i>Gürtel</i>	39

Ideenfindung

Für die Veranstaltung *Interaction Design 2* im dritten Semesters, war es die Aufgabe, ein Konzept zu entwickeln und dieses im Anschluss umzusetzen. Die Besonderheit hierbei war allerdings, dass der Fokus auf *Wearable Devices*, also tragbaren Geräten lag. Es war bei dem Konzept zu beachten, dass das Produkt mit einer Applikation auf einem Smartphone kommunizieren soll und auch über selbige gesteuert wird. Die Entwicklung dieser App fällt in den Aufgabenbereich der Veranstaltung *Webtechnologien 2* und wurde somit auch als Hausarbeit zum Abschluss des Semesters festgelegt.

Zum Start des Arbeitsprozess haben wir uns in einer Gruppe mit drei Personen zusammengefunden und unsere Ideen gesammelt, die wir bisher in der Veranstaltung *Interaction Design 2* entwickelt haben. Wir mussten aber leider ziemlich schnell feststellen, dass unsere bisherigen Ideen nicht wirklich innovativ waren, oder wie z.B. ein Feuerzeug das die Rauchaktivitäten des Nutzers misst, schon als ähnliches Produkt auf dem Markt zu finden sind.

Wir haben daraufhin in der Gruppe noch einmal begonnen neue Ideen zu sammeln und sind uns dann nach kurzer Zeit einig geworden, dass unser *Wearable Device* ein intilligenter Gürtel werden soll.

Konzept

Nachdem wir uns auf das Grundlegende Objekt für unser Produkt festgelegt hatten, begannen wir damit, uns über die mögliche Funktionalität gedanken zu machen. Hier stand für uns im Vordergrund, dass ein nützlicher Aspekt für den Alltag in dem Device integriert sein soll und somit ein täglicher Gebrauch gegeben ist.

Als erstes kam uns natürlich in den Sinn die Funktionalität an den Bauchumfang des Nutzers zu koppeln. So hätte sich der Gürtel automatisch enger gezogen, um den Nutzer darauf hinzuweisen, dass sein Bauchumfang in einer gewissen Zeit zugenommen hat.

Nach Rücksprache mit unserem Professor kamen wir dann aber zu der Einsicht, dass dies zu eindeutig und nicht innovativ genug für solch ein Produkt wäre.

Nach einem gemeinsamen Brainstorming kam uns die Redewendung „*den Gürtel enger schnallen*“ in den Sinn, die sich ja bekanntlich auf den finanziellen Status bezieht. Wir waren uns schnell einig, dass dies ein sinnvoller Faktor wäre auf den wir uns beziehen können und begannen damit unsere Idee zu konkretisieren.

Nach Abschluss dieser Phase sind wir zu dem folgenden Konzept gekommen: Durch die App auf dem Smartphone wird erfasst wieviel Geld der Nutzer im Monat zur Verfügung hat und es wird errechnet welche Summe pro Tag ausgegeben werden kann. Wird dieser Tagessatz vom Nutzer erschöpft, zieht sich der Gürtel automatisch enger um dem Nutzer dies zu signalisieren.

Zielgruppe

Die Zielgruppe für unser Produkt ist weit gefächert. So möchten wir die Altersgruppen zwischen 18 und 60 ansprechen. Dies resultiert daraus, dass als Bedingung für die Nutzung ein Bankkonto mit Online Zugang vorhanden sein muss, sowie ein Smartphone oder ein Tablet, was bei Kindern und Jugendlichen nicht zwangsläufig gegeben ist. Die Obergrenze beim Alter resultiert daraus, dass ein

gewisses Maß an Offenheit gegenüber neuen Technologien bestehen sollte, was bei den älteren Generationen zum Teil Schwierigkeiten erzeugen könnte. Im Grunde ist unser Produkt daher für alle geeignet, die über einen Online-Banking Account, sowie ein onlinefähiges Mobilgerät verfügen und Spaß an neuen Technologien haben.

Einsatzfelder

Wie schon erwähnt, war es unser Ziel ein nützliches Gadget für den Alltag zu entwickeln. Da ein Gürtel von vielen Menschen aus nützlichen, sowie aus modischen Gründen getragen wird, wäre es mit BOB auf jeden Fall möglich eine große Käuferschicht anzusprechen. Bei einer späteren Vermarktung könnte der Nutzer den Gürtel selbst nach dem persönlichen

Geschmack anpassen. So könnten ausgefallene Designs entstehen. Für Geschäftsleute könnten schlichtere Designs angeboten werden. Der Gürtel bietet zudem einen enormen Mehrwert, da er dem Nutzer auf eine intuitive Art und Weise dazu bringt, bewusster mit seinem Geld umzugehen. So ist BOB ein intelligenter Helfer für jedermann.

Gestaltung

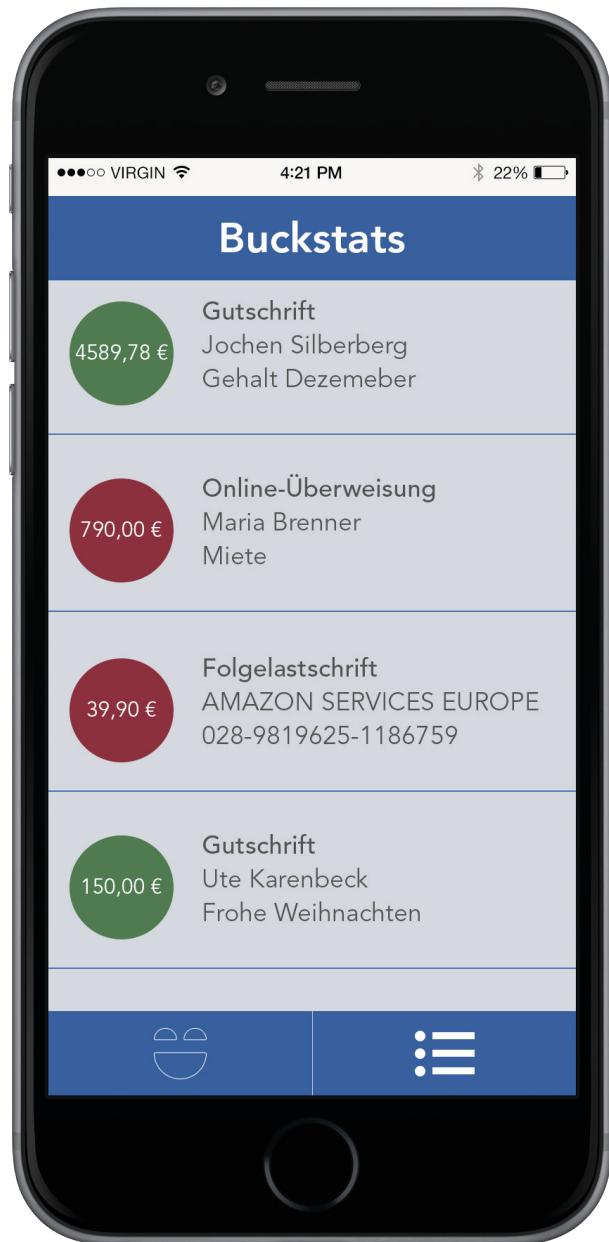
Bob

Bei der Gestaltung haben wir uns grundsätzlich vorgenommen einen recht minimalistischen und somit funktionalen Stil einzuhalten. Somit soll gewährleistet werden, dass unsere breit gefächerte Zielgruppe mit dem Endprodukt umgehen kann und es nicht zu Verwirrungen oder Unsicherheiten kommt. Im Folgendem wollen wir einen Überblick über unsere Gestaltungslösungen geben.

Wir haben für die App eine Figur mit dem Namen BOB entwickelt, welche wir für die Vermittlung der Daten verwenden. BOB kann als persönlicher Begleiter gesehen werden, welcher dem Nutzer der App einen Überblick über die eigenen Finanzen gibt, indem er, als Füllstand dargestellt, das für den heutigen Tag zu Verfügung stehende Geld anzeigt. Die Farbkodierung von BOB lässt zudem unmittelbar erkennen, ob der Nutzer noch Geld ausgeben kann oder ob er dies lieber sein lassen sollte.

Zusätzlich zur Darstellung des zur Verfügung stehenden Geldes übernimmt BOB auch die Aufgabe dem Produkt eine Identität zu geben, indem er ihm einen Wiedererkennungswert verleiht. So wird trotz des schlichten und technisch angehauchten Erscheinungsbildes ein persönlicher Charme versprüht. Ein sehr persönliches Thema wird so mit einer gewissen Leichtigkeit vermittelt.





