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

**Vývoj aplikácií pre mobilné zariadenia**

Semestrálna práca

Matúš Mištrik 5ZYI23

Školský rok 2021/2022

# Špecifikácia zadania

Ako tému na semestrálnu prácu som si vybral vytvorenie aplikácie To-Do list. Ide o jednoduchú aplikáciu, ktorej cieľom je pomôcť užívateľovi si rozvrhnúť plánované úlohy, ktoré musí alebo chce v konkrétny deň stihnúť urobiť.

# Prehľad aplikácii podobného zamerania

Graphical user interface, text, application, chat or text message

Description automatically generatedGraphical user interface, application, Teams

Description automatically generated with medium confidenceGraphical user interface, application, Teams

Description automatically generatedGraphical user interface, text, application, chat or text message

Description automatically generatedNa trhu sa nachádza obrovské množstvo aplikácii s podobným zameraním. Medzi najznámejšie určite patria To Do od Microsoftu alebo TickTick. Obidve aplikácie podporujú základné ovládacie prvky, ako sú: pridávanie, modifikácia a mazanie úloh, značenie dokončených úloh a pod. Zopár obrázkov z aplikácii (zľava prvé 2 obrázky To Do, sprava TickTick):

Osobne mi príde zložité pri týchto aplikáciách si zvoli dátum, alebo nejaký popis k mojej úlohe (dá sa to, ale človek sa tam musí preklikať). V oboch prípadoch po pridaní úlohy sa úloha zaradí medzi ostatné podľa dátumu, ale vizuálne ich neoddelí nejakou kategóriou, takže to pôsobí trošku neprehľadne. Ba čo viac, v To Do sa úloha nedá vyhľadať (teda aspoň som nenašiel), čo mi príde, ak má človek veľa úloh, celkom dôležitá funkcia.

Presne na tieto veci som sa zameral ja v mojej aplikácii.

Moja aplikácia sa líši od hore uvedených v týchto bodoch:

* Úlohy sú utriedené podľa dátumu, pričom sú zobrazené pod kategóriami, pekne prehľadne, tak ako to má byť
* Úlohy sa dajú vyhľadať (v inej obrazovke), odkiaľ sa dajú tiež modifikovať, alebo mazať
* Užívateľ si môže vytvoriť šablóny na úlohy, ktoré sa mu často opakujú, napríklad ak si často značí, že musí niečo zaplatiť, tak si vytvorí šablónu a potom pri vytváraní, alebo modifikácii, si ju môže vybrať a automaticky mu to vyplní políčka pre meno a popis úlohy.
* Haptický feedback po stlačení hocijakého tlačidla
* Graphical user interface, text, application, email

  Description automatically generatedGraphical user interface, application

  Description automatically generatedGraphical user interface

  Description automatically generated with medium confidenceJednoduchý dizajn, v ktorom sa človek nestratí

Zopár obrázkov z mojej aplikácie:

# Analýza navrhovanej aplikácie

## Krátka analýza:

Používateľ môže: - pridať úlohu, kde je: - zadanie názvu  
 - zadanie popisu  
 - zvolenie dátumu  
 - výber zo šablón  
 - modifikovať úlohu, kde je: - zadanie názvu  
 - zadanie popisu  
 - zvolenie dátumu  
 - výber zo šablón  
 - vymazanie úlohy  
 - označenie, či je (ne)splnená

- vymazať úlohu  
 - označiť za splnenú/nesplnenú  
 - pridať šablónu, kde je: - zadanie názvu  
 - zadanie popisu  
 - upraviť šablónu, kde je: - zadanie názvu  
 - zadanie popisu  
 - vymazanie šablóny  
 - vymazať šablónu  
 - vyhľadanie úlohy, odkiaľ môže: - upraviť úlohu  
 - vymazať úlohu  
 - označiť za (ne)splnenú

# Návrh aplikácie

# Popis implementácie

Na začiatku som si vytvoril model, ktoré budú predstavovať jednotlivé úlohy. Pre prácu s úlohami, ako napr. ukladanie, pridávanie a pod., som si vytvoril manažéra, ktorý to bude mať na starosti. Nakoľko zoznam úloh v aplikácii bude len jeden, tak som tohoto manažéra dal ako singleton. Nakoľko sa na takýto projekt neoplatí požívať nejakú relačnú databázu, rozhodol som sa ja po konzultácii s prednášajúcim so stáže využiť lokálnu databázu UserDefaults. Aby som predišiel komplikáciám pri ukladaní objektov do tejto databázy, musel som objekty pred uložením zakódovať do JSON. Pri načítavaní z databázy sa musí zase dekódovať.

Text

Description automatically generated

Na zobrazovanie úloh som si vytvoril novú obrazovku (storyboard), kde som si ako hlavný view zvolil TableView, čo mi zabezpečí, že môžem si pohodlne dopĺňať celly a budú stále pekne usporiadané. Nakoľko som si vzhľad mojej celly chcel prispôsobiť, tak som si aj pre ňu vytvoril osobitný xib súbor, kde som si ju namodeloval podľa vlastných predstáv. Cella je navrhnutá tak, aby svoju veľkosť dynamicky menila. To znamená, že ak sa zadá príliš dlhý názov úlohu, tak sa dá na dva riadky a cella sa zväčší.

Graphical user interface, application

Description automatically generated

Prvý problém však nastal, keď som chcel zachytiť v hlavnom ViewControlleri, či bolo stlačené tlačidlo v celle. Na vyriešenie tohto problému som si vytvoril vlastného delegáta pre danú cellu, vďaka ktorému mi bola požadovaná funkcia umožnená.

Text

Description automatically generated

Ďalší problém nastal pri volaní druhého okna, ktoré sa malo zobraziť modálne. V tomto prípade išlo o problém, že po vrátení (ukončení okna) mi nezavolalo metódu viewDidAppear, kde som si obnovoval stav tableView. Tu som musel zdediť metódu od správcu animácii, ktorá tento proces registrovala.

Text

Description automatically generated

Čo sa týka šablón, implementačný postup bol veľmi podobný, ako pri ukladaní úloh.

Pri implementácii vyberania si šablón pri vytváraní alebo modifikácii úloh sa znova vyskytol problém, ako dostať z obrazovky dáta zo šablóny, na ktorú užívateľ klikol, naspäť na hlavnú obrazovku. Tu by sa mali vyplniť do textových políčok. Jedným z riešení bolo urobiť si znova delegáta. Ale keďže ja som človek lenivý, rozhodol som sa vyskúšať druhý spôsob, kratší na naprogramovanie. A to sú closures. Ešte som nevidel jazyk, kde by sa toľko využívali tie lambdy, ako vo swifte. Ale je to magické. Tu je príklad:

Najprv som si closure zadefinoval ako atribút obrazovky so šablónami.  


Do metódy, ktorá sleduje interakcie s cellami som si pridal túto closure, kde som si dal ako parameter šablónu s daným indexom.

Text

Description automatically generated with medium confidence

V hlavnej obrazovke, ktorá volala túto, som už len zavolal closure s týmto menom

Text

Description automatically generated

Ešte som implementoval aj haptickú odozvu, ktorého implementácia bola prekvapivo jednoduchá





Aplikáciu som dokonca lokalizoval do slovenčiny a angličtiny.

# Použité zdroje:

<https://developer.apple.com/documentation/uikit>

https://www.hackingwithswift.com/100

<https://cocoacasts.com/>

Maroš Novák – konzultácie ☺