



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

APLIKACE PRO VYHLEDÁVÁNÍ A POROVNÁVÁNÍ PRODUKTŮ NA IPAD

APPLICATION FOR SEARCH AND COMPARISON OF PRODUCTS FOR IPAD

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

PETR JAŠÍČEK

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. VÍTĚZSLAV BERAN, Ph.D.

BRNO 2016

Abstrakt

Cílem bakalářské práce je navrhnout grafické uživatelské rozhraní mobilní aplikace na zařízení iPad, která bude sloužit obchodníkům firmy GÜDEL a.s. pro zlepšení efektivity prodeje planetových převodovek. Návrh aplikace postupoval podle metodologie User-centered design. Řešení bylo vytvořeno na základě prototypu otestovaného obchodníky firmy GÜDEL a.s.

Abstract

The aim of the project is to design graphical user interface for mobile application on iPad which will aid salesmen from company GÜDEL a.s. in improving effectivity of selling planetary gearboxes. Design of the application followed User-centered design methodology. Application was based on prototype tested by salesmen from company GÜDEL a.s.

Klíčová slova

iOS, iPad, Mobilní aplikace, User Experience, UCD, Objective C

Keywords

iOS, Mobile application, iPad, User Experience, UCD, Objective C

Citace

Petr Jašíček: Aplikace pro vyhledávání a porovnávání produktů na iPad, bakalářská práce, Brno, FIT VUT v Brně, 2016

Aplikace pro vyhledávání a porovnávání produktů na iPad

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Vítězslava Berana, Ph.D.

Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Petr Jašíček
17. května 2016

Poděkování

Tímto bych rád poděkoval mému vedoucímu bakalářské práce, panu Ing. Vítězslavu Beranovi, Ph.D za poskytnutou podporu a odborné vedení.

© Petr Jašíček, 2016.

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

1 Úvod	2
2 Teoretický úvod	3
2.1 User-centered design	3
2.2 Uživatelské testování	9
2.3 Úvod do programování v iOS	11
2.4 Existující řešení	15
3 Návrh řešení	16
3.1 Definování problému	16
3.2 Návrh grafického uživatelského rozhraní	19
4 Realizace, testování a vyhodnocení	25
4.1 Postup práce	25
4.2 Implementace	25
4.3 Testování a vyhodnocení	30
5 Závěr	34
A Obsah CD	36
B Testovací formulář	37

Kapitola 1

Úvod

V současné době dochází čím dál tím častěji k využívání informačních technologií pro obchodní účely. Můžou to být jednoduché programy pro správu e-mailů nebo telefonních čísel na osobní počítače, ale také komplexní aplikace na mobilní zařízení vyvinuté přímo pro zákazníka na základě jeho potřeb pro podporu prodeje.

Cílem této práce je navrhnout aplikaci na iPad sloužící k podpoře prodeje planetových převodovek firmě GÜDEL a.s. a následně aplikaci podle návrhu vyvinout. Uživatelé budou schopni pomocí aplikace vyhledat nejbližší planetové převodovky, které vyhovují zadaným vstupním parametrům. U vybrané konkrétní převodovky si budou uživatelé schopni otevřít její produktový katalog, 3d model a budou schopni vybranou převodovku nakonfigurovat a přímo z aplikace vytvořit nabídku zákazníkovi pomocí vygenerovaného e-mailu. Dále budou mít uživatelé možnost procházet si všechny dostupné planetové převodovky firmy GÜDEL a.s. Aplikace bude také umožňovat zobrazení kontaktů na všechny pobočky firmy GÜDEL a.s. po celém světě.

V Kapitole 2 se nachází teoretický úvod do problematiky moderního návrhu aplikací se zaměřením na uživatele, seznámení s technologií použité k vývoji aplikace a popsání existující řešení.

Kapitola 3 sestává ze samotného návrhu aplikace zahrnující definování problému, který aplikace bude řešit, definování budoucích uživatelů a uživatelských požadavků a následný popis postupu návrhu grafického uživatelského rozhraní aplikace.

Poslední kapitola 4 je věnována způsobu implementace klíčových prvků aplikace a popisu testování na uživateli.

Kapitola 2

Teoretický úvod

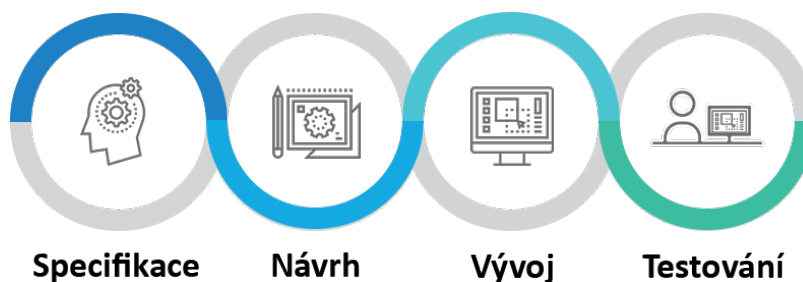
Návrh softwaru je proces implementování softwarového řešení k jednomu nebo více problémům. Důležitou částí návrhu softwaru je analýza požadavků na software. Pokud je vyvíjený software "polo-automatický" nebo na uživatele zaměřený, návrh softwaru může zahrnovat návrh UX (User Experience), na jehož základě se vytvoří prototyp, který pomůže zjistit dané požadavky na software. Pokud je software kompletně automatický (tj. nemá uživatele nebo uživatelské rozhraní), návrh softwaru může být velice jednoduchý ve formě diagramu toku dat nebo ve formě slovního popisu plánovaných sekvencí událostí.

V dnešní době existuje mnoho metodologií vývoje softwaru, v této práci se zaměřím na vývoj softwaru zaměřený na uživatele (User-centered design = UCD).

2.1 User-centered design

User-centered design (UCD) je návrhová metodologie ve které je kladen důraz na požadavky, potřeby a limitace koncových uživatelů tak, že jsou promítnuty do všech etap vývoje [4]. Produkty vyvinuté za použití UCD metodologie jsou optimalizované pro koncové uživatele a důraz je kladen na to, jak koncový uživatel chce nebo potřebuje používat produkt namísto přinucení koncového uživatele používat tak, jak byl produkt navržen, proti filozofii uživatele.

User-centered design je běžná metodologie v softwarovém vývoji. Proces UCD je většinou rozdělen do čtyř hlavních částí životního cyklu softwaru, který je zobrazen na obrázku 2.1.



Obrázek 2.1: Typický UCD cyklus¹

¹Převzato z <http://colossom.com/user-centric-design.html>

2.1.1 Specifikace kontextu použití

Specifikace kontextu použití [4] se dá definovat jako identifikování uživatelů, kteří budou produkt používat, k jakému účelu ho budou používat a v jakých situacích ho budou používat.

K získání těchto znalostí se v praxi využívají jednotlivé techniky, mezi které patří:

- tvorba a údržba projektového plánu,
- definování uživatelů za využití tvorby person,
- sběr uživatelských požadavků,
- definování účelu použití produktu v určitých situacích pomocí tvorby scénářů.

Projektový plán

Projektový plán [4] bere v úvahu přístup, jakým se tým vydá a pomůže týmu dokumentovat rozhodnutí týkající se cíle, rozsahu, plánování a prostředků.

Rozsah projektu definuje čeho je potřeba dosáhnout pro úspěšné splnění projektu. Při diskutování rozsahu je důležité definovat:

- Jaký produkt se vyvíjí?
- Jak rozsáhlý bude produkt?
- Vytváří se produkt pro konkrétního zákazníka nebo pro běžné uživatele?

Je tedy nutné si definovat jasně měřitelné cíle, kterých se při vývoji produktu snažíme dosáhnout. Když se jasně definují měřitelné cíle, je možné vyhodnotit úspěšnost při dokončení projektu.

Persony

Persona [4] je fiktivní osoba, která zosobňuje reálné uživatele. Persona si připomínat, pro koho je vyvíjený produkt určen. K vytvoření person je nutné uživatelům pokládat typově následující otázky:

- Jaký je váš oblíbený produkt? Proč je lepší než ostatní podobné produkty?
- Používání jakého produktu vás nejvíce frustruje? Proč? Jak byste ho vylepšili?
- Pokud byste měl vytvořit aplikaci, který vám má pomoci s Vaším úkolem, jak by vypadala? Co by měla dělat?

Těmito otázkami se dá získat lepší pochopení, co je uživatelům příjemné a co je motivuje produkt používat. Se získanými informacemi od uživatelů se vytvářejí persony. Person by měla obsahovat následující položky:

- Jméno
- Věk

- Rodinný stav
- Bydliště
- Zaměstnání
- Záliby a oblíbené věci
- Potřeby
- Slasti a strasti

Tvorba person může být důležitá zejména ve situacích, kde je určitá vzdálenost mezi návrhářem a uživatelem. Klasickým příkladem je vývoj aplikace pro hromadný trh na některý ze smartphonů.

Uživatelské požadavky

Uživatelské požadavky [4] se dají definovat jako to, co uživatel od vyvíjeného produktu potřebuje.

Sbírání uživatelských požadavků je jednou z nejdůležitějších částí UCD. Uživatelské požadavky jsou základem celého návrhového procesu, je nemožné jít správným směrem v návrhu, aniž by byly uživatelské požadavky jasně definované.

Je vhodné si tento proces dokumentovat - např. při interview si postupně psát shrnutí, jaké požadavky uživatel má a poté tuto dokumentaci uživateli ukázat a ujistit se, že jde opravdu o to, co tím uživatel myslel.

Uživatelské požadavky nejsou technické, během fáze sbírání uživatelských požadavků se nemusí napsat jediný řádek kódu.

Scénáře

Scénáře [4] jsou krátké příběhy, které reflektují situace, ve kterých se můžou uživatelé vyskytnout. Ve scénáři se zaměřujeme na to, jak vyvíjený produkt pomůže uživateli v dané situaci.

Tvorbou scénářů se tedy jasně definuje účel použití vyvíjeného produktu i situace, ve kterých budou uživatelé produkt používat.

2.1.2 Návrh

Návrh je druhou hlavní částí životního cyklu produktu, předchází samotné implementaci. Cílem návrhu je mít zpracované veškeré podklady o které se bude opírat samotný vývoj, tj. mít připravené modely databáze, prototyp produktu a data nebo workflow modely.

Prototypování

Prototypování [7] je proces vytváření málo či více kvalitních mockupů návrhu aplikace za účelem mít něco hmotného pro testování na uživatelích, aniž bychom museli už psát kód. Prototypování je velice mocný nástroj pro vizualizaci návrhu aplikace budoucím uživatelům.

I když prototypování zabere čas, má obrovské výhody. Prototypování může pomoci ušetřit mnoho hodin přímého vývoje aplikace, který ve výsledku nefunguje nebo výsledek není takový, jaký koncový uživatelé chtějí.

Když se přistupuje k vývoji aplikace formou takovou, že se začne ihned psát kód, často se stává, že se častěji hledají řešení, které více sedí kódu než uživateli. Prototypování uvolňuje od závazku myslet na implementační kód a zaměřuje se na interakci mezi uživatelem a vyvíjenou aplikací.

Rozlišujeme tři hlavní výstupy prototypování: Wireframy [7], Mockupy [7], Prototypy [7].

