

Hochschule Worms
Fachbereich Informatik
Studiengang Angewandte Informatik B.Sc/M.Sc

Dokumentation zur Haushaltsmanager-App

vorgelegt von
Chiara Police und Nico Vogel

Gutachter:	Prof. Dr. Stephan Kurpjuweit
Bearbeitungszeitraum:	Wintersemester 2020/21
Abgabedatum:	14. Februar 2021

Inhaltsverzeichnis

Anfängliche App Idee	3
Erste Grundzüge	4
Barcodescanner	5
Neubeginn	7
Screenshots unserer App	11

Anfängliche App Idee

Nach der ersten Vorlesung haben wir uns Gedanken über mögliche Apps gemacht und konnten uns schnell auf eine App-Idee einigen.

Unsere App hilft ihren Benutzern eine Übersicht über deren Einkaufsprodukte zu bekommen.

Die Benutzer können nach dem Einkaufen mithilfe eines Barcode Scanners Produkte einscannen und Name, Anzahl und Preis des Produktes angeben. Die Produkte werden dann zu einer Datenbank hinzugefügt, wodurch man eine Liste erhält, in der man sehen kann was für Produkte man zu Hause hat.

Sobald der App-Nutzer dann ein Produkt leer macht, kann er dieses wieder einscannen und aus der Datenbank löschen. Anschließend soll der Benutzer gefragt werden, ob das Produkt auf eine Einkaufsliste soll. So kann sich der Benutzer neben der Liste von Produkten, die er bereits im Haushalt hat, auch noch eine Einkaufsliste erstellen.

Erste Grundzüge

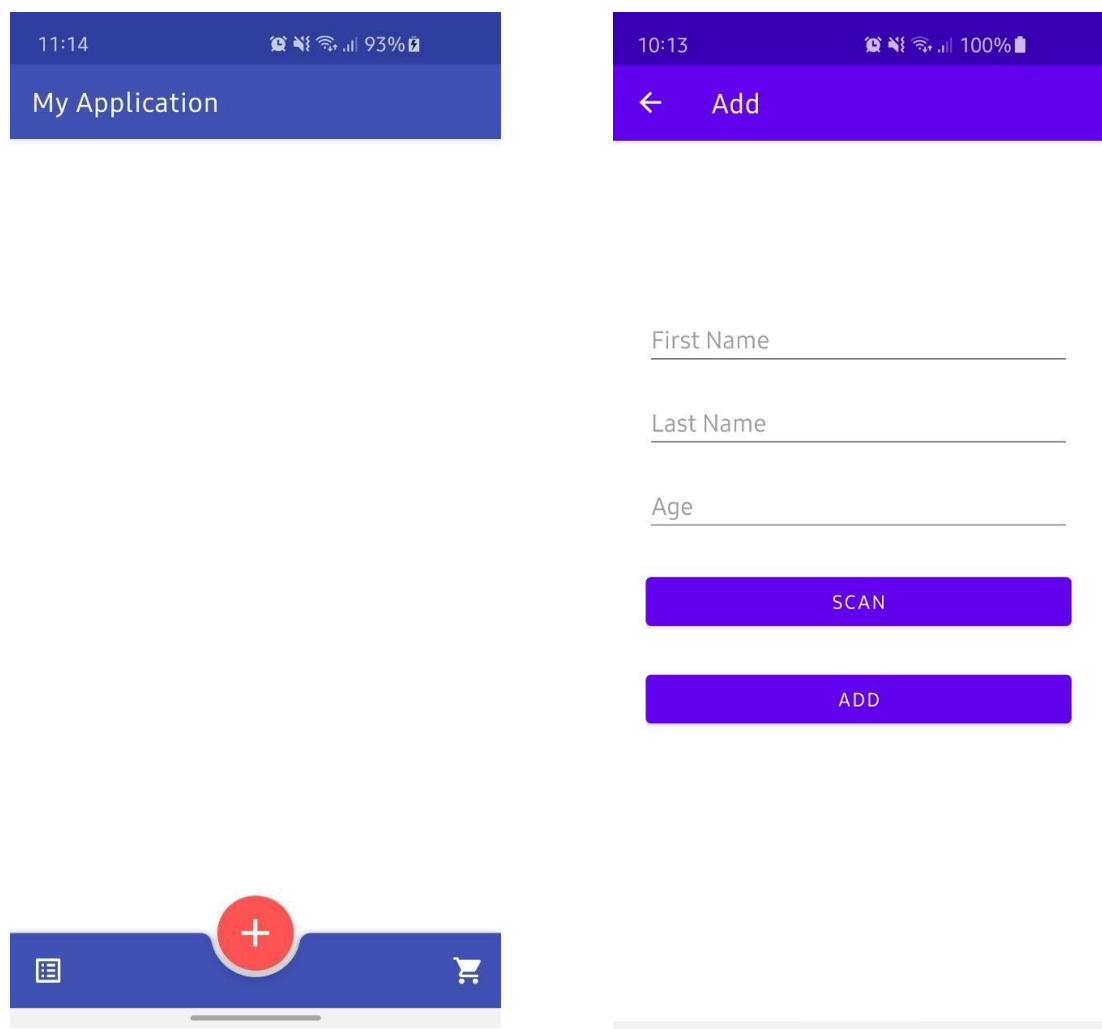
Nachdem wir nun wussten, was unsere App machen soll haben wir auch schon angefangen uns über das Erstellen einer Datenbank zu informieren. Hierzu haben wir uns diese Videoreihe auf Youtube ([Link](#)) angeschaut und zunächst eins zu eins nachgebaut, um erstmal ein Gefühl dafür zu bekommen.

