



**FAKULTA
INFORMAČNÍCH
TECHNOLOGIÍ
ČVUT V PRAZE**

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

| | |
|--------------------------|--------------------------------------|
| Název: | Mobilní aplikace pro komunitní účely |
| Student: | Jakub Pečenka |
| Vedoucí: | Ing. Josef Pavlíček, Ph.D. |
| Studijní program: | Informatika |
| Studijní obor: | Informační systémy a management |
| Katedra: | Katedra softwarového inženýrství |
| Platnost zadání: | Do konce letního semestru 2019/20 |

Pokyny pro vypracování

Cílem práce je tvorba mobilní aplikace, která bude navrhována pro křesťanské dorosty s cílovou věkovou skupinou 11-15 let. Jedná se o motivační aplikaci na způsob modrých životů z Rychlých Šípů, ve které bude seznam různých úkolů a pokud uživatel úkol splní, vybarví si vždy okénko. Úspěšnost splnění úkolů je možné sdílet s dalšími uživateli. Aplikace navíc zaznamenává uživatelův pokrok, připomíná mu pomocí upomínek plnění úkolů a nabízí další podpůrné funkce.

1. Proveďte průzkum a zhodnocení podobných existujících mobilních aplikací.
2. Analyzujte uživatelské požadavky na aplikaci.
3. Navrhněte hlavní případy užití a uživatelské scénáře.
4. Navrhněte funkcionalitu aplikace.
5. Implementujte prototyp mobilní aplikace.
6. Doporučte další vývojový postup aplikace.
7. Aplikaci manažersky zhodnoťte, tj. proveďte průzkum trhu, určete její pozici na trhu, definujte její finanční náročnost z hlediska údržby a budoucího rozvoje. Vyčíslete předpokládané příjmy v následujících 5 letech.

Seznam odborné literatury

Dodá vedoucí práce.

Ing. Michal Valenta, Ph.D.
vedoucí katedry

doc. RNDr. Ing. Marcel Jiřina, Ph.D.
děkan

V Praze dne 5. února 2019



**FAKULTA
INFORMAČNÍCH
TECHNOLGIÍ
ČVUT V PRAZE**

Bakalářská práce

Mobilní aplikace pro komunitní účely

Jakub Pečenka

Katedra softwarového inženýrství

Vedoucí práce: Ing. Josef Pavlíček, Ph.D.

12. května 2019

Poděkování

Rád bych poděkoval Ing. Josefovi Pavlíčkovi, Ph.D., za cenné rady, věcné připomínky a vstřícnost při konzultacích. Dále chci poděkovat rodině za vytrvalou a trpělivou podporu při studiu.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval(a) samostatně a že jsem uvedl(a) veškeré použité informační zdroje v souladu s Metodickým pokynem o etické přípravě vysokoškolských závěrečných prací.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů. V souladu s ust. § 46 odst. 6 tohoto zákona tímto uděluji nevýhradní oprávnění (licenci) k užití této mojí práce, a to včetně všech počítačových programů, jež jsou její součástí či přílohou, a veškeré jejich dokumentace (dále souhrnně jen „Dílo“), a to všem osobám, které si přejí Dílo užít. Tyto osoby jsou oprávněny Dílo užít jakýmkoli způsobem, který nesnižuje hodnotu Díla, a za jakýmkoli účelem (včetně užití k výdělečným účelům). Toto oprávnění je časově, teritoriálně i množstevně neomezené. Každá osoba, která využije výše uvedenou licenci, se však zavazuje udělit ke každému dílu, které vznikne (byť jen zčásti) na základě Díla, úpravou Díla, spojením Díla s jiným dílem, zařazením Díla do díla souborného či zpracováním Díla (včetně překladu), licenci alespoň ve výše uvedeném rozsahu a zároveň zpřístupnit zdrojový kód takového díla alespoň srovnatelným způsobem a ve srovnatelném rozsahu, jako je zpřístupněn zdrojový kód Díla.

V Praze dne 12. května 2019

.....

České vysoké učení technické v Praze

Fakulta informačních technologií

© 2019 Jakub Pečenka. Všechna práva vyhrazena.

Tato práce vznikla jako školní dílo na Českém vysokém učení technickém v Praze, Fakultě informačních technologií. Práce je chráněna právními předpisy a mezinárodními úmluvami o právu autorském a právech souvisejících s právem autorským. K jejímu užití, s výjimkou bezúplatných zákonných licencí a nad rámec oprávnění uvedených v Prohlášení na předchozí straně, je nezbytný souhlas autora.

Odkaz na tuto práci

Pečenka, Jakub. *Mobilní aplikace pro komunitní účely*. Bakalářská práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2019.

Abstrakt

Tato bakalářská práce se zabývá návrhem motivační mobilní aplikace a jejím zhodnocením z ekonomicko-manažerského hlediska. V práci se porovnávají podobné existující aplikace, analyzují se funkční požadavky, které jsou poté navrženy pomocí diagramu případů užití a je vytvořen a otestován prototyp navrhované aplikace. Na závěr jsou spočteny náklady metodami PERT a Use case points, analyzovány příjmy aplikace a projekt je manažersky zhodnocen. Vytvořené řešení poskytuje podklad pro následnou implementaci aplikace. Náklady na vývoj jsou odhadovány na 440 000 – 560 000 Kč. Práce je určena pro Dorostovou unii, která na základě výsledků práce posoudí návratnost investice a případně schválí implementaci v souladu s návrhem. V příloze lze nalézt vytvořený prototyp mobilní aplikace.

Klíčová slova funkční prototyp, návrh mobilní aplikace, motivační aplikace, ekonomicko-manažerské zhodnocení, Android, komunita mladých, denní návyky

Abstract

This bachelor thesis deals with the design of the motivational mobile application and its evaluation from the economic and managerial points of view. The objectives of this thesis are the comparison of similar already existing applications and the analysis of given functional requirements, which are then visualized via use case diagrams. Based on this, the prototype of the proposed application is created and tested. Finally, the costs are calculated using the PERT and Use case points methods, the application profit is analysed and the project is evaluated from a managerial point of view. The proposed solution provides the basis for the implementation of the app. The estimated cost of the development is 440 000 – 560 000 CZK. This thesis is intended for the Dorostová unie, which will assess the return of the investments on the basis of its outcomes. The approval of this proposal will be followed with the app implementation accordingly. A created prototype of the application can be found in the attachments.

Keywords function prototype, mobile app design, motivation app, economic-managerial summary, Android, youth club community, daily habits

Obsah

| | |
|--|-----------|
| Úvod | 1 |
| 1 Cíl práce | 3 |
| 2 Teoretická část | 5 |
| 2.1 Způsoby vývoje mobilních aplikací | 5 |
| 2.1.1 Typy mobilních aplikací | 5 |
| 2.2 Softwarový proces | 6 |
| 2.2.1 Unifikovaný proces vývoje aplikací | 7 |
| 2.2.2 Proces vývoje mobilních aplikací | 7 |
| 2.3 Analýza požadavků | 8 |
| 2.4 Modelování případů užití | 9 |
| 2.5 Wireframe | 9 |
| 2.6 Prototyp aplikace | 10 |
| 2.7 Nástroje | 10 |
| 2.7.1 Justinmind | 10 |
| 2.7.2 Visual-paradigm.com | 11 |
| 2.7.3 Mockflow | 11 |
| 2.8 SWOT analýza | 11 |
| 2.9 Odhad nákladů projektu | 12 |
| 2.9.1 Kalkulátor nákladů | 14 |
| 2.9.2 Metoda PERT | 14 |
| 2.9.3 Use case points | 15 |
| 2.10 Grafické rozhraní aplikace | 17 |
| 2.10.1 Jakob Nielsen a jeho přístup | 19 |
| 2.11 Závěr | 20 |
| 3 Analýza | 21 |
| 3.1 Zvolený způsob vývoje | 21 |
| 3.2 Požadavky zadavatele | 21 |

| | | |
|----------|---|-----------|
| 3.3 | Existující řešení | 23 |
| 3.3.1 | Mé cíle, návyky a předsevzetí | 23 |
| 3.3.2 | Habit Tracker | 24 |
| 3.3.3 | Návykovač | 25 |
| 3.3.4 | HabitHub – Habit and Goal Tracker | 25 |
| 3.3.5 | Závěr | 25 |
| 3.4 | Funkční požadavky | 26 |
| 3.4.1 | Registrace | 26 |
| 3.4.2 | Online sdílení | 27 |
| 3.4.3 | Vyplňování a zobrazování návyků | 27 |
| 3.4.4 | Přidávání návyků | 28 |
| 3.4.5 | Ostatní | 28 |
| 3.4.6 | Nastavení | 29 |
| 3.5 | Nefunkční požadavky | 29 |
| 3.6 | Závěr | 30 |
| 4 | Návrh | 31 |
| 4.1 | Případy užití | 31 |
| 4.1.1 | Seznam aktérů | 31 |
| 4.1.2 | UC1: Registrace | 31 |
| 4.1.3 | UC2: Vytvořit zvyk | 33 |
| 4.1.4 | UC3: Vyplnit úspěšnost zvyku | 33 |
| 4.1.5 | UC4: Zobrazit výsledky plnění zvyků | 34 |
| 4.1.6 | UC5: Změna nastavení zvyků | 35 |
| 4.1.7 | UC6: Zobrazit oznámení | 36 |
| 4.1.8 | UC7: Editovat oznámení | 36 |
| 4.1.9 | UC8: Požádat o přátelství | 36 |
| 4.1.10 | UC9: Přidat vedoucího | 37 |
| 4.1.11 | UC10: Schválit přítele | 37 |
| 4.1.12 | UC11: Odebrat přítele | 37 |
| 4.1.13 | UC12: Odebrat vedoucího | 38 |
| 4.1.14 | UC13: Změna obecného nastavení | 38 |
| 4.1.15 | UC14: Změna na účet vedoucího | 39 |
| 4.1.16 | UC15: Změna na běžný účet | 39 |
| 4.2 | Wireframe aplikace | 39 |
| 4.3 | Závěr | 40 |
| 5 | Vytvoření prototypu | 41 |
| 6 | Testování | 45 |
| 6.1 | Průběh | 45 |
| 6.2 | Výsledky | 45 |
| 6.3 | Závěr | 46 |

| | |
|---|-----------|
| 7 Ekonomicko-manažerské zhodnocení | 47 |
| 7.1 Náklady na vývoj | 47 |
| 7.1.1 Kalkulátory nákladů | 47 |
| 7.1.2 PERT | 47 |
| 7.1.3 Use Case Points | 48 |
| 7.1.4 Závěr | 52 |
| 7.2 Náklady na údržbu | 53 |
| 7.3 Předpokládané příjmy | 53 |
| 7.3.1 Finanční příjmy | 54 |
| 7.3.2 Nefinanční příjmy | 54 |
| 7.4 SWOT analýza | 54 |
| 7.5 Shrnutí | 54 |
| 8 Doporučení dalšího postupu | 57 |
| Závěr | 59 |
| Bibliografie | 61 |
| A Seznam použitých zkratk | 67 |
| B Obrazovky prototypu | 69 |
| C Obsah příloženého CD | 75 |

Seznam obrázků

| | | |
|------|--|----|
| 2.1 | Časový diagram unifikovaného procesu vývoje | 7 |
| 2.2 | SWOT analýza | 12 |
| 2.3 | Kužel nejistoty | 13 |
| 3.1 | Ukázky z aplikace „Mé cíle, návyky a předsevzetí“ | 24 |
| 3.2 | Ukázky z aplikace „HabitHub – Habit and Goal Tracker“ | 26 |
| 4.1 | Diagram případů užití | 32 |
| 4.2 | Wireframe: Denní zobrazení a Přidávání návyků | 40 |
| 5.1 | Prototyp: Standardní menu, Menu následovníka, Menu vedoucího | 42 |
| 5.2 | Prototyp: Denní, Týdenní a Měsíční zobrazení | 42 |
| 5.3 | Prototyp: Statistiky, Přidávání návyku a Oznámení | 43 |
| B.1 | Registrace | 69 |
| B.2 | Dokončení | 69 |
| B.3 | Standardní menu | 70 |
| B.4 | Menu následovníka | 70 |
| B.5 | Menu vedoucího | 70 |
| B.6 | Denní zobrazení | 70 |
| B.7 | Týdenní zobrazení | 71 |
| B.8 | Měsíční zobrazení | 71 |
| B.9 | Přidat návyk | 71 |
| B.10 | Oznámení | 71 |
| B.11 | Přidat oznámení | 72 |
| B.12 | Statistiky | 72 |
| B.13 | Přátelé | 72 |
| B.14 | Vedoucí | 72 |
| B.15 | Nastavení | 73 |
| B.16 | Upravit návyk | 73 |
| B.17 | Nastavení | 74 |

Seznam tabulek

| | | |
|-----|--|----|
| 2.1 | Metoda UCP: ohodnocení UC | 17 |
| 2.2 | Metoda UCP: ohodnocení aktérů | 17 |
| 2.3 | Metoda UCP: faktory technické komplexnosti | 18 |
| 2.4 | Metoda UCP: faktory prostředí | 18 |
| 7.1 | Kalkulátory ceny implementace | 48 |
| 7.2 | Aplikace PERT metody | 48 |
| 7.3 | Metoda UCP: ohodnocení UC – výpočet | 50 |
| 7.4 | Metoda UCP: ohodnocení aktérů – výpočet | 51 |
| 7.5 | Metoda UCP: faktory technické komplexnosti – výpočet | 51 |
| 7.6 | [Metoda UCP: faktory prostředí – výpočet | 52 |
| 7.7 | SWOT analýza | 55 |

Úvod

Mobilní telefony používáme hojně a často. A jestli někdo obzvlášť, pak jsou to mladí lidé a teenageři. Starší lidé si často, a někdy oprávněně, stěžují na pasivitu a nezodpovědnost dnešní mladé generace. I vedoucí klubů a kroužků pro mladé lidi mají často problém povzbudit mladé ke smysluplné aktivitě. Tento aktuální problém pomůžu vyřešit způsobem, jaký je mladým blízký – pomocí chytré mobilní aplikace.

Aplikace je určena především pro křesťanské dorostové kluby. Jedná se o síť oddělených klubů obvykle s 5–20 členy do 15 let, které různě spolupracují a jsou zastřešeny společným vedením. Aplikace ale bude vhodná i pro další podobné organizace, například pro skauty. Pomocí mobilní aplikace, kterou v bakalářské práci navrhnu, pomůžu zábavnou formou motivovat nejen mladé lidi k aktivnímu životu.

Kromě již zmíněných důvodů je mi navíc práce s mladými lidmi osobně blízká, a proto jsem se rozhodl podílet na vývoji této mobilní aplikace. V rámci bakalářské práce provedu analýzu požadavků, zhodnotím postup návrhu a mobilní aplikaci navrhnu. Poté implementuji a otestuji prototyp mobilní aplikace. Zhodnotím také ekonomicko-manažerská hlediska. Na bakalářskou práci následně budu moci navázat samotnou implementací aplikace.

Cíl práce

Prvním cílem této bakalářské práce je navrhnout mobilní aplikaci pro operační systém Android, implementovat a otestovat její prototyp. S tím se pojí také analýza a zvolení technologií pro postup. Aplikace bude pracovně nazývána *Dorůstáček*. Jedná se o motivační aplikaci, jejíž primární cíl je povzbudit uživatele zábavnou formou k plnění jednotlivých úkolů. Úkoly jsou plněny na denní bázi, tedy úspěšnost splnění se hodnotí pro každý den. Jednotlivé úkoly jako takové by měly být časově nenáročné a jednoduché. Náročnost spočívá v dodržení pravidelnosti, například každý den 10 minut cvičit nebo přečíst jednu kapitolu knihy.

Uživatel je motivován především formou vlastní zpětné vazby. Pokud daný úkol splní, odškrtně si ten den úkol jako splněný. Vhodnou grafickou formou bude evidována historie splnění. Aplikace navíc podporuje formu sdílení s přáteli. Uživatelé se tak mohou motivovat i vzájemně.

Dalším cílem práce je provést průzkum trhu – najít a zhodnotit podobné již existující aplikace. Výsledek průzkumu bude jedním z podkladů pro návrh mobilní aplikace. Navíc na základě průzkumu definuji pozici navrhované aplikace na trhu.