Wireframe

Wireframe je statická nepřesná reprezentace návrhu. Wireframe by měl jasně obsahovat:

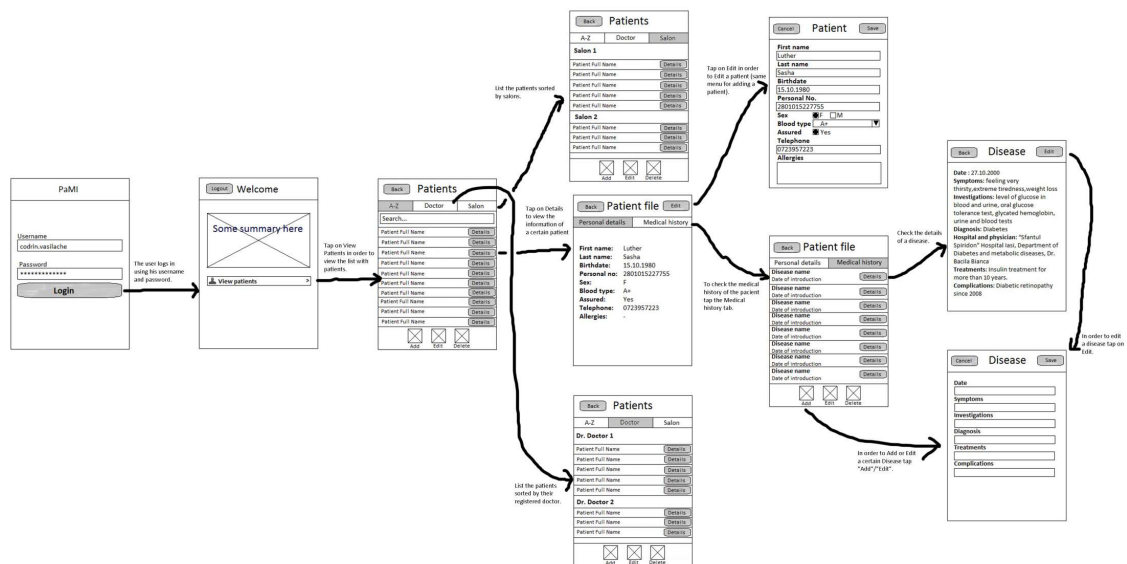
- hlavní skupiny obsahu (co?),
- strukturu (kde?),
- popis a práci s rozhraním (jak?).

Wireframy jsou základní stavební kameny návrhu a měly by obsahovat všechny důležité části výsledného produktu.

U wireframů není vhodné jít do úplných detailů, ale je nutné, aby nechyběly žádné důležité části. Wireframe by se dal přirovnat k mapě města - každá ulice je na mapě zobrazena, ale je zjednodušena.

Samotné vytváření wireframů by nemělo zabrat mnoho času, většina času by měla být strávena komunikací s ostatními členy týmu nebo se zákazníkem.

Wireframy se typicky používají v projektové dokumentaci. Jsou statické a měly by být krátce popsány. Nejsou určeny pro testování na uživateli, ale dá se na nich získat zpětná vazba pro další návrh. Typický wireframe mobilní aplikace je zobrazen na obrázku 2.2.



Obrázek 2.2: Wireframe²

²Převzato z <https://patientmo.wordpress.com/category/wireframes/>

Mockup

Mockup je statická středně - vysoce přesná reprezentace návrhu. Velice často je mockup používán pro vizualizaci výsledného vzhledu aplikace. Mockup by měl obsahovat:

- reprezentaci struktury aplikace,
- vizualizaci obsahu,
- statickou demonstraci základní funkcionality.

Mockupy jsou zejména důležité, pokud je potřeba předvést vizuální návrh pro získání zakázky.

Díky jejich vzhledu jsou mockupy na pohled profesionálnější než wireframy a také mnohem rychlejší na vytvoření než prototypy. Jsou dobrým zdrojem získávání zpětné vazby a je dobré je mít v projektové dokumentaci. Typický mockup mobilní aplikace je zobrazen na obrázku 2.3.



Obrázek 2.3: Mockup³

Prototyp

Prototyp je středně až vysoce přesná reprezentace výsledného produktu, který simuluje uživatelské rozhraní. Umožňuje uživatelům:

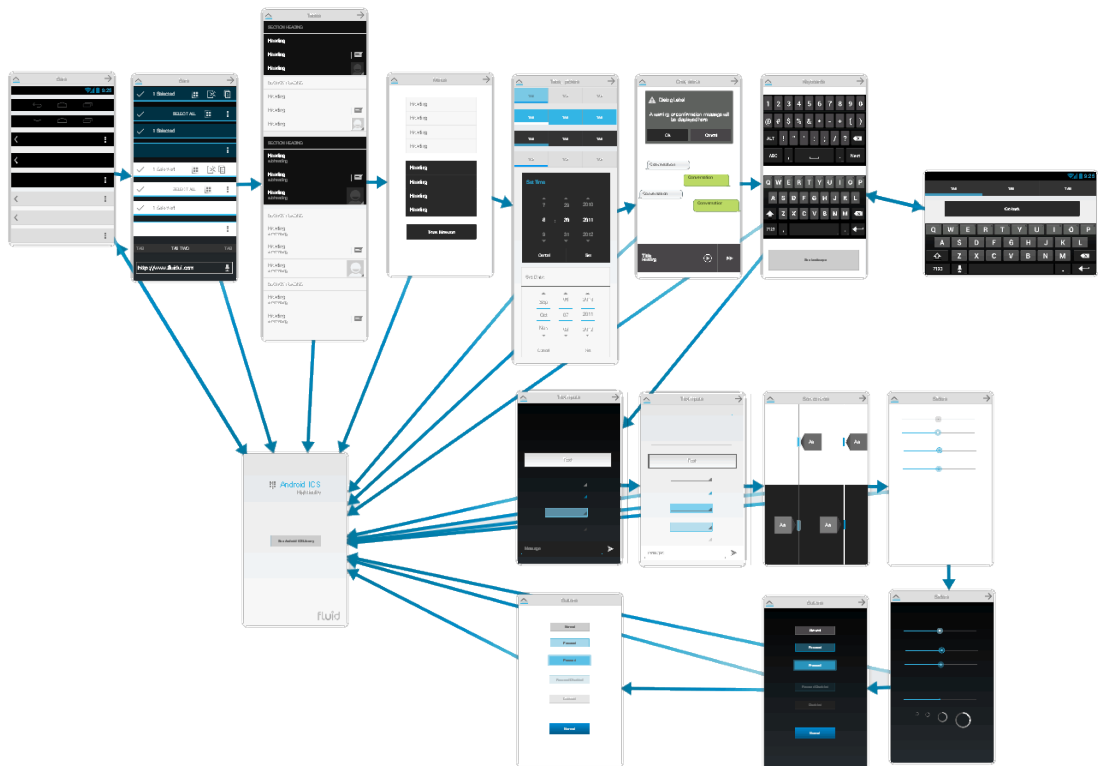
³Převzato z <http://husk.eecs.berkeley.edu/courses/cs294-84-fall12/index.php/P3-Drinke>

- projít si strukturu aplikace,
- pracovat s uživatelským rozhraním,
- testovat nejdůležitější prvky aplikace podobně jako u výsledného produktu.

Prototyp simuluje finální pracování uživatele s uživatelským rozhraním aplikace. Nemusí vypadat naprosto stejně jako výsledná aplikace, ale měl by být z velké části hodně podobný. Návaznost backendu na uživatelské rozhraní se většinou nedělá - šetří to čas i peníze ve vývoji.

Prototypy se používají zejména pro uživatelské testování. Dají se tak hodně dobře získávat informace k ověření použitelnosti rozhraní před tím, než vůbec začne fáze implementace.

Vytváření prototypů je poměrně drahé a časově náročné, ale spolu s následujícím uživatelským testováním se tvorba prototypů v mnoha případech značně vyplatí. Typický prototyp mobilní aplikace je zobrazen na obrázku 2.2.



Obrázek 2.4: Prototyp⁴

Shrnutí typů prototypování

⁴Převzato z <http://wireframes.linowski.ca/2012/08/fluid-ui-mobile-app-prototyping/>

	Přesnost reprezentace	Cena, čas	Využití	Popis
Wireframe	Malá	\$	Dokumentace, získání uživatelských požadavků	Černobílé skicy, reprezentace UI
Mock-up	Střední - vysoká	\$\$	Získání zpětné vazby, získání zakázky	Statická vizualizace návrhu
Prototyp	Střední - vysoká	\$\$\$	Uživatelské testování, někdy znovupoužitelný základ pro implementaci	Interaktivní

Tabulka 2.1: Shrnutí prototypování

2.2 Uživatelské testování

Uživatelské testování je měřené pozorování chování uživatelů používající testovaný produkt [4]. Uživatelské testování je založeno na měřeních a datech pro potvrzení předpokladů o vyvíjeném produktu.

Obecný cíl uživatelského testování je změřit efektivitu funkcí nebo sad funkcí vyvíjeného produktu. K tomu je zapotřebí určit metriky jako čas potřebný ke splnění určitých úkonů nebo maximální počet chyb při plnění úkonu. Testování může také být kombinováno s dotazníkem ke zjištění špatně pozorovaných věcí, jako míra je míra spokojenosti prací s produktem.

K provedení uživatelského testování je za potřebí vytvořit plán.

Testovací plán

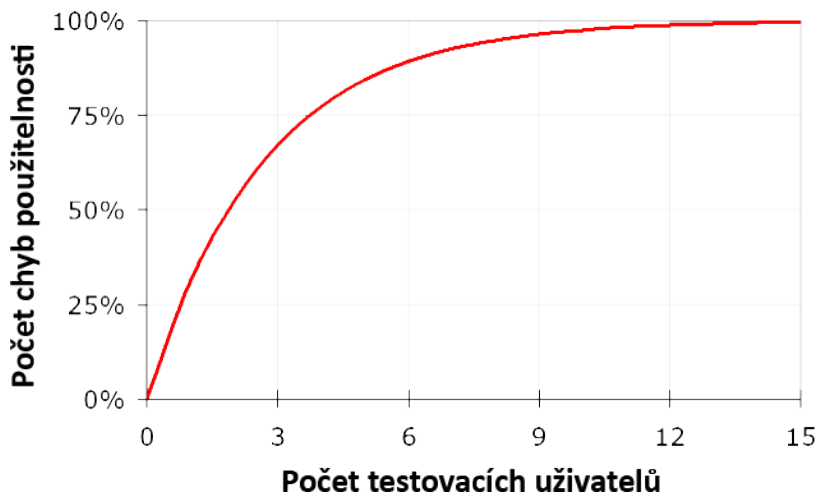
Před provedením uživatelského testování je nutné vědět co je vůbec potřebné otestovat a definovat, kdo je správný testovací subjekt. Dále je nutné vytvořit samotný testovací scénář s dílčími pokyny, což zvýší konzistenci výsledků uživatelského testování.

Testovací scénář by měl obsahovat následující části [4]:

- **Představení:** Je nutné představit koncept testování a jeho účel.
- **Ujištění:** Některé osoby mohou být nervózní. Je vhodné ujistit, že se testuje aplikace, nikoliv osoba, která jej testuje.
- **Testovací vodítka:** Testovací vodítka pomáhají definovat jak bude testování probíhat. Provede se nejdříve školení osob? Budou se moci osoby ptát během testování? Tyto otázky by měly být pokryty v testovacích vodítkách.
- **Úkoly:** Úkoly jsou základem uživatelského testování. Výstupem testování zadaných úkolů jsou konkrétní data. například můžeme měřit čas potřebný pro změnu profilového obrázku nebo počet vyhledávání pro nalezení vhodné obuvi.
- **Závěr:** Vyhradit si čas na konci testování pro odpovězení případných otázek.
- **Poděkování:** Je nutné vyhradit si čas na poděkování osobám, které se podíleli na uživatelském testování.

2.2.1 Prerekvizity

Nejnutnější prerekvizitou k uživatelskému testování je mít samotné testovací subjekty, tedy osoby, které budou dle testovacího plánu provádět testování produktu. Ideální počet testovacích subjektů neexistuje, ale obecně platí [4], že při testování na 5 až 10 subjektech se objeví většina problémů, jak je zobrazeno na obrázku 2.5.



Obrázek 2.5: Počet testovacích subjektů na počet objevených problémů⁵

Mimo testovací subjekty je nutné mít další věci:

Stopky

Pokud je výsledkem uživatelského testování čas, který uživatelům trvá splnit zadané úkoly, je nutné použít stopky nebo jiné zařízení pro měření času.

Poznámkový blok

Používání poznámkového bloku a pera je efektivnější než využití notebooku nebo jiného mobilního zařízení. Testovací subjekty mluví rychle a ideálním cílem je zdokumentovat veškeré chování testovacích subjektů. Rychle nakreslený symbol nebo diagram je mnohdy efektivnější než psaní na mobilním zařízení.