App

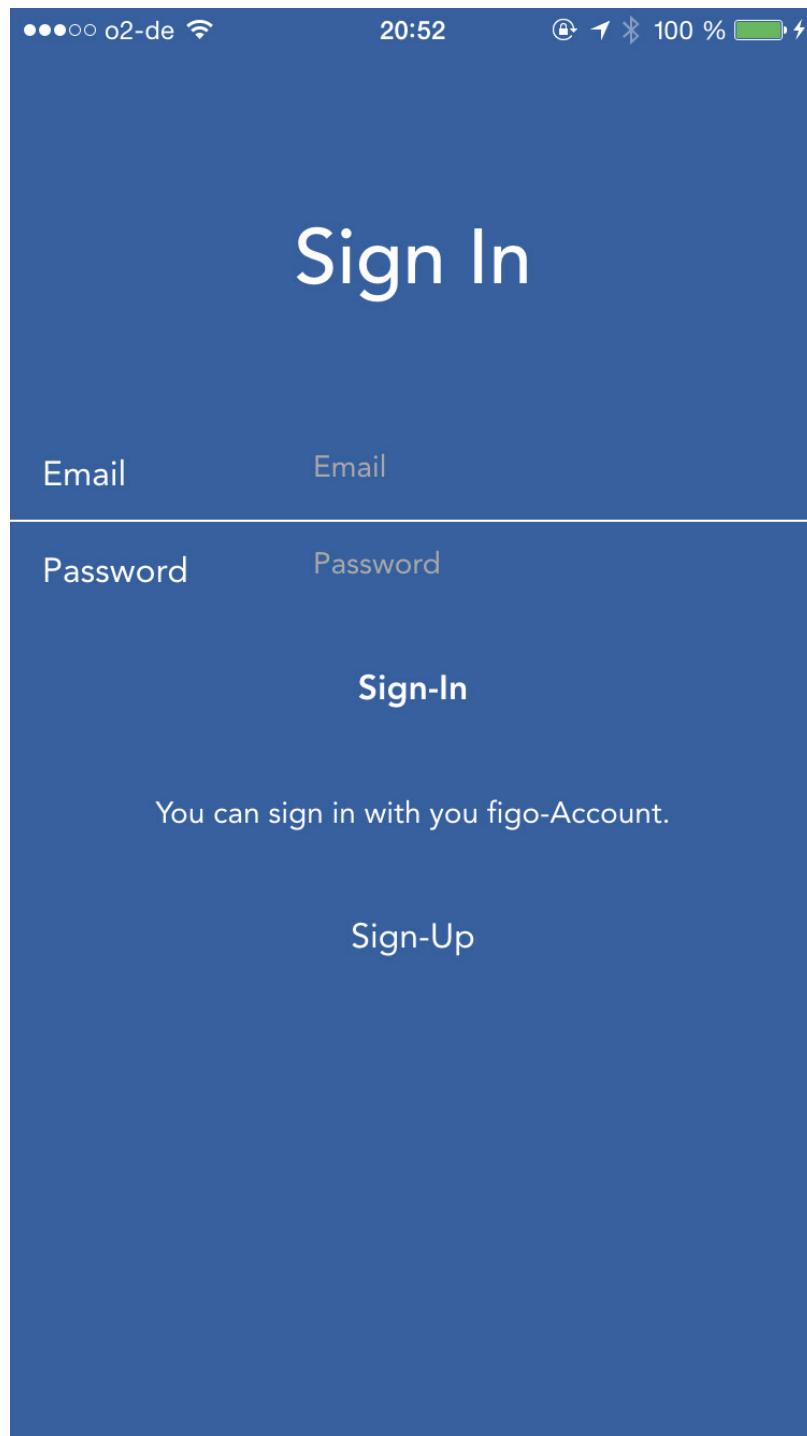
Unsere App ist als App mit Tabbasierter Navigation aufgebaut. Das heißt, dass der Nutzer einzelne Tabs über eine Navigationsleiste am unteren Rand aufrufen kann. Unsere App besteht aus drei Tabs. Der erste Tab zeigt dem Nutzer, wieviel Geld er noch ausgeben kann. Der zweite Tab gibt dem Nutzer eine Übersicht über seine getätigten Transaktionen. Diese sind nach dem Datum sortiert. Der Betrag wird als rot oder grün gefärbte Blase angezeigt. Der dritte Tab bietet dem Benutzer die Möglichkeit Einstellung zu tätigen und zu ändern. Er kann zum Beispiel seine monatlichen Ausgaben angeben.

Wenn er dies tut werden diese nicht mehr aus den Transaktionen abgeleitet. Auch hat er die Möglichkeit die Einstellungen zurückzusetzen und eine Verbindung zum Gürtel wiederherzustellen.

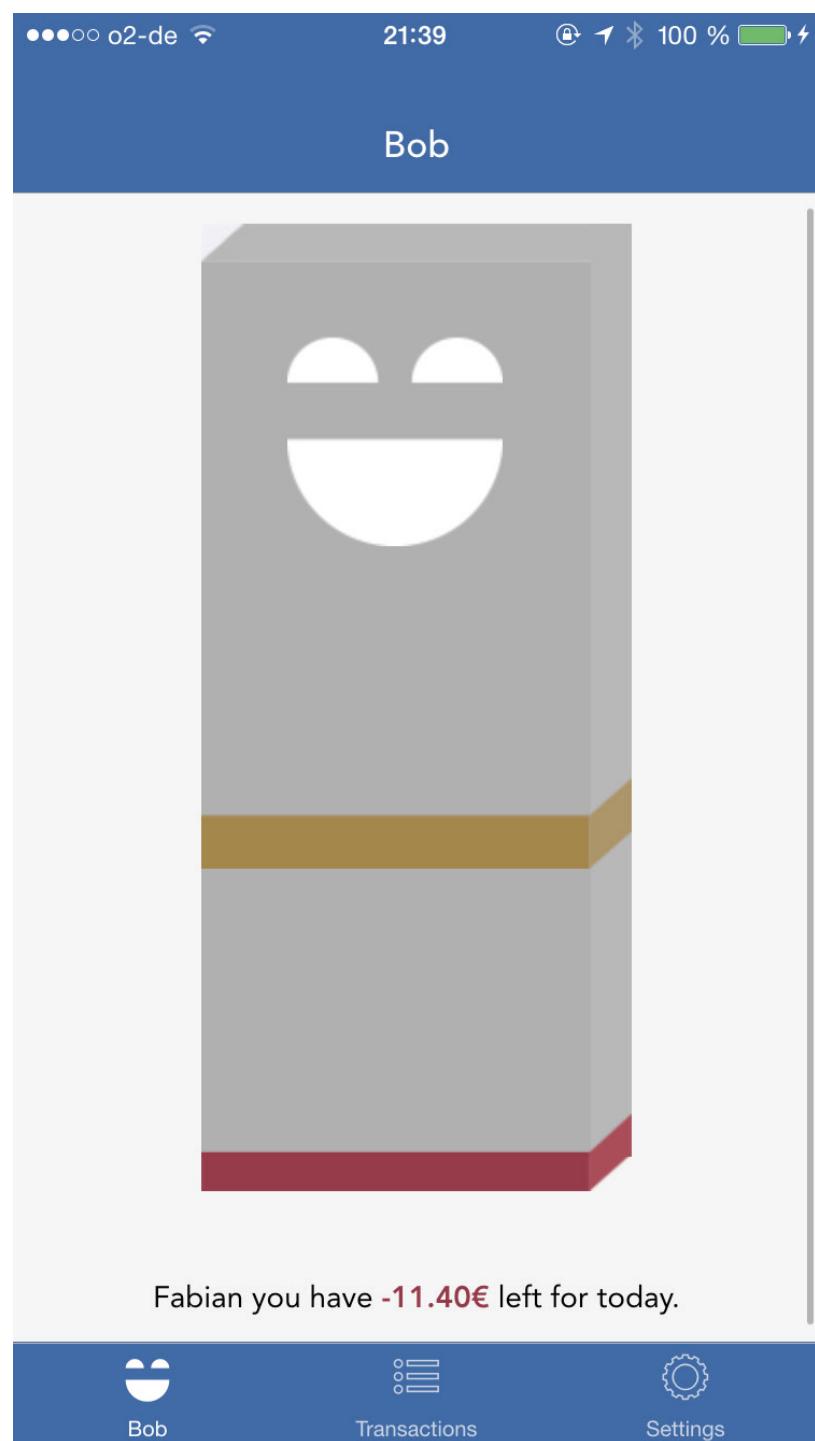
Wir haben uns entschieden unsere App in einem möglichst nativen Stil zu halten. Viele Cordova-basierte Apps haben leider den Nachteil, dass diese von der Gestaltung her ein nicht natives Gefühl vermitteln können. Unser Ziel war es daher von Anfang an, dieses Gefühl zu vermitteln. Dies hat uns natürlich auch hingehend unseres minimalistischen Stils stark beeinflusst. Insgesamt muss gesagt werden, dass

wir uns mehr auf iOS als auf Android konzentriert haben.

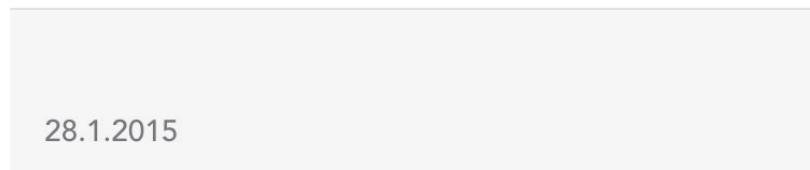
Die Farbauswahl der App basiert auf den Farbtönen des Zehn- und Hundert-Euroscheins. Der Blauton, welchen wir als Hauptfarbe verwendet haben, ist vom Zwanzig-Euroschein abgeleitet. Weiterhin haben wir Abstufungen dieser Farben verwendet. Das Rot des Zehn-Euroscheins benutzen wir, um negative Ausgaben darzustellen und um dem Nutzer zu signalisieren, dass er kein Geld mehr ausgeben sollte. Das Grün des Hundert-Euroscheins benutzen wir hingegen für positive Ausgaben.