Posledním cílem bude aplikaci zhodnotit z ekonomicko-manažerského pohledu. Definovat náklady na vývoj a údržbu a zvážit přínosnost aplikace.

Nejdříve pološím teoretický základ pro praktickou část práce (tzn. popíši potřebné pojmy a rozeberu postupy aplikované v praktické části), následně prozkoumám již existující podobné Android aplikace, poté zanalyzuji a popíši přesné požadavky na aplikaci, kterou následně navrhnu, otestuji a zhodnotím z ekonomicko-manažerského pohledu.

Teoretická část

Tato teoretická kapitola popisuje způsoby a postupy softwarového vývoje se zaměřením na mobilní aplikace. Vysvětluje pojmy wireframe a prototyp. Popisuje nástroje, které jsou v praktické části použity. Vysvětluje postup analýzy a objasňuje použité metody na odhadování nákladů vývoje. Kapitola se také zabývá grafickým rozhraním aplikace.

2.1 Způsoby vývoje mobilních aplikací

Vývoj mobilních aplikací značně komplikuje fakt, že lidé používají mobilní telefony s různými operačními systémy. V současnosti, podle statistiky z března 2019, je světově jednoznačně nejrozšířenější operační systém Android, jež používá okolo 75 % uživatelů. Druhým nejrozšířenějším systémem je iOS s podílem přibližně 22 %. [1] Přímo v České republice je statistika velmi podobná a příliš se neliší od globálního trhu. Podle poslední statistiky mají v Česku asi 3 % uživatelů operační systém Windows, asi 74 % Android a skoro 20 % iOS. [2] Poslední statistika pro Českou republiku je však z prosince roku 2016. Navíc Windows 10 Mobile – nejnovější verze operačního systému Windows – obdrží poslední aktualizaci 10.12.2019 a vývoj mobilního operačního systému Windows skončí. Proto většina jeho uživatelů přechází na Android, nebo iOS. Existují samozřejmě i další operační systémy, ale jejich uživatelské zastoupení je pod 1 %, a proto je nebudu brát v potaz.

Při vývoji mobilní aplikace musíme tedy jistě vzít v úvahu oba rozšířené operační systémy: Android, i iOS. Nicméně není samozřejmostí, že aplikace bude dostupná pro oba zmíněné systémy. Pokud aplikaci chceme vyvíjet pro více než jeden operační systém, respektive platformu, mluvíme o tzv. multiplatformním vývoji.

2.1.1 Typy mobilních aplikací

Existují tři základní typy mobilních aplikací:

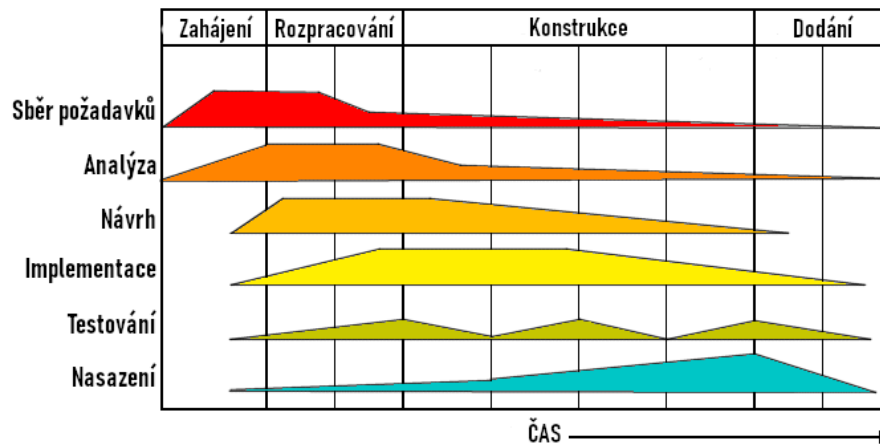
- **Nativní** aplikace je aplikace vytvořená pouze pro jednu platformu. Pro další platformu je potřeba vytvořit další samostatnou aplikaci, která samozřejmě může mít stejnou funkcionalitu a např. může i komunikovat se stejným serverem. Nativní aplikace jsou nejrychlejší, nejlépe využívají funkce zařízení a jejich vzhled je přesně přizpůsoben operačnímu systému. Nevýhodou je časová náročnost vývoje a horší udržitelnost a správa kódu.
- **Webová** aplikace je v podstatě webová stránka přizpůsobená, aby vypadala jako nativní aplikace. Lze k ní obvykle přistoupit i přímo přes webový prohlížeč. Webové aplikace jsou nejlevnější alternativou a mají jednoduchou správu. Ačkoliv omezení webových aplikací se postupem času zmenšují, mají stále značně omezený přístup k funkcím zařízení a jsou náročnější na hardware.
- **Hybridní** aplikace je kombinací nativní a webové. Jedná se o zajímavou a v současnosti rychle se rozvíjející alternativu. Výhodou je, že aplikace je multiplatformní. Stačí tedy jeden kód pro iOS, i Android. Zároveň nemá tak omezené možnosti využití funkcí telefonu jako webová aplikace. Mezi hlavní nevýhody patří vyšší požadavky na hardware, absence nativních UI prvků a někdy náročnější vývoj.

Otázka, který typ aplikace zvolit, nemá jednoznačnou odpověď. Záleží na konkrétní aplikaci. Navíc možnosti vývoje se postupně mění. Přibývají a vyvíjejí se nové technologie apod. A tak obzvláště hybridní, ale i webové aplikace jsou používanější, než dříve. [3], [4], [5], [6]

2.2 Softwarový proces

Samotná implementace (programování) mobilní aplikace je jen jedna z fází tvorby softwaru. Kromě implementace je potřeba provést řadu dalších kroků. Martin Hlavatý uvádí, že „*Softwarový proces/Model životního cyklu vývoje software (SDLC, Software Development Lifecycle)* je množina aktivit nutných k tomu, aby software vznikl. Jejich souslednost, opakování, vstupy a výstupy jednotlivých aktivit a nároky na jejich provedení.“ [7] A dodává, že mezi kroky, které je vždy potřeba udělat, patří:

- specifikace – co bude systém vykonávat, případně jak se změní,
- architektura a design – z jakých částí a jakým způsobem se bude systém skládat,
- implementace – vlastní výroba systému,
- validace – ověření, že systém správně vykonává požadované funkcionality.



Obrázek 2.1: Časový diagram unifikovaného procesu vývoje [10] (překlad autora)

Při procesu vývoje softwaru si musíme odpovědět na otázky, jak postupovat, jaké fáze softwarového produktu přesně rozlišovat, apod. Sadu doporučení, která odpovídají na tyto otázky, nazýváme metodikou. Problém je, že projekty jsou různorodé a každý projekt je něčím specifický. Co může dobře fungovat u jednoho projektu, nemusí fungovat u jiného. Navíc úspěšnost projektu závisí i na mnoha dalších faktorech. Určitě tedy nelze metodiky chápat jako univerzální návod na úspěch. [8]

2.2.1 Unifikovaný proces vývoje aplikací

Jednou z nejužších metodik pro softwarový proces je metodika unifikovaného procesu vývoje aplikací, anglicky *Unified Process – UP*. Pro vyjádření metodiky používá vlastní jazyk – UML. Metodika popisuje 4 fáze (zahájení – *inception*, rozpracování – *elaboration*, konstrukce – *construction* a dodání – *transition*) a 6 pracovních postupů (sběr požadavků – *business Modeling*, analýza – *requirements*, návrh – *analysis and design*, implementace – *implementation*, testování – *test* a nasazení – *deployment*). Diagram 2.1 zobrazuje zmíněné fáze a pracovní postupy v průběhu času. [8], [9]

2.2.2 Proces vývoje mobilních aplikací

Proces vývoje mobilních aplikací pochopitelně vychází z obecného softwarového procesu. Přesný popis kroků procesu vedoucího od počáteční myšlenky až k úspěšnému provozu mobilní aplikace se podle různých zdrojů [11], [12], [13] liší. Nebudu popisovat konkrétní metodiku, ale v následujícím seznamu jen uvedu hlavní body procesu, které předcházejí samotnému vývoji.

1. průzkum trhu,
2. definice požadavků (viz sekce 2.3),
3. wireframe (viz sekce 2.5),
4. prototyp aplikace (viz sekce 2.6),
5. UX design aplikace (viz sekce 2.10).

2.3 Analýza požadavků

Když společnost PM Solutions zjišťovala, proč 37 % všech projektů skončí neúspěchem, mezi 5 nejčastějšími důvody byla právě analýza požadavků. Komplikace nastanou, pokud jsou požadavky nejasné, neshodnou se na nich obě strany, jsou nepřesné, nejsou jednoznačné, nebo si vzájemně odporují. [14]

Cíle analýzy požadavků jsou: „*vymezit hranice systému, umožnit přesnější odhad pracnosti, vyjasnit si zadání se zákazníkem a zachytit omezení, která jsou na systém kladena.*“ [9] Zachycují se pomocí strukturovaného textu, nebo graficky pomocí diagramů. Základní kategorizace dělí požadavky na funkční a nefunkční (obecné). Požadavky se píší jednoznačně a tak, aby byly ověřitelné a splnitelné.

Funkční požadavky popisují funkce, které bude navrhovaný software vykonávat, například formátovat text, poskytovat různé informace; nebo co všechno bude systém evidovat.

Nefunkční požadavky jsou takové, které určují omezení kladená na systém, mají zásadní dopad na návrh architektury a určují dodržování standardů. Vedle funkčních požadavků „*popisují další nezbytné vlastnosti potřebné vzhledem k prostředí a kontextu. Např. se jedná o požadavky na spolehlivost, bezpečnost, výkonnost, podporu během provozu atp.*“ [9]

Ve výukové prezentaci od společnosti Profinit [15] je proces specifikace uživatelských požadavků rozdělen do 4 fází:

1. **vyzvidání** – zjišťování a shromažďování informací,
2. **analýza** – získání představy o tom, co je potřeba,
3. **specifikace** – zdokumentování analýzy tak, jak je vyžadováno,
4. **ověření** – ověření, zda jsou specifikované požadavky věcně i formálně správné.

2.4 Modelování případů užití

Model případů užití, též označován jako model jednání, nebo anglicky *Use Case model*, vyjadřuje, kdo a jakým způsobem bude systém používat. V modelu zachycujeme pouze funkční požadavky. Nelze tedy pouze na jeho základě správně odhadnout cenu vývoje softwaru apod. Cílem modelování je detailní specifikace funkčních požadavků. Využívá se jako:

- detailní specifikace požadavků na systém,
- základ pro tvorbu uživatelské příručky,
- podklad k tvorbě akceptačních testů,
- podklad pro zpřesnění odhadů pracnosti.

A je složen ze:

- seznamu účastníků,
- diagramu případů užití,
- popisu případů užití,
- scénářů případů užití (hlavních a alternativních).

Seznam účastníků jmenuje a stručně popíše účastníky, kteří do systém z vnějšku zasahují. Nemusí se vždy jednat o reálné osoby, respektive role, jako např. „Knihovník“, nebo „Zákazník“. V některých případech je účastníkem i jiný systém, nebo např. čas.

Diagram případů užití zobrazuje účastníky a jejich oprávnění. Patří do skupiny diagramů chování.

Případ užití je, dle referenční příručky jazyka UML, „*specifikace posloupností, které systém, podsystém nebo třída může vykonat prostřednictvím interakce s vnějšími (externími) aktéry*“. [16] (překlad Bogdan Kiszka) Aktéry jsou účastníci ze seznamu účastníků.

Scénáře případů užití detailněji popisují interakci systému a uživatele (např. „přihlášení uživatele do systému“). Při tvorbě scénářů se popisuje *co* má systém dělat. Nikoliv však *jak* to má dělat. U běžných scénářů stačí jednoduchý popis. Jednotlivé scénáře označujeme obvykle zkratkou UC, číslem a popisem. např. UC1: první scénář. [9], [17]

2.5 Wireframe

Wireframe, česky tzv. „drátěný model“, je užíván k vytvoření základního návrhu struktury a funkčnosti aplikace. Neklade se důraz na detailní funkčnost ani grafický vzhled aplikace. Jedná se o přibližný náčrt všech obrazovek aplikace. [18]

2.6 Prototyp aplikace

Prototyp mobilní aplikace je v různé míře funkční model, který se tváří jako skutečná aplikace. Ve skutečnosti však nemusí být napsána ani čárka kódu. Základní funkcí je zobrazení vzhledu jednotlivých stránek. Obvykle jsou prototypy „klikací“, což znamená, že lze klikáním měnit jednotlivé stránky. Někdy se lze setkat i s dalšími prvky prototypu, jako jsou např. jednoduché animace, možnost testovat prototyp v mobilním zařízení apod. Prototyp ale nemá vnitřní logiku, jedná se jen o grafické znázornění.

Obvykle jej vytváříme pomocí specializovaného prototypovacího nástroje, kterých je na trhu k dispozici celá řada. Liší se způsobem vytváření prototypů, možnostmi návrhu apod.

Prototyp mobilní aplikace vytváříme po rozpracování funkčních požadavků a načrtnutí wireframe, ale před samotnou implementací. Pokud se postupuje podle unifikovaného procesu vývoje aplikací (viz 2.2.1), spadá implementace prototypu do fáze *zahájení*. [9]

Díky prototypu můžeme aplikaci otestovat a získat zpětnou vazbu ještě dříve, než začneme s programováním. To je klíčová výhoda, protože vývoj a úpravy prototypu jsou mnohem jednodušší, rychlejší a levnější, než úpravy skutečné aplikace. Čím dříve se ve vývojovém procesu objeví nedostatek, tím lépe.

Mezi hlavní výhody prototypu tak patří možnost:

- včas nalézt chyby návrhu,
- rychle porovnat varianty návrhu,
- rychlé zpětné vazby vývojářů i uživatelů,
- použít prototyp jako prezentační nástroj. [12], [19]

Náklady na vývoj prototypu, který obvykle trvá maximálně několik dní, se tak rozhodně vyplatí.

2.7 Nástroje

V této sekci představím hlavní nástroje, které jsem použil. Výběr nástrojů zde již popisovat nebudu.

2.7.1 Justinmind

Nástroj Justinmind [20] použiji pro tvorbu wireframe. Někdy se pro nakreslení wireframe používá pouze papír a tužka. Já jsem z praktických důvodů použil raději specializovaný softwarový nástroj pro tvorbu modelů a prototypů. Podobných nástrojů je na trhu mnoho a vyzkoušel jsem jich hned několik. Justinmind poskytuje profesionální možnosti pro tvorbu wireframe, i pro tvorbu

prototypů. Pro mě dostačující funkcionality nabízí i při bezplatné licenci, a proto jsem jej pro tvorbu wireframe zvolil.

2.7.2 Visual-paradigm.com

Visual-paradigm [21] je nástroj s mnoha funkcemi pro podporu vývoje softwaru. Jedná se např. o modelování diagramů, vytváření analýz apod. Já jsem použil online verzi pro modelování diagramu případů užití, protože pro tuto funkci stačí bezplatná licence.

2.7.3 Mockflow

Ačkoliv prototyp aplikace bych mohl vytvořit i v již zmíněném nástroji Justinmind, na doporučení vedoucího jsem se pro samotné vytvoření prototypu rozhodl použít nástroj Mockflow. [22] Musel jsem zakoupit placenou licenci.

Nástroj Mockflow podporuje návrhy pro různá zařízení od mobilních a webových aplikací, až k chytrým hodinkám, nebo televizi. Nástroj nabízí velké množství elementů, které lze doplnit i vlastními prvky. Elementy lze pak různě upravit, změnit barvu apod. Elementy se pak navzájem kombinují a vytvoří jednu stránku jako celek. Prototyp má obvykle několik stránek a lze vytvořit i „klikací“ odkazy mezi stránkami. Nástroj podporuje také export prototypu v HTML formátu, nebo např. jako PDF.

2.8 SWOT analýza

Jedná se o zkratku čtyř anglických slov:

- **strengths** = silné,
- **weaknesses** = slabé,
- **opportunities** = příležitosti,
- **threats** = hrozby.