⁵Převzato z [4]

2.3 Úvod do programování v iOS

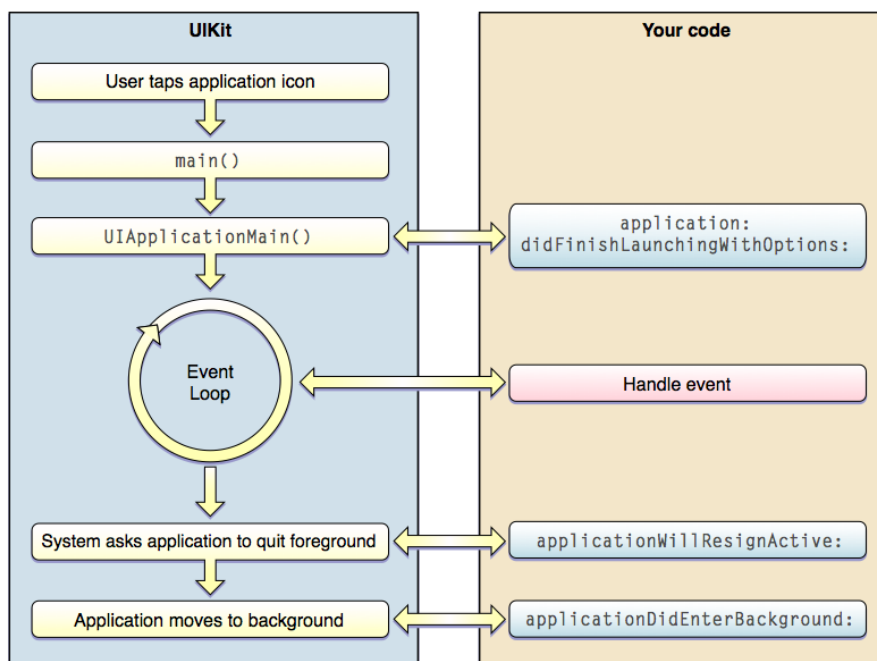
Pro programování aplikací pro platformu iOS⁶ je nutné splňovat určitá kritéria. Nejprve je nutné vlastnit zařízení se systémem OS X⁷. Dále, pokud předpokládáme vývoj nativní aplikace, je nutné mít vývojové prostředí Xcode⁸. Pro umístění vytvořené aplikace na Apple iStore nebo pro privátní distribuci je nutné vlastnit Apple vývojářský účet. V případě pouhého testování na simulátoru integrovaného v prostředí Xcode to není potřeba.

2.3.1 Základní vlastnosti iOS aplikací

Aplikace založené na iOS mají omezený přístup k souborovému systému [6]. Pro každou nainstalovanou aplikaci je v souborovém systému vytvořena samostatná část nazývaná sandbox, do které má aplikace přímý přístup. Aplikace má přímý přístup pouze do této části.

Všechny aplikace mají společný soubor nazvaný info.plist. Jedná se o strukturovaný textový dokument obsahující nezbytné informace o výsledné binární aplikaci jako minimální verze operačního systému, na kterém může aplikace fungovat, podporované natočení zařízení - vertikální nebo horizontální a jiné. Vývojář je zodpovědný za jeho správné vyplnění.

Všechny iOS aplikace mají společný životní cyklus reprezentován obrázkem 2.6, který začíná spuštěním aplikace a končí ukončením aplikace.



Obrázek 2.6: Životní cyklus iOS aplikace⁹

⁶[https://cs.wikipedia.org/wiki/iOS_\(Apple\)](https://cs.wikipedia.org/wiki/iOS_(Apple))

⁷https://cs.wikipedia.org/wiki/OS_X

⁸<https://cs.wikipedia.org/wiki/Xcode>

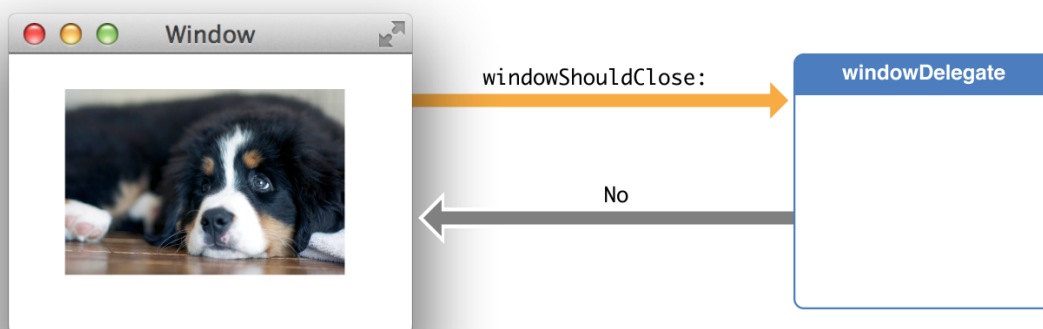
⁹Převzato z <http://www.brainvire.com/cross-platform-mobile-application-corporate-world/>

2.3.2 Delegace a protokoly

Delegace [1] je mohutně využívaný vzor využívaný v iOS aplikacích, který dokáže nahradit řadu případů nešikovného dědění. Princip delegace je jednoduchý: namísto toho, aby se třída sama o všechno starala prostřednictvím svých metod (jež její dědicové mohou reimplementovat), instance třídy obsahuje odkaz na spolupracující objekt, tzv. delegáta, s nímž nejrůznější akce prostřednictvím odpovídajících zpráv *konzultuje*.

Zásadní a klíčovou výhodou tohoto přístupu oproti dědění je to, že můžeme funkčně odlišné bloky kódu skutečně rozdělit do různých tříd.

Například když se chystáme zavřít okno, delegát je o stavu informován a vrací hodnotu typu BOOL zda může být okno zavřeno. Pokud tedy editujeme neuložený text a pokusíme se okno ukončit, může delegát vrátit hodnotu false a zobrazí výzvu k uložení dat.



Obrázek 2.7: Proces delegace¹⁰

2.3.3 Tvorba grafického uživatelského rozhraní

Nativní grafické uživatelské rozhraní pro iOS aplikace se vytváří obecně jedním ze tří způsobů:

- **iOS Storyboards** [8]: Grafický nástroj pro tvorbu uživatelského rozhraní využívající více pohledů a umožňující tvorbu přechodů mezi jednotlivými pohledy,
- **NIBs** [8]: Grafický nástroj pro tvorbu uživatelského rozhraní reprezentující jediný pohled.
- **Kód**: Tvorba uživatelského rozhraní pomocí programového kódu, tedy bez využití grafických nástrojů.

Obecně neexistuje nejlepší řešení. Volba optimálního řešení záleží na potřebách projektu.

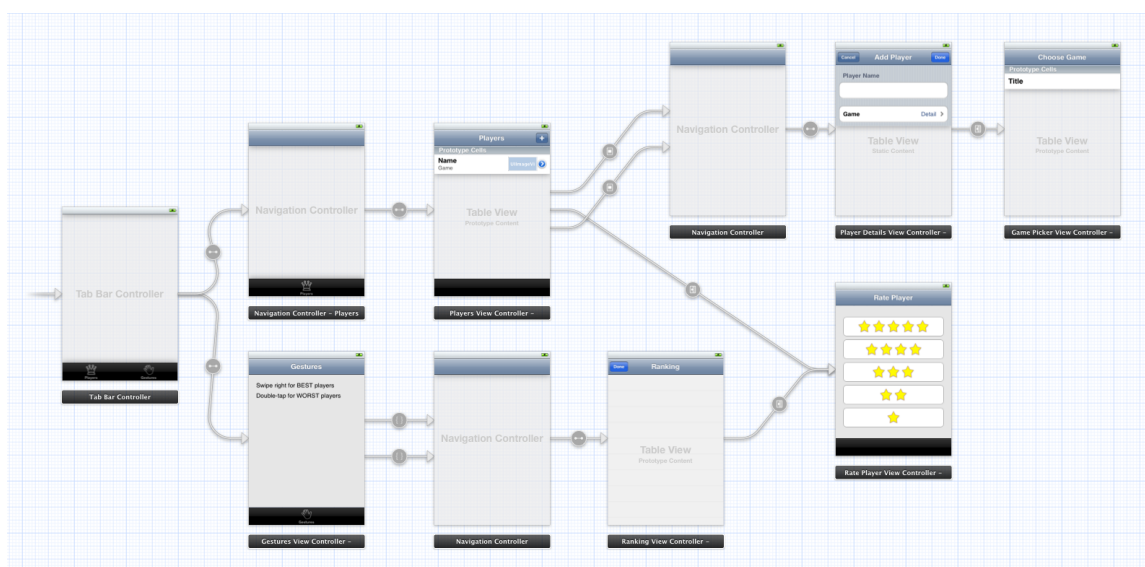
¹⁰Převzato z [6]

Storyboards

Storyboards jsou nejmladší ze zmíněných tří metod pro tvorbu uživatelského rozhraní. Jedná se o plátno, na kterém vytváříme vzhled jednotlivých pohledů a můžeme pohledy spojovat a tím tvořit přechody mezi nimi.

Hlavní výhodou storyboards je jejich vizuální explicitnost a tedy možnost jejich využití pro rychlou tvorbu prototypu uživatelského rozhraní aplikace - stačí vytvořit jednotlivé pohledy a minimum vlastního kódu.

Nevýhodou storyboards je jejich znovupoužitelnost. Pokud je pohled vytvořen ve storyboard svázaný s jinými pohledy, nemůže se pouze nakopírovat do jiného projektu jako samostatný pohled. Dále je složité aktualizovat storyboard v systému pro správu verzí, jelikož storyboard je ve skutečnosti XML dokument popisující všechny prvky, které obsahuje, což nevyhnutelně způsobuje slučovací konfliktky. Obrázek 2.8 obsahuje typický storyboard mobilní aplikace pro zařízení iPhone.



Obrázek 2.8: Storyboard¹¹

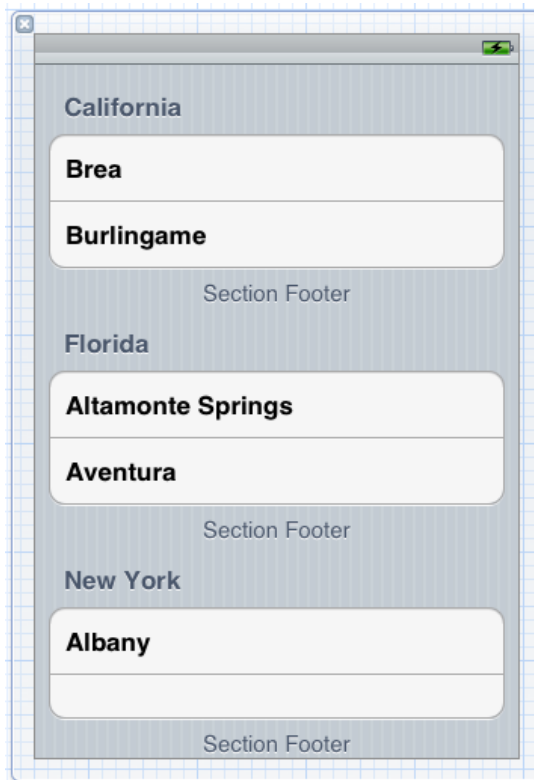
NIBs

NIBs jsou předchůdcem storyboards. NIBs se využívají pro tvorbu uživatelského rozhraní reprezentující jediný pohled, který je možné zobrazit v hlavním pohledu podle potřeby. Ukázka typického NIB pohledu je zobrazena na obrázku 2.9. Optimální využití NIBs zahrnuje řešení následujících uživatelských rozhraní:

- modální okna,
- jednoduché okna na registraci nebo přihlášení,
- nastavení,
- vyskakující okna,

¹¹Převzato z <https://www.raywenderlich.com/5138/beginning-storyboards-in-ios-5-part-1>

- znovupoužitelné pohledy.



Obrázek 2.9: NIB¹²

Kód

Veškerý návrh uživatelského rozhraní pomocí grafických nástrojů (storyboards, NIBs) se dá vždy udělat programaticky pomocí kódu. Některé problémy se také jinak než programaticky vyřešit nedají, například dynamické rozvržení, kde se jednotlivé elementy pohledu výrazně mění na základě stavů.