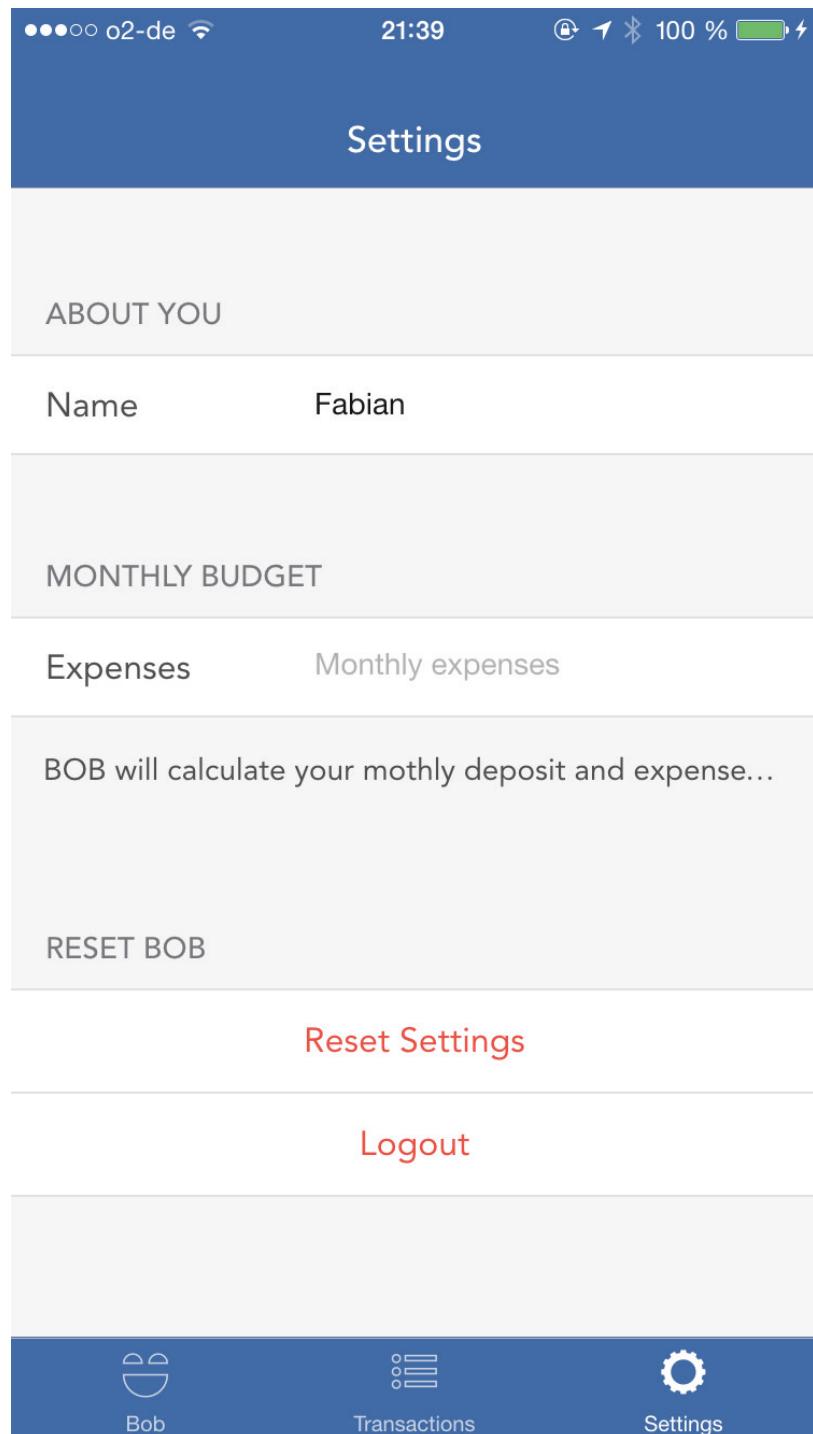
Loginscreen der App.



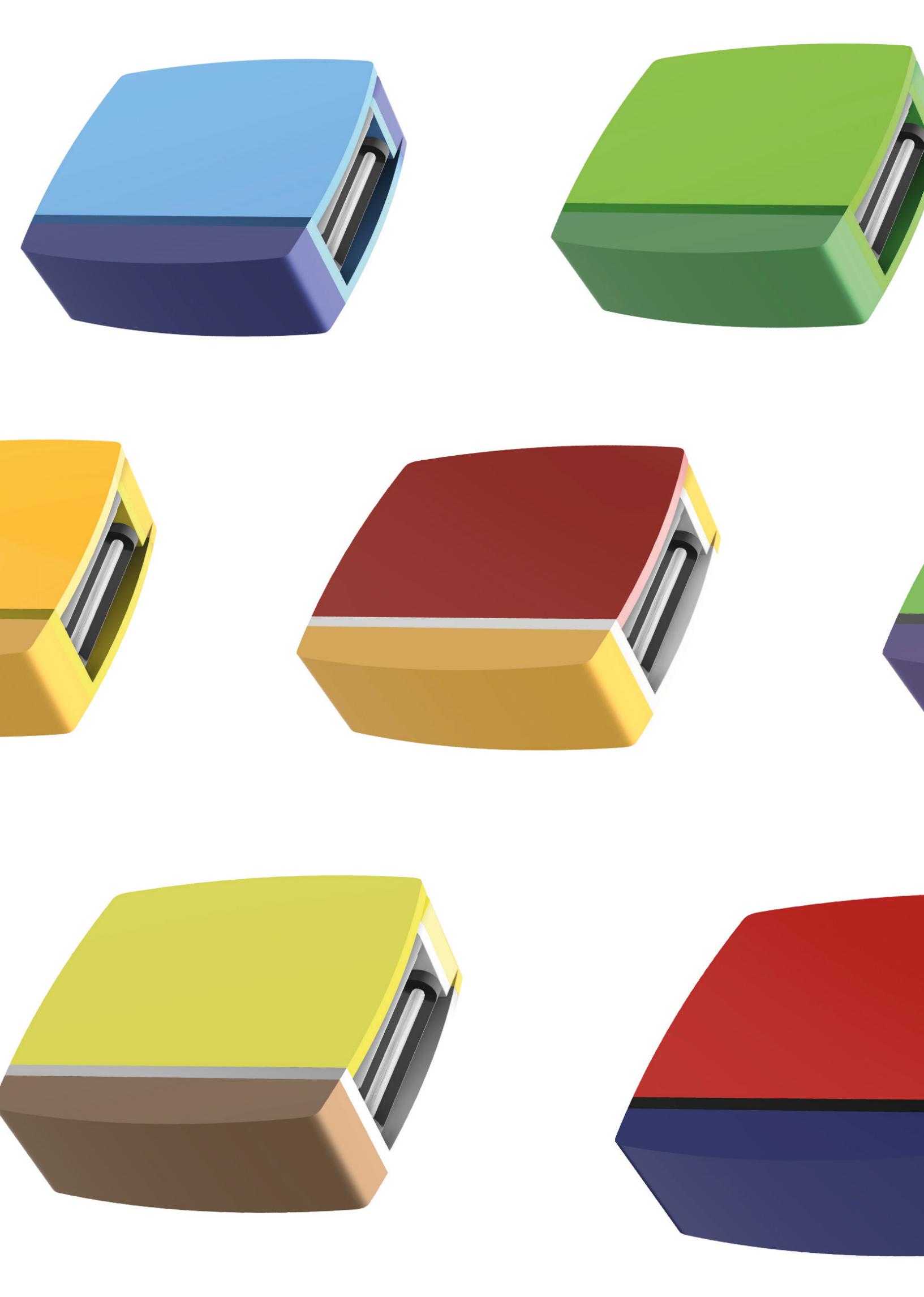
Dieser Screen zeigt dem Nutzer, wie viel er noch ausgeben kann.

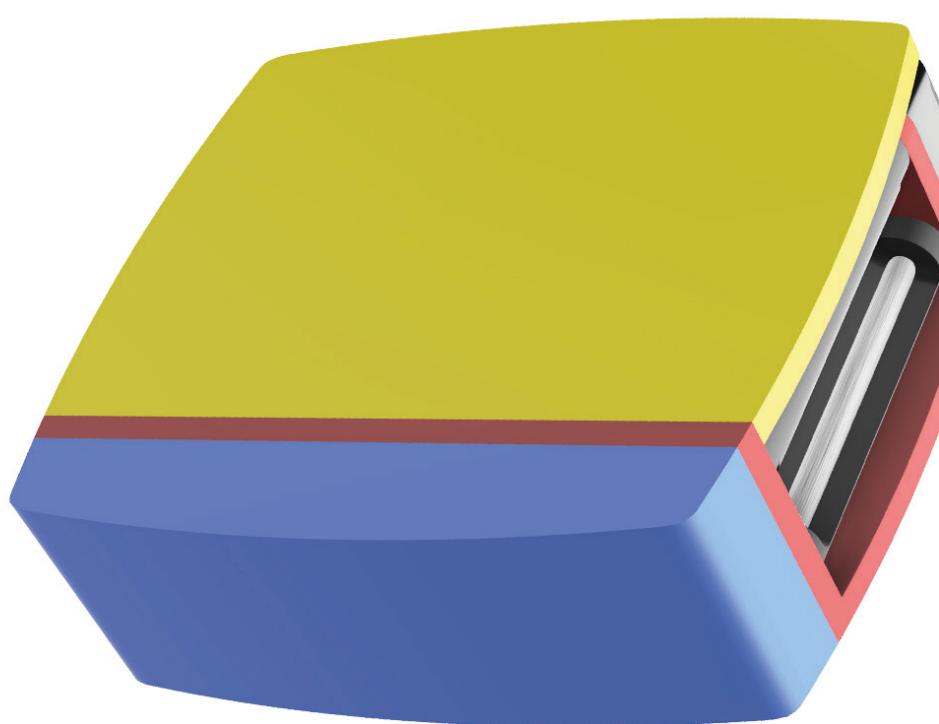
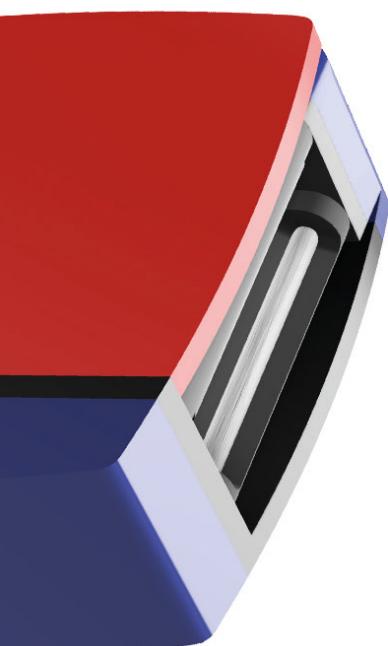
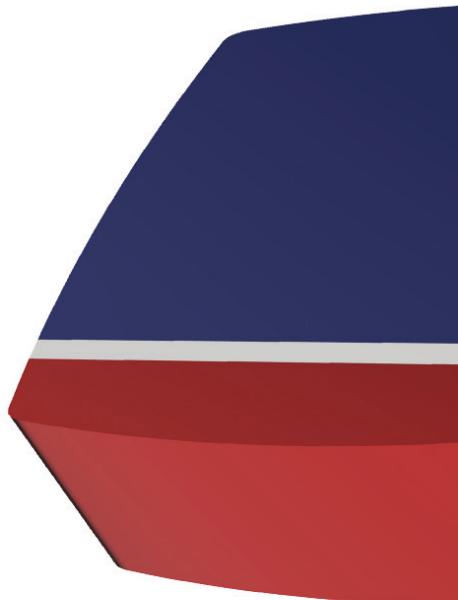


Die Transaktionen werden nach dem Datum geordnet angezeigt.