SWOT analýza si klade za cíl jednak zanalyzovat externí příležitosti a hrozby, jednak interní silné a slabé stránky. Obvykle se dělí do čtyř kvadrantů (viz obrázek 2.2). Tvoří dobrý základ pro formulování strategie vývoje. Účelem analýzy vnějších příležitostí a hrozeb je zhodnotit, zda podnik může využít příležitosti a vyhnout se hrozbám, které nemůže přímo ovlivnit. Mezi vnější faktory patří například kolísavé ceny, stav trhu, ekonomická situace, politická destabilizace, sociální přechod, změna právního státu atd. Účel analýzy interních silných a slabých stránek je zhodnotit, jak podnik plní svou vnitřní práci, jako je řízení, efektivita práce, výzkum a vývoj, atd. [24]



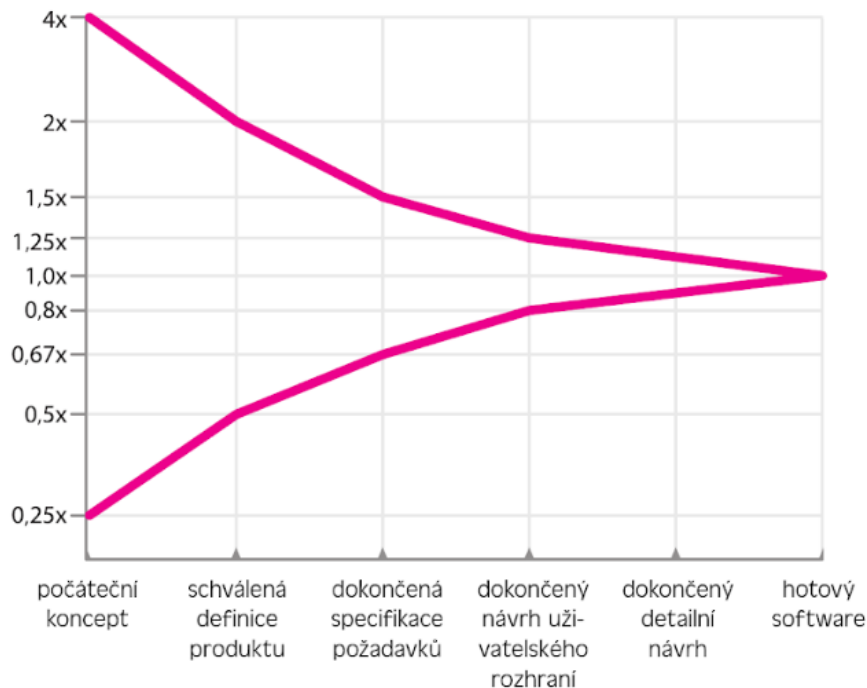
Obrázek 2.2: SWOT analýza [23]

Na webu managementmania.com je uvedeno, že „*SWOT analýza je univerzální analytická technika zaměřená na zhodnocení vnitřních a vnějších faktorů ovlivňujících úspěšnost organizace nebo nějakého konkrétního záměru (například nového produktu či služby).*“ [25] Analýza je využívána při řízení změn strategie ve společnosti, ale i pro méně rozsáhlé aplikace. Např. přípravu strategie dílčího projektu, strategie pro vítězství v soutěži, nebo i lékařské operace.

Přestože se zdá být SWOT analýza jednoduchá, je potřeba dodržovat základní postupy, aby měla skutečně smysl. Existuje řada podrobnějších metod pro tvorbu SWOT analýzy, které mohou pomoci. V zásadě je potřeba se zaměřit pouze na klíčové a důležité faktory, zahrnout pouze fakta a objektivní faktory, ideálně pracovat v týmu a v neposlední řadě vyhodnotit, jak s analýzou dále pracovat. [25], [26], [27]

2.9 Odhad nákladů projektu

Aby investor mohl zvážit, zda se mu vyplatí do projektu investovat, potřebuje znát cenu vývoje. Znat přesnou cenu softwaru před jeho dokončením ale není možné. Proto můžeme pouze odhadovat. Odhad je prezentován jako rozsah. Uvádí se minimum, maximum a expertní předpoklad. A odhady jsou pak v průběhu vyvíjení softwaru čím dál tím přesnější, jak to popisuje tzv. kužel



Obrázek 2.3: Kužel nejistoty [28]

nejistoty (viz obrázek 2.3).

„Jedna chybná implikace z tohoto běžného zobrazení kužele nejistoty je, že to vypadá, jako by trvalo velice dlouho, než se kužel zúží - jako by nebylo možné vyhotovit dost přesný odhad až do chvíle, kdy je projekt skoro hotov. Naštěstí je tento dojem způsoben tím, že jednotlivé milníky jsou od sebe na horizontální ose vzdáleny stejně a přirozeně předpokládáme, že na této ose je kalendářní čas. Ve skutečnosti jsou tyto milníky v rozvrhu projektu spíše na začátku.“ [28]

Píše ve své knize McConnell a dodává, že v prvních 30 % projektu se tak rozptýl přesnosti odhadu snižuje z $4\times$ až na $1,25\times$, kdy je návrh uživatelského rozhraní již dokončený.

Existuje množství různých metod odhadu. Pro přesnější odhad je dobré v projektu použít více metodik a vzájemně je kombinovat. Záleží nicméně na konkrétním projektu, případně na jeho aktuální fázi. Odhadování je účinnější, pokud jsou k dispozici historické údaje z vývoje podobných produktů. Kvalita odhadu velmi záleží i na zkušenostech experta. Metodiky mohou být velkou pomocí, nelze jim ale absolutně důvěřovat. [28], [29]

2.9.1 Kalkulátor nákladů

Nejjednodušší možností, jak odhadnout cenu mobilní aplikace, je použít nějaký kalkulátor nákladů na vývoj mobilní aplikace. Bývají dostupné online zdarma od společností zabývajících se vývojem mobilních aplikací. Po zodpovězení několika základních otázek dostaneme cenu aplikace. Pro osoby neznalé problematiky se může jednat o způsob, kdy během několika minut získají alespoň základní, rámcový odhad ceny aplikace. Takovýto způsob odhadu je nicméně poněkud amatérskou a nepřesnou možností.

2.9.2 Metoda PERT

Metoda PERT – anglicky *Program (Project) Evaluation and Review Technique* – se používá pro výpočet odhadu doby trvání projektu. „Její základ je postaven na váženém průměru a jeho směrodatné odchylce. Metoda PERT vypočítá dobu trvání úkolu ze zadaných trojic vstupních údajů váženým průměrem. Pro výpočet je nutné zadat odhad optimistické, očekávané a pesimistické doby trvání a váhy.“ [30]

Pro výpočet doby trvání jednoho úkolu se používá rovnice 2.1. Pokud je potřeba získat součet doby trvání více úkolů, jednoduše se sečtou jejich jednotlivé výsledky. Použité zkratky jsou vysvětleny v následujícím seznamu:

- T_v – doba trvání úkolu,
- T_o – optimistický časový odhad doby trvání úkolu,
- T_{oc} – reálně očekávaný odhad doby trvání úkolu,
- T_p – pesimistický časový odhad doby trvání úkolu,
- $váha_1$ – váha pro optimistický časový odhad,
- $váha_2$ – váha pro reálně očekávaný časový odhad,
- $váha_3$ – váha pro pesimistický časový odhad.

$$T_v = \frac{T_o \times váha_1 + T_{oc} \times váha_2 + T_p \times váha_3}{váha_1 + váha_2 + váha_3} \quad (2.1)$$

Váhy dávají možnost ovlivnit k jakému odhadu se expert více kloní. Např. pokud se předpokládá pesimistický průběh vývoje projektu, lze nastavit váhy: $váha_1 = 1$, $váha_2 = 1$, $váha_3 = 4$. Součet vah musí vždy nabývat čísla 6. Výchozí nastavení bývá: $váha_1 = 1$, $váha_2 = 4$, $váha_3 = 1$. [30], [31]

Metoda PERT obsahuje ještě další části a možné výpočty. Pomocí této metody lze také např. vypočítat s jakou pravděpodobností bude projekt ukončen v zadaném čase. Pro rozsah bakalářské práce však postačí výpočet doby trvání úkolu.

2.9.3 Use case points

Use case points (UCP) je metoda odhadování vytvořena pro vývojáře, kteří pro vývoj softwaru používají Unified Modeling Language (UML), nebo Rational Unified Process (RUP). Metodu vyvinul Gustav Karner roku 1993. Jak název napovídá, metoda je založena na odhadování na základě Use Case (UC) (viz odstavec v sekci 2.4). Pro použití této metody tedy je potřeba již mít sepsány Use Case. [32]

Podle [32], [33], [34] se při výpočtu metodou UCP postupuje v následujících krocích:

- **krok 1 – výpočet čísla UUCW:**

Každému UC se přiřadí váha podle tabulky 2.1. Váhu lze vybrat buď na základě počtu transakcí, na základě předpokládaného počtu tříd, nebo zvážení obou faktorů. Např. tedy pro UC1 se odhadne 6 transakcí a váha pro UC1 je tedy 10. Sečtením všech UC vah se získá číslo UUCW (Unadjusted Use-Case Weight). Tedy $UUCW = \sum_{k=1}^n váhaUCk$, kde n je celkový počet UC.

- **krok 2 – výpočet čísla UAW:**

Každému aktérovi v systému je přiřazena váha podle tabulky 2.2. Sečtením všech vah aktérů se získá číslo UAW (Unadjusted Actor Weight). Platí tedy rovnice: $UAW = \sum_{k=1}^n váhaAktéra_k$, kde n je celkový počet aktérů.

- **krok 3 – výpočet čísla UUCP:**

Číslo UUCP (Unadjusted Use-Case Points), které vyjadřuje neupravenou velikost systému, se získá prostým součtem UAW a UUCW. Platí tedy rovnice: $UUCP = UUCW + UAW$.

- **krok 4 – výpočet čísla TFactor:**

Každý ze 13 faktorů z tabulky 2.3 je ohodnocen celým číslem z rozsahu 0–5, kde 0 znamená nerelevantní a 5 znamená velmi důležitý.

Pro každý faktor se poté vypočte číslo jeho dopadu vynásobením váhy faktoru s přiřazenou hodnotou. Tedy např. faktor T1, který má fixně danou váhu 2, je ohodnocen důležitostí pro konkrétní systém číslem 3. Dopad faktoru T1 se poté vypočte jako 2×3 . Platí tedy následující rovnice:

$dopadFaktoru = váhaFaktoru \times ohodnoceniFaktoru$. Číslo $TFactor$ (Technical Factor - technický faktor) je potom součet všech čísel $dopadFaktoru$. Platí tedy rovnice: $TFactor = \sum_{k=1}^{13} dopadFaktoru.k$.

- **krok 5 – výpočet TCF:**

Číslo TCF (Technical Complexity Factor) je vypočteno podle rovnice: $TCF = 0,6 + (0,01 \times TFactor)$.

- **krok 6 – výpočet čísla EFactor:**

Postupujeme obdobně jako v kroku 4, jen podle jiné tabulky. Každý z 8 faktorů z tabulky 2.4 je ohodnocen celým číslem z rozsahu 0-5, kde 0 znamená nerelevantní a 5 znamená velmi důležitý.

Pro každý faktor se poté vypočte faktor dopadu vynásobením váhy faktoru s přiřazenou hodnotou: $dopadFaktoru = váhaFaktoru \times ohodnoceniFaktoru$. Číslo EFactor (Environment Factor - faktor prostředí) je potom součet všech čísel $dopadFaktoru$. Platí tedy: $EFactor = \sum_{k=1}^8 dopadFaktoru.k$.

- **krok 7 – výpočet čísla EF:**

Číslo EF (Environmental Factor) se vypočte podle následující rovnice: $EF = 1,4 + (-0,03 \times EFactor)$.

- **krok 8 – výpočet čísla UCP:**

Číslo UCP (Use Case Points) je vypočítáno vynásobením čísel UUCP, TCF a EF. Platí tak rovnice: $UCP = UUCP \times TCF \times EF$. Číslo UCP je výsledek metody Use Case Points.

- **krok 9 – co dál?**

Číslo UCP vyjadřuje jakýsi výsledný faktor této metodiky. Pro získání počtu pracovních hodin projektu je potřeba toto číslo vynásobit počtem hodin potřebných na 1 UCP. Karner, autor metody, doporučuje počítat 20 hodin pro každý UCP. Číslo lze upravit, ale mělo by zůstat v rozsahu 15-30 hodin pro 1 UCP.

| UC komplexnost | počet transakcí | počet tříd | UC váha |
|----------------|-----------------|------------|---------|
| jednoduchá | ≤ 3 | ≤ 5 | 5 |
| průměrná | 4–7 | 6–10 | 10 |
| komplexní | > 7 | > 10 | 15 |

Tabulka 2.1: Metoda UCP: ohodnocení UC

| komplexnost aktéra | příklad | váha aktéra |
|--------------------|---|-------------|
| jednoduchá | Další systém komunikující přes API. | 1 |
| průměrná | Další systém komunikující přes nějaký protokol, nebo člověk komunikující přes textové rozhraní. | 2 |
| komplexní | Člověk komunikující přes uživatelské rozhraní (GUI). | 3 |

Tabulka 2.2: Metoda UCP: ohodnocení aktérů

2.10 Grafické rozhraní aplikace

Téměř každá pátá stažená mobilní aplikace je použita uživatelem pouze jednou během prvních 6 měsíců od stažení. [35], [36] Pokud aplikace „padá“, „zamrzá“ apod., ani ne 20 % uživatelů ji dá více šancí než dvě. [37] Ze zahraničních statistik se také dozvídáme, že 21 % Američanů ve věku 18-34 let někdy smazalo aplikaci, protože se jim nelíbila ikona na domovské obrazovce. [38]

Uživatelé, jak vyplývá z výše uvedených statistik, jsou nároční. Vyvinout uživatelsky oblíbenou aplikaci není jednoduché. A zdaleka nezáleží jen na samotných funkcionalitách aplikace.

Touto problematikou se zabývá tzv. UX Design. UX (user experience) je označení pro uživatelskou přívětivost/zkušenost. „*Pocity a reakce, které mají uživatelé při interakci s produktem (...), se nazývají uživatelská zkušenost (...). UX je to, co uživatel pocítí a pamatuje si v důsledku použití aplikace, systému nebo webu.*“ UX Design úzce souvisí s UI (User Interface) a GUI (Grafical User Interface), což je uživatelské rozhraní, respektive grafické uživatelské rozhraní. „*Pomocí takového rozhraní může uživatel komunikovat se zařízením,*

2. TEORETICKÁ ČÁST

| faktor | popis | váha |
|---------------|--------------------------------|-------------|
| T1 | distribuovaný systém | 2 |
| T2 | doba odezvy | 1 |
| T3 | efektivita koncového uživatele | 1 |
| T4 | složitost vnitřních operací | 1 |
| T5 | znovupoužitelnost kódu | 1 |
| T6 | jednoduchá instalace | 0,5 |
| T7 | snadné použití | 0,5 |
| T8 | přenositelnost | 2 |
| T9 | jednoduchá změna systému | 1 |
| T10 | souběžnost | 1 |
| T11 | bezpečnostní funkce | 1 |
| T12 | přístup pro aplikace 3. stran | 1 |
| T13 | speciální trénink uživatelů | 1 |

Tabulka 2.3: Metoda UCP: faktory technické komplexnosti

| faktor | popis | váha |
|---------------|--|-------------|
| E1 | obeznámenost se zvoleným vývojovým procesem | 1,5 |
| E2 | zkušenost s obdobnými aplikacemi | 0,5 |
| E3 | zkušenost s objektově orientovaným programováním | 1 |
| E4 | schopnosti vedoucích analytiků | 0,5 |
| E5 | motivace týmu | 1 |
| E6 | stabilní požadavky na systém | 2 |
| E7 | zaměstnanci na částečný úvazek | -1 |
| E8 | obtížné programovací jazyky | -1 |

Tabulka 2.4: Metoda UCP: faktory prostředí

počítačem a programy.“ [39]

2.10.1 Jakob Nielsen a jeho přístup

Jednou z ikon v oblasti UX Design je Jakob Nielsen, narozen v Kodani. Je významnou osobností v oboru použitelnosti webů (web page usability), byl označen jako „guru webové použitelnosti“ (The New York Times), „nejchytřejší osoba na webu“ (ZDNet AnchorDesk), nebo „světově vedoucí expert na webovou použitelnost“ (U.S. News & World Report). [40] Ačkoliv se Nielsen zaměřuje převážně na webové aplikace, jeho obecná pravidla lze jistě použít i pro mobilní aplikace. Navíc svět webových a mobilních aplikací není tak moc rozdílný a je značně propojený.