Da wir die App am eigenen Smartphone testen möchten mussten wir uns noch eine Anleitung suchen, die erklärt wie man sein Smartphone mit Android Studio verbinden kann. Hierfür haben wir uns ein Video ([Link](#)) angeschaut.

Nachdem wir nun die komplette Videoreihe durchgearbeitet hatten, haben wir ein zweites Projekt erstellt und den Code für unsere Zwecke entsprechend angepasst.

Anschließend machten wir uns Gedanken über die Gestaltung unserer App und haben uns dazu entschlossen eine Navigationsleiste einzufügen. Dies konnten wir mithilfe eines Videos ([Link](#)) umsetzen.

So sah unsere App nun aus:



Barcodescanner

Als Nächstes haben wir den Barcodescanner eingefügt, der sich öffnen soll sobald man auf das Plus drückt. Das haben wir mithilfe eines Videos ([Link](#)) umsetzen können.



Nun wollten wir noch sogenannte "Tabs" hinzufügen, um vom Inventar bzw. der Bestandsliste auf die Einkaufsliste zu wechseln. Das konnten wir mit diesem Video ([Link](#)) umsetzen. Leider wurde die Liste anfangs nicht richtig angezeigt.



Daraufhin haben wir ein neues Projekt erstellt und zunächst das "Tab Layout" und die "Navigation Bar" eingefügt. Anschließend haben wir die Datenbank übertragen und die entsprechenden Fragmente für die Liste, das Hinzufügen und Ändern von Produkten erstellt.

Leider war das Programm zwar syntaktisch, aber nicht semantisch korrekt.

Wir haben natürlich versucht den Fehler zu finden, jedoch mussten wir dabei feststellen, dass wir mit der App grundsätzlich nicht wirklich zufrieden waren.

Wir wollten eine App erstellen, die mit Hilfe des Barcode Scanners Produkte erkennt und in das Inventar einfügt. Bei der aktuellen App musste man jedoch die Produkte selbst einfügen. Der Barcodescanner hatte eigentlich nicht wirklich eine Funktion. Darum haben wir uns entschieden nochmal von vorne anzufangen und es mit einer bereits gefüllten Datenbank umzusetzen.

Das heißt es soll mit dem Barcode Scanner möglich sein Produkte einzuscannen und diese dann aus der Datenbank direkt in das Inventar hinzuzufügen. Hierfür haben wir uns entschieden eine Beispieldatenbank mit einigen Beispiel Produkten selbst zu erstellen.

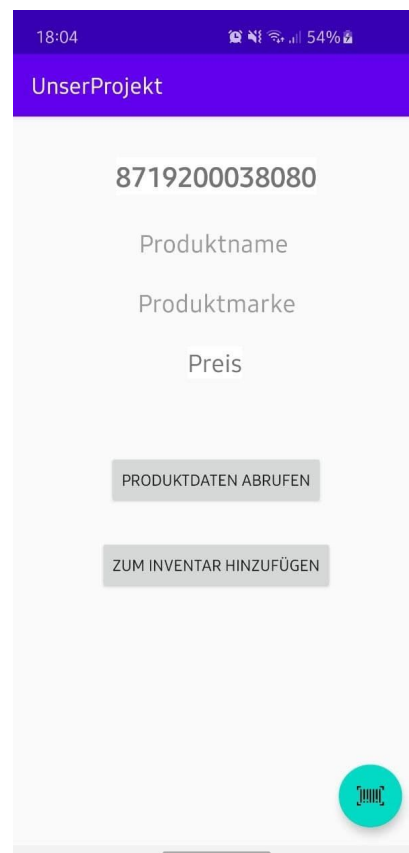
Neubeginn

Wir haben also wieder ein neues Projekt erstellt und diesmal mit dem Barcodescanner angefangen, der aufgerufen wird sobald man auf den "Floatingaction Button" klickt.