Der Nutzer kann verschiedene Aspekte der App personalisieren.





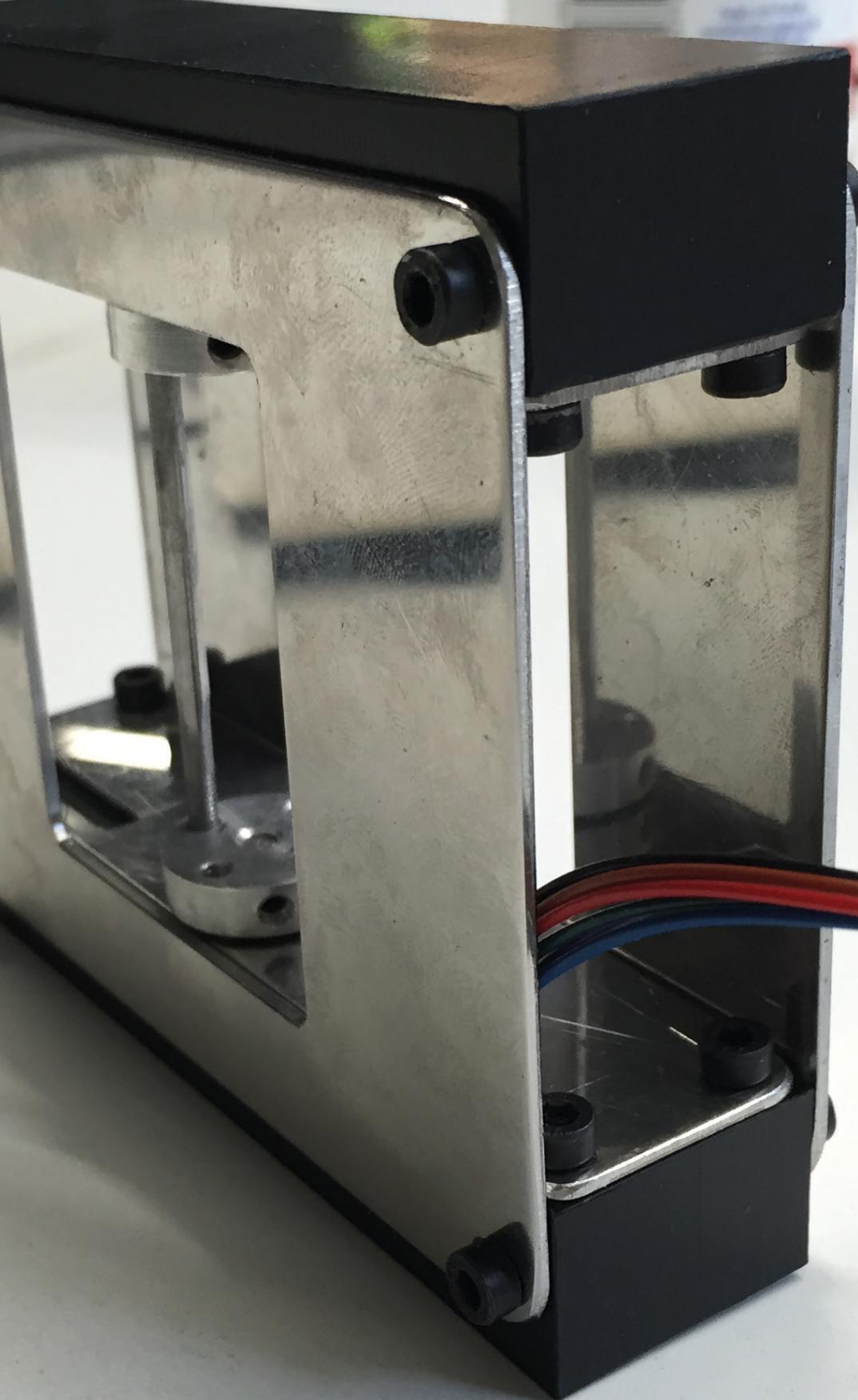
Gürtelschnalle

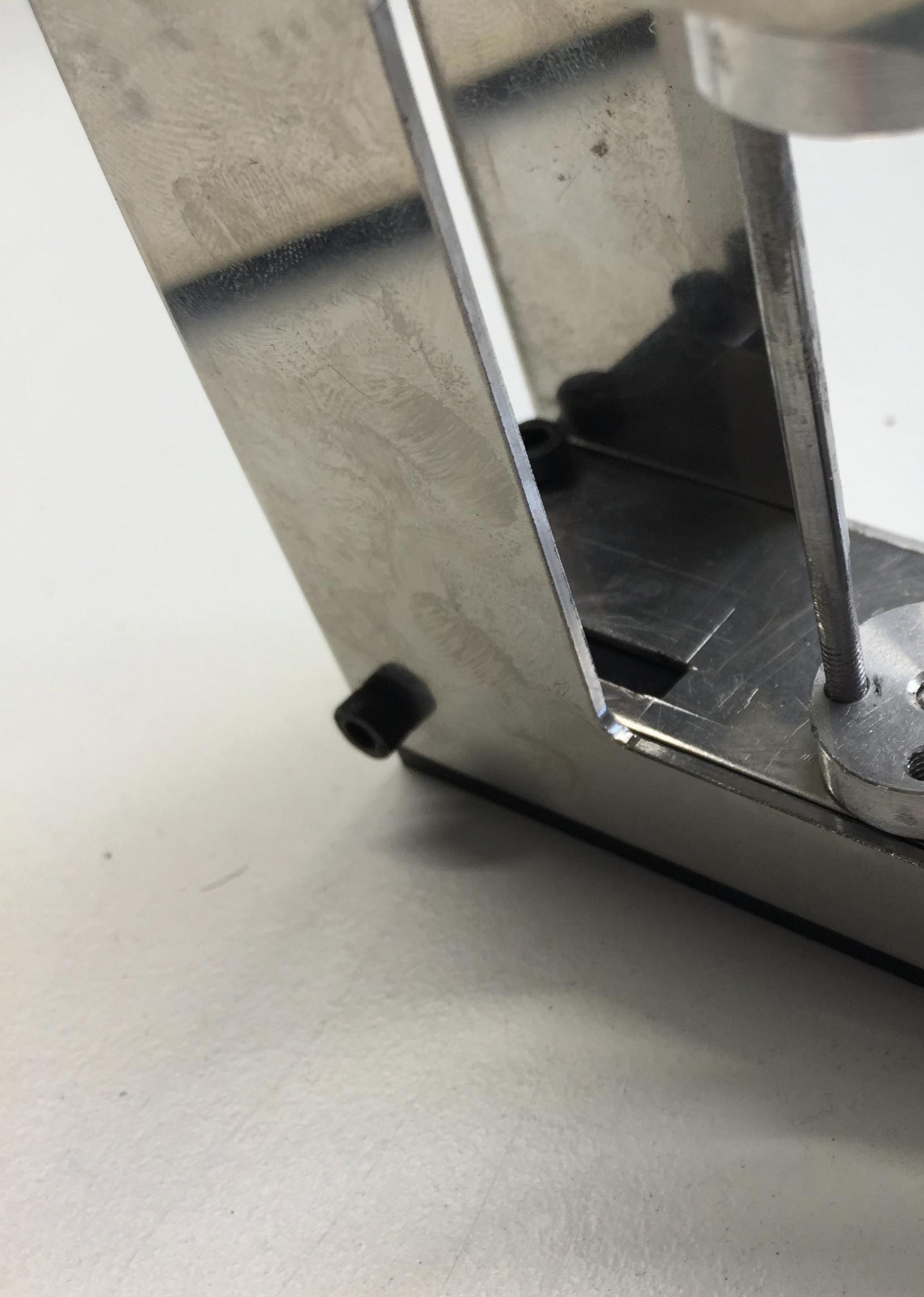
Die Gestaltung der ersten Gürtelschnalle hat sich hauptsächlich aus den funktionalen Anforderungen sowie der eingesetzten Technik ergeben. So entwickelten wir mit der Software Inventor einen ersten Prototypen, wozu die eingesetzte Technik wie Motoren, Gestänge, Batterien und Arduino Controller entsprechende Aussparungen in dem Modell bekamen. Dieses erste Layout ließen wir in den Werkstätten der Hochschule Osnabrück fertigen. Für den Prototypen wurden die Materialien *POM* und *Stahlblech* verwendet. Beim testen der Funktionalität dieses Modells fiel uns aber auf, dass der Aufrollmechanismus für den Gürtel nicht optimal war. Dieser bestand aus zwei parallel montierten Stangen, zwischen denen der Gürtel durchgeführt und durch die Drehung eben dieser Stangen aufgerollt wurde. Leider ging durch den Abstand