Nielsen vyslovil tzv. *Jakob's Law* (Jakobův zákon): „*Uživatel stráví více času na jiných webových stránkách, než na tvé webové stránce.*“ (překlad autora). Z čehož plyne, že pokud je naše webová stránka navržena podobně jako ostatní, uživatel který přijde na naši stránku, rychle ví, jak ji používat a zaměří se na to podstatné – naše produkty, služby, nabídky, obsah, sdělení apod. [41]

Nielsen popsals 10 bodů (heuristik) použitelnosti pro UI Design:

1. Viditelnost aktuálního stavu systému – systém by měl vždy informovat uživatele, co se děje.
2. propojení systému a reálného světa – systém by měl přizpůsobit svůj jazyk a způsob komunikace uživateli.
3. Uživatelská kontrola a svoboda – uživatel často zvolí nějakou funkci omylem a potřebuje rychle odejít z nežádaného dialogu, vrátit zpět nechtěnou aktivitu apod. To mu musí systém efektivně umožnit.
4. Konzistence a standardizace – uživatel by si neměl klást otázku, zda různá slova, akce a situace znamenají totéž. Systém by měl dodržovat konvence dané platformy.
5. Prevence chyb – je potřeba co nejvíce eliminovat chyby, zabránit vzniku potenciálního problému.
6. Rozpoznávání namísto vzpomínání – systém by měl co nejméně zatěžovat uživatelskou paměť. Uživatel by neměl být nucen pamatovat si informace z předchozího dialogu apod.
7. Flexibilita a efektivita používání – systém by měl být dobře použitelný pro zkušené i nezkušené uživatele. Různé urychlovací elementy mohou zkušenému uživateli urychlit interakci se systémem.
8. Estetický a minimalistický design – systém by měl obsahovat co nejméně zbytečných informací.

9. Pomoci uživatelům rozpoznat, diagnostikovat a vzpamatovat se z chyb – chybové hlášky by měli jednoduše vyjádřit problém a konstruktivně navrhnout řešení.
10. Pomoc a dokumentace – přestože nejlepší je používat systém bez dokumentace, může být nezbytně potřebné dokumentaci a další pomoc poskytnout. [42]

2.11 Závěr

Z této kapitoly plyne potřeba zvolit nativní, hybridní, nebo webový způsob vývoje. Popsal jsem celý proces, jakým aplikace při vývoji prochází a nástroje, které použiji. Nejprve musím zanalyzovat požadavky, na jejichž základě poté navrhnu případy užití, wireframy a prototyp aplikace, který otestuji. Na závěr podle popsaných metod odhadnu náklady na vývoj aplikace, zhodnotím přínosy a provedu SWOT analýzu. Investovat čas do kvalitní analýzy, návrhu a vývoje prototypu se vyplatí, dobré GUI je klíčové. Popsal jsem základní heuristiky Jakoba Nielsena. V souladu s nimi budu postupovat.

Analýza

Tato kapitola je prakticky zaměřená. Vysvětluje zvolený postup vývoje a popisuje konkrétní požadavky zadavatele. Kapitola dále obsahuje analýzu existujících řešení a analýzu funkčních a nefunkčních požadavků.

3.1 Zvolený způsob vývoje

Pro navrhovanou aplikaci *Dorůstáček* jsem zvolil nativní způsob vývoje. Jelikož Android je nerozšířenější operační systém v cílové skupině, bude aplikace nejprve vyvinuta pro Android. Nativní aplikace nenabízí dostatek možností pro kvalitní, rychlou a graficky dobře přizpůsobenou aplikaci. Hybridní vývoj je pro *Dorůstáček* velmi zajímavá volba a určitě stála za zvážení. V aplikaci se totiž nebudou používat složité animace, ani nějak zvlášť náročné výpočetní operace apod. Nativní vývoj jsem nakonec zvolil z následujících důvodů:

1. Vzhledem k cílové skupině dětí do 15 let předpokládám, že nemalé procento uživatelů bude používat telefon se starším operačním systémem. U hybridní aplikace by se mohly vyskytnout problémy s kompatibilitou, případně i rychlostí.
2. Je klíčové, aby byla aplikace velmi intuitivní a jednoduchá na ovládání. Absence nativních UI prvků u hybridního vývoje by byla omezující.
3. V cílové skupině očekávám ještě větší zastoupení operačního systému Android, než v uvedených globálních statistikách. Vývoj Android nativní aplikace tedy bude v první fázi projektu stačit a po zhodnocení úspěchu se až v další fázi projektu vyvine aplikace pro iOS.

3.2 Požadavky zadavatele

Možnost podílet se na vývoji mobilní aplikace jsem dostal od vedení Dorostové unie. Dohodli jsme se, že v rámci bakalářské práce aplikaci konkrétně navrhnu

3. ANALÝZA

a rozpracuji.

Mobilní aplikace *Dorůstáček* je primárně navrhována pro konkrétní cílovou skupinu – křesťanské dorosty. Členové dorostů jsou obvykle děti věku 11–16 let, kteří se pod vedením několika dospělých vedoucích scházejí obvykle každý týden a někdy podniknou speciální akce – víkendové aktivity, tábory apod. Dorosty jsou pak sdruženy pod celorepublikové vedení, které dorosty podporuje, poskytuje materiály, organizuje různé akce a sjezdy pro dorosty, vyučuje vedoucí a podobně. Věřím že aplikaci by ale využily i další skupiny uživatelů. Navrhnu tedy aplikaci způsobem, který bude umožňovat jednoduchou modifikaci pro cílové uživatele.

Komunikoval jsem s Mgr. Josefem Kosteckým. [43] Kromě obecného konceptu mi sdělil i některé konkrétní požadavky. Ty jsou uvedeny v následujícím seznamu:

- registrace a přihlašování uživatele e-mailem,
- vyhledávání přátel přes telefonní čísla (podobně jako aplikace „what-sapp“),
- možnost sdílet s přáteli své úkoly,
- podpora sdílení v rámci jednoho konkrétního klubu,
- možnost používat aplikaci offline,
- několik pevně daných návyků,
- možnost uživatele přidat si i vlastní návyky,
- upomínky pro vyplnění,
- možnost zvolit si různé barvy pro různé úkoly,
- možnost uživatele zvolit si cíl, kterého chce dosáhnout,
- pochvala pro uživatele za zvládnutí úkolů,
- statistiky denních, týdenních a měsíčních výsledků,
- jednoduché, vtipné a přehledné motivační grafické rozhraní,
- nějakým způsobem zapojit i vedoucí konkrétního jednoho klubu, který by přidával úkoly pro jeho klub.

3.3 Existující řešení

Existující řešení jsem analyzoval ze dvou důvodů. Jednak abych nevyvíjel aplikaci, která již existuje. Jednak abych se mohl inspirovat z podobných existujících aplikací a na jejich provedení navázat.

Jen ve svém osobním mobilním telefonu jsem objevil 4 aplikace, které mají nějakým způsobem motivační funkci. Aplikací, které mají své uživatele motivovat, je na Google Play ke stažení nejméně několik desítek. Nemalý počet aplikací se zaměřuje na motivační citáty, takovými aplikacemi se zabývat vůbec nebudu.

Nejprve jsem vytvořil seznam 19 uživatelsky rozšířených motivačních aplikací, které by pro mě mohly být zajímavé. Aplikace se v zásadě snaží motivovat uživatele k dosažení dlouhodobých úkolů které si zvolí, pomoci mu soustředit se, změnit dlouhodobé návyky, nebo mít kvalitní životní styl. Některé motivují již ke konkrétní činnosti (četba, sport). Každou z devatenácti aplikací jsem nainstaloval, prohlédl si ji a vyzkoušel.

Z vyzkoušených aplikací byly čtyři velmi blízké aplikaci, kterou budu navrhopvat, proto je blíže porovnám. Zabývaly se změnou dlouhodobých návyků, a jsou založené na stejném principu denního posuzování splnění cílů. I dalšími aplikacemi se ale můžu inspirovat – jejich provedením, kreativitou a způsobem motivace.

Například z vyzkoušených aplikací nejlépe uživatelsky hodnocená aplikace *Productivity Challenge Timer* [44] vyniká vtipnými hláškami, aplikace *Zombies, Run!* [45] motivuje běžce pocitem, že je honí zombie, několik aplikací motivuje uživatele sbíráním bodů pro charitu či vysazováním stromů v Africe, aplikace *Habitica: Gamify Your Tasks* [46] zase motivuje uživatele formou RPG hry. Aplikace *Fabulous - Self Care* [47] pak vyčnívá komplexností a profesionálním zpracováním.

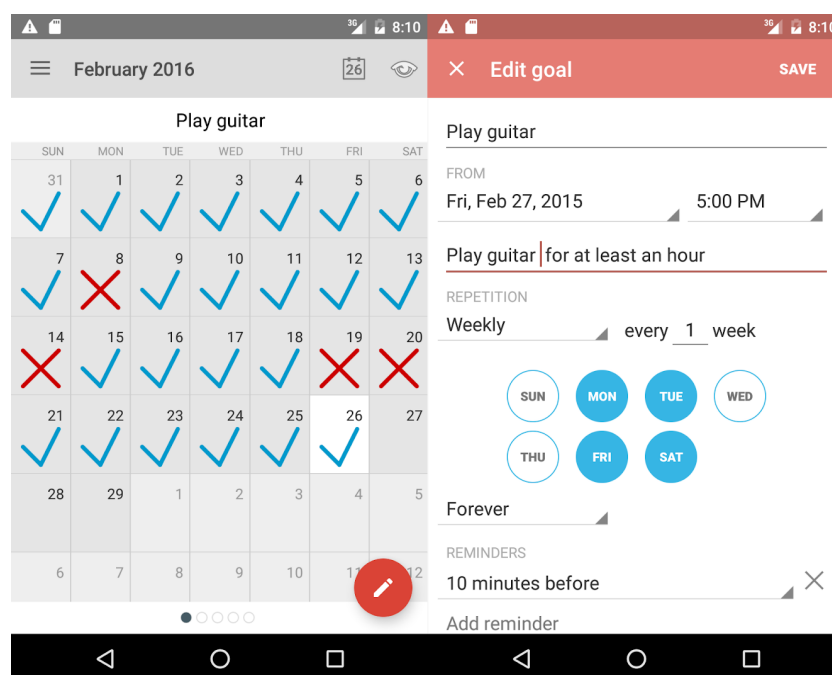
3.3.1 Mé cíle, návyky a předsevzetí

Princip aplikace *Mé cíle, návyky a předsevzetí* [48] spočívá v nastavení cílů, respektive návyků. Při zadání cíle uživatel zvolí, jak často (které dny, nebo kolik dní v týdnu, či každý několikátý den) je nutné návyk splnit. Pro každý den uživatel následně zaškrtně, zda návyk splnil, či nikoli. Aplikace má měsíční, týdenní a detailní zobrazení. V týdenním a detailním zobrazení může uživatel vidět několik návyků pod sebou. Na obrázku 3.1 lze vidět měsíční zobrazení a možnosti při přidávání nového návyku.

Aplikace je z blíže porovnávaných aplikací uživatelsky nejlépe hodnocená, přívětivá, jednoduchá, velmi přehledná a efektivní. V základních funkcích je velmi podobná aplikaci, kterou budu navrhopvat. Je tedy jistě vhodné z aplikace *Mé cíle, návyky a předsevzetí* při návrhu vycházet.

Nevýhodou může být, že aplikace vůbec nepodporuje nějaké sdílení, design nezaujme a chybí grafické znázornění řetězení splněných úkolů. Uživateli tak

3. ANALÝZA



Obrázek 3.1: Ukázky z aplikace „Mé cíle, návyky a předsevzetí“ [48]

chybí motivace neporušit návyk, aby neporušil rozvinutý řetěz. Aplikace dobře splňuje základní funkce, ale nenabízí žádnou přidanou hodnotu.

3.3.2 Habit Tracker

V aplikaci *Habit Tracker* [49], na rozdíl od výše zmíněné aplikace, jsou navíc následující možnosti: zobrazování citátů a statistik pod kalendářem splnění, možnost navolení různých barev pro konkrétní návyky, možnost zadání denní odpovědi i formou čísla (např. 5 hodin učení), možnost příjmu vstupních dat z aplikace *Google Fit*, dostupné widgety a bledě znázorněný den nesplnění návyku v případě, že splněn být nemusel.

Aplikace *Habit Tracker* je navíc jedinou z blíže porovnávaných aplikací, které podporují nějakou formu sdílení – i když poměrně omezenou formou. Komunita funguje na jednoduchém způsobu sdílení příspěvků a možnosti jejich komentování a hodnocení. Nelze tedy přímo vidět úspěchy dalších uživatelů, nelze ani sledovat příspěvky konkrétního přítele.

Posledním zajímavým rozdílem je grafické znázornění řetězu úspěchů. Mezi úspěšnými dny se graficky zobrazuje pruh a uživatel je tak motivován k tomu, aby řetěz nepřerušil.

3.3.3 Návykovač

Další alternativou je aplikace *Návykovač* [50]. Nabízí pouze základnější funkce. Přestože se jedná o jednoduchou aplikaci, její ovládání pro mě nebylo intuitivní. Zobrazení měsíčního kalendáře pro jeden konkrétní návyk se aktivuje dlouhým stisknutím návyku, což ale nikde v aplikaci není naznačeno. Nelze navíc zobrazit historii úspěchů u více návyků najednou. S uživatelským hodnocením na *Google Play* 4,0 hvězdičky je *Návykovač* nejhůře hodnocenou aplikací ze čtyř blíže zkoumaných.

Nicméně, po překonání zprvu náročnějšího ovládání aplikace, *Návykovač* splní základní funkce dobře. Výhoda aplikace je v rychlém zápisu denní sebereflexe, možnosti upomínek a možnosti zápisu neutrálního puntíku, který nereprezentuje ani úspěch, ani neúspěch. Neutrální puntík uživatel použije pokud prakticky nemohl daný návyk splnit, například byl nemocný.

3.3.4 HabitHub – Habit and Goal Tracker

Poslední srovnávanou aplikací je *HabitHub – Habit and Goal Tracker* [51]. Aplikace nabízí kvalitně zpracovanou většinu již zmíněných funkcí. Zaujme povedeným designem s možností zvolit si vlastní téma. Podporuje graficky znázorněné řetězení úspěchů (viz obrázek 3.2), měsíční a týdenní zobrazení, barevné rozlišení návyků, upomínky, zálohování. Aplikace též nabízí různé možnosti opakování hodnocení návyku a jejich nastavení je velmi intuitivní a přehledné. Nevýhodou je, že plná verze aplikace není dostupná zdarma.

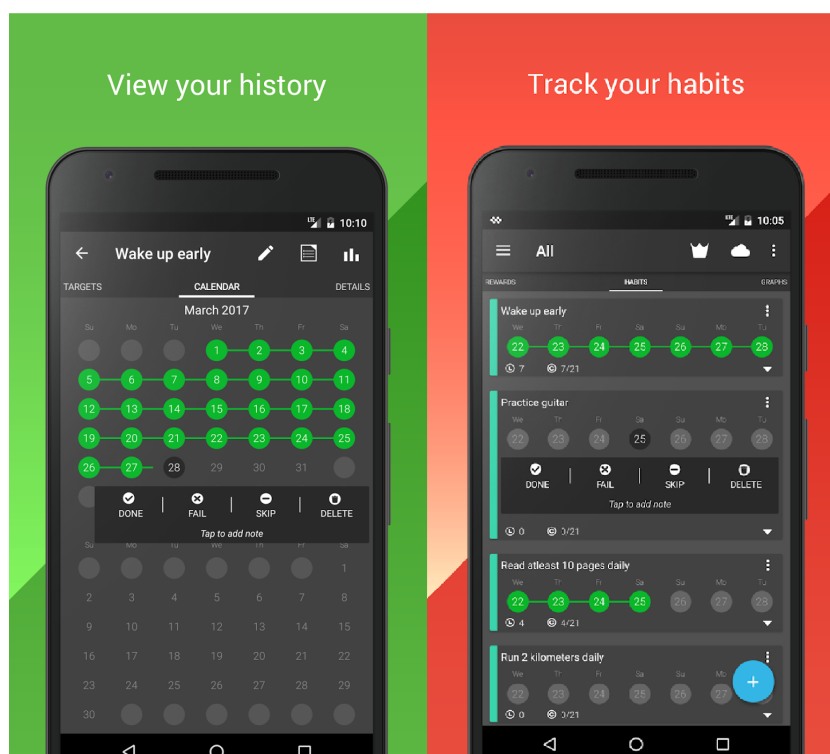
Zajímavou funkcí aplikace je možnost nastavit si kromě návyků také vlastní odměnu. Za úspěšné zvládnutí návyků uživatel dostává body. Za body si pak koupí odměnu, již si předtím sám nastavil.