Dafür konnten wir den Code aus unserem vorherigen Projekt nutzen.

Im Anschluss daran wollten wir umsetzen, dass der gescannte Barcode automatisch in ein Textfeld in der "MainActivity" eingefügt wird.

Hierfür mussten wir uns erstmal darüber informieren, wie man Daten zwischen "Activities" übergibt. Das konnten wir schließlich mit einem Video ([Link](#)) herausfinden.



Nun ging es weiter mit einer neuen Datenbank, welche wir mit SQLite erstellten, und in diese erstmal ein paar Produkte mit den Werten: Barcode (PK), Produktname, Produktmarke und Preis hinzugefügt haben. Zur Erklärung und Umsetzung dessen schauten wir uns ein Video ([Link](#)) an.

Mithilfe eines weiteren Videos ([Link](#)) konnten wir schließlich noch überprüfen, ob der eingescannte Barcode in der Datenbank enthalten ist oder nicht.

Im Anschluss folgte die Umsetzung zum Erhalten der Produktinformationen (z.B. Produktname), mit der wir feststellen konnten, ob das Produkt bereits in der Datenbank enthalten ist.

Wir hatten jedoch zunächst Probleme damit die Produktinformationen, aus der Datenbank, als "TextView" anzeigen zu lassen. Wir konnten jedoch eine Lösung hierfür anhand dieses Videos ([Link](#)) finden.

Ziel war es nun die Datensätze der Produkttabelle in eine andere Tabelle zu kopieren und diese anschließend im "RecyclerView" anzeigen zu lassen.

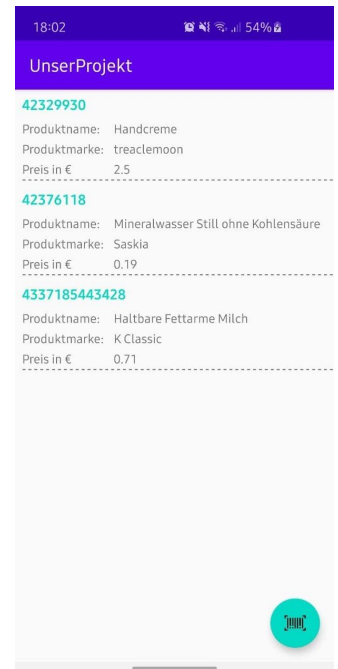
Die App lief zwar zu Beginn, stürzte aber immer wieder ab.

Auf dem Weg der Lösungsfindung stießen wir auf ein weiteres Video ([Link](#)), in dem gezeigt wird wie man sich bestehende Datensätze anzeigen lassen, neue hinzufügen und auch Bestimmte suchen kann.

Jetzt konnte man die Produktinformationen abrufen, nachdem man den Barcode eingescannt hat.



Außerdem konnten wir es nun umsetzen die zweite Tabelle "Inventar" zu erstellen und diese mit den entsprechenden Produktinformationen aus der anderen Tabelle zu füllen, wenn man den entsprechenden Button klickt. Die Tabelle "Inventar" wird nun im RecyclerView angezeigt.

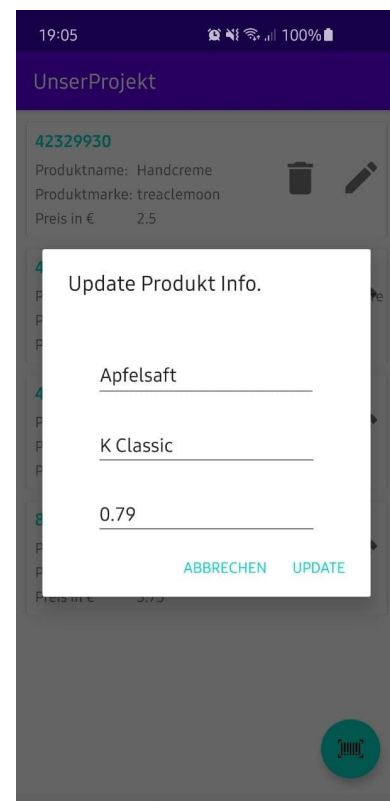
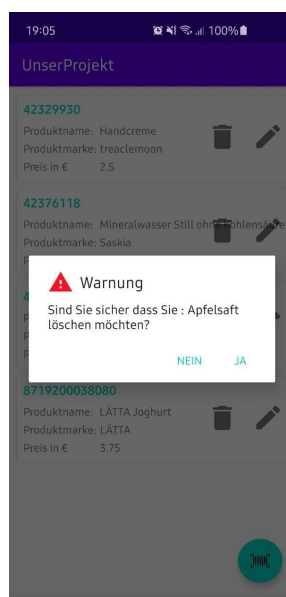
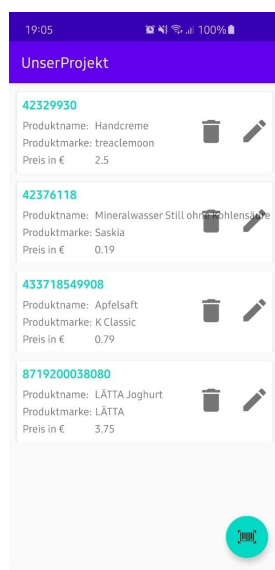