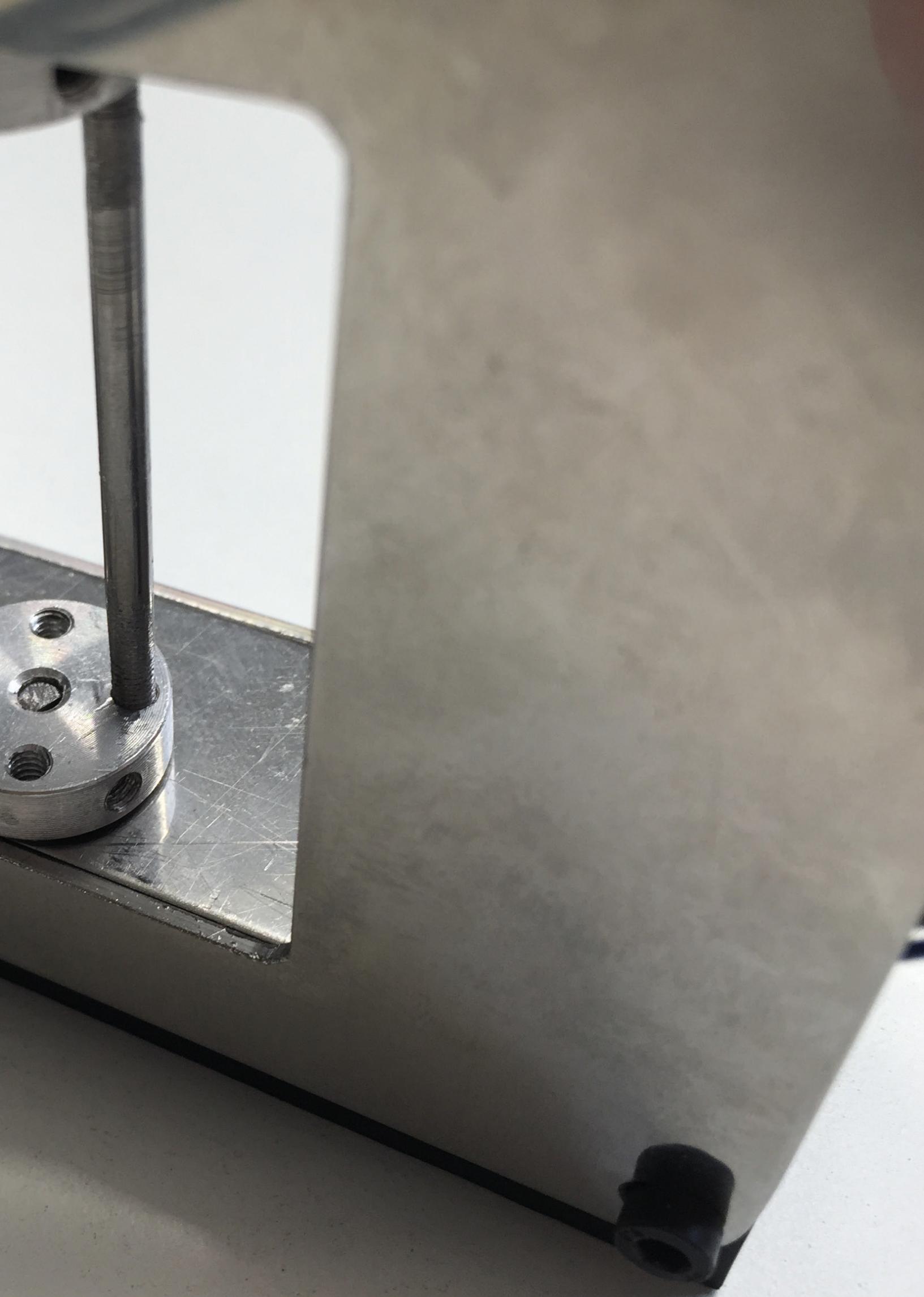
dieser Stangen zu viel Kraft verloren. Außerdem musste der Gürtel beim öffnen manuell zurückgesetzt werden. Wir entschieden uns daher dazu noch einen zweiten Prototypen zu entwickeln.

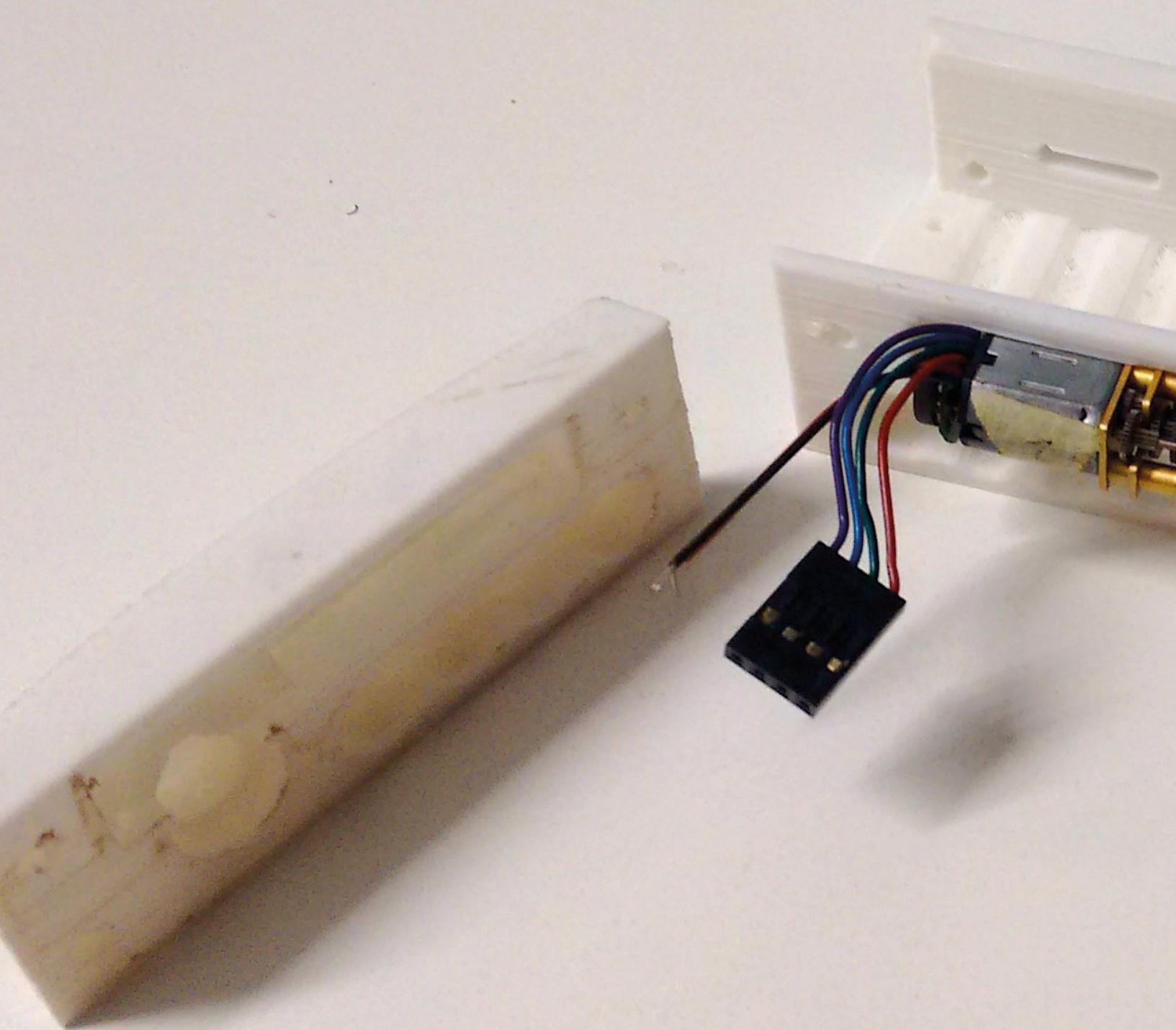
Bei dem zweiten Prototypen besteht der Aufrollmechanismus nur noch aus einer Stange mit angebrachten Zahnrad, an der ein Gürtel vorbeigeführt und aufgerollt wird. Durch diese Neukonstruktion war es möglich auch einiges an Größe der gesamten Gürtelschnalle einzusparen, was gerade bei *Wearable Devices* einen immensen Vorteil bietet. Zudem nutzen wir einen ausgeklügelten Hebelmechanismus zum verschließen des Gürtels, so dass der eingezogene Gürtel fest auf das Zahnrad gepresst wird. Der zweite Prototyp wurde sowohl aus MDF-Platten gefertigt, wie auch 3D gedruckt.