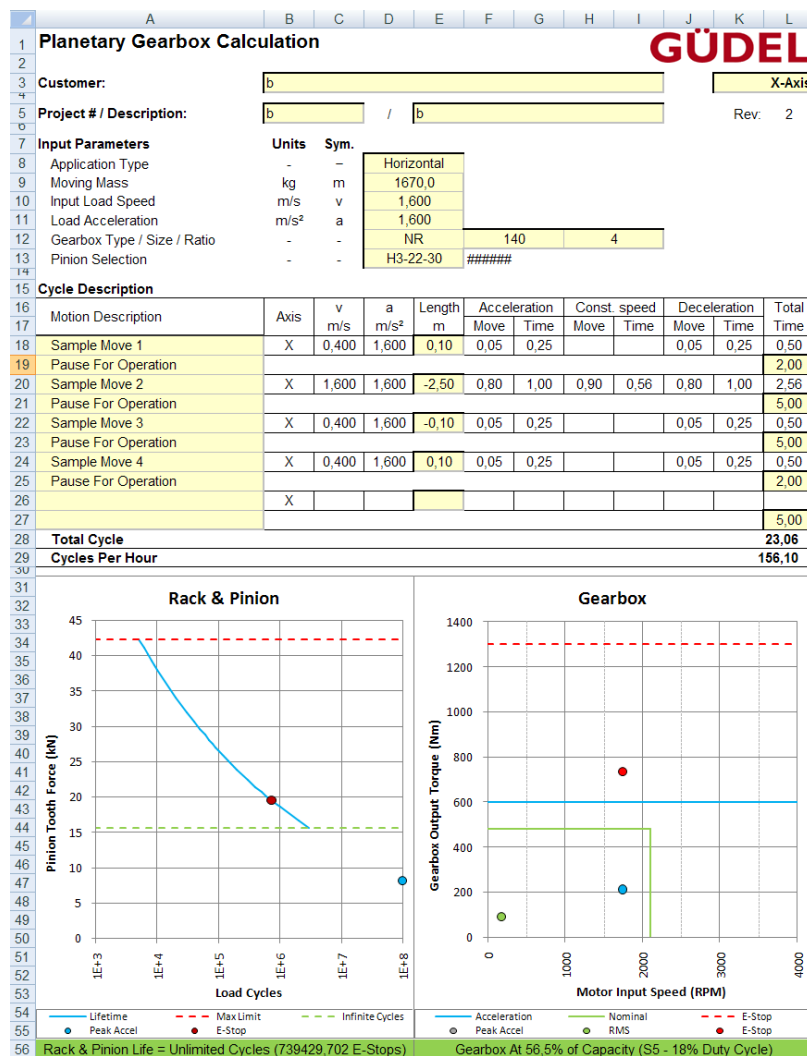
¹²Převzato z <http://gabriel-tips.blogspot.cz/2011/10/uitableview-display-and-hide-cells-as.html>

2.4 Existující řešení

Nejbližší řešení navrhovaného produktu existuje aktuálně v podobě excel sešitu s makry 2.10. Sešit umožňuje zkontrolovat, zda určitá vybraná převodovka vyhovuje zadaným vstupním parametrům aplikace¹³.

Toto řešení není dostatečné z několika důvodů:

- špatně využitelné, pokud obchodník komunikuje přímo se zákazníkem,
- nenavrhuje samo vhodnou převodovku, pouze kontroluje, zda zadaná převodovka vyhovuje vybraným parametrům,
- chybí další požadovaná funkcionalita.



Obrázek 2.10: Kalkulace převodovky GÜDEL

¹³ aplikace v tomto kontextu znamená řešení problému v reálném světě, tj. např. využití převodovky pro pohon břemena těžkého 1000kg se zrychlením 2m/s-2.

Kapitola 3

Návrh řešení

Návrh aplikace se řídil metodologií user-centered design, popsanou v kapitole 2.1. Na základě několika schůzek s budoucími uživateli tak byla provedena analýza a návrh vyvíjené aplikace. V této kapitole je definován problém, který má aplikace řešit, dále jsou definovány uživatelské požadavky kladené na aplikaci, definování uživatelé v podobě person a účely využití aplikace za pomoci scénářů.

Dále je zde popsán proces návrhu grafického uživatelského rozhraní, který se opírá o získané uživatelské požadavky.

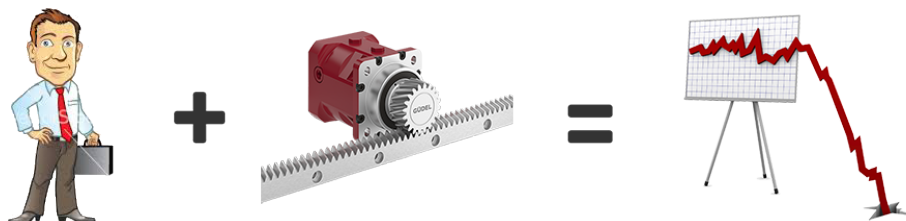
3.1 Definování problému

Firma GŮDEL a.s. koupila Francouzskou firmu, která vyrábí velice kvalitní planetové převodovky. Bohužel je velice obtížné tyto převodovky prodat zákazníkům a to ve většině případů proto, že obchodník firmy GŮDEL není schopný při návštěvě zákazníka přesně určit, jakou převodovku by mohl nabídnout pro zákazníkovi potřeby. Nejčastěji se obchodníci ocitají v následujících situacích:

- Obchodník je u zákazníka. Zákazník má nějaké fungující průmyslové řešení aplikace (např. automatizace výroby karoserií), kde využívá převodovky. Obchodník tedy chce na základě fungujícího řešení zákazníka mu nabídnout převodovku firmy GŮDEL a.s. a zákazníka přesvědčit, že převodovka firmy GŮDEL a.s. je lepší, kvalitnější apod. Ovšem nastává problém: Obchodník neví, jaká převodovka může na dané řešení sedět, takže zákazníkovi nemůže na místě ihned nic nabídnout, ale musí si sepsat parametry dané průmyslové aplikace a poslat dotaz do Francie, jaká převodovka by se na to hodila. Jenže obchodníkovi přijde odpověď až za několik dní a to už si sám zákazník skoro nepamatuje, že byl s obchodníkem na schůzce.
- Obchodník je na jednání s velkou firmou, která právě řeší návrh průmyslové aplikace. Firma prochází různé návrhy řešení, kde jsou různé požadavky na převodovku. Obchodník tak nemůže ihned reagovat.

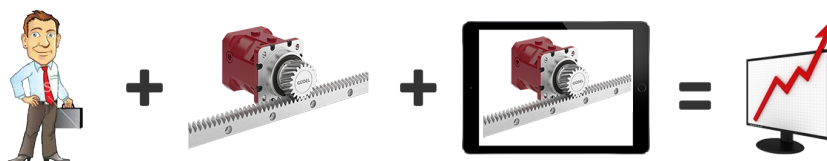
Výše zmíněné situace jsou ty nejdůležitější. Momentální stav prodeje nových převodovek zákazníkům je takový, že jsou bohužel neprodejné i když jsou velice kvalitní - kvalitnější, než většina převodovek využívaná zákazníky.

Cílem této práce je navrhnout a vyvinout takovou aplikaci, která obchodníkům umožní všechny výše popsané problémy efektivně a profesionálně vyřešit přímo na místě při schůzce



Obrázek 3.1: Aktuální stav

se zákazníkem a tím zvýšit prodejnost převodovek. Cíl je tedy dostat obchodníky ze stavu 3.1 do stavu 3.2.



Obrázek 3.2: Cílový stav

Persony

Persona, která představuje typického uživatele, který bude aplikaci využívat, byla vytvořena na základě schůzky s obchodníky firmy GÜDEL a.s.

- **Jméno:** Petr
- **Věk:** 49 let
- **Rodinný stav:** Ženatý
- **Zaměstnání:** Obchodník s patnácti letou praxí
- **Oblíbené věci:** iPad, iPhone, výkonné počítače
Potřeby: Petr má iPad neustále u sebe. Rád by ho využíval do plného potenciálu, tedy využívat jej v obchodních povinnostech, aby mu usnadnil a zefektivnil jeho práci.

Petr přešel z notebooku na iPad, protože u zákazníka je nepraktické vytahovat notebook a ukazovat zákazníkovi na počítači PDF katalogy, výkresy, případně CAD¹ modely. Mnohem raději by pro obchodní účely využíval iPad - je praktičtější a při prezentaci u zákazníka profesionálnější.

¹https://cs.wikipedia.org/wiki/Computer_aided_design

Uživatelské požadavky

Na základě vstupních rozhovorů s budoucími uživateli, obchodníky firmy GÜDEL a.s., vznikly uživatelské požadavky kladené na vyvíjenou aplikaci. Uživatelské požadavky jsou popsány v tabulce 3.1.

#	Popis uživatelského požadavku
1	Aplikace musí pracovat v režimu offline - bez potřeby přístupu k internetu
2	Uživatel aplikace bude mít možnost zadat parametry, které specifikují pracovní podmínky, ve kterých by navržená převodovka pracovala. To zahrnuje specifikaci následujících parametrů: Hmotnost břemena, rychlost a zrychlení, převodový poměr převodovky, typ aplikace (horizontální nebo vertikální), typ výstupu převodovky a specifikace pracovního cyklu.
3	Uživateli aplikace se zobrazí přehled převodovek, které zadaným parametrům nejlépe vyhovují i převodovky, které nesplňují některé potřebné parametry.
4	Po výběru konkrétní převodovky z přehledu navržených převodovek se uživateli zobrazí přehled zadaných parametrů a srovnání s parametry převodovky
5	Uživatel aplikace bude mít možnost procházet všechny typy GÜDEL převodovek
6	Uživatel aplikace bude mít možnost zobrazit katalog konkrétní převodovky
7	Uživatel aplikace bude mít možnost zobrazit kompletní katalog převodovek v jakémkoliv okamžiku při práci s aplikací
8	Uživatel aplikace bude mít možnost zobrazit 3D model převodovky
9	Uživatel aplikace bude mít možnost u vybrané převodovky provést její tzv. specifikaci - tj. určit typ vstupu / výstupu, délku výstupní hřídele apod.
10	Uživatel má možnost vygenerovat e-mail obsahující technické parametry převodovky a jeho odeslání
11	Uživatel bude mít možnost zobrazit fotogalerii převodovek
12	Uživatel aplikace bude mít alternativní možnost navigace v aplikaci pomocí stromově strukturovaného menu
13	Uživateli bude zobrazeno GÜDEL logo na všech stranách aplikace
14	Uživatel bude mít přístup ke kontaktům ke všem pobočkám firmy GÜDEL na celém světě

Tabulka 3.1: Uživatelské požadavky

Scénáře

Uvažujme typického uživatele Petra popsaného personou z podkapitoly 3.1.

- Petr je u zákazníka. Zákazník potřebuje pro nadcházející projekt automatizace výroby ocelových nožů planetové převodovky pro roboty pohybující se na lineárním vedení. Zákazník zná specifikace, které musí aplikace splňovat, tj převodový poměr, kroutící moment apod. Petr potřebuje najít nejbližší převodovku, která zadané parametry splňuje. Jak může Petrovi vyvíjená aplikace pomoci?
- Petr je u zákazníka. Zákazník má funkční převodovky pro své aplikace, ale Petrovi se zdá, že převodovky jsou předimenzované na danou aplikaci, tj. pro danou aplikaci by stačila o 1 nebo o 2 třídy slabší převodovka. Petr by chtěl zákazníkovi navrhnout lepší řešení dané aplikace. Jak může Petrovi vyvíjená aplikace pomoci?

- K Petrovi přijede zákazník. Zákazník by si rád prohlédl, jaké převodovky může Petr nabídnout. Rád by také viděl fotky nebo videa, kde jsou převodovky aplikovány. Petr chce mít všechny tyto věci po ruce. Jak může Petrovi vyvíjená aplikace pomoci?
- Petr je u zákazníka. Zákazník chce vidět reálné fotky převodovky a jejich využití v praxi. Jak může Petrovi vyvíjená aplikace pomoci?
- Petr je u zákazníka. Zákazník si vyžáda kontakt na hlavní pobočku ve Švýcarsku. Jak může Petrovi vyvíjená aplikace pomoci?