Wir konnten ebenfalls herausfinden, dass manche Barcodes zu groß für den Datentyp "INTEGER" waren, weswegen wir nun "LONG" verwendeten.

Desweiteren haben wir dafür gesorgt, dass man nach dem Hinzufügen von Produkten automatisch zur Übersicht der Produkte gelangt.

Im Anschluss wollten wir herausfinden wie man einzelne Einträge aus dem Inventar löschen und verändern kann. Hierbei konnte uns ein Video ([Link](#)) helfen.

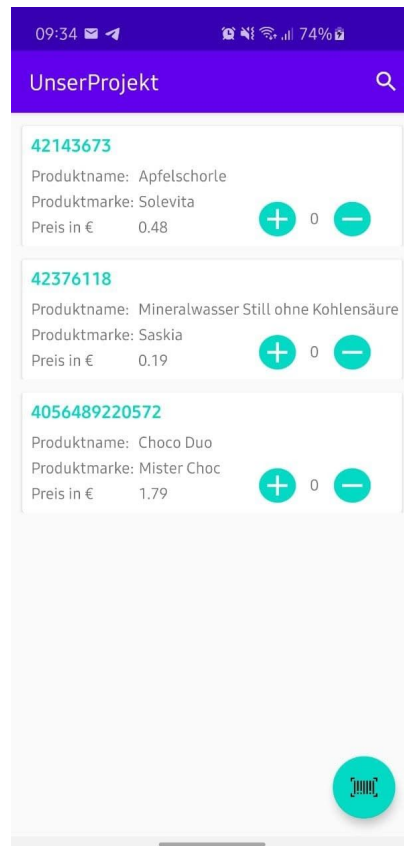
Zusätzlich änderten wir noch das Design des "RecyclerView", indem wir "CardView" eingefügt haben und dafür sorgten, dass man durch einfaches anklicken der Produkte diese verändern kann.



Als Nächstes wollten wir umsetzen, dass man durch Wischen die einzelnen Produkte wieder löschen kann. Dabei hat ein Blogbeitrag ([Link](#)) sehr geholfen.

Der nächste Punkt war die Suche nach Produkten, welches wir mit Hilfe eines Videos ([Link](#)) in unsere "TopBar" einfügen konnten.

Zudem wollten wir angeben, welche Anzahl man von welchem Produkt Zuhause hat und haben dies entsprechend umgesetzt.



Uns ist aufgefallen, dass die Anzahl der Produkte nicht richtig gespeichert und deshalb bei jedem Öffnen der App auf "0" gesetzt wurde.

Das konnten wir lösen, indem wir die Spalte "InvNumber" der Tabelle "Inventar" hinzugefügt haben und nun beim Inserieren eines Produkts die Möglichkeit geben die Anzahl einzugeben und im Update Fenster diese auch wieder zu ändern. So kann man die Anzahl auch nicht versehentlich auf dem Hauptscreen ändern.

Zuletzt haben wir unsere App noch mit der Einkaufsliste und einer "Bottom Navigation Bar" ergänzt. Dadurch konnten wir ermöglichen, dass man durch anklicken der entsprechenden Elemente zwischen den "Activities" wechseln kann. Orientierung holten wir uns hierbei bei einem StackOverflow Eintrag ([Link](#)).

Wir haben festgestellt, dass sich die Einkaufsliste automatisch leert, wenn man den Preis mit einem Komma anstatt mit einem Punkt trennt. Deshalb haben wir uns überlegt wie man das verhindern könnte. Wir sind zum Entschluss gekommen, dass ein "NumberPicker" dafür sehr gut geeignet ist und haben deshalb zwei davon in unserem Projekt implementiert. Dazu haben wir uns für die Umsetzung dieses kurze Video ([Link](#)) angesehen. Der letztendliche Abschluss bildet das finale Design - dunkel mit orangenen Akzenten.

Quelle des Barcode Icons: [Link](#)

Screenshots unserer App