3.3.5 Závěr

Dobře zpracovaných aplikací, které formují návyky uživatele pomocí denní zpětné vazby, již několik existuje. Jsou poměrně často stahované a dobře hodnocené. Budu se jimi inspirovat a navážu na ně.

Zadávání návyků a jejich základní zobrazování není důvod implementovat zásadně jinak. Dále je mnoho zajímavých funkcí, které stojí za zvážení: připomínkování, zobrazování citátů, neutrální puntík, systém odměn, statistiky, možnost příjmu dat z jiné aplikace, konkrétní design a další.

Zásadním rozdílem pro navrhovanou aplikaci *Dorůstáček* bude zaměření na užívání v komunitě s cílovou věkovou skupinou a sdílení v rámci komunity. Navrhnou tedy aplikaci s podobnými funkcemi jako má *Mé cíle, návyky a předsevzetí* a přidám funkce pro sdílení v rámci komunity. Design bude jednoduchý a atraktivní, aby se aplikace dala především jednoduše a rychle používat, ale zároveň design zaujal.



Obrázek 3.2: Ukázky z aplikace „HabitHub – Habit and Goal Tracker“ [51]

3.4 Funkční požadavky

Funkční požadavky vychází z požadavků zadavatele a z řešerše existujících aplikací. Ačkoliv zadavatel navrhoval přihlašování pomocí e-mailu, kvůli požadavku vyhledávání přátel přes telefonní čísla je praktičtější provést i registraci přes telefonní číslo.

3.4.1 Registrace

F1: Způsob registrace

Uživatel se při prvním přihlášení registruje pomocí telefonního čísla, které je ověřeno zasláním SMS.

F2: Účet „Účastník“

Při registraci se každý účet registruje jako základní členský účet.

F3: Účet „Vedoucí“

V aplikaci bude možnost zažádat zdarma o povýšení účtu účastníka na účet vedoucího. Pokud administrátor žádost schválí, stane se z členského účtu účet vedoucího. Každý oddíl může mít pouze jednoho vedoucího.

Uživatelé s účastnickým účtem mohou následovat jednoho vedoucího. Účet vedoucího má stejné možnosti jako účastnický účet a navíc může přidávat úkoly a sdílet oznámení všem svým následovníkům. Vedoucí už ale nemůže následovat dalšího vedoucího.

3.4.2 Online sdílení

F4: Vyplňování offline

Aplikace musí fungovat i offline. V offline režimu může uživatel vyplňovat úspěšnost návyků, přidávat si nové návyky a zobrazovat historii a statistiky. Po připojení k síti se aktualizují sdílená data.

F5: Vyhledávání přátel

Každý uživatel může dle telefonního čísla vyhledat svého přítele. Aplikace nebude sama nabízet potenciální přátele.

F6: Sdílení s přáteli

Uživatel sdílí se svými přáteli výsledky plnění návyků. Tedy u každého přítele může uživatel vidět, jak se mu daří. Úspěšnost přátel se zobrazuje pod zobrazeními vlastních úspěchů.

3.4.3 Vyplňování a zobrazování návyků

F7: Vyplňování úspěšnosti návyku

Úspěšnost splnění denních návyků si uživatel vyplní sám. Vyplňování je možné až 3 dny zpětně. Každý návyk lze vyplnit jako „splněn“, „nesplněn“, nebo „neutrální“.

F8: Denní zobrazení

V denním zobrazení uživatel vidí všechny své návyky pod sebou a jednoduše může vyplnit jejich úspěšnost pro aktuální den.

F9: Týdenní zobrazení

V týdenním zobrazení uživatel na jedné obrazovce vidí pro každý svůj návyk kalendář pro aktuální týden. Návyky jsou zobrazeny pod sebou. I v týdenním zobrazení lze vyplňovat úspěšnost splnění návyku.

F10: Měsíční zobrazení

V měsíčním zobrazení uživatel vidí kalendář pro celý aktuální měsíc. Ovšem pouze pro jeden zvolený návyk. I v měsíčním zobrazení lze vyplňovat úspěšnost splnění návyku.

F12: Zobrazení statistik úspěchů

Aplikace poskytuje uživateli zpětnou vazbu i formou grafického zobrazení různých statistik úspěšnosti. Kromě osobních statistik uživatele se zobrazují i statistiky členů jednoho konkrétního klubu – tedy všech následovníků jednoho vedoucího.

3.4.4 Přidávání návyků

F13: Nastavení opakování

Při vytváření návyku lze nastavit jak často jej má uživatel splnit. Některé návyky tedy nemusí být splněny každý den. Lze vyplnit konkrétní dny v týdnu (např. každé pondělí a čtvrtek), nebo maximální možný počet vynechaných dnů (např. každý 2. den).

F14: Nastavení barvy

U každého návyku lze z vybrané škály barev nastavit pro návyk vlastní barvu.

F15: Nastavení připomínky

Ke každému návyku lze nastavit připomínání vyplnění.

F16: Nastavení názvu a popisu

Ke každému návyku musí uživatel zadat název. Volitelně může přidat také popis zvyku.

F17: Nastavení cílových uživatelů

Pokud návyk zadává vedoucí, musí navíc nastavit, zda přidávaný návyk je pouze pro něj osobně, nebo i pro všechny jeho následovníky.

3.4.5 Ostatní

F18: Pevně dané návyky

V rámci aplikace je několik návyků již přednastavených, pevně daných pro všechny. Ihned po instalaci aplikace má tedy uživatel již nastavených několik návyků.

F19: Editace návyku

Již vytvořené návyky lze editovat. U návyků, které jsou pevně dané v rámci aplikace, nebo které se uživateli zobrazují kvůli vedoucímu kterého následuje, lze měnit pouze barva a upomínky.

F20: Oznámení vedoucího

Pokud má uživatel přidaného svého vedoucího, v menu se mu objeví navíc odkaz na zobrazení „Oznámení“. Vedoucí může prostřednictvím této funkce oznamovat následovatelům důležité oznámení. Například co si mají přinést na setkání, nebo co se naučit. Vedoucí má v menu místo odkazu na zobrazení „Oznámení“ odkaz na „Přidat oznámení“.

3.4.6 Nastavení

 F21: Globální upomínka

V rámci aplikace lze nastavit jedno hlavní připomenutí pro vyplnění všech návyků.

 F22: Nastavení upomínek

Uživatel bude moci nastavit zvuk a vibrace upomínek.

 F23: Nastavení profilového obrázku

Uživatel bude moci dobrovolně nahrát profilový obrázek. Jinak se zobrazuje jen neutrální ikona uživatele.

3.5 Nefunkční požadavky

 N1: Grafické rozhraní

Grafické rozhraní musí být pro uživatele jednoduché a intuitivní. Bude vytvořeno podle standardů Android. Design ale nesmí působit stroje – do aplikace budou přidány hezké grafické prvky.

 N2: Zobrazení řetězu úspěchů

V týdenním a měsíčním zobrazení jsou dny graficky propojeny „řetězcem úspěšností“. Řetězec se přeruší až v případě neúspěchu plnění zvyku. Tedy např. pokud je potřeba zvyk plnit každý 2. den, jeden neúspěšný den sérii nepřerušuje.

 N3: Dostupnost

Aplikace bude dostupná ze zařízení se systémem Android. Nebude dostupná jako webová aplikace.

 N4: Podpora systémů

Aplikace vyžaduje Android 4.4 a vyšší.

 N5: Rozšiřitelnost, modifikovatelnost a udržitelnost

Z důvodů jednoduchých úprav v aplikaci by jednotlivé komponenty měly být co nejméně provázány.

□ **N6: Bezpečnost**

Aplikace musí být bezpečná. Osobní data uživatelů musí být bezpečně uložena a chráněna.

3.6 Závěr

V této kapitole jsem pro *Dorůstáček* zvolil nativní způsob vývoje a operační systém Android. Zanalyzoval jsem existující řešení a zjistil, že podobných aplikací pro Android několik existuje. *Dorůstáček* je ale unikátní zaměřením a funkcemi pro podporu komunity. Na základě existujících řešení a požadavků zadavatele jsem vytvořil funkční a nefunkční požadavky. Z funkčních požadavků budu vycházet při tvorbě uživatelských scénářů. Nefunkční požadavky N1 a N2 se projeví až při tvorbě prototypu. Ostatní nefunkční požadavky se projeví až při samotné implementaci aplikace.

Návrh

Tato kapitola je prakticky zaměřená. Zabývá se již konkrétním návrhem aplikace. Popisuje případy užití a aktéry aplikace. Dále je v této kapitole popsána tvorba wireframe.

4.1 Případy užití

Případy užití vychází z předchozí analýzy požadavků. Diagram případů užití je zobrazen na obrázku 4.1).

4.1.1 Seznam aktérů

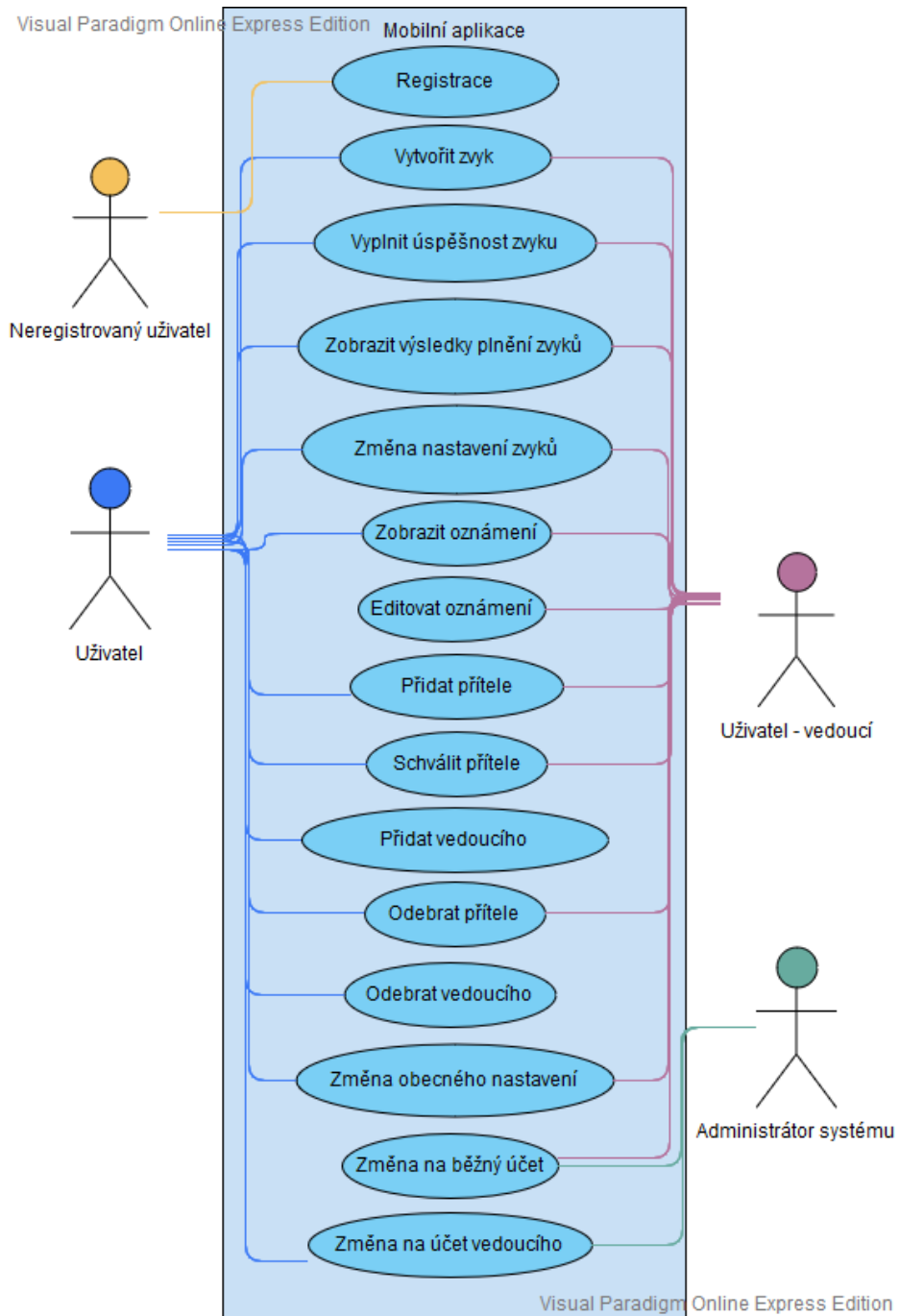
- Neregistrovaný uživatel** nemá prakticky přístup do aplikace. Má pouze takový přístup, aby se mohl zaregistrovat. Neregistrovaný uživatel nemůže aplikaci používat.
- Uživatel** je standardně zaregistrovaný uživatel se všemi základními právy. Při opětovném zapnutí aplikace se nemusí znovu přihlašovat. Uživatel může následovat jednoho svého vedoucího.
- Uživatel-vedoucí** má všechny možnosti uživatele, kromě následování, obohacené o funkce pro vedoucí. Tedy zatímco standardní uživatel může následovat vedoucího, tak vedoucí už nemůže nikoho následovat.
- Administrátor systému** pouze vyřizuje změny účtů z běžného účtu na účet vedoucího.

4.1.2 UC1: Registrace

- **Cíl:**

Cílem je zaregistrovat neregistrovaného uživatele.

4. NÁVRH



Obrázek 4.1: Diagram případů užití

- **Hlavní scénář:**

1. Neregistrovaný uživatel otevře nainstalovanou aplikaci.
2. Uživatel zadá své telefonní číslo.
3. Systém zašle uživateli v SMS autorizační kód.
4. Uživatel zadá autorizační kód a uživatelské jméno a začne aplikaci používat.

4.1.3 UC2: Vytvořit zvyk

- **Cíl:**

Cílem je, aby uživatel vytvořil nový zvyk.

- **Hlavní scénář:**

1. Uživatel se rozhodne přidat nový zvyk.
2. Systém po kliknutí zobrazí formulář umožňující zadat: název, popis, barvu, způsob opakování a připomínky.
3. Uživatel vyplní údaje. Všechny kromě popisu a připomínkování jsou povinné.
4. Jestliže se jedná o vedoucího, musí navíc povinně vyplnit, zda je zvyk pouze pro něj, nebo i pro jeho následovníky.
5. Systém přidá zvyk mezi ostatní zvyky k vyplňování a zobrazování. Pokud zvyk přidal vedoucí i pro následovníky, pak systém přidá zvyk i všem uživatelům, kteří vedoucího následují.

4.1.4 UC3: Vyplnit úspěšnost zvyku

- **Cíl:**

Cílem je, aby uživatel vyplnil u vybraného zvyku a dne, zda zvyk splnil, nesplnil, nebo ani jedno (neutrálně – zvyk nemohl splnit).

- **Hlavní scénář:**

1. Uživatel se rozhodne vyplnit úspěšnost zvyku.
2. Systém uživateli v dennímobrazení (na domovské obrazovce) zobrazí všechny jeho návyky v seznamu.

3. Uživatel vybere daný návyk a jedním kliknutím, případně několika kliknutími, vyplní úspěšnost pro aktuální kalendářní den.