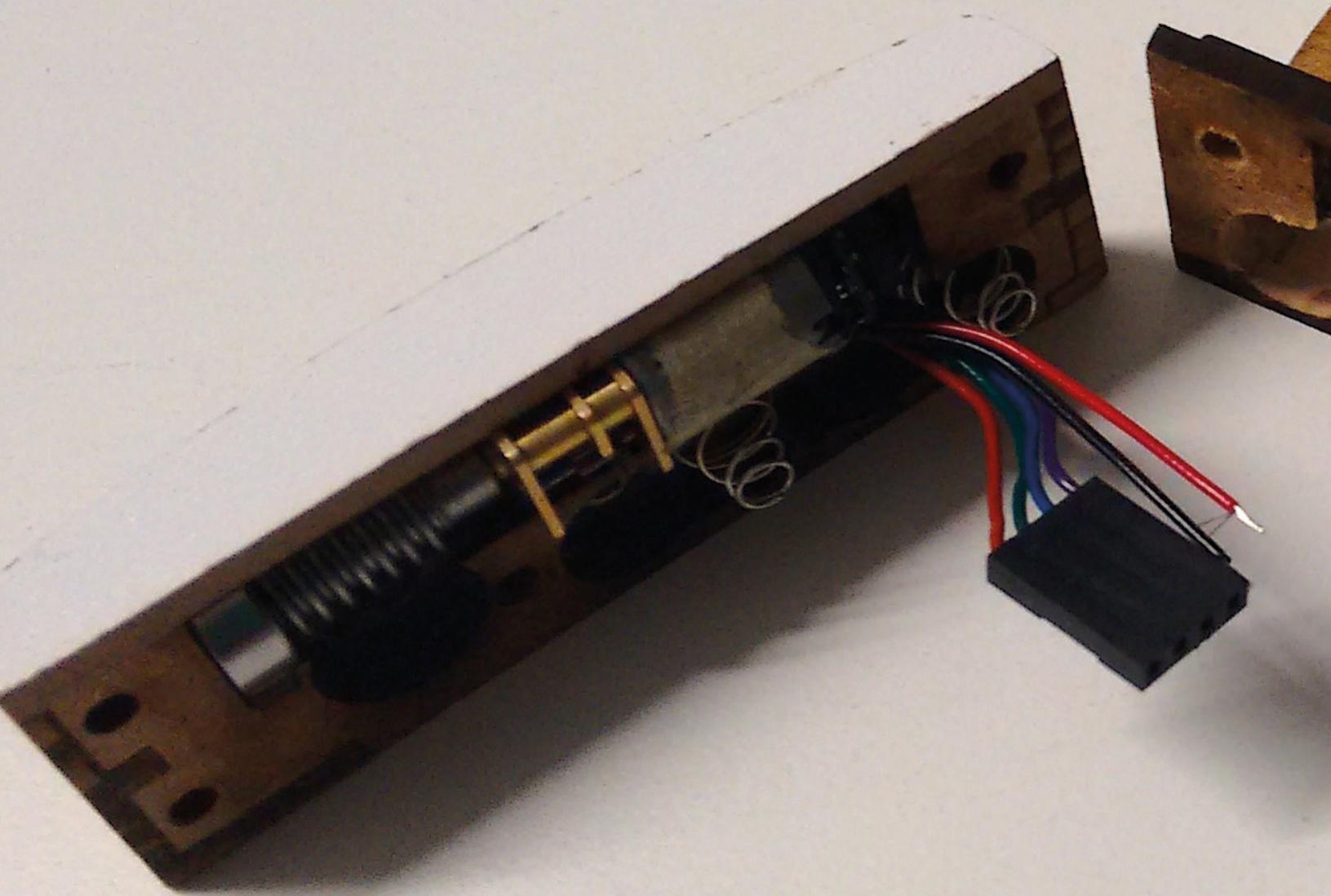


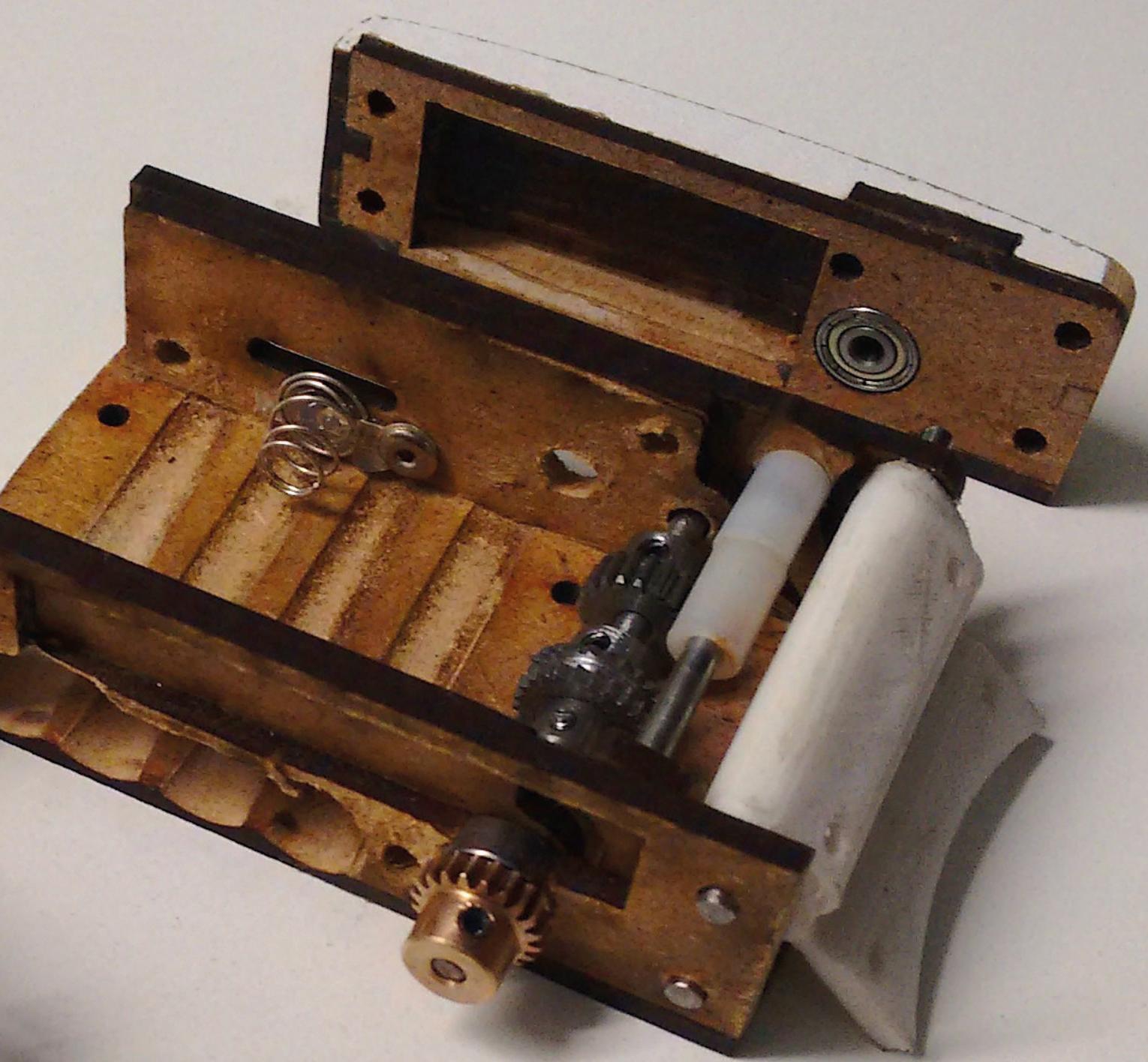


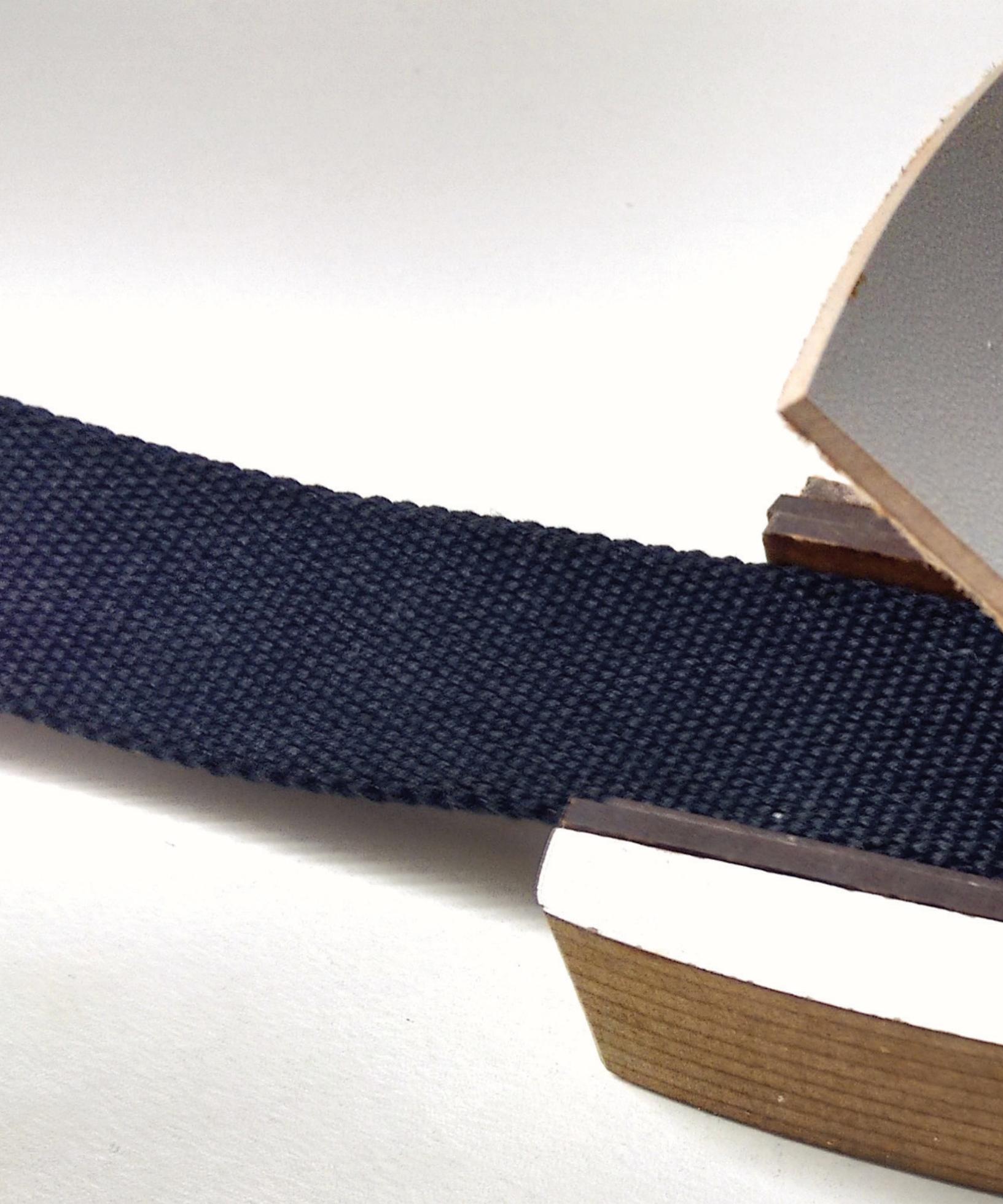




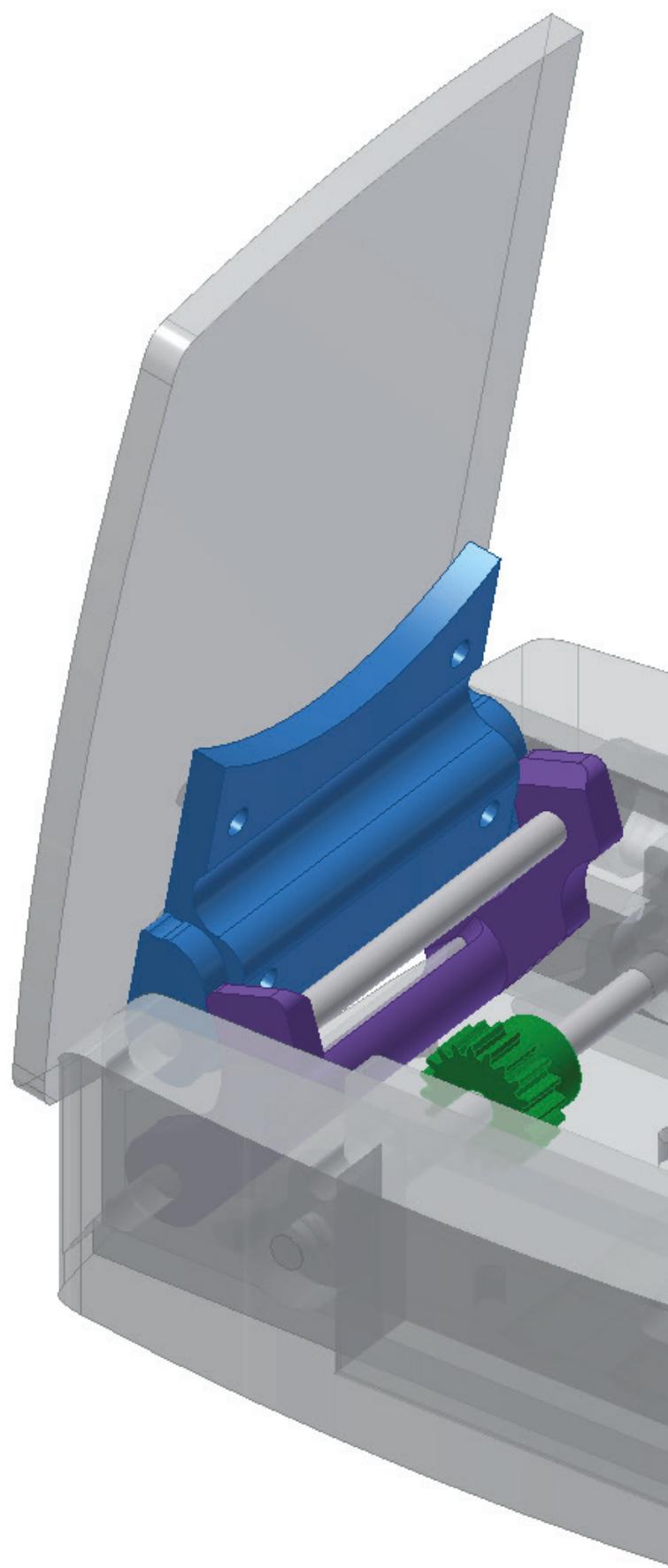


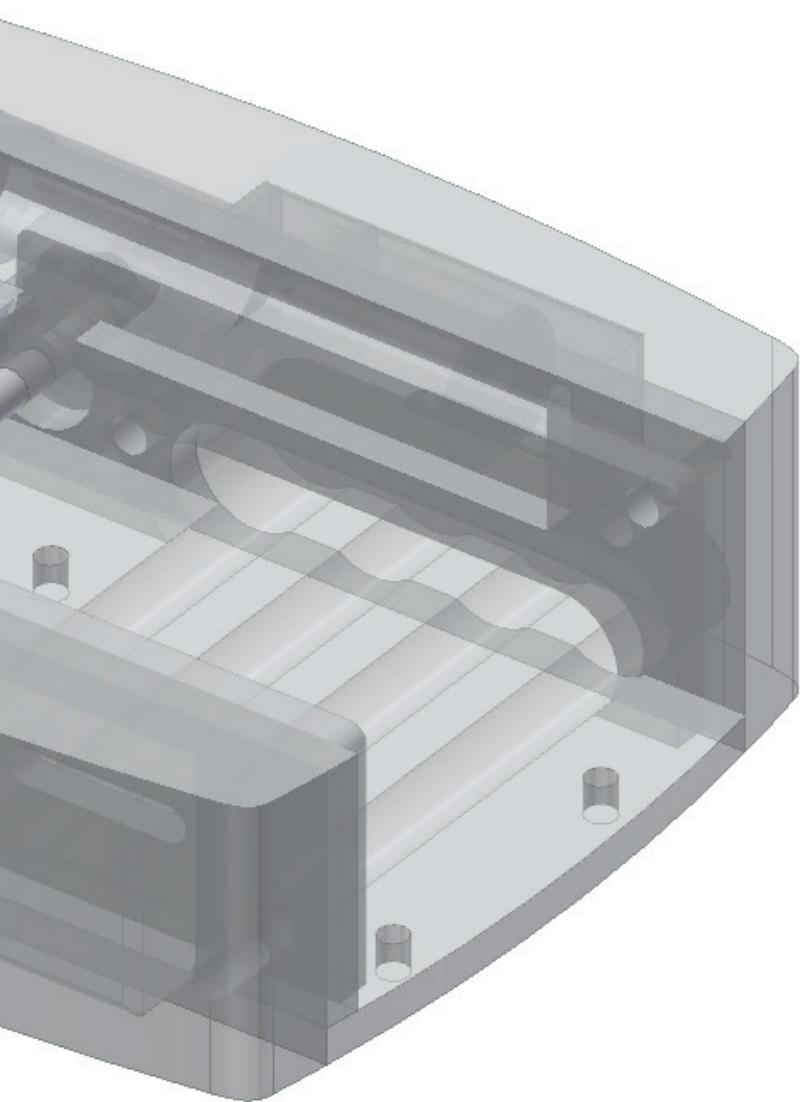


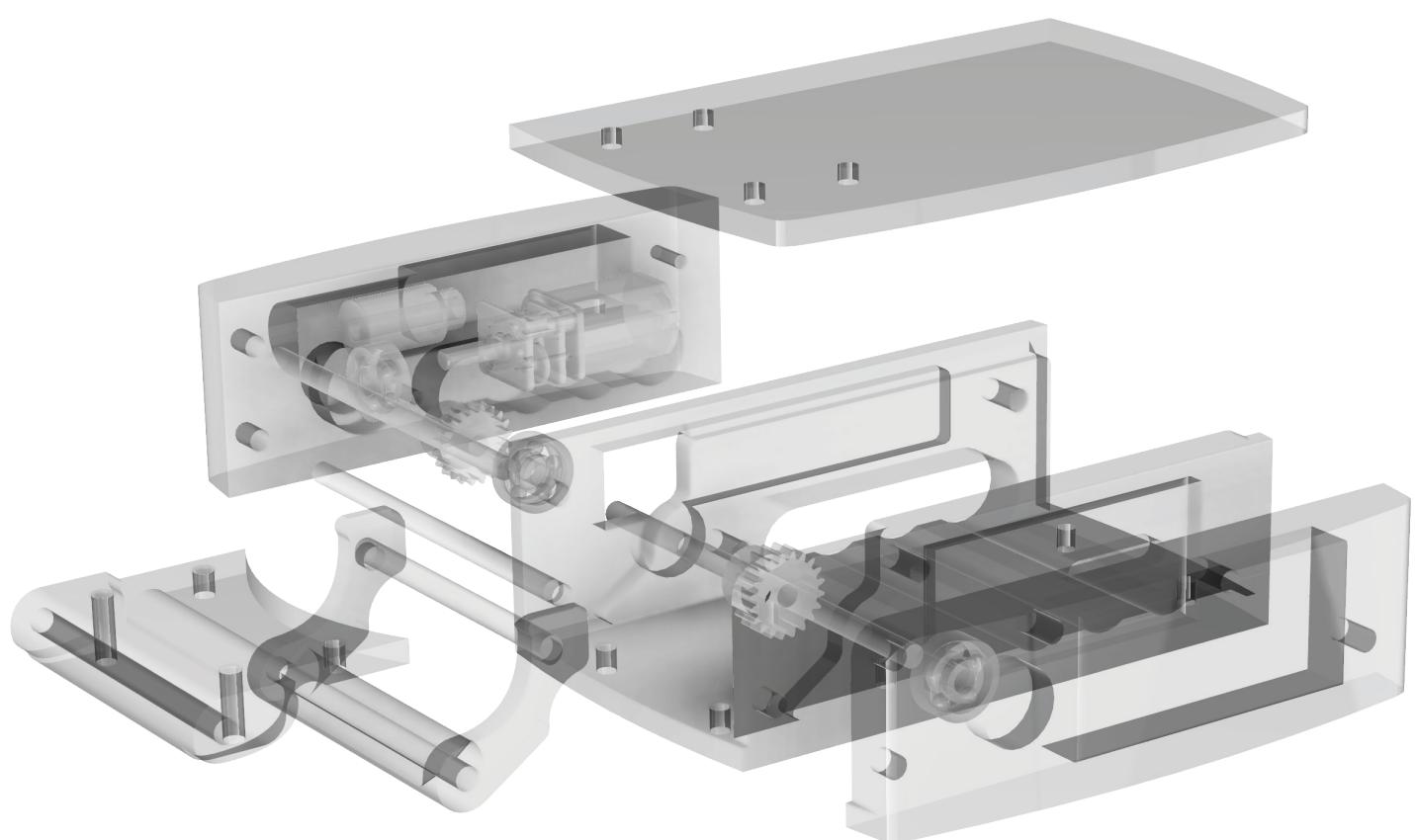


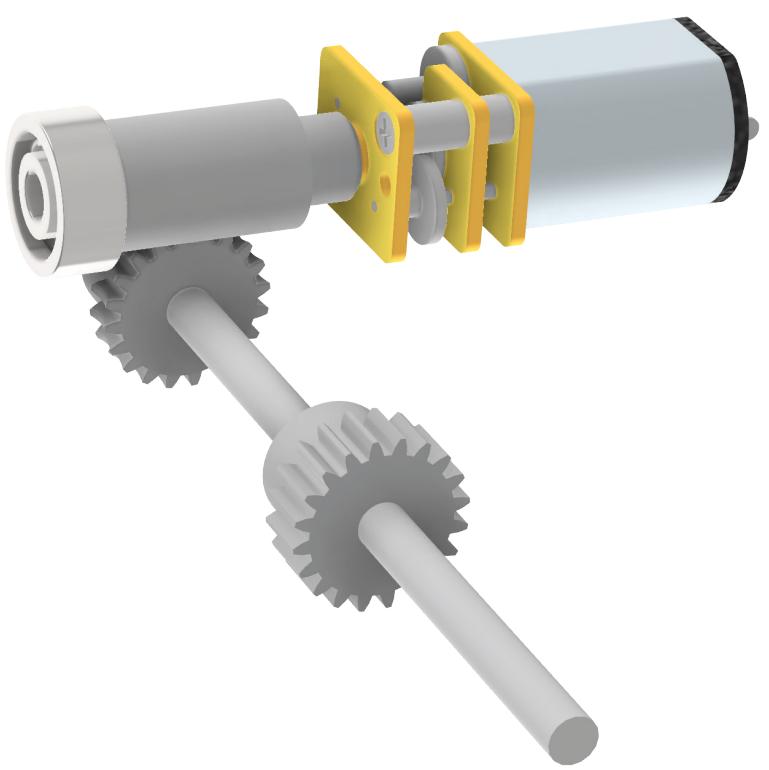
















Programmierung

Framework 7

Für die Umsetzung der App haben wir nach einem Framework gesucht, welches unseren Wunsch, ein natives Gefühl zu erzeugen, gerecht wird. Als erstes haben wir uns für *Framework 7* entschieden. Später sind wir jedoch auf das Framework *Ionic* gewechselt. Im Folgenden wollen wir die beiden Frameworks vorstellen und unsere Entscheidungen darlegen.

Framework 7 ist ein Javascript-Framework, welches darauf ausgerichtet ist ein natives Look and Feel zu erzeugen. Es ist speziell auf iOS ausgerichtet, bietet also keine Gestaltungslösung für Android. *Framework 7* zeichnet sich durch den einfachen Einstieg in das Framework aus. Man kann schnell gutaus sehende Prototypen erstellen. Jedoch bietet es keine eigene Java Script-Struktur, so dass es sehr schnell unübersichtlich wird, damit zu arbeiten. *Framework 7* bietet eine eigene Bibliothek für den Zugriff

auf das DOM (*Document Object Model*) und für Templates. Die Template-Engine *Template 7* ist eine Ableitung von *Handlebars*, einer auf *Mustache.js* basierenden Template-Engine. Die einzelnen Templates für die App sind in script-Tags in der *index.html* gespeichert. Sie werden über JavaScript eingelesen und zu JavaScript kompiliert. Es kann dann ein Kontext, also verschiedene Daten, zum Template geschickt werden, welches dann zu HTML umgewandelt und in die *index.html* injiziert wird.

Ionic

Die Finale Umsetzung der App erfolgte mithilfe des *Ionic-Frameworks*. Dieses beruht auf *Angular.js*, einem Framework für Clientseitige JavaScript Anwendungen. Der Vorteil gegenüber *Framework 7* ist, dass es im Gegensatz zu diesem auch für Android funktioniert und wir dadurch für beide Systeme entwickeln konnten. Außerdem ist die Dateistruktur deutlich übersichtlicher, da Ionic eine klare Struktur in *Controller*, *Factories* und *States* macht. Controller stellen die Logik der einzelnen States dar und werden aufgerufen, wenn ein State aktiv ist. Ein

