**ŽILINSKÁ UNIVERZITA V ŽILINE  
FAKULTA RIADENIA A INFORMATIKY**

**Vývoj aplikácií pre mobilné zariadenia**

Semestrálna práca

Alexander Krajči 5ZYI23

Školský rok 2021/2022

# Špecifikácia zadania

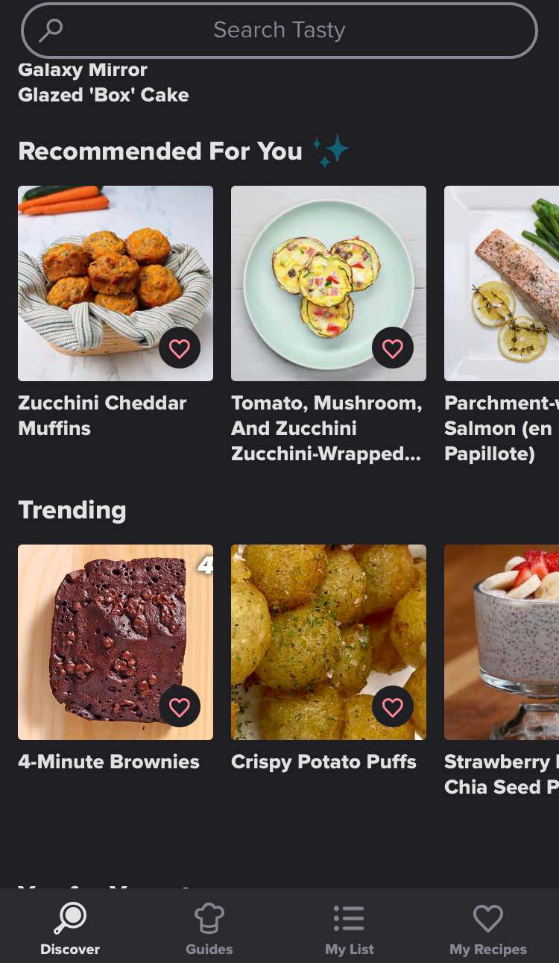
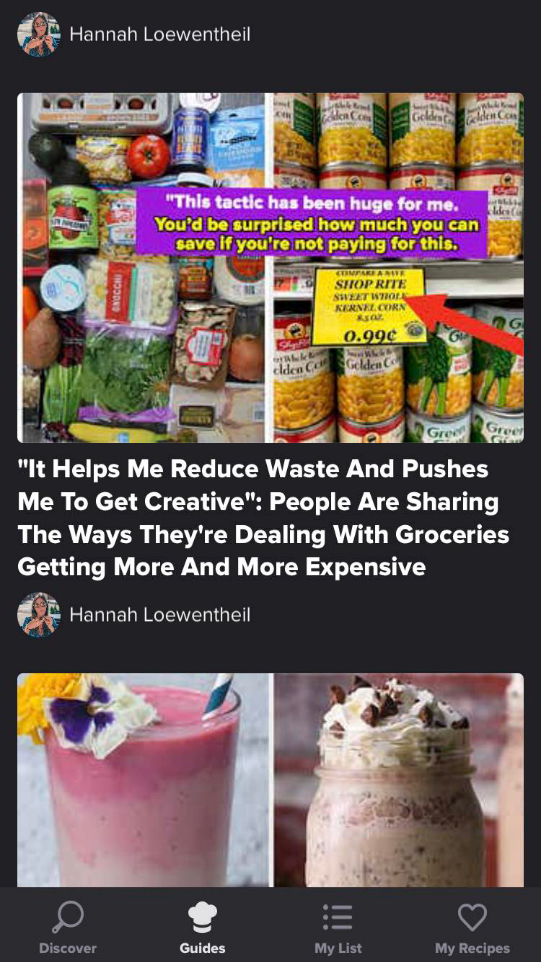
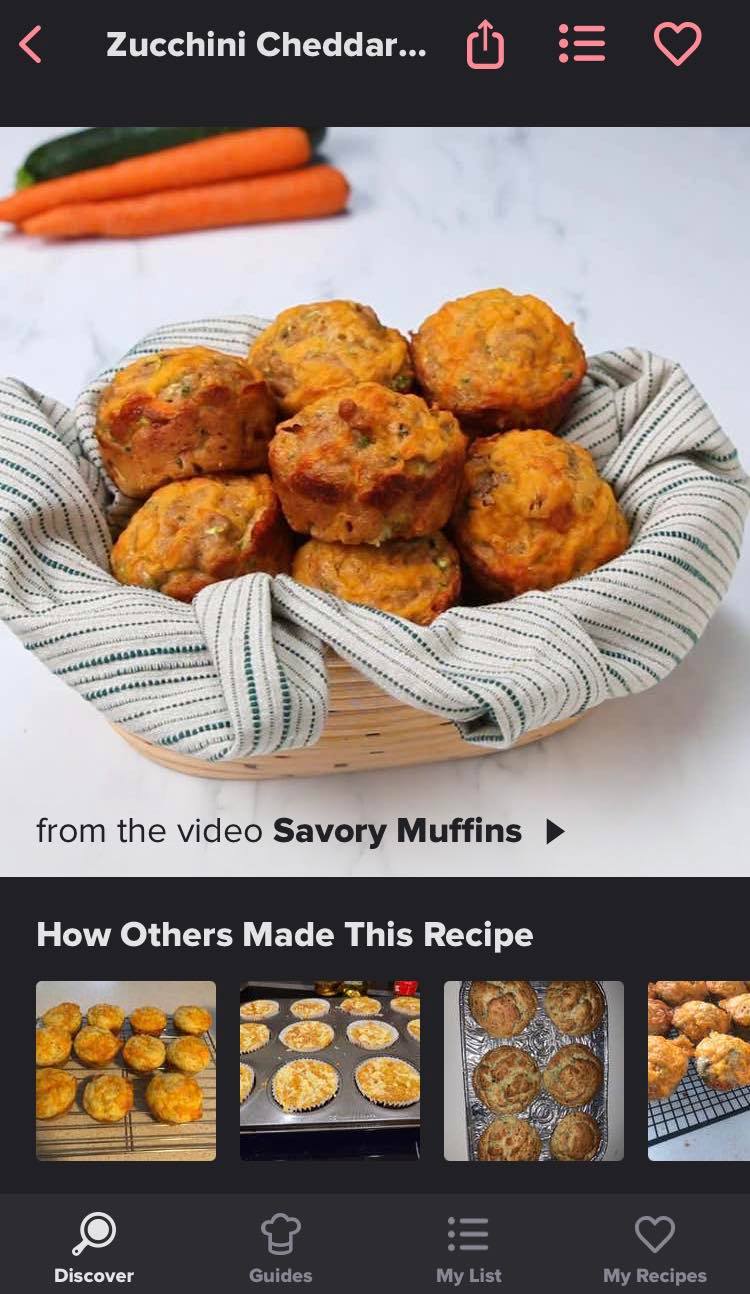
Ako tému pre semestrálnu prácu som si vybral Recept aplikáciu, ktorej účelom je zobraziť používateľovi jedlo a aké ingrediencie na prípravu potrebuje, poprípade si vytvoriť vlastný nákupný zoznam, kde si zapíše tieto ingrediencie. Taktiež zobrazuje posledných 5 receptov, ktoré užívateľ naposledy navštívil.