- **1. alternativní scénář:**

1. Uživatel se rozhodne vyplnit úspěšnost zvyku v týdenním zobrazení.
2. Systém uživateli v týdenním zobrazení zobrazí všechny jeho návyky v seznamu. A u každého návyku vidí dny pro celý aktuální kalendářní týden. Uživatel případně může vybrat jiný týden. Uživatel může vyplnit zvyk až 3 dny zpětně.
3. Uživatel v seznamu vybere daný návyk a vybraný den a jedním kliknutím, případně několika kliknutími, vyplní úspěšnost pro vybraný kalendářní den.

- **2. alternativní scénář:**

1. Uživatel se rozhodne vyplnit úspěšnost zvyku v měsíčním zobrazení.
2. Systém uživateli v měsíčním zobrazení zobrazí kalendář vybraného zvyku aktuálního měsíce. Uživatel případně může vybrat jiný měsíc. Uživatel může vyplnit zvyk až 3 dny zpětně.
3. Uživatel vybere kalendářní den a jedním nebo několika kliknutími vyplní úspěšnost pro vybraný den.

4.1.5 UC4: Zobrazit výsledky plnění zvyků

- **Cíl:**

Cílem uživatele je, aby shlédnul výsledky plnění svých návyků, i návyků svých přátel.

- **Hlavní scénář:**

1. Uživatel se rozhodne zobrazit své výsledky a výsledky svých přátel pro aktuální kalendářní den.
2. Systém uživateli v denním zobrazení (na domovské obrazovce) zobrazí denní úspěšnost uživatele a úspěšnost jeho přátel v seznamu pod uživatelovými návyky.

- **1. alternativní scénář:**

1. Uživatel se rozhodne zobrazit historii svých výsledků a výsledků svých přátel v týdenním zobrazení.
2. Systém uživateli v týdenním zobrazení zobrazí historii všech jeho návyků v seznamu pro vybraný kalendářní týden. A pod seznamem uživatelských návyků také seznam a úspěšnost návyků jeho přátel pro vybraný kalendářní týden.

- **2. alternativní scénář:**

1. Uživatel se rozhodne zobrazit historii svých výsledků a výsledků svých přátel v měsíčním zobrazení.
2. Systém uživateli v měsíčním zobrazení zobrazí historii vybraného návyku pro vybraný kalendářní měsíc. A pod tím také seznam a úspěšnost návyků jeho přátel pro vybraný kalendářní měsíc. Nejprve se zobrazí nejrelevantnější návyky přátel. Například pokud někteří přátelé uživatele mají stejný návyk, zobrazí se nejprve historie přátel pro stejný návyk.

- **3. alternativní scénář:**

1. Uživatel se rozhodne zobrazit historii statistiky výsledků a výsledků komunity.
2. Systém uživateli v seznamu zobrazí statistiky osobní i komunitní.

4.1.6 UC5: Změna nastavení zvyků

- **Cíl:**

Cílem je, aby uživatel změnil nastavení vybraného zvyku.

- **Hlavní scénář:**

1. Uživatel se rozhodne upravit nastavení zvyku.
2. Systém zobrazí formulář umožňující upravit nastavení zvyku.
3. Uživatel vyplní údaje. U zvyků, které uživatel nevytvořil sám (byly vytvořeny jeho vedoucím, nebo jsou pevně dané aplikací), může uživatel změnit nastavení připomínek a barvy. U zvyků vytvořených přímo uživatelem lze navíc změnit i nastavení názvu, popisu a způsobu opakování návyku.
4. Systém změní uživatelské nastavení a provede patřičné změny.

4.1.7 UC6: Zobrazit oznámení

- **Cíl:**

Cílem je, aby Uživatel, který následuje nějakého vedoucího, zobrazil jeho oznámení.

- **Hlavní scénář:**

1. Uživatel se rozhodne zobrazit oznámení vedoucího.
2. Systém zobrazí oznámení vedoucího.

4.1.8 UC7: Editovat oznámení

- **Cíl:**

Cílem je, aby Uživatel-vedoucí změnil oznámení pro uživatele, kteří jej následují.

- **Hlavní scénář:**

1. Vedoucí se rozhodne nastavit nebo změnit oznámení.
2. Systém zobrazí stránku s textovým vstupem pro editaci oznámení.
3. Vedoucí vyplní nové oznámení a potvrdí změnu.
4. Systém nastaví nové oznámení. Jestliže již předtím vedoucí nějaké oznámení sdílel, systém jej přepíše.

4.1.9 UC8: Požádat o přátelství

- **Cíl:**

Cílem je, aby Uživatel, nebo Uživatel-vedoucí přidal ke svému účtu nového přítele.

- **Hlavní scénář:**

1. Uživatel se rozhodne požádat o přátelství nového přítele.
2. Systém zobrazí stránku s možností zadat telefonní číslo přítele.
3. Uživatel vyplní telefonní číslo přítele a volbu potvrdí.
4. Systém podle zadaného telefonního čísla přidá vybranému uživateli novou žádost o přátelství.

4.1.10 UC9: Přidat vedoucího

- **Cíl:**

Cílem je, aby uživatel přidal ke svému účtu nového vedoucího.

- **Hlavní scénář:**

1. Uživatel nemá přidaného vedoucího a rozhodne se vedoucího přidat.
2. Systém zobrazí stránku s možností zadat telefonní číslo nového vedoucího.
3. Uživatel vyplní telefonní číslo vedoucího a volbu potvrdí.
4. Systém podle zadaného telefonního čísla přidá nového vedoucího. Nyní se uživateli budou zobrazovat i vytvořené návyky vedoucího, bude moci zobrazit výsledky plnění jeho návyků, i jeho oznámení.

4.1.11 UC10: Schválit přítele

- **Cíl:**

Cílem je, aby uživatel přijal žádost o přátelství.

- **Hlavní scénář:**

1. Uživatel se rozhodne přijmout žádost o přátelství. Rozhodl např. na základě zobrazené notifikace.
2. Systém zobrazí stránku, kde se nachází možnost vybrat ze seznamu uživatele ke schválení.
3. Uživatel vybere přítele a volbu potvrdí.
4. Systém zaregistruje nové přátelství.

4.1.12 UC11: Odebrat přítele

- **Cíl:**

Cílem je, aby uživatel odebral některého ze svých aktuálních přátel.

- **Hlavní scénář:**

4. NÁVRH

1. Uživatel se rozhodne odebrat přítele.
2. Systém zobrazí stránku s možností vybrat ze seznamu aktuálních přátel přítele k odebrání.
3. Uživatel vybere požadovaného přítele a volbu potvrdí.
4. Systém zruší přátelství mezi vybranými přáteli.

4.1.13 UC12: Odebrat vedoucího

- **Cíl:**

Cílem je, aby uživatel přestal následovat svého vedoucího.

- **Hlavní scénář:**

1. Uživatel se rozhodne odebrat vedoucího.
2. Systém zobrazí stránku s možností odebrat vedoucího.
3. Uživatel potvrdí, že opravdu chce přestat následovat svého vedoucího.
4. Systém uživateli odebere vedoucího.

4.1.14 UC13: Změna obecného nastavení

- **Cíl:**

Cílem je, aby uživatel změnil některé ze svých nastavení.

- **Hlavní scénář:**

1. Uživatel se rozhodne změnit obecné nastavení.
2. Systém zobrazí stránku s možnostmi nastavení
3. Uživatel upraví požadované vlastnosti. Může změnit uživatelské jméno, zvuk oznámení, vibrace, výchozí čas připomenutí návyků a profilový obrázek.
4. Uživatel potvrdí provedené změny.
5. Systém uživateli změní nastavení.

4.1.15 UC14: Změna na účet vedoucího

- **Cíl:**

Cílem je, aby uživatel změnil svůj typ účtu na účet vedoucího.

- **Hlavní scénář:**

1. Uživatel se rozhodne změnit svůj účet na účet vedoucího.
2. Systém zobrazí stránku s možností zažádat o účet vedoucího.
3. Uživatel potvrdí svou žádost.
4. Administrátor žádost zhodnotí, zkontroluje, že za daný oddíl je pouze jeden vedoucí a žádost schválí.
5. Systém uživateli změni typ účtu.

4.1.16 UC15: Změna na běžný účet

- **Cíl:**

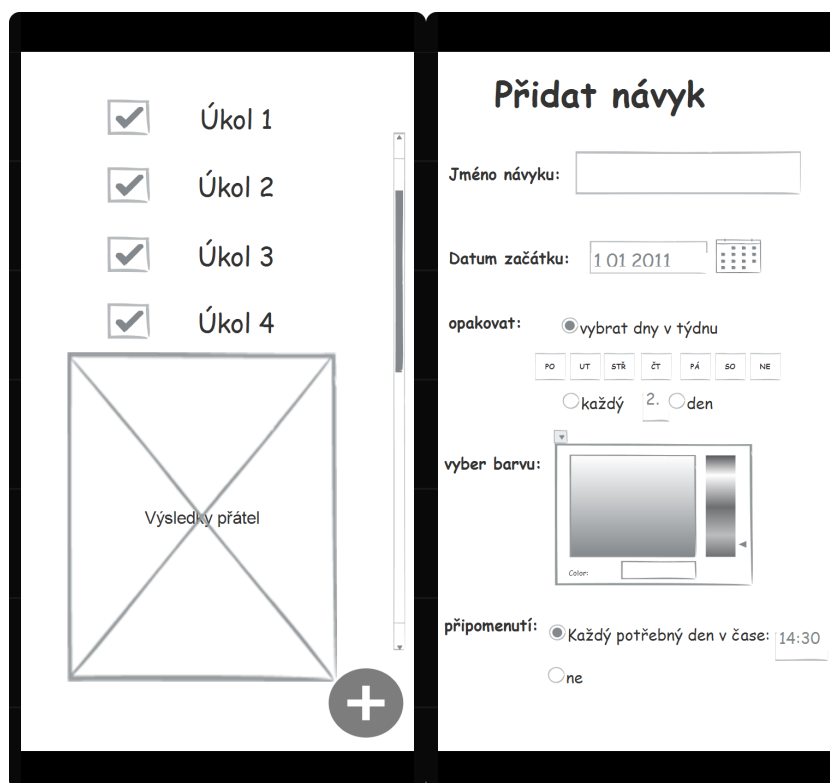
Cílem je, aby uživatel změnil svůj typ účtu z účtu vedoucího na běžný účet.

- **Hlavní scénář:**

1. Uživatel se rozhodne změnit svůj účet na běžný účet.
2. Systém zobrazí stránku s možností zažádat o změnu na běžný účet.
3. Uživatel potvrdí svou žádost.
4. Systém uživateli změni typ účtu.

4.2 Wireframe aplikace

Při tvorbě wireframe jsem vycházel z předchozí analýzy. Stránky jsem navrhl pouze pro orientaci mobilu na výšku, protože předpokládám, že takto se aplikace bude především zobrazovat. Navíc otočení mobilu na šířku nezmění funkcionality stránky, ale pouze se změni způsob zobrazení. Celkem jsem vytvořil 16 náčrtů obrazovek. Všechny wireframe jsou k dispozici v elektronické příloze. Na obrázku 4.2 je vlevo zobrazen wireframe úvodní obrazovky (denní zobrazení) a v pravo wireframe přidávání návyků.



Obrázek 4.2: Wireframe: Denní zobrazení a Přidávání návyků

4.3 Závěr

V této kapitole jsem popsal případy užití. Načrtl jsem také drátěné modely (wireframe) aplikace, které vychází z případů užití a analýzy požadavků. Na základě návrhu vytvořím prototyp.

Vytvoření prototypu

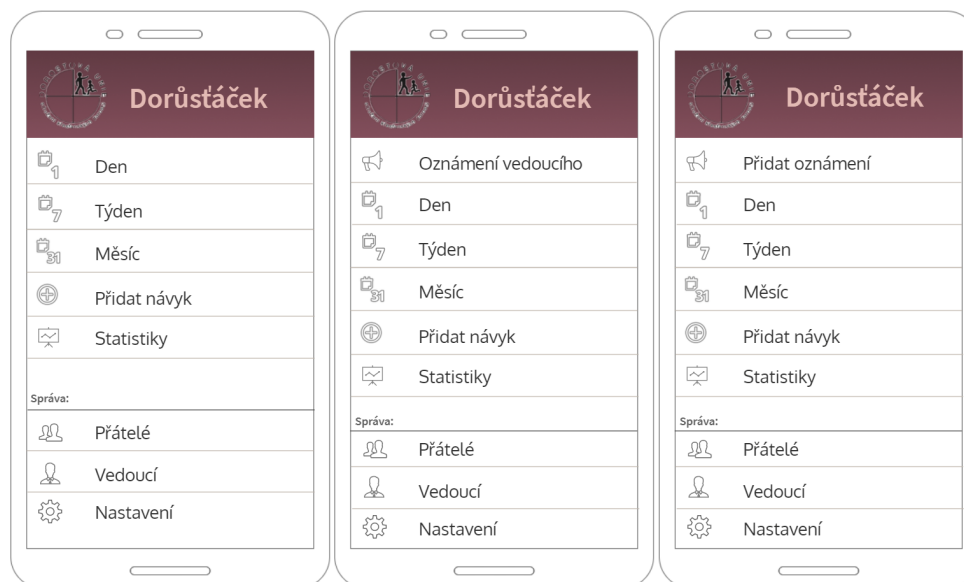
Prototyp jsem vytvořil na základě předchozího návrhu a analýzy. Vycházel jsem z teoretických poznatků o grafickém rozhraní aplikace. Zaměřil jsem se tedy především na to, aby aplikace byla jednoduchá, přehledná a srozumitelná. V rámci vymezených hranic jsem se pak snažil o design, který cílovou skupinu zaujme. Pro návrh prototypu jsem použil prototypovací nástroj Mockflow (viz 2.7.3). Uvedu a popíšu několik zobrazení pro ilustraci základního fungování. Všechny obrazovky prototypu jsou k dispozici v příloze. Celý prototyp je součástí elektronické přílohy.

Na obrázku 5.1 lze vidět 3 různá menu v závislosti na typu účtu. Menu se liší jen podle oznámení. Standardně menu zobrazuje možnosti, které má uživatel ihned po registraci (levá obrazovka). Pokud si uživatel přidá svého vedoucího, přibude v menu možnost zobrazit jeho oznámení (prostřední obrazovka). V případě, že uživatel změnil účet na účet vedoucího, v menu mu přibude možnost přidávat oznámení. Zobrazení menu v prototypu zabírá celou obrazovku, při implementaci se však bude jednat o klasické vysouvací menu.

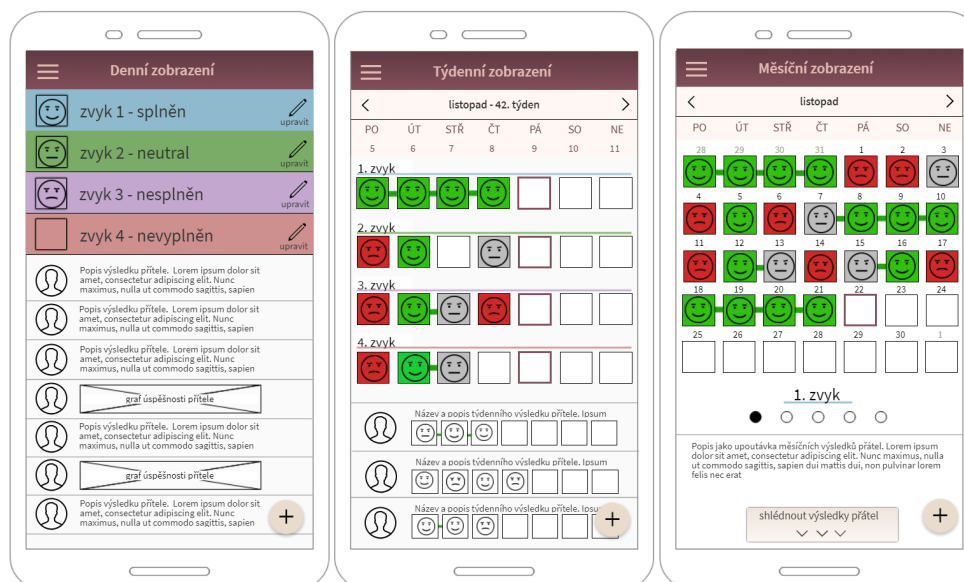
Na obrázku 5.2 lze vidět 3 různé typy zobrazení. Ve všech je možné kliknutím na čtvereček příslušného dne upravit hodnocení. Jednotlivé návyky jsou barevně odlišeny.