State entspricht einer URL, welche sich durch Interaktionen ändern kann.

Eine Besonderheit von Ionic ist das *data-Binding*. Dieses bietet einen einfachen Datenaustausch zwischen View/Template und Controller. Dies ist besonders hilfreich, wenn man z.B. die Daten eines Formularfeldes auswerten möchte, da der Wert bereits im Controller vorhanden ist und nicht erst ausgelesen werden muss.

Aufbau

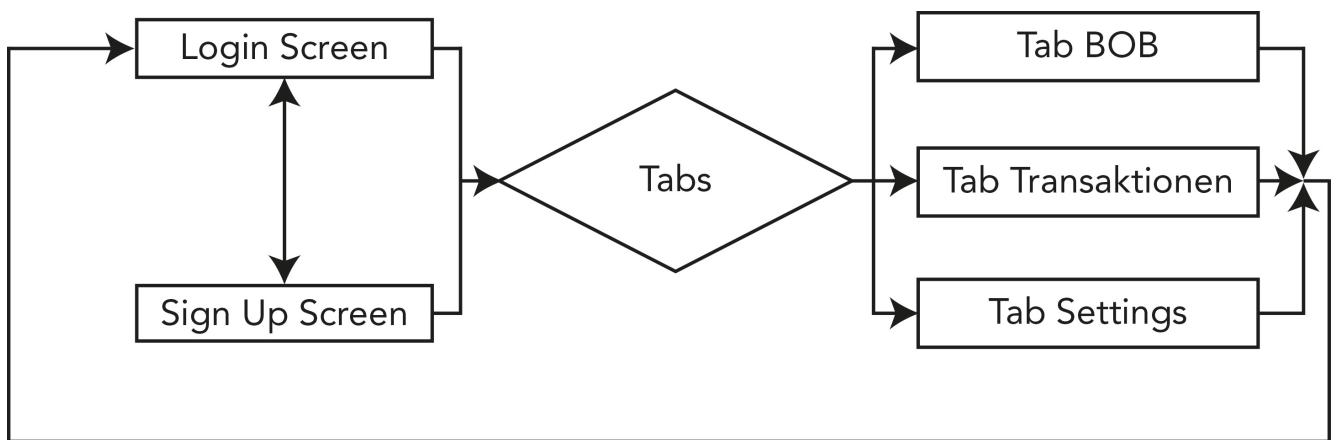
Der Nutzer sieht nach dem Starten der App einen Loginscreen, auf dem er sich einloggen kann oder zu einem Screen zum registrieren bei *figo* wechseln kann. Die Daten nach dem Login oder der Registrierung werden direkt an die *figo-API* gesendet. Die API sendet dann ein *Access Token* zurück. Dieser kann dann innerhalb von einer Stunde zur Authentifizierung von Anfragen verwendet werden. Wenn der Token nicht mehr valide ist versucht die App sich ein neuen Token mithilfe eines *Refresh Tokens* zu besorgen. Dieser ist im

Gegensatz zum Access Token dauerhaft gültig. Sollte dies fehlschlagen gelangt der Nutzer wieder zum Loginscreen und muss sich neu einloggen.

Wir nutzen die *figo-API*, da diese die einzige kostenfreie Lösung ist, auf Bankdaten und Transaktionen zuzugreifen.

In einem Intervall von zehn Minuten lädt die App neue Transaktionen über die API und berechnet das noch zur Verfügung stehende Guthaben. Anhand des Guthabens werden BOB dann verschiedene Klassen zugewiesen, welche die Darstellung

beeinflussen. Die Transaktionen werden, bevor sie als Liste gerendert werden, nach dem Datum sortiert. Zudem werden die dreistelligen Währungscodes zu den entsprechenden Schriftzeichen umgewandelt. Aus EUR würde beispielsweise ein €-Zeichen generiert werden. Zum Anzeigen der Einstellungsseite lädt die App zuerst die bisherigen bzw. die Standard Einstellungen aus dem localStorage. Die geänderten Einstellungen werden direkt nach der Änderung im localStorage geändert.