3.2 Návrh grafického uživatelského rozhraní

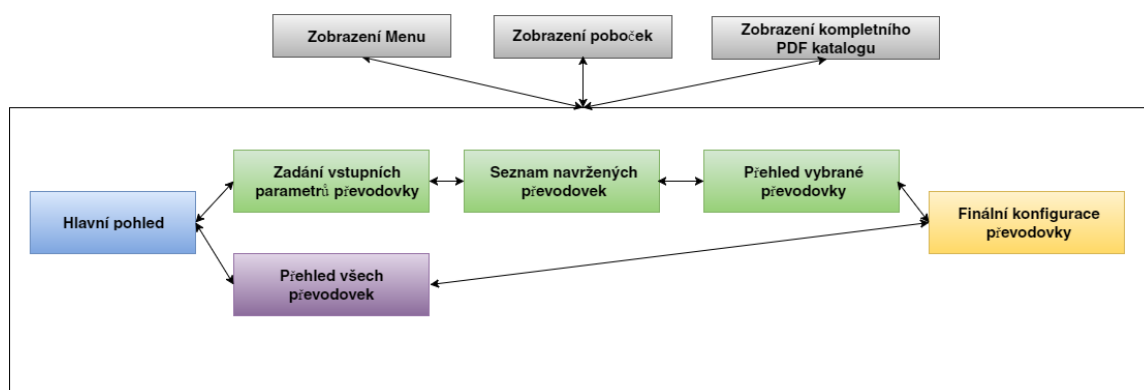
Vstupním bodem aplikace musí být hlavní pohled, který bude sloužit jako rozcestník pro hlavní funkční celky plynoucí z uživatelských požadavků:

- navržení převodovky ze zadaných parametrů, které převodovka musí splňovat,
- procházení dostupných GÜDEL převodovek

Dále jsou v aplikaci prvky, které musí být dostupné ve všech pohledech aplikace:

- zobrazení menu,
- zobrazení poboček,
- zobrazení kompletního PDF katalogu.

Diagram 3.3 vizualizuje přechody a závislosti mezi jednotlivými pohledy aplikace.



Obrázek 3.3: Relace mezi jednotlivými funkčními celky aplikace

3.2.1 Návrh převodovky ze zadaných parametrů

Proces vyhledání a návrhu převodovek se skládá z několika dílčích kroků, které by v aplikaci měly být funkčně odděleny - měly by být na samostatných pohledech:

1. zadání požadovaných parametrů kladených na převodovku,
2. výběr vhodné převodovky na základě seznamu nejbližších převodovek,
3. zobrazení detailu vybrané převodovky a přehled zadaných parametrů,
4. finální konfigurace vybrané převodovky.

Vyhledání převodovky ze zadaných parametrů

První uživatelský požadavek kladený na aplikaci je vyhledání vhodné převodovky na základě parametrů, které musí splňovat. GUI musí obsahovat prvky pro následující úkony:

1. zvolení a zadání vstupních parametrů: Typ aplikace převodovky, hmotnost přenášeného břemena, rychlost, zrychlení a převodový poměr převodovky,
2. zvolení typu výstupu převodovky: Pastorek nebo jiný,
3. definování pracovního cyklu převodovky,

Zvolení a zadání vstupních parametrů

Parametr typu aplikace může nabývat poute dvou předdefinovaných hodnot - Horizontální nebo vertikální. Pro volbu mezi dvěma předdefinovanými hodnotami je vhodné použít přepínací tlačítko, jehož využití je zobrazeno na obrázku 3.4 - z přepínacího tlačítka je patrné, jaká možnost se přepíná, jelikož obsahuje text.



Obrázek 3.4: Přepínací tlačítko

Hodnoty parametrů hmotnost přenášeného břemena, rychlosti a zrychlení nejsou předem známé, není tedy možné předdefinovat hodnoty. Uživatel aplikace tyto hodnoty bude muset ručně zadat. Tento účel nejlépe řeší textové pole zobrazené na obrázku 3.5, na které uživatel klikne a vyplní pomocí tlačítek na nabídnuté klávesnici.

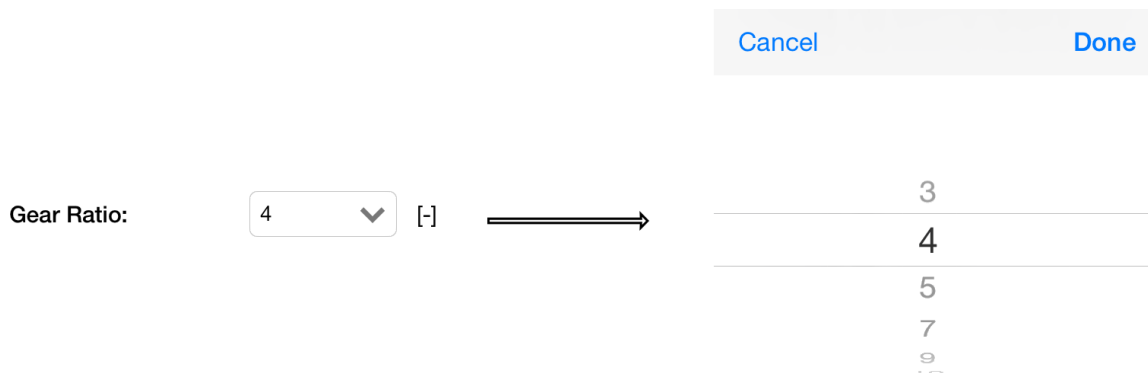
Moving Mass:	<input type="text" value="0"/>	[kg]
Input Load Speed:	<input type="text" value="0"/>	[m/s]
Load Acceleration:	<input type="text" value="0"/>	[m/s ²]

Obrázek 3.5: Textová pole

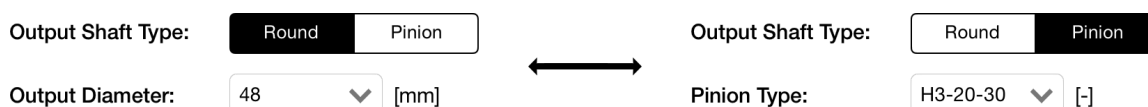
Převodový poměr je nutné zvolit z předem definovaných hodnot. Jelikož možných hodnot převodových poměrů je řádově desítky, není vhodné použít přepínací tlačítko. Řešení výběru z velkého počtu předdefinovaných hodnot se realizuje pomocí rozbalovacího menu, ze kterého si uživatel vybere požadovanou hodnotu. Vzhled a princip rozbalovacího menu je zobrazen na obrázku 3.6.

Volba typu výstupu převodovky

Převodovka může mít dva druhy výstupu: hřídel nebo pastorek. Na volbu mezi dvěma předdefinovanými typy je vhodné využít přepínací tlačítko. Pro výběr z velkého množství předdefinovaných hodnot je vhodné využít rozbalovací menu. Princip přepínání mezi výstupním typem hřídele a pastorkem je zobrazen na obrázku 3.7.



Obrázek 3.6: Rozbalovací menu



Obrázek 3.7: Výběr typu výstupu převodovky

Definice pracovního cyklu převodovky

Pro popis pracovního cyklu je nutné zadat délku operace po kterou bude převodovka břemeno převážet a čas potřebný na provedení operace. Z těchto zadaných hodnot a z hodnot definovaných ve vstupních parametrech převodovky se vypočítají doplňující informace o dílčích pracovních cyklech: Rychlost, zrychlení, dílčí časy pohybů. Počet pracovních cyklů je proměnlivý, nejčastěji jsou však tři až čtyři. Je tedy vhodné navrhnout řešení problému definování pracovního cyklu převodovky jako tabulku, jejíž řádky jsou pracovní cykly.

Buňky v tabulce, které je možné vyplnit je nutné graficky odlišit od buněk, které se automaticky vypočítají. Nabízí se tedy možnost realizovat vyplnitelné buňky jako textová pole, která budou mít odlišnou barvu od ostatních buněk. Vzhled navržené tabulky je zobrazen na obrázku 3.8.

Work Cycle Description

Motion Description	v [m/s]	a [m/s ²]	Length [m]	Acceleration		Const. Speed		Deceleration		Total Move Time	Pause for operation
				Move	Time	Move	Time	Move	Time		
Movement 1	1.60	1.60	2.50	0.80	1.00	0.90	0.56	0.80	1.00	2.56	2.00
Movement 2	0.89	1.60	0.50	0.25	0.56	0.00	0.00	0.25	0.56	1.12	5.00
Movement 3	1.60	1.60	-3.00	0.80	1.00	1.40	0.88	0.80	1.00	2.88	2.00



Obrázek 3.8: Tabulka pro definici pracovního cyklu

3.2.2 Vizualizace seznamu navržených převodovek

Dle uživatelského požadavku č. 3 je nutné vhodně vizualizovat převodovky nejlépe vyhovující i nevyhovující požadavkům vypočítaných ze zadaných parametrů.

Vypočítané parametry na jejichž základě se hledají nejbližší převodovky:

- pohotovostní kroutící moment,

- nominální kroutící moment,
- zrychlující kroutící moment,
- rychlost motoru,
- typ pracovního cyklu.

Je tedy vhodné zobrazit výše uvedené hodnoty do tabulky 3.9 symbolizující požadavky, které převodovka musí splňovat a zobrazit dva sloupce navržených převodovek:

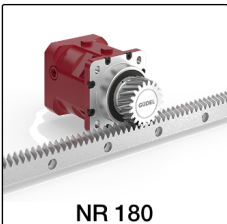
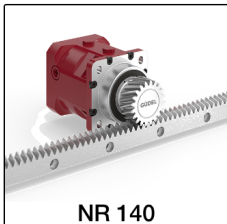
- převodovky, které nejlépe vyhovují minimálním požadavkům,
- převodovky, které nejlépe nevyhovují minimálním požadavkům

Každý prvek sloupce bude obsahovat tlačítko s názvem převodovky a grafy reprezentující míru splnění minimálních požadavků. Navržená vizualizace seznamu vyhovujících a nevyhovujících převodovek je zobrazena na obrázku 3.10.

Minimal gearbox requirements:

E-Stop torque:	1707.6 [Nm]	Motor Input Speed:	1440.0 [RPM]
Nominal torque:	272.9 [Nm]	Duty Type:	S5 [-]
Acceleration torque:	490.7 [Nm]	Gear Ratio:	4 [-]

Obrázek 3.9: Tabulka minimálních požadavků kladených na převodovku

Closest suitable gearboxes:		Closest unsuitable gearboxes:	
 <p>NR 180</p>	Emergency Stop Torque	 <p>NR 140</p>	Emergency Stop Torque
	+1072.4 (+62.8%)		-407.6 (-23.9%)
	Acceleration Torque		+109.3 (+22.3%)
	+659.3 (+134.3%)		+207.1 (+75.9%)
	Nominal Torque		+497.1 (+182.2%)
Motor Input Speed	+2060.0 (+143.1%)	Motor Input Speed	+2560.0 (+177.8%)

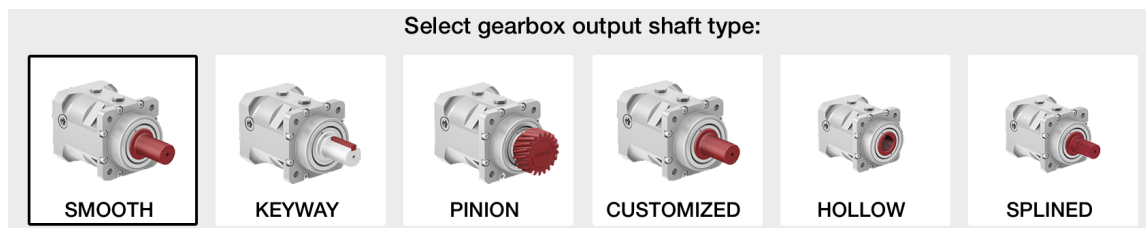
Obrázek 3.10: Nejlépe vyhovující a nejlépe nevyhovující převodovky

3.2.3 Finální specifikace převodovky

Finální specifikace převodovky zahrnuje několik dílčích kroků:

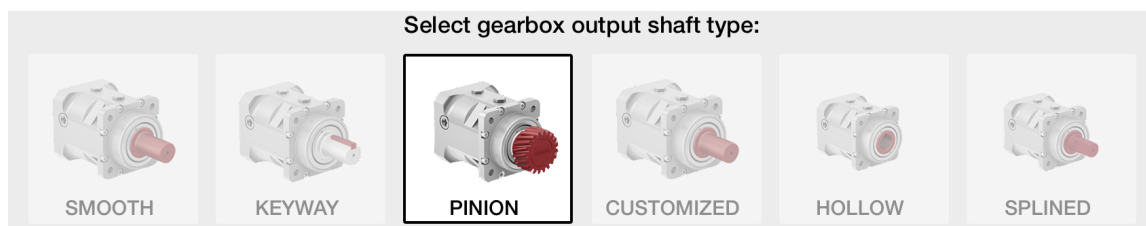
- výběr typu výstupu převodovky,
- výběr typu pozice převodovky,
- definice výstupní hřídele nebo pastorku,
- výběr typu vstupu převodovky,
- výběr typu příruby na motor,
- zvolení přesnosti,
- zvolení převodového poměru.