Na obrázku 5.3 lze vidět obrazovku zobrazení statistik, přidání nového zvyku a zobrazení oznámení od vedoucího. U statistik jsou nahoře zobrazeny obecné souhrny, níže lze zobrazit statistiky pro konkrétní zvyk.

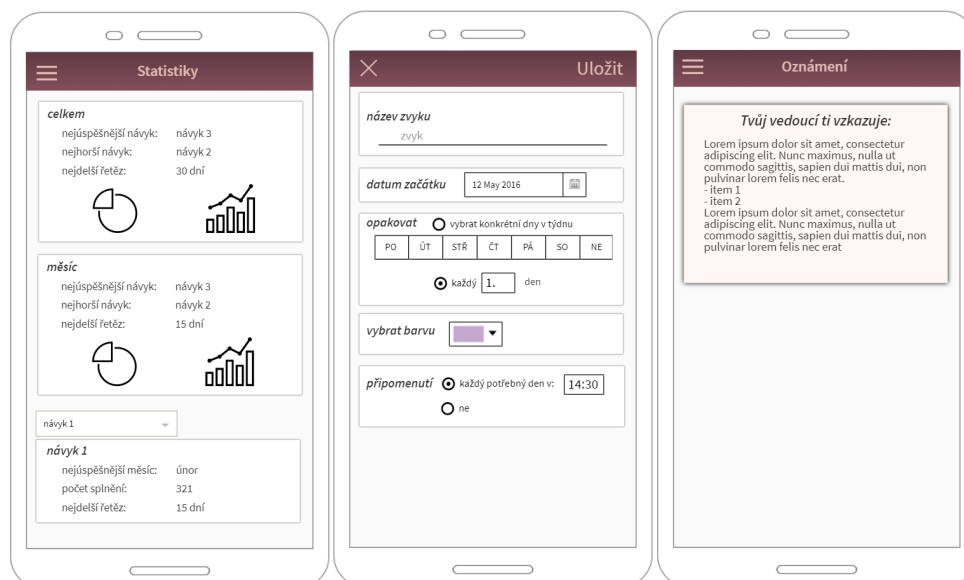
5. VYTVOŘENÍ PROTOTYPU



Obrázek 5.1: Prototyp: (zleva) Standardní menu, Menu následovníka, Menu vedoucího



Obrázek 5.2: Prototyp: (zleva) Denní zobrazení, Týdenní zobrazení, Měsíční zobrazení



Obrázek 5.3: Prototyp: (zleva) Statistiky, Přidávání návyku, Oznámení

Testování

Tato kapitola se zabývá testováním prototypu mobilní aplikace. Popisuje jak testování probíhalo, jaké výsledky přineslo a jaké závěry byly z testování vyvozeny.

6.1 Průběh

Prototyp aplikace testovalo 6 dobrovolníků. Jednalo se o členy křesťanského dorostu, vedoucí, nebo další osoby, které vědí jak dorosty fungují.

Klikací prototyp jsem upravil pro testování, vyexportoval jako HTML z nástroje Mockflow a zpřístupnil na [52]. Testovalo se menu a obrazovky uživatele následovníka (klasický uživatel, který si ale již přidal svého vedoucího). Na úvodní stránce jsem stručně popsal, co má aplikace dělat, a jak funguje prototyp. Poté měl dobrovolník aplikaci projít a otestovat. Někteří dobrovolníci testovali sami online. U některých dobrovolníků jsem byl osobně přítomen a mohl tak navíc sám získat přímou zpětnou vazbu. Po otestování jsem dobrovolníkům položil několik otázek.

6.2 Výsledky

Účel aplikace i způsob používání dobrovolníci rychle pochopili. Ovládání je velmi intuitivní a srozumitelné. Uživatelé chápou, na které elementy lze kliknout a co dělají. Celkově se jim aplikace líbí.

Jeden uživatel vyjádřil nejasnost ohledně přidávání zvyků (kdo návyky přidává) a ohledně nastavení opakování návyku (co znamená nastavení opakování při přidání nového návyku). Dva uživatelé neměli jasno k čemu je neutrální ohodnocení zvyku. Reakce na grafický design byly většinou pozitivní. Pouze 2 dobrovolníci byli spokojeni jen průměrně.

Dostalo se mi také dvou zajímavých postřehů jaké funkce přidat a co vylepšit. Jednalo se o potřebu přidat upomínku pro nové žádosti o přátelství

nebo oznámení a přidání plánu akcí.

6.3 Závěr

Z testování jsem vyvodil následující závěry:

1. Aplikace je celkově dobře navržená. Ovládání je intuitivní.
2. Je potřeba přidat upomínku pro nové žádosti o přátelství nebo oznámení.
3. Je potřeba dovysvětlit některé funkce a způsoby, jak aplikace funguje. Zodpovědět otázky, kdo může přidávat zvyky a jak to funguje?, jak funguje opakování zvyku?, k čemu je neutrální hodnocení? apod. Současné popisky jsou srozumitelné. Je ale potřeba přidat další dovysvětlující elementy.
4. Grafický design je potřeba ještě zkonzultovat s expertem a poté případně upravit. Celkový grafický návrh ale pochopitelně zůstane stejný, bude se jednat o drobné změny a navržení loga a ikony.

Ekonomicko-manažerské zhodnocení

Tato kapitola popisuje použití několika metod pro odhad nákladů na budoucí vývoj aplikace a jaké z odhadů plynou závěry. Shrnuje také náklady na údržbu, možné finanční příjmy, i nefinanční přínosy aplikace. Dále je v kapitole uvedena a vysvětlena SWOT analýza. Na konci kapitoly jsou výsledky ekonomicko-manažerského hodnocení shrnuty.

7.1 Náklady na vývoj

7.1.1 Kalkulátory nákladů

Použil jsem 3 kalkulátory nákladů. V každém jsem odpověděl na několik základních otázek o aplikaci a kalkulátor mi spočítal odhadovanou cenu aplikace, za kterou by mi společnost aplikaci implementovala. Výsledky (viz tabulka 7.1) vyšly v rozsahu 570–660 tisíc českých korun. Společnosti ale jistě nechtějí pouze pokrýt náklady na zaměstnance a provoz. Počítají i s čistým ziskem. Zdůrazňuji, že se jedná o velmi základní a hrubý způsob odhadu, který mi dal jen jakousi rámcovou představu.

7.1.2 PERT

Pomocí metody PERT (viz 2.9.2) se odhaduje doba trvání projektu. Všechny číselné jednotky jsou v tzv. člověkodnech (zkratka MD, anglicky man-day). MD je tedy čas odpovídající jednomu pracovnímu dni jedné osoby. V odhadu jsem již nepočítal s návrhem, tvorbou prototypu apod. Tyto činnosti jsou součástí této bakalářské práce a byly již vypracovány. Budoucí vývoj jsem rozdělil do 6 činností a pro každou odhadl T_o (optimistický odhad času), T_{oc} (očekávaný odhad času) a T_p (pesimistický odhad času). Ze zadaných údajů mi pro každou činnost vyšel T_v (výsledný čas). V tabulce 7.2 jsou zapsány

7. EKONOMICKO-MANAŽERSKÉ ZHODNOCENÍ

zadané údaje a výsledky. Výsledný odhadovaný součet všech činností je roven 67,17 MD.

| Kalkulátor společnosti | Výsledek | přepočet do CZK |
|------------------------|------------|-----------------|
| Z1 [53] | 25 200 USD | 579 600 CZK |
| Oozou [54] | 28 350 USD | 652 100 CZK |
| Yeeply [55] | 21 900 GBP | 652 100 CZK |

Tabulka 7.1: Kalkulátory ceny implementace

| činnost | To | Toc | Tp | Tv |
|-----------------------------|----|-----|----|-------|
| úpravy GUI návrhu | 0 | 1 | 2 | 1 |
| návrh SW architektury | 3 | 5 | 6 | 4,83 |
| implementace GUI | 10 | 20 | 28 | 19,67 |
| implementace vnitřní logiky | 20 | 25 | 36 | 26 |
| implementace databáze | 3 | 5 | 8 | 5,17 |
| testování a ladění | 8 | 10 | 15 | 10,5 |

Tabulka 7.2: Aplikace PERT metody

7.1.3 Use Case Points

Use Case Points (viz 2.9.3) je také metoda, pomocí které se odhaduje délka trvání projektu. Výsledek je počet tzv. UCP bodů. Ty se poté převedou na odhad odpracovaných hodin. Metoda vychází z navržených případů užití (UC). Postupovat budu dle jednotlivých kroků, jak jsou popsány v teoretické části.

- **krok 1 – výpočet čísla UUCW:**

Každému UC jsem v tabulce 7.3 přiřadil váhu podle komplexnosti 5, 10, nebo 15. Sečtením všech UC vah jsem získal výsledek: $UUCW = 85$.

- **krok 2 – výpočet čísla UAW:**

Každému aktérovi jsem v tabulce 7.4 přiřadil váhu podle komplexnosti. Aktéra „Neregistrovaný uživatel“ jsem zanedbal. Sečtením všech vah jsem získal výsledek: $UAW = 7$.

- **krok 3 – výpočet čísla UUCP:**

Pro získání čísla UUCP (Unadjusted Use-Case Points) jsem sečetl čísla UUCW a UAW. Platí tedy: $UUCP = 85 + 7 = 92$.

- **krok 4 – výpočet čísla TFactor:**

Každý z 13 faktorů z tabulky 7.5 jsem ohodnotil celým číslem z rozsahu 0–5 podle důležitosti (5 je nejdůležitější). Pro každý faktor jsem pak vypočetl číslo jeho dopadu vynásobením váhy faktoru s jeho ohodnocením. Sečtením všech výsledných faktorů dopadu jsem získal výsledek: $TFactor = 21,5$.

- **krok 5 – výpočet TCF:**

Číslo TCF (Technical Complexity Factor) jsem vypočítal podle rovnice: $TCF = 0,6 + (0,01 \times 21,5) = 0,815$.

- **krok 6 – výpočet čísla EFactor:**

Každý z 8 faktorů z tabulky 7.6 jsem ohodnotil celým číslem z rozsahu 0–5 podle důležitosti (5 je nejdůležitější). Pro každý faktor jsem pak vypočetl číslo jeho dopadu vynásobením váhy faktoru s jeho ohodnocením. Sečtením všech výsledných faktorů dopadu jsem získal výsledek: $EFactor = 18,5$.

- **krok 7 – výpočet čísla EF:**

Číslo EF (Environmental Factor) jsem vypočítal podle následující rovnice: $EF = 1,4 + (-0,03 \times 18,5) = 0,845$.

7. EKONOMICKO-MANAŽERSKÉ ZHODNOCENÍ

| případ užití | váha |
|-------------------------------------|------|
| UC1: Registrace | 5 |
| UC2: Vytvořit zvyk | 10 |
| UC3: Vyplnit úspěšnost zvyku | 5 |
| UC4: Zobrazit výsledky plnění zvyků | 5 |
| UC5: Změna nastavení zvyků | 5 |
| UC6: Zobrazit oznámení | 5 |
| UC7: Editovat oznámení | 5 |
| UC8: Požádat o přátelství | 5 |
| UC9: Přidat vedoucího | 5 |
| UC10: Schválit přítele | 5 |
| UC11: Odebrat přítele | 5 |
| UC12: Odebrat vedoucího | 5 |
| UC13: Změna obecného nastavení | 5 |
| UC14: Změna na účet vedoucího | 10 |
| UC15: Změna na běžný účet | 5 |

Tabulka 7.3: Metoda UCP: ohodnocení UC – výpočet

- **krok 8 – výpočet čísla UCP:**

Číslo UCP (Use Case Points), což je výsledek metody Use Case Points, jsem vypočítal podle rovnice: $UCP = 92 \times 0,815 \times 0,845 = 63,3581$.

- **krok 9 – co dál?**

Abychom získali počet pracovních hodin projektu, musíme počet UCP vynásobit počtem hodin potřebných na 1 UCP. Autorem je doporučeno počítat 20 hodin pro každý UCP. Číslo je možné změnit, mělo by ale zůstat v rozsahu 15–30 hodin. Jelikož komplexnost systému není velká, snížím číslo na 15. Počet pracovních hodin jsem tedy vypočítal: $63,3581 \times 15 = 950,3715$. Což je asi 119 MD.

| aktér | váha |
|-----------------------|-------------|
| Uživatel | 3 |
| Uživatel-vedoucí | 2 |
| Administrátor systému | 2 |

Tabulka 7.4: Metoda UCP: ohodnocení aktérů – výpočet

| faktor | popis | váha | přiřazená váha | výsledný faktor dopadu |
|---------------|--------------------------------|-------------|-----------------------|-------------------------------|
| T1 | distribuovaný systém | 2 | 1 | 2 |
| T2 | doba odezvy | 1 | 1 | 1 |
| T3 | efektivita koncového uživatele | 1 | 4 | 4 |
| T4 | složitost vnitřních operací | 1 | 1 | 1 |
| T5 | znovupoužitelnost kódu | 1 | 2 | 2 |
| T6 | jednoduchá instalace | 0,5 | 3 | 1,5 |
| T7 | snadné použití | 0,5 | 4 | 2 |
| T8 | přenositelnost | 2 | 1 | 2 |
| T9 | jednoduchá změna systému | 1 | 2 | 2 |
| T10 | souběžnost | 1 | 1 | 1 |
| T11 | bezpečnostní funkce | 1 | 2 | 2 |
| T12 | přístup pro aplikace 3. stran | 1 | 0 | 0 |
| T13 | speciální trénink uživatelů | 1 | 1 | 1 |

Tabulka 7.5: Metoda UCP: faktory technické komplexnosti – výpočet

7. EKONOMICKO-MANAŽERSKÉ ZHODNOCENÍ

| faktor | popis | váha | přiřazená váha | výsledný faktor dopadu |
|--------|--|------|----------------|------------------------|
| E1 | obeznámenost se zvoleným vývojovým procesem | 1,5 | 4 | 6 |
| E2 | zkušenost s obdobnými aplikacemi | 0,5 | 0 | 0 |
| E3 | zkušenost s objektově orientovaným programováním | 1 | 3 | 3 |
| E4 | schopnosti vedoucích analytiků | 0,5 | 3 | 1,5 |
| E5 | motivace týmu | 1 | 4 | 4 |
| E6 | stabilní požadavky na systém | 2 | 4 | 8 |
| E7 | zaměstnanci na částečný úvazek | -1 | 3 | -3 |
| E8 | obtížné programovací jazyky | -1 | 1 | -1 |

Tabulka 7.6: Metoda UCP: faktory prostředí – výpočet

7.1.4 Závěr

Metodou PERT vyšel časový odhad na projekt 67 MD, zatímco metodou Use Case Points 119 MD. Rozdíl není malý. Obě metody pochopitelně závisí na schopnostech experta. Osobně mám malé zkušenosti s odhadováním ceny projektů a nemám k dispozici historická data. Musím tedy počítat s větší mírou nepřesnosti. Metoda Use Case Points se mi zdá být určena spíše pro větší projekty a tak se přikloním k nižšímu číslu a odhadnu výsledný počet MD na 70–90.

Podle [56] je průměrný hrubý plat Android vývojáře v České republice 47 734 Kč. Náklady na jednu pracovní hodinu by tak bez dalších nákladů stály zaměstnavatele asi 400 Kč. S připočtením dalších nákladů budu zjednodušeně počítat 750 Kč jako hodinový náklad, tedy 6 000 Kč jako náklad na jeden pracovní den (MD).

Připočtu ještě dalších 20 tisíc korun na zaplacení externího spolupra-

covníka pro grafický design. Vynásobením odhadnutých člověkodní denním platem programátora získám odhad budoucích nákladů na vývoj 420–540 tisíc korun. Po přičtení zmíněných nákladů na grafický design získám výsledný odhad nákladů 440–560 tisíc korun. V kalkulátorech nákladů vycházela aplikace nejlevněji za 580 tisíc korun. Odhad nákladů vývoje vlastními silami tedy vychází podle očekávání o něco levněji. Zanedbávám náklady na analýzu, návrh a vytvoření prototypu. Tyto náklady jsou součástí bakalářské práce a činily by dalších asi 25 tisíc korun.