Wenn nicht eingeloggt und neuer Access Token
nicht abgerufen werden kann

Probleme

Ein Problem, auf das wir gestoßen sind ist, dass wir erst einen eigenen Login-Server schreiben mussten, da der figo-Login nicht zu einer App weiterleiten kann, wir jedoch den Access Token, welcher beim Login zurückgegeben wird, für die App abfangen mussten. Dieses Problem könnten wir nach einem netten Gespräch mit figo jedoch glücklicherweise lösen. Figo hat uns einen direkten Zugang zu der Login-API ermöglicht, so dass wir diesen direkt von der App aus ausführen können.

Gürtel

Für den Gürtel haben wir beim ersten Prototypen einen *Arduino Mini* in Kombination mit einem *Bluetooth 2.1* Chip verwendet. Zudem benutzen wir einen *DRV8835 Dual Motor Controller* mit 1.2 A Leistung pro Kanal. Die Kanäle werden zusammen geschlossen, um dem Motor genügend Leistung zur Verfügung zu stellen. An den Motor Controller ist ein 1.6 A, 6V *Microgetriebemotor* mit Encoder angeschlossen, welcher den Gürtel zusammenzieht oder wieder öffnet. Beim zweiten Prototypen

haben wir den Arduino durch einen *RedBearLab Blend Micro* mit eingebautem *Bluetooth Low Energy* ersetzt, da dieser weniger Platz verbraucht und durch die Verwendung von Bluetooth Low Energy nicht nur mit Android, sondern auch mit iOS läuft. Die App übermittelt den aktuellen Stand in Prozent über Bluetooth an den Gürtel, das heißt wenn der Gürtel zu 50 Prozent zugezogen ist sendet die App eine 50 an den Arduino. Der Arduino übersetzt diesen Befehl dann in Steuersignale an

den Motorcontroller. Über den Encoder werden im Folgendem die Umdrehungen des Motors gezählt und dadurch die aktuelle Position des Gürtels errechnet. Bei öffnen des Gürtels wird die Position des Gürtels zurückgesetzt und beim schließen neu angefahren.

