INEZ / EZML

Die INEZ (Intelligenter Einkaufszettel) oder auch genannt EZML (Einkaufszettel mit Maschine Learning) ist eine Einkaufsliste die dir helfen soll schneller deine Einkaufsliste zu schreiben. Die App läuft auf MacOS (Beta 10.15 19A546d), iOS (iOS 13) und iPadOS. Die ganze app ist in swiftUI geschrieben und nutzt zwei ML Models um ein möglichst großes Produkt Portfolio abzudecken.

Features:

- Autovervollständigung von den Mengeneinheiten (Liter, Stück, Packung Flasche, Mal) durch ML
- Vervollständig der stück Anzahl und automatisches Addieren bei doppelter Eingabe
- Edit function um Tippfehler zu korrigieren
- Produktvorschläge durch ML
- Quick add um schneller Produkte hinzuzufügen
- Check button um die Produkte als eingekauft zu markieren
- Delete button um Produkte von der liste zu entfernen
- Erkennen von Plural und Singular
- Deutsche und Englische version (Produkte bleiben immer deutsch)
- Onboarding Screen (Created by GitHub - BLCKBIRDS/SwiftUI-Onboarding-Screen: How to create an Onboarding View by embedding an UIPageViewController)

Bugs:

- Quick add: Addieren der Stückzahl wird nicht in der View geupdatet (Dieses Feature braucht @ObservableObject welches unter der aktuellen macOS beta leider kaputt ist und die app zum Absturz bringt)
- Bei Eingabe von z.b. 1 Packung Brot wird 1 Stück Packung Brot zu liste hinzugefügt.
- Products von Unit tests bleiben in CoreData
- Onboarding ist nur auf Englisch (.strings file wird nicht in arrays gelesen :( )

Über die App:

Die app ist innerhalb von 10 tagen entstanden. Dabei wurden die ML model erstellt und die dazugehörigen Datensets. Die Ideen für die features und die app stammt von August Wettbewerb von it-taltens.de. Die Datensatz basieren auf einer englischen Lebensmittel Datenbank die ich sortiert, gefiltert und auf deutsch übersetzt habe. Für die Mengeneinheiten ML habe die die ganzen Daten per hand kategorisiert um ein möglichst akkurates Ergebnis zu erziehen. Für die Produktvorschläge habe ich die Daten durch eine simulation genieren lassen, der Datensatz basiert auf Zufälligen kaufverhalten und herstellen. Das ML model analysiert dieses Datensatz und gibt auf dieser Grundlage Produkt Empfehlungen.

Ich habe mich für ML entschieden da es deutlich effizenter ist und ich mit kleinen datensetz ein möglichst breites feld abdecken kann.

Das UI dieser App basiert auf einen mix von den Design des App Stores und Apple Music. Die navigation von der macOS app ist custom build da die navigationView von swiftUI komplett umbrauchbar ist. Das Design ist in light mode und dark mode verfügbar! Die Add view sollte eigentlich live Produktvorschläge geben und eine Autovervollständigung bieten. Leider ist beides nicht möglich da ich dafür @ObservableObject nutzen müsste was wie bereits erwähnt aktuell kaputt ist. Die Einkaufsliste wird in CoreData gespeichert um den Datenschutz zu gewährleisten. CoreData ist apples eignes Datenbank framework welches die Daten local speichert. Eine synchronisation ist leider noch nicht möglich. Die App enthält einige Unit tests die die Verbindung zu CoreData prüfen, lessen, Schreiben, Löschen und die ML models testen, auch wird die Funkion der user Eingabe Prüfung geprüft (splitString).

Die Struktur der app ist MVVM (Mode lView ViewModel) ich halte dies für ein gute Struktur für swiftUI die auch sehr gut und ohne problem funktioniert hat umzusetzen. MVVM passt sehr gut das es in Swift UI keinen klassichen kontroller gibt, es ist wie react auch eine DSL spache/ framework, da für react auch MVVM empfohlen wird lässt sich dies eins zu eins auf swiftUI übertragen. Ich bin zufrieden mit der entscheidung.

Die Erkennung von Plural und Singular kommt von einen swift Package, welches es uns recht einfach macht diese Funkion hinzuzufügen. Ich habe ein paar custom rules abgestellt damit es für die deutschen Wörter zuverlässig klappt. GitHub - Cosmo/GrammaticalNumber: 1️⃣➡️🔢 Turns singular words to plural and vice-versa in Swift. Das Package wurde genau am 31. raus gebracht genau richtig das ich es noch einbauen konnte :) danke @cosmo!

Wie baue ich die app ?

Okay das erfordert etwas Vorbereitung, benötigt wird ein mac mit MacOS (Beta 10.15 19A546d) dies ist die aktuellste beta die mir zu verfügung steht. Ich kann nicht garantieren das bei einer neueren funktioniert da ich nicht weiß was apple ändern wird. Ältere sollten auch nicht laufen. Dazu wird die beta des programmes Xcode benötigt Version 11.0 beta 6 (11M392r), auch hier neuere oder ältere Versionen können Probleme machen (sollten Probleme machen). Falls ihr nicht die passenden Versionen zu verfügung habt bitte kontaktiert mich, dann passe ich das Programm an die für euch zu

Verfügung stehenden Versionen an. Downloads zu finden unter developer.apple.com/download (bei beiden die beta 7 downloaden unter Xcode wurde vergessen die Versionsnummer hoch zu drehen aber die website zeigt es richtig an, werdet ihr dann sehen :) sonst auf die build Nummer schauen)

Sobald diese beiden dinge installiert sind, öffnet ihr die `smarte_einkaufsliste.xcodeproj` unter in den Projekt Einstellungen (erreichbar in obersten Eintrag des File trees auf smarte_einkaufsliste). Dort wählt ihr unter `Targets` den Punkt `smarte_einkaufsliste` geht dann auf `Signing & Capabilities` und passt das team an und den iOS bundle identifier, der MacOS identifier sollte sich automatisch anpassen. Wenn nicht kann man ihn unter `Build Settings` anpassen.

Jetzt wählen wir als Build device MyMac aus.

Nachdem ihr das getan habt braucht ihr nur noch Run zu drücken (oben links)

Wie lasse ich die unit tests laufen ?

Die tests befinden sich in den Ordner `smarte_einkaufslisteTests` dort liegt die `smarte_einkaufslisteTests.swift` nun haben wir bei der Klasse ein run button und bei jeder einzelnen Funkion, die klasse führt alle testes nacheinander aus, die bei der Funkion nur die jeweilige Funkion.

Wo ist der edit mode ?

Der edit mode ist leicht versteckt, In your list musst du auf den produkt namen klicken um in den edit mode von den item zu kommen.