# Prehľad aplikácii podobného zamerania

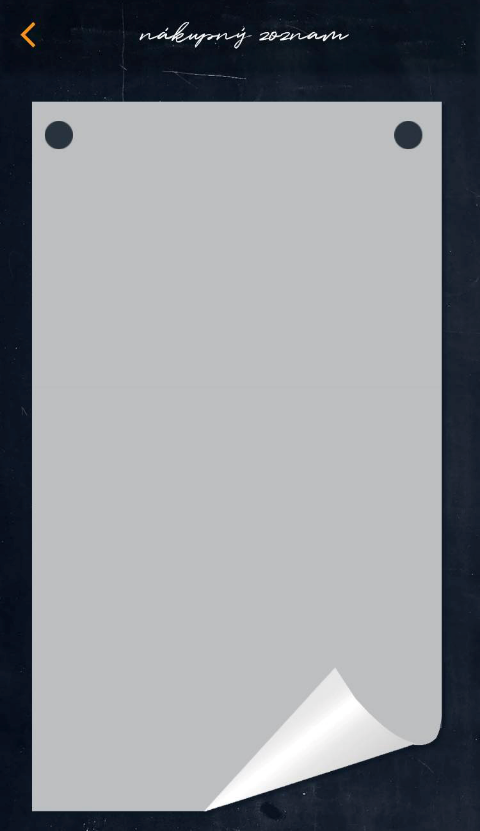
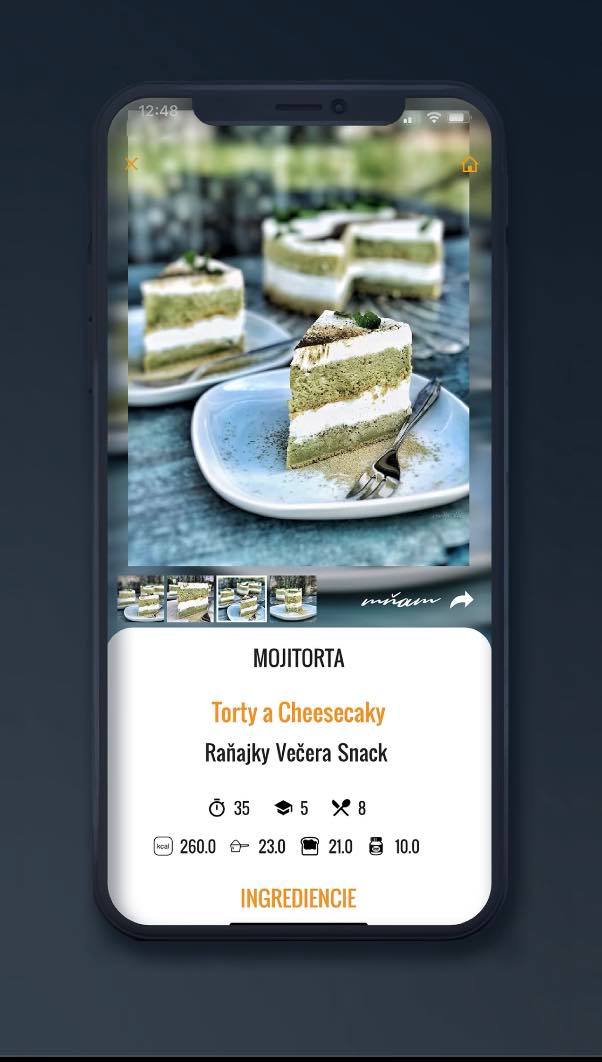
Aplikácie na AppStore podobné tej mojej :

TASTY:

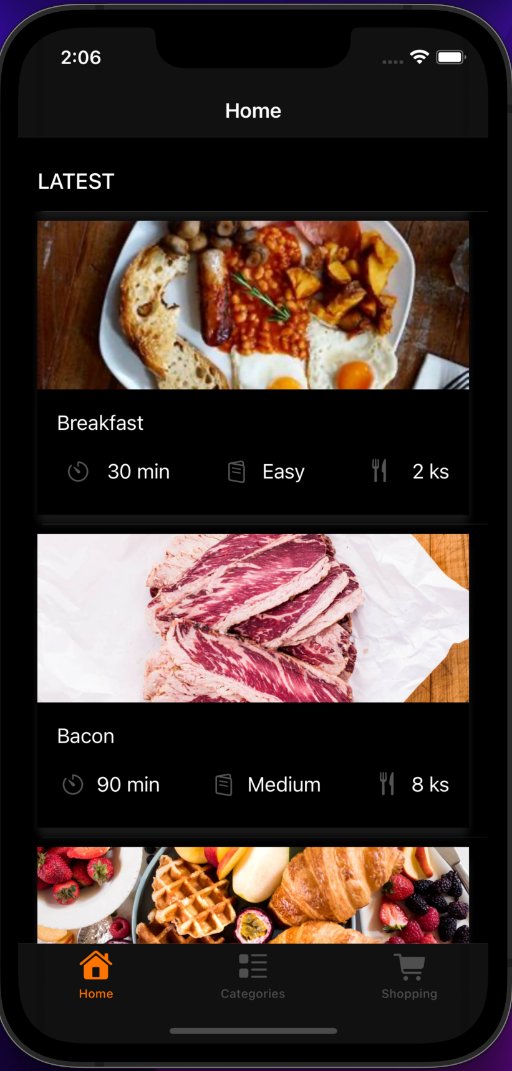
Jedná sa o appku s rôznymi receptami. Medzi svojimi službami ponúka odporúčanie nových receptov, vyhľadávanie receptov, návody na prípravu receptov, vytvorenie si vlastného nákupného košíka teda inventár potravín, ktoré si treba zadovážiť pred prípadným varením, a položka my recipes, kde si užívateľ môže ukladať obľúbené recepty alebo pridať vlastné.  
Zopár screenov z aplikácie Tasty:



MňamMňam:   
Jedná sa o appku, ktorá je nabitá zaujímavým UI dizajnom, ponúka rôzne funkcie ako: vyhľadávanie receptov, návody ako recepty pripraviť, počítanie kalórii, vedenie si vlastného nákupného zoznamu.   
Zopár screenov z aplikácie MňamMňam:



Moja aplikácia sa líši od hore uvedených v týchto bodoch:  
- Moja aplikácia ponúka možnosť pozrieť si 5 posledných receptov, ktoré užívateľ navštívil.  
- , nemá prešpekulovaný dizajn, kedy človek ani poriadne nevie, na čo sa díva

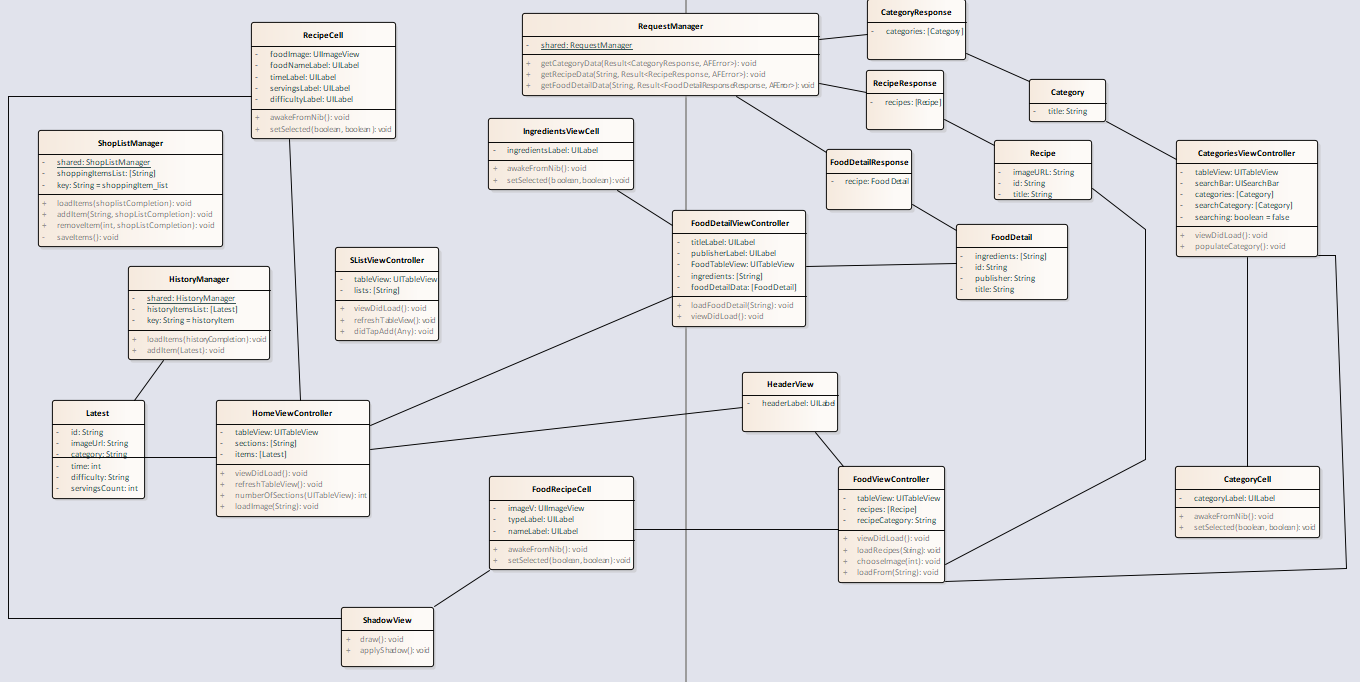
Zopár obrázkov z mojej aplikácie:

# Analýza navrhovanej aplikácie

## Krátka analýza:

Používateľ môže: - filtrovať kategórie podľa názvu  
 - prehľadávať jedlá danej kategórie  
 - zistiť informácie o ingredienciách nutných pre prípravu daného jedla  
 - prehľadávať 5 posledne navštívených receptov  
 - pridať prvok do nákupného zoznamu  
 - vymazať prvok z nákupného zoznamu

# Návrh aplikácie



# Popis implementácie

Implementáciu som začal vytváraním jednotlivých storyboardov pre jednotlivé screeny. Teda začal som najprv grafickými prvkami, ku kódeniu som sa dostal neskôr. Vytváral som si vlastné celly na ukladanie dát, v ktorých som musel riešiť constrainty, aby sa mi nerozchádzal text. V receptCell (obrázok naľavo) som prvky pridával do stackView konkretne obrazok a popis. U FoodCell (obrázok napravo) som popisy len oddelil constraintami.