Příruba na motor, pozice, vstup a výstup převodovky mají opět definovaný počet typů, dalo by se proto znovu použít přepínací tlačítko, ovšem všechny tyto typy mají určitou grafickou reprezentaci - a jelikož nejsme na aktuálním pohledu omezeni místem, je vhodnější zvolit tlačítka s odpovídající grafickou reprezentací. Ukázka přepínacího tlačítka s grafickou reprezentací je uvedena na obrázku 3.11.



Obrázek 3.11: Přepínací tlačítka s grafickou reprezentací u výběru typu výstupu

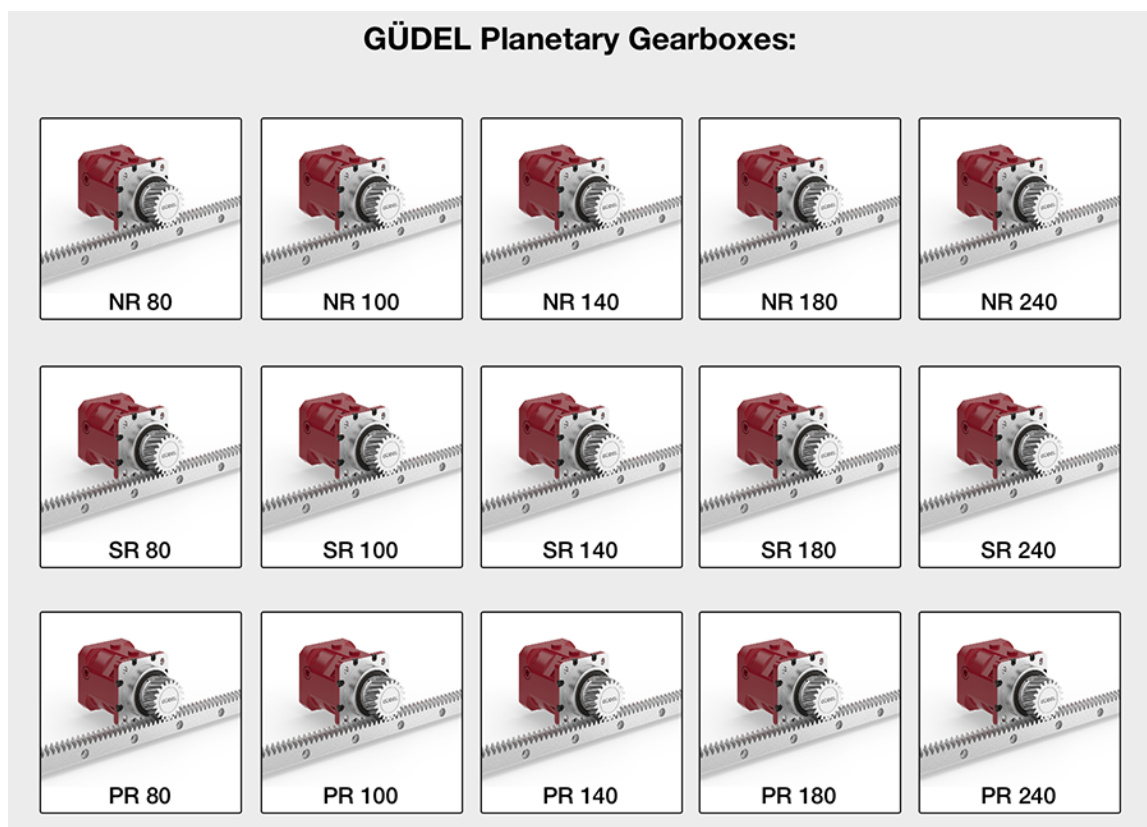
Pokud finální specifikace převodovky následuje po kroku 3.2.1, tak jsou již některé parametry předvolené - nesmí být změněny. Je tedy nutné zamezit přepínání mezi ostatními tlačítky. To se dá graficky realizovat pomocí změny alfa kanálu. Výsledný efekt změny alfa kanálu u přepínacího tlačítka je uveden na obrázku 3.12.



Obrázek 3.12: Předvolený typ výstupní hřídele

3.2.4 Přehled všech dostupných převodovek

Uživatelský požadavek č. 5 vyžaduje, aby bylo možné v aplikaci vybrat konkrétní převodovku bez zadávání parametrů. Převodovky se dělí na tři typy: NR, SR a PR a mají dílčí velikostní řady 80, 100, 140, 180 a 240, tedy celkově 15 základních typů převodovek. Vhodné řešení je udělat 15 tlačítek, které budou reprezentovat jednotlivé převodovky. Toto řešení je zobrazeno na obrázku 3.13. Možnou alternativou je rozdělit pohledy na tři dílčí strany, které by reprezentovaly jednotlivé typy: NR, SR a PR, ale tím by přibylo zbytečné zanoření do aplikace.



Obrázek 3.13: Zobrazení všech základních typů převodovek

Kapitola 4

Realizace, testování a vyhodnocení

Tato kapitola ve své první polovině dokumentuje užitý pracovní postup, použité nástroje, knihovny a důležité části implementace aplikace. Druhá polovina se zabývá průběhem a vyhodnocením uživatelského testování.

4.1 Postup práce

Před zahájením a během implementace jsem absolvoval několik konzultačních sezení s obchodníky firmy GÜDEL a.s. Počáteční schůzky se týkali zejména definování cílové skupiny uživatelů a definování uživatelských požadavků na aplikaci tak, aby co nejvíce vyhovovala potřebám obchodníků. Během absolvování úvodních konzultací a stanovení základních požadavků na aplikaci byly vytvořeny jednoduché wireframy aplikace, které byly základem tvorby prototypu. Prototyp aplikace obsahoval všechny důležité prvky uživatelského rozhraní, které vyšly z definovaných uživatelských požadavků a umožňoval navigaci v aplikaci mezi jednotlivými pohledy. Prototyp jsem představil na konzultaci obchodníkům firmy GÜDEL a.s. jako základní návrh aplikace. Z této schůzky vzešla upřesnění uživatelských požadavků. Následně jsem dle upřesněného návrhu popsaného v kapitole 3.2 implementoval výslednou aplikaci. Posledním krokem bylo provedení závěrečného testování aplikace a vyhodnocení jejich výsledků.

Cílová platforma aplikace je operační systém iOS na tabletu iPad, jelikož všichni obchodníci firmy GÜDEL a.s. mají tato zařízení. Vybranou minimální verzi pro běh aplikace je verze iOS 6.1, kterou podporují všechna zařízení iPad 2 a novější¹.

4.2 Implementace

Pro implementaci aplikace jsem zvolil vývojové prostředí Xcode 6.2, jelikož je to oficiální vývojové prostředí od firmy Apple určené pro vývoj aplikací operačního systému iOS. Vývojové prostředí Xcode obsahuje všechny potřebné funkce pro správu projektu a také simulátor iOS zařízení. Měl jsem na výběr mezi programovacím jazykem Objective C [1] nebo jazykem swift. Vybral jsem implementační jazyk Objective C, jelikož k němu existuje mnohem větší množství tutoriálů a informací ve srovnání s novějším jazykem Swift.

¹<http://www.everymac.com/systems/apple/ipad/ipad-faq/ipad-operating-system-info-update-cost-app-support.html>

4.2.1 Použité knihovny

Pro většinu aplikace byla plně dostačující základní knihovna operačního systému iOS při použití verze 6.1. Aplikace však obsahuje i nestandardní funkcionalitu, která není v základní knihovně zahrnuta. Prvním problémem je otevírání dokumentů. Základní knihovna operačního systému iOS sice obsahuje dostupnou funkcionalitu pro zobrazování PDF dokumentů, ale je z několika důvodů nedostačující. Proto byla použita knihovna třetí strany PDF Reader Core for iOS². Pro zobrazení fotogalerie byla použita knihovna třetí strany MWPhotoBrowser³.

4.2.2 Využití externí aplikace

Jedním z hlavních uživatelských požadavků byla možnost zobrazovat 3D modely převodovek. Zobrazení 3D modelu je netriviální problém a na jeho řešení neexistuje žádná volná knihovna. K zobrazení 3D modelu byla tedy využita externí aplikace eDrawings, kterou musí mít uživatel aplikace nainstalovanou, pokud chce zobrazit 3D model.

Implementace otevření 3D modelu aplikací eDrawings je řešena pomocí *UIDocumentInteractionController*, což je controller v iOS API pro pomoc při práci s lokálními soubory v aplikaci. Způsob otevření je zobrazen na obrázku 4.1.

Listing 4.1: Implementace způsobu otevírání modelu v externí aplikaci

```
- (void)openEdrawing:(NSString*)edrawingName
                        fromButton:(UIButton*)button
{
    NSURL *URL = [[NSBundle mainBundle]
                  URLForResource:edrawingName withExtension:@"eprt"];

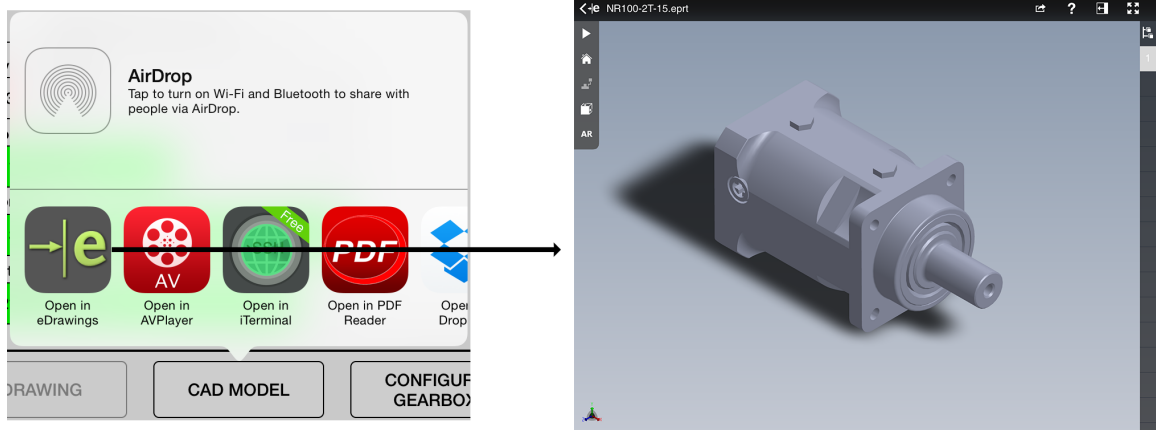
    if (URL) {
        // Initialize Document Interaction Controller
        self.UIDController = [UIDocumentInteractionController
                              interactionControllerWithURL:URL];

        // Configure Document Interaction Controller
        [self.UIDController setDelegate:self];

        // Present Open In Menu
        [self.UIDController
         presentOpenInMenuFromRect:[button frame]
         inView:self.view animated:YES];
    }
}
```

²<https://github.com/vfr/Reader>

³<https://github.com/mwaterfall/MWPhotoBrowser>



Obrázek 4.1: Otevření 3D modelu v aplikaci eDrawings

4.2.3 Načtení dat převodovek

Všechna data převodovek GÜDEL, nad kterými se provádí hledání, jsou uložena ve formátu CSV⁴ (Comma Separated Values). Jejich načtení proběhne pouze jednou a to při spuštění aplikace. Jelikož načtení dat trvá v průměru 450 milisekund, není nutné provádět načítání asynchronně. Pro jednoduchou serializaci formátu CSV byla použita knihovna třetí strany CHCSVParser⁵. Tato knihovna poskytuje metodu pro načtení jednotlivých záznamů CSV souboru při specifikování cesty k CSV souboru a oddělovacího znaku.