A picture containing text, screenshot, monitor

Description automatically generatedGraphical user interface, application

Description automatically generated

Text

Description automatically generatedPo vytvorení UI prvkov som potreboval začať nejako ťahať tie údaje do mojich vytvorených ciell. Údaje o receptoch sa ťahajú cez api, problém je však že väčšina Receptových api je buď platená alebo nie moc optimálna, preto som sa rozhodol pre recipesapi.herokuaopp.,com, ktoré sú student friendly a netreba za ne platit. Je to však na margo množstva a kvality aké ponúkajú. Keď som si teda už vybral stránku, z ktorej budem api ťahať, potreboval som si pripraviť model, preto som si pridal niekoľko viewControllerov konkrétne CategoryViewController, FoodViewController a FoodDetailViewController (ano, posledné dve sú veľmi originálne ale bohužial niesom umelecká duša na vytváranie názvov). Začal som teda prípravou tohto screenu, kde som využil TableView na prezentovanie možných typov kategórii, jednotlivé názvy kategórii sa teda uložia do TableViewCell a zobrazia na screene. Teraz som potreboval riešiť problém, ako jednotlivé kategórie dotiahnúť z apička. Vytvoril som si preto manažéra, ktorý bude mať na starosti všetky tieto api volania triedu RequestManager, ktorý pomocou balíčku alamofire vie robiť jednotlivé cally.

Text

Description automatically generatedKeď už mám metódu, ktorá ťahá api, potrebujem ich teraz naloadovať do daného CategoryViewControllera. Naimplementoval som si na to vlastnú metódu, ktorá naťahané api vloží zoznamu, ktorý má daný Controller, v prípade, že sa api nenaťahali vypíše error.

Následne zobrazím celly na obrazovku, buď všetky kategórie alebo tie ktoré vyhovujú filtru = vyhľadávaniu.

Text

Description automatically generated

Podobne ťahám a zobrazujem aj konkrétne jedlá a konkrétne detaily jedál. Medzi jednotlivými screenami prechádzam pri kliknutí na jednotlivú tableViewCellu.

Text

Description automatically generated

Rovnako to funguje aj vo FoodViewControllery s navigáciou na novú screenu, jediný rozdiel že tu sa to prezentuje modálne, teda iba vyskakovacie okno a vo FoodViewControllery prekrytim.

Po implementovaní ťahania api som začal implementovať nákupný zoznam. Pridávanie nákupných položiek som spravil cez alerty, nakoľko mi prišlo zbytočné vytvárať celý nový screen na obyčajné pridanie prvku. Keď používateľ klikne na tlačítko plus tak vyskočí alert o pridaní nového predmetu do zoznamu. Je tam textfield, do ktorého užívateľ napíše prvok, ktorý potrebujem nakúpiť dajme tomu 10kg ryže. Tento string sa mu uloží do pamäti UserDefaults.

Text

Description automatically generated

Prvok môžeme odstrániť ikonickým swipe gesture

Text

Description automatically generated

Pre pridávanie prvkov do UserDefaults som si musel vytvoriť opäť vlastného manažéra ShopListManager, ktorý pridávané prvky kóduje do JSON a následne ich podľa kľúča uloží do UserDefaults. Manažér sa stará o pridávanie prvkov, ukladanie prvkov, načítavanie prvkov a odstraňovanie prvkov.

Text

Description automatically generated

Text

Description automatically generatedAko posledné som implementoval históriu posledných navštívených receptov. Pre túto funkcionalitu som si opäť musel vytvoriť vlastného manažéra, ktorý ukladá objekty do UserDefaults. Manažér dokáže ukladať objekty, loadovať ich a pridávať ich, svojvolnú funkcionalitu odstránenia som sem nepridal nakoľko som nemal zámer spraviť nejaké favorites, ale práve históriu, objekty sa odstráňujú vlastne pri pridávaní, ak by prekročili určitý počet prvkov, v mojom prípade som to nastavil na 5 prvkov.

Objekty histórie loadujem do HomeViewControllera, kde sa ukladajú do tableViewCell. Po zobrazení cell na ne môžme kliknúť a zobrazia nám opäť konkrétny detail jedla.

Aplikácia je pripravená na lokalizáciu, má nachystané prvky, v ktorých text môžeme lokalizovať do angličtiny.

# Použité zdroje:

<https://www.hackingwithswift.com/100>  
<https://developer.apple.com/documentation/uikit>  
https://recipesapi.herokuapp.com/  
Konzultácie v GoodRequeste