Listing 4.2: Načtení dat všech GÜDEL převodovek

```
// Load Gudel gearbox database from CSV
NSURL *URL = [[NSBundle mainBundle] URLForResource:@"GBData"
            withExtension:@"csv"];
NSArray *rows = [NSArray arrayWithContentsOfDelimitedURL:URL
                delimiter:','];
for (NSArray *row in rows) {
    NSInteger i = 0;
    GudelGearbox* gearbox = [[GudelGearbox alloc] init];
    for (NSString* string in row) {
        switch (i) {
            case 0: gearbox.name = string; break;
            // ...
        }
        i++
    }
    [gudelGearboxes addObject:gearbox];
}
```

⁴<https://cs.wikipedia.org/wiki/CSV>

⁵<https://github.com/davedelong/CHCSVParser>

4.2.4 Implementace grafického uživatelského rozhraní

Uživatelské rozhraní je implementováno pomocí grafického nástroje Storyboard, který je popsán v podkapitole 2.3.3. Všechny pohledy nacházející se v prvku Storyboard musí mít unikátní identifikátor - řetězec. Některé grafické prvky jsou dále upraveny programově v závislosti na stavu, ve kterém se aplikace právě nachází. K programovému přístupu ke grafickým prvkům vytvořených v grafickém nástroji Storyboard se využívají makra IBOutlets (Interface Builder Outlets) a IBAction (Interface Builder Action). Všechny pohledy jsou instance stejné třídy - *DetailViewController*.

Implementace pohledu pro zadání vstupních parametrů

Pohled pro zadání vstupních parametrů obsahuje několik grafických prvků, které jsou využity pro zadání nebo volbu vstupních parametrů. Prvním z nich je přepínací tlačítko implementované pomocí iOS widgetu *UISegmentedControl*, dalším je textové pole implementované pomocí iOS widgetu *UITextField*, posledním z nich je rozbalovací menu, které je implementované pomocí knihovny třetí strany *DownPicker*⁶.

UISegmentedControl

UISegmentedControl je jednoduché přepínací tlačítko obsahující předvolený počet sekcí s textem. Sekce *UISegmentedControl* jsou indexovány vzestupně zleva od 0. Příklad vzhledu *UISegmentedControl* je zobrazen na obrázku 3.4, implementace metody zaznamenávající změnu hodnoty typu aplikace je uvedena v ukázce 4.3.

Listing 4.3: Metoda zaznamenávající změnu indexu přepínacího tlačítka u typu aplikace

```
- (IBAction) valueChangedApplicationType :
    (UISegmentedControl *) sender
{
    inputGearboxParameters.applicationType =
        sender.selectedSegmentIndex;
}
```

UITextField

UITextField je textové pole určené pro uložení uživatelského vstupu. Při kliknutí na textové pole se zobrazí klávesnice, jejíž pomocí uživatel zadá požadovaný vstup. Pro použití *UITextField* je nastaven *Data Source* na controller pohledu *DetailViewController*. Dále jsou implementovány delegační metody, které se zavolají při kliknutí na textové pole a při potvrzení zadání vstupu do textového pole. Jelikož je na pohledu více textových polí, je nutné je od sebe odlišit při jejich zpracování v delegačních metodách. K odlišení se používá parametr tag, který se nastaví tak, aby byl u každého textového pole odlišný. Implementace těchto metod je zobrazena v ukázce 4.4.

⁶<https://github.com/Darkseal/DownPicker>

Listing 4.4: Delegační metody pro zpracování textových polí

```

// Metoda volana pri kliknuti na textove pole
- (BOOL) textFieldShouldBeginEditing:(UITextField *)textField
{
    if ([textField.text compare:@"0"] == 0 ||
        [textField.text compare:@"0.0"] == 0 ||
        [textField.text compare:@"0.00"] == 0) {
        textField.text = @"";
    }
    return TRUE;
}

// Metoda volana pri dokonceni editace textoveho pole
- (BOOL) textFieldShouldEndEditing:(UITextField *)textField
{
    switch (textField.tag)
    {
        // Moving Mass text field
        case 15: inputGearboxParameters.movingMass =
                [textField.text doubleValue]; break;
        // Input Load Speed text field
        case 16:
            [self UpdateAllWorkCycles];
            inputGearboxParameters.inputLoadSpeed =
                [textField.text doubleValue];
            break;
        //...
    }

    return YES;
}

```

Implementace pohledu pro zobrazení nejbližších převodovek

Pohled pro zobrazení nejbližších převodovek následuje pohledu pro zadání a volby vstupních parametrů. Při zobrazení pohledu se vypočítají ze zadaných parametrů minimální požadované hodnoty kladené na převodovku a aktualizuje se tabulka, která tyto hodnoty vizualizuje. Ukázka této tabulky je zobrazena na obrázku 3.9. Dále se na základě vypočítaných minimálních požadavků kladených na převodovku vyhledají vyhovující převodovky a nevhovující převodovky, které jsou následně zobrazeny. Záznam zobrazené převodovky sestává z tlačítka s obrázkem a názvem převodovky a grafu složeného ze čtyřech pruhů, které obsahují text definující rozdíl mezi minimální požadovanou hodnotou a hodnotou parametru převodovky. Ukázka tohoto záznamu je zobrazena na obrázku 3.10. Všechny pruhové grafy jsou generovány dynamicky programově.

Navigační lišta

Všechny pohledy mají společný *UINavigationController*, což je horní navigační lišta. Na této liště jsou vlevo umístěna tlačítka, která musí být zobrazena na všech pohledech aplikace a také logo firmy GÜDEL. Navigační lišta také obsahuje text, který popisuje účel aktuálního pohledu. Tlačítka na navigační liště jsou vytvořena programově.

4.3 Testování a vyhodnocení

Tato podkapitola se zabývá testováním aplikace a vyhodnocením výsledků. Je zde popsán postup přípravy testů, podoba testování a průběh testování. V druhé části podkapitoly jsou zhodnoceny výsledky a zdokumentovány zjištěné nedostatky.

4.3.1 Prvotní testování

Aplikace byla během vývoje testována pùběžně s obchodníky firmy GÜDEL a.s. Cílem bylo otestovat základní princip navigace v aplikaci a volbu grafických prvků pro zadávání parametrů a zobrazení vyhledaných převodovek. V tomto testování jsem se zaměřil zejména na tři hlavní otázky:

1. **Navigace v aplikaci** - jelikož se jedná o aplikaci s více než 1 pohledem, musel se vyřešit způsob navigace mezi jednotlivými pohledy. Zvolil jsem způsob navigace tlačítka s popisem nebo i obrázkem po jejichž kliknutí se přepne na další nebo předchozí stranu. V rámci testování jsem chtěl ověřit, zda je tento způsob navigace pro obchodníky intuitivní. Ukázalo se, že obchodníci měli s tímto způsobem navigace zkušenosti, takže v této otázce padl jednotný souhlas. Jeden obchodník ovšem chtěl, aby existoval alternativní způsob navigace v aplikaci. Jako alternativní řešení jsme zvolili formu stromového menu.
2. **Způsob zadávání parametrů** - způsob zadávání parametrů byl řešen pomocí typických widgetů platformy iOS. Všichni obchodníci vlastnili firmením iPad, měli tedy zkušenosti s většinou grafických prvků. Obchodníkům nečinil problém zadat všechny vstupní parametry, tedy vyplnit textová pole a použít přepínací tlačítko. Při tomto testování se ale ukázalo, že se některé parametry nemohou zadat volně pomocí textových polí, nýbrž musí být zvoleny z předem předdefinovaných hodnot.
3. **Způsob zobrazení nejbližších převodovek** - u způsobu zobrazení nejbližších převodovek jsem chtěl v rámci testování ověřit zejména porozumění vizualizace a porozumění zobrazeným pruhovým grafům. Obchodníci ihned pochopili, že první zobrazená převodovka je nejbližší vyhovující díky pruhovým grafům a jejímu umístění. Obchodníci ocenili přítomnost tabulky se zobrazením minimálních požadavků na převodovku. Při tomto testování vznikl další uživatelský požadavek a to zobrazení nejen nejbližších vyhovujících převodovek, ale také zobrazení nejbližších nevyhovujících převodovek.

4.3.2 Závěrečné testování

Závěrečné testování proběhlo celkově dvakrát. Nejdříve byla finální aplikace představena a poskytnuta obchodníkům a následně proběhlo první testování na základě připraveného testovacího scénáře obsahující úkoly týkající se práce s aplikací. Testovací úkoly jsou uvedeny v příloze B. Další testování proběhlo po dvou týdnech za využití stejné sady testovacích úkolů jako při prvním testování. Během těchto dvou týdnů obchodníci využívali aplikaci u zákazníků, pracovali tedy s aplikací v praxi. Očekávaným výstupem těchto dvou závěrečných testování bylo časové zlepšení při vypracovávání zadaných úkolů po dvou týdenním využívání aplikace. Jako hlavní měřená metrika při testování byl tedy zvolen čas potřebný pro provedení definovaných úkolů v testovacím scénáři.

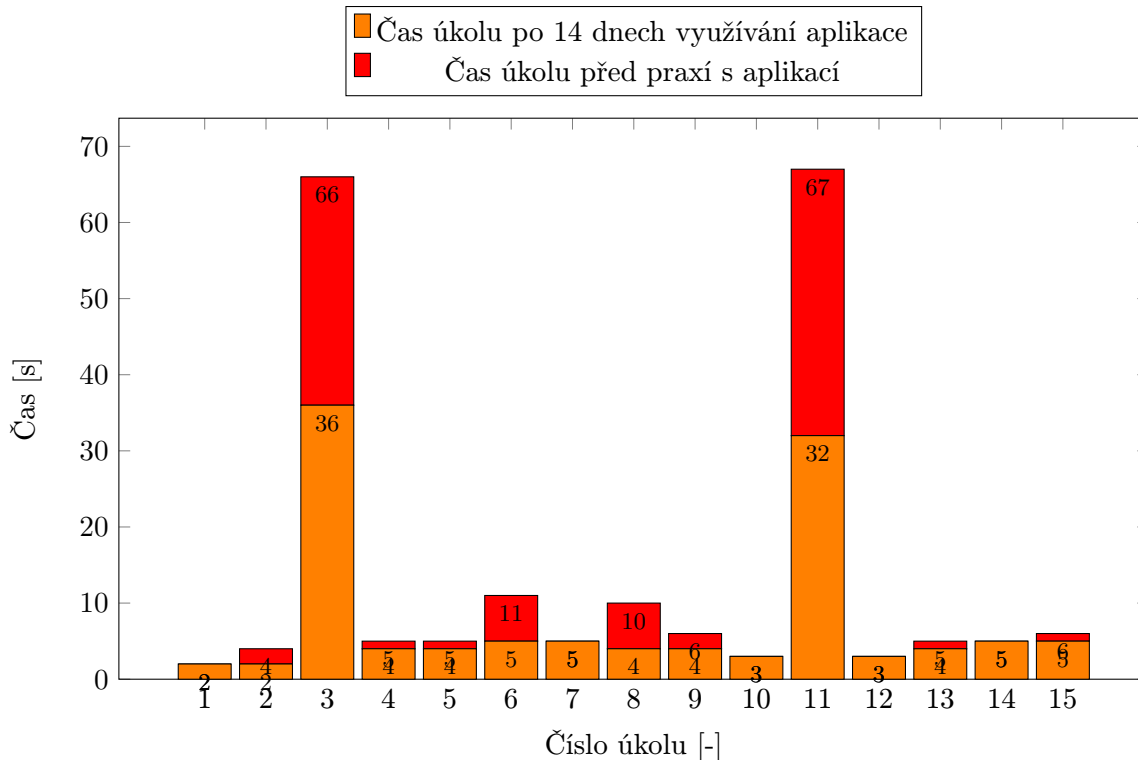
Během prvního testování proběhlo také testování funkčnosti výpočtů, zda jsou vypočítané hodnoty ze zadaných parametrů validní.

Obchodník č. 1

Obchodník č. 1 při testování po předání aplikace trvala nejdéle práce s rozbalovacím menu při zadávání vstupních parametrů, což se projevilo na čase potřebném pro provedení úkolu č. 3. Dále hledal obchodník č. 1 delší dobu po vygenerování emailu s parametry převodovky tlačítko pro jeho odeslání. Navigace v aplikaci nečinila žádný problém.

Při dalším testování, které následovalo po dvou týdnech od předání aplikace, kdy obchodník využíval aplikaci v praxi u zákazníků se znatelně snížil potřebný čas pro zadání vstupních parametrů a pro finální konfiguraci převodovky. Vykonávání jednotlivých úkolů bylo plynulé.

Naměřené hodnoty z testování jsou zobrazeny v grafu 4.2.



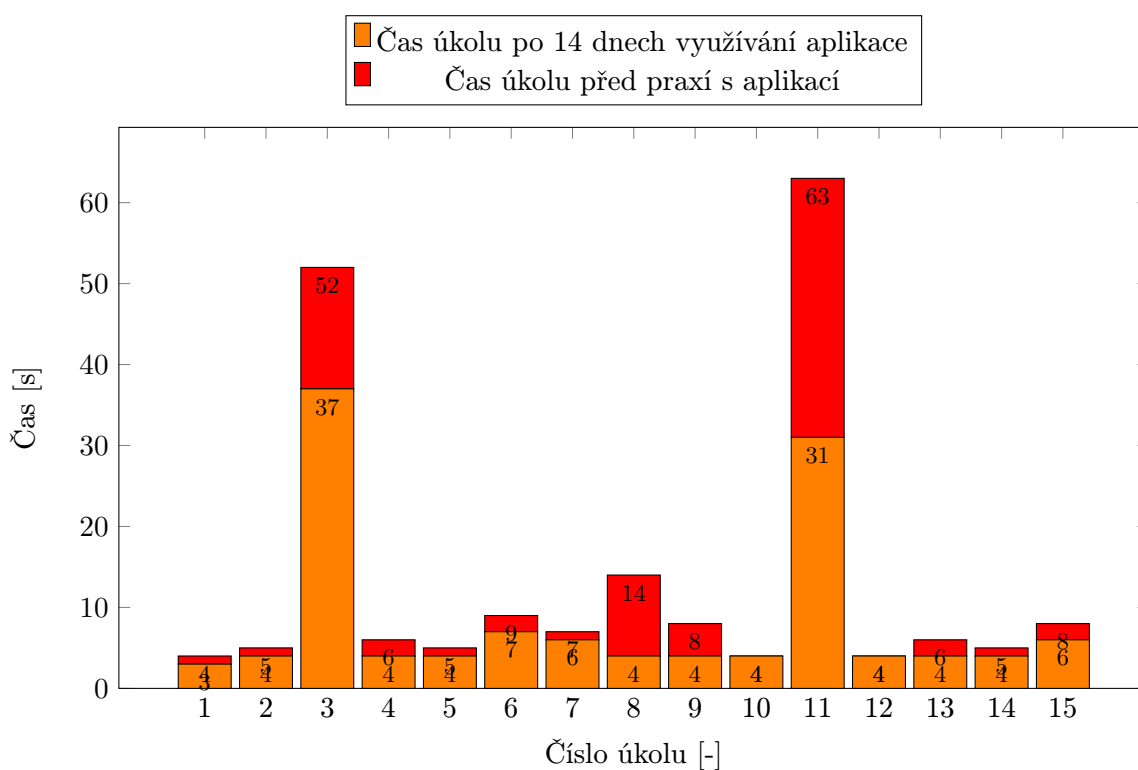
Obrázek 4.2: Naměřené časy jednotlivých úkolů obchodníka č. 1

Obchodník č. 2

Obchodníku č. 2 nedělalo potíže porozumění uživatelskému rozhraní pro zadání vstupních parametrů, ale spíše pomalé psaní. Číslice a desetinná čísla zadával na klávesnici pomaleji než obchodník č. 1. Dále trvalo dlouhou dobu přepnutí se z aplikace eDrawings do původní aplikace, jelikož si neuvědomil, že se 3D model otevřel v externí aplikaci. Při psaní adresáta ve vygenerovaném emailu hledal delší dobu znak pro zavínáč, tím se prodloužil čas pro vykonání úkolu 11. Navigace v aplikaci mezi jednotlivými pohledy nedělala obchodníku č. 2 problém.

Při dalším testování obchodník č. 2 zadával znaky z klávesnice rychleji při vyplňování textových polí. Dále přišel na rychlý způsob přepnutí mezi aplikací eDrawings a výchozí aplikací pro návrh převodovek a to použitím tříprstového gesta, tím se znatelně snížil čas potřebný pro vykonání úkolu č. 8.

Naměřené hodnoty z testování jsou zobrazeny v grafu 4.3.



Obrázek 4.3: Naměřené časy jednotlivých úkolů obchodníka č. 2

4.3.3 Vyhodnocení testování

Pozorovaný problém při práci s rozbalovacím menu (obrázek 3.6) při prvotním testování by se dal vyřešit zvýrazněním zobrazeného kontextového menu, například snížením alfa kanálu všech ostatních grafických prvků. Problém orientace při otevření 3d modelu v externí aplikaci se v momentální situaci nedá zlepšit, jelikož není k dispozici otevřená knihovna třetí strany, která by umožnila zobrazit model přímo v aplikaci.

Celkově dopadlo testování velmi pozitivně. Obchodníci porozuměli napoprvé téměř všem prvkům uživatelského rozhraní, což nejspíš souvisí s faktem, že byli součástí návrhové fáze

aplikace. Po dvou závěrečných testování bylo patrné u obou obchodníků znatelné zlepšení v čase při vykonávání klíčových úkolů - zadání parametrů kladených na převodovku a finální konfigurace převodovky.

Kapitola 5

Závěr

Cílem této bakalářské práce bylo prostudovat problematiku a navrhnout řešení grafického uživatelského rozhraní mobilní aplikace pro zařízení iPad, jejíž účelem je zefektivnění prodeje planetových převodovek obchodníkům firmy GÜDEL a.s.

V první části návrhu proběhly schůzky s budoucími uživateli, kde se specifikovaly požadavky na aplikaci. Dále byla na základě rozhovorů vytvořena persona reprezentující typického uživatele aplikace a byli vytvořeny scénáře použití pro definování účelu aplikace. Poté byl vytvořen prototyp aplikace, který se otestoval na uživatelích. Testování prototypu přineslo nové uživatelské požadavky, zejména přidání alternativní možnosti navigaci v aplikaci pomocí stromového menu a přidání zobrazení nejbližších nevhodných převodovek. Na základě připomínek a nových uživatelských požadavků vzniklých při testování prototypu byla aplikace implementována ve vývojovém prostředí Xcode 6.2 za použití programovacího jazyku Objective C.

Po dokončení realizace byla aplikace předána koncovým uživatelům a proběhly dvě kola finálního testování, obsahovalo splnění patnácti úkolů práce s aplikací. První kolo testování proběhlo ihned po předání aplikace, druhé proběhlo po čtrnácti dnech, kdy uživatelé využívali aplikaci v praxi. Bylo sledováno časové zlepšení při plnění předepsaných úkolů a také byly sledovány reakce uživatelů při jejich práci s aplikací. Potřebný čas pro vykonání klíčových úkolů klesl u testovaných uživatelů téměř o polovinu u druhého testování, kdy byly uživatelé s aplikací řádně seznámeni a nevznikla žádná vážná nepochopení při jejím používání.

Realizované řešení se koncovým uživatelům, obchodníkům firmy GÜDEL a.s., velice líbí a aplikace nahradila nutnost obchodníků s sebou neustále nosit produktové katalogy a notebooky při komunikaci se zákazníkem. V budoucnu by obchodníci rádi měli podobné nástroje pro další produkty firmy GÜDEL a.s., jako například lineární vedení.

Literatura

- [1] S. G. Kochan: *Programming in Objective-C 2.0*. 2nd ed. Upper Saddle River: Addison Wesley Professional, 2009. ISBN 978-0-321-56615-7, 600 s.
- [2] D. Stone, C. Jarrett, M. Woodroffe, S. Minocha: *User Interface Design and Evaluation*. Morgan Kaufmann, 2005, ISBN 0120884364.
- [3] Steve Krug: *Don't Make Me Think! A Common Sense Approach to Web Usability*, 2nd ed. USA: New Riders, 2006, ISBN 0-321-34475-8.
- [4] Travis Lowdermilk: *User-Centered Design*. USA: O'Reilly, 2013, ISBN 978-1-449-35980-5.
- [5] Apple Inc.: *iOS Human Interface Guidelines* [Online]. 2016. Dostupné z: <https://developer.apple.com/library/ios/documentation/UserExperience/Conceptual/MobileHIG/>.
- [6] Apple Inc.: *iOS Development Library* [Online]. 2016. Dostupné z: <https://developer.apple.com/library/ios/navigation/>.
- [7] Marcin Treder: Wireframing, Prototyping, Mockuping – What's the Difference? *designmodo.com* [Online]. 2012. Dostupné z: <http://designmodo.com/wireframing-prototyping-mockuping/>.
- [8] Antonio Bello: iOS User Interfaces: Storyboards vs. NIBs vs. Custom Code. *toptal.com* [Online]. 2014. Dostupné z: <https://www.toptal.com/ios/ios-user-interfaces-storyboards-vs-nibs-vs-custom-code>.

Příloha A

Obsah CD

Příložené CD obsahuje:

- adresář se s projektem aplikace pro vývojové prostředí Xcode;
- adresář se zdrojovými kódy této technické zprávy;
- tuto technickou zprávu v PDF;
- video a plakát prezentující tuto práci;

Příloha B

Testovací formulář

1. Spusťte aplikace
2. Přejděte na pohled pro zadání vstupních parametrů
3. Vyplňte následující parametry aplikace, ve kterých bude převodovka pracovat:
 - Hmotnost břemena: 3200kg
 - Vstupní rychlost: 1.6 m/s
 - Zrychlení: $1.9m/s^2$
 - Převodový poměr: 4
 - Poloha převodovky: Horizontální
 - Výstupní typ: Pastorek, typ H4-20-40
 - Pracovní cyklus: 1. Pohyb 3.2m, pauza pro operaci 5s, 2. Pohyb 3.2m, pauza pro operaci 4s
4. Zobrazte pohled seznamu nejbližších navržených převodovek
5. Zobrazte detail nejvhodnější navržené převodovky
6. Zobrazte produktový katalog vybrané převodovky a následně ho zavřete
7. Otevřete CAD model vybrané převodovky v aplikaci eDrawings
8. Přejděte zpět do aplikace GEARBOX SIZING TOOL
9. Zobrazte kompletní katalog všech produktů z aktuální strany
10. Přejděte na pohled konfigurace převodovky
11. Zvolte přesnost převodovky $P=3$. Vygenerujte E-Mail obsahující vygenerované parametry převodovky. Napište na místo zákazníka "GÜDEL". Zvolte jako adresáta e-mailovou adresu pjasicek@gmail.com. E-Mail odešlete.
12. Vraťte se na hlavní pohled aplikace
13. Přejděte na pohled seznamu všech GÜDEL převodovek
14. Přejděte na pohled převodovky SR 100
15. Pomocí Menu tlačítka zobrazte stranu převodovky NR 180