7.2 Náklady na údržbu

Kromě nákladů na vývoj a nasazení je třeba zvážit také náklady na následnou údržbu a správu systému. Jedná se o náklady na server a administraci.

Na serveru bude uložena vnitřní logika aplikace (tzv. backend aplikace) a databáze uživatelských dat. Například společnost Snackhost [57] nabízí server s dostačujícím výkonem (20G SSD, 3G RAM, 3vCPU) za 340 Kč měsíčně. Budu s rezervou počítat částku 500 Kč měsíčně.

Administrace aplikace po zaběhnutí zabere odhadem 2 hodiny měsíčně. S hodinovým nákladem externího spolupracovníka 500 Kč odhaduji měsíční náklad na administraci 1000 Kč.

Pro získání celkových nákladů na měsíční údržbu sečtu náklady na server a administraci. Výsledných 1500 Kč je odhad měsíčních nákladů na provoz aplikace.

7.3 Předpokládané příjmy

Dorůstáček není komerční aplikací vyvíjenou za účelem přímého zisku. Vyčíslit příjmy pouze finanční částkou tak není možné, neboť přínosy aplikace jsou spíše nefinančního charakteru. Nicméně je jistě vhodné možné finanční příjmy zvážit a pokrýt tak alespoň část nákladů.

S příjmy přímo od dorostenců, případně členů dalších klubů, nepočítám. Cílová skupina je věkem příliš mladá a odporovalo by to vizi projektu. Peníze na projekt je potřeba pokrýt vlastním kapitálem a získat od dalších organizací, které by aplikaci využívaly. Další část nákladů pokryjí sponzorské dary a státní granty. Organizace samozřejmě můžou finance získat od členů nepřímo, např. formou členských příspěvků, sbírky apod. Dorostová unie tedy bude muset část nákladů zaplatit sama, část pokryje grant a sponzoři a nějaké finance získá ze spolupráce s dalšími organizacemi. S čistým ziskem se nepočítá. Je potřeba pokrýt náklady, které se částečně zaplatí z fondu Dorostové unie.

7.3.1 Finanční příjmy

Jsou 2 možnosti finančních příjmů:

1. jednorázové příjmy od organizací podobného charakteru, které by aplikaci chtěly také používat;
2. udělení státního grantu pro neziskový projekt.

Klíčovým potenciálním partnerem jsou skauti, kteří mají 64 383 členů. [58] Dále jsou možnými partnery např. Royal Rangers, další sportovní a zájmové kluby apod.

7.3.2 Nefinanční příjmy

Výhody, které by aplikace komunitě přinesla, jsou následující:

- motivace k dobrým návykům (čtení, spolupráce atd.),
- růst disciplíny, trpělivosti a cílevědomosti,
- usnadnění efektivní komunikace vedoucích a členů klubu.

7.4 SWOT analýza

SWOT analýza (viz tabulka 7.7) vychází z předchozí analýzy a je podpořena mou osobní znalostí problematiky. Jedná se o komplexní analýzu projektu vývoje a nasazení mobilní aplikace.

SWOT analýza reflektuje skutečnost, že projekt je vytvářen pro neziskovou organizaci. Na hrozby se nabízí jako nejlepší reakce využít silné stránky. Je potřeba aplikaci efektivně propagovat a vytvořit ji na základě znalosti cílové skupiny a obecně znalosti fungování dorostů. Je také potřeba důkladně zvážit náklady, aby projekt například neskončil rozpracovaný a nedokončený.

7.5 Shrnutí

Závěry z ekonomicko-manažerského zhodnocení jsem shrnul do následujících bodů:

- Náklady na budoucí vývoj aplikace činí 440–560 tisíc Kč.
- Měsíční náklady na provoz budou činit asi 1 500 Kč.
- Aplikace nemusí být finančně zisková.
- Aplikace bude pro Dorostovou unii nefinančním přínosem (např. podpora dobrých návyků, růst disciplíny atd.).

- Náklady lze (částečně) pokrýt žádostí o státní grant a spoluprací s dalšími organizacemi.
- Partnerské organizace je potřeba oslovit ještě před zahájením vývoje aplikace.

| silné stránky | slabé stránky |
|---|------------------------------------|
| jasná vize | nedostatečný kapitál |
| znalost cílových uživatelů | nezkušenost s vývojem softwaru |
| dlouhodobé vztahy s potenciálními uživateli | malá personální základna odborníků |
| možnost efektivní propagace | |

| příležitosti | hrozby |
|---|-------------------------------------|
| stabilní základna potenciálních zájemců | nezájem dorostenců |
| možnost rozšířit aplikaci do dalších organizací | špatná spolupráce vedoucích |
| unikátní produkt | prolomení bezpečnosti třetí stranou |
| expanze na mezinárodní trh | malá cílová skupina |
| možné získání státního grantu | |

Tabulka 7.7: SWOT analýza

Doporučení dalšího postupu

Postup, jak doporučuji v projektu dále pokračovat, jsem rozdělil do 8 fází:

1) Uvážit, zda v projektu pokračovat a oslovit potenciální partnerské organizace

Nejprve je potřeba rozhodnout, zda v projektu pokračovat. Náklady jsou pro Dorostovou unii vysoké. Proto doporučuji oslovit potenciální partnerské organizace (skauty, Royal Rangers apod.) a i na základě jejich reakce rozhodnout o budoucím postupu.

2) Provést úpravy návrhu

Poté doporučuji provést nutné úpravy návrhu, které vzešly z výsledků testování. Je potřeba:

- implementovat notifikace při přidání nové žádosti o přátelství, nebo nového oznámení,
- přidat dovysvětlující elementy pro lepší pochopení, jak aplikace funguje (např. ikona otazníku, která po kliknutí zobrazí vysvětlující text).

3) Ověřit návrh se zadavatelem

Následně je potřeba podrobně zkontrolovat návrh a prototyp se zadavatelem. Případně zvážit požadavky i podle dostupnosti finančních zdrojů.

4) Provést potřebné úpravy návrhu

V poslední fázi před zahájením implementace ještě doporučuji:

- ve spolupráci s externím grafikem navrhnout nové logo, zkonzultovat grafický design aplikace a případně jej upravit,

8. DOPORUČENÍ DALŠÍHO POSTUPU

- technicky navrhnout způsob platby a distribuce aplikace (i v závislosti na výsledcích jednání s partnerskými organizacemi),
- vyřešit, aby způsob uložení a zpracování uživatelských dat byl v souladu s nařízením o ochraně osobních údajů.

5) Aplikaci implementovat, otestovat a nasadit

Po posledních úpravách doporučuji zahájit implementaci aplikace a poté *Dorůstáček* otestovat a nasadit do provozu. I během implementace doporučuji komunikovat se zadavatelem a ověřovat postup. Při testování bude vhodné zapojit co nejvíce dobrovolníků z cílové skupiny uživatelů. Po nasazení do reálného provozu doporučuji dále získávat zpětnou vazbu od uživatelů a rychle odladit chyby.

6) Aplikaci propagovat

Aby aplikace v co největší míře zasáhla cílovou skupinu, bude klíčovým faktorem efektivní propagace. S propagací je vhodné začít ještě před nasazením aplikace do provozu.

7) Přidání nových funkcí

Po úspěšném nasazení, vyladění chyb a rozšíření aplikace, lze zvážit přidání dalších funkcí:

- widgety,
- možnost přidání vlastních cílů (např. série 20 úspěšných dnů v kuse),
- přidání odměn za splnění zvyků nebo cílů (např. sbírání virtuálních samolepek, nebo stavění domečku),
- kalendář akcí,
- hlasování uživatelů,
- a další funkce, které uživatelé navrhnou formou zpětné vazby.

8) Vývoj iOS aplikace a expanze do zahraničí

Po nějaké době úspěšného provozu doporučuji vývoj iOS aplikace a poté expanzi do zahraničí. Po úspěšném zvládnutí této fáze předpokládám i finanční ziskovost projektu.

Závěr

V práci jsem se zabýval analýzou, návrhem a implementací prototypu mobilní aplikace *Dorůstáček*, a s tím souvisejícím ekonomicko-manažerským shrnutím. V rámci analýzy jsem prozkoumal a zhodnotil podobné existující aplikace a vytvořil uživatelské funkční požadavky. Poté jsem navrhl případy užít a wireframe aplikace. Na základě wireframe jsem následně vytvořil funkční prototyp. Poté jsem zhodnotil aplikaci z ekonomicko-manažerského hlediska. Finančně jsem se především zaměřil především na zhodnocení nákladů vývoje a následné údržby. Také jsem použil SWOT analýzu pro strategické zhodnocení projektu. Poté jsem doporučil další postup.

U finančního zhodnocení bylo těžké zhodnotit příjmy, protože projekt nepočítá s příjmem z reklam, ani od koncových uživatelů. Aplikace bude financována Dorostovou unií, případně dalšími organizacemi. Příjmy se tak nedají jednoduše vyčíslit, jsou spíše nefinančního charakteru. *Dorůstáček* podpoří komunitní život a osobní rozvoj uživatelů. Dobrovolníci, kteří prototyp testovali, i zástupci Dorostové unie, zájem o aplikaci projeví. Otázkou je, zda se podaří zaplatit náklady.

Postupoval jsem standardními postupy podle unifikovaného procesu vývoje a dle obvyklých kroků při vývoji mobilní aplikace. Prototyp jsem vytvořil v prototypovacím nástroji Mockflow.

Nyní je potřeba zvážit, zda aplikaci dále vyvíjet. Přitom je potřeba neopomenout rizika a nedostatky plynoucí ze SWOT analýzy. Vzhledem k nemalým nákladům doporučuji oslovit potenciální partnerské organizace a náklady zaplatit společně. Poté aplikaci implementovat, otestovat a nasadit.

Bibliografie

1. Mobile Operating System Market Share Worldwide. In: *StatCounter* [online]. 2019 [cit. 2019-04-10]. Dostupné z: <http://gs.statcounter.com/os-market-share/mobile/worldwide>.
2. Mobile Operating System Market Share Czech Republic. In: *StatCounter* [online]. 2019 [cit. 2019-04-10]. Dostupné z: <http://gs.statcounter.com/os-market-share/mobile/czech-republic/2016>.
3. Jak vyvíjet mobilní aplikace. In: *Jan Václavík* [online]. 2015 [cit. 2019-04-11]. Dostupné z: <http://janvavclavik.cz/jak-vyvijet-mobilni-aplikace/>.
4. Hybridní, nativní nebo webové aplikace? In: *iQuest* [online]. © 2018 [cit. 2019-04-11]. Dostupné z: <http://www.hybridniaplikace.cz/srovnani.html>.
5. SACCOMANI, Pietro. Native Apps, Web Apps or Hybrid Apps? What's the Difference? In: *MobiLoud* [online]. Fifty Pixels Ltd, 2019 [cit. 2019-04-11]. Dostupné z: <https://www.mobiloud.com/blog/native-web-or-hybrid-apps/>.
6. DUA, Kinjal. The Complete Guide to Mobile App Development: Web vs. Native vs. Hybrid. In: *ClearBridge* [online]. 2019 [cit. 2019-04-11]. Dostupné z: <https://clearbridgemobile.com/mobile-app-development-native-vs-web-vs-hybrid/>.
7. HLAVATÝ, Martin. *Softwarový proces* [online]. 2018 [cit. 2019-04-11]. Dostupné z: https://moodle.fit.cvut.cz/pluginfile.php/172/course/section/11326/1_SoftwareProcess.pdf [Soubor přístupný po přihlášení do sítě ČVUT].
8. MLEJNEK, Jiří. *Metodiky vývoje* [online]. 2018 [cit. 2019-04-11]. Dostupné z: https://moodle.fit.cvut.cz/pluginfile.php/88297/mod_resource/content/0/11_prednaska.pdf [Soubor přístupný po přihlášení do sítě ČVUT].

9. ARLOW, Jim; NEUSTADT, Ila. *UML 2 a unifikovaný proces vývoje aplikací: průvodce analýzou a návrhem objektově orientovaného softwaru*. 2. vydání. Brno: Computer Press, 2007. ISBN 978-80-251-1503-9.
10. *History* [online obrázek] [cit. 2019-05-04]. Dostupné z: <http://www.agilemethod.net/p/waterfall.html>.
11. BLAIR, Ian. 8 Steps to Understanding The Mobile App Development Lifecycle. In: *BuildFire* [online] [cit. 2019-04-15]. Dostupné z: <https://buildfire.com/understanding-mobile-app-development-lifecycle/>.
12. HALL, Reuben. How To Prototype A Mobile App And Leave A Lasting Impression On Investors. In: *MindSea* [online]. [2018] [cit. 2019-04-15]. Dostupné z: <https://mindsea.com/how-to-prototype-a-mobile-app/>.
13. ASHWINI, Amit. What Are The Various Phases Of Mobile App Development? In: *TheStartup* [online]. 2017 [cit. 2019-04-15]. Dostupné z: https://medium.com/swlh/what-are-the-various-phases-of-mobile-app-development-4f0a1748e619?fbclid=IwAR1kNmL0wwW58r-2Ucu4i-C0C2r8_DPxAxZnKRwn9SVzjMGgxTeriMTets.
14. KRIGSMAN, Michael. CIO analysis: Why 37 percent of projects fail. In: *ZDNet* [online]. CBS Interactive, 2011 [cit. 2019-04-12]. Dostupné z: <https://www.zdnet.com/article/cio-analysis-why-37-percent-of-projects-fail/>.
15. PROFINIT. *Requirements Engineering: Sběr a analýza požadavků* [online]. 2018 [cit. 2019-04-12]. Dostupné z: https://moodle.fit.cvut.cz/pluginfile.php/172/course/section/11326/03_Requirements_Engineering.pdf [Soubor přístupný po přihlášení do sítě ČVUT].
16. RUMBAUGH JAMES, Jacobson Ivar; GRADY, Booch. *The Unified Modeling Language Reference Manual*. Essex, UK, UK: Addison-Wesley Longman Ltd., 1999. ISBN 0-201-30998-X.
17. MLEJNEK, Jirí. *Analýza a sběr požadavků* [online]. 2018 [cit. 2019-04-12]. Dostupné z: https://moodle.fit.cvut.cz/pluginfile.php/88280/mod_resource/content/3/03_prednaska.pdf [Soubor přístupný po přihlášení do sítě ČVUT].
18. What is wireframing? In: *Experience UX* [online]. © 2019 [cit. 2019-04-20]. Dostupné z: <https://www.experienceux.co.uk/faqs/what-is-wireframing/>.
19. Why Prototyping is Essential to Your Design Process. In: *WebFX* [online]. © 1995-2019 [cit. 2019-04-17]. Dostupné z: <https://www.webfx.com/blog/web-design/prototyping-is-essential/>.
20. JUSTINMIND. *Justinmind Prototyper. version 8.7.0* [software]. 2019 [cit. 2019-04-11]. Dostupné z: <https://www.justinmind.com/>.

21. *Visual Paradigm* [online]. Visual Paradigm, © 2019 [cit. 2019-04-08]. Dostupné z: <https://www.visual-paradigm.com/>.
22. A PRODULE SYSTEMS PVT LTD. *Desktop App for MockFlow WireframePro. version 3.6.2.0* [software]. 2019 [cit. 2019-04-11]. Dostupné z: <https://mockflow.com>.
23. *Diagram SWOT analýzy v češtině* [online obrázek] [cit. 2019-04-06]. Dostupné z: https://cs.wikipedia.org/wiki/SWOT#/media/File:SWOT_cs.svg.
24. CHANG, Hsu-Hsi; HUANG, Wen-Chih. Application of a quantification SWOT analytical method. *Mathematical and Computer Modelling* [online]. 2006, roč. 43, č. 1 [cit. 2019-04-05]. ISBN 0895-7177. Dostupné z DOI: 10.1016/j.mcm.2005.08.016.
25. SWOT analýza. In: *ManagementMania.com* [online]. © 2011-2016 [cit. 2019-04-05]. Dostupné z: <https://managementmania.com/cs/swot-analyza>.
26. ZIKMUND, Martin. Kde se vzala a k čemu všemu je vlastně SWOT analýza. In: *Business vize* [online]. Nitana s.r.o., 2010 [cit. 2019-04-05]. Dostupné z: <http://www.businessvize.cz/planovani/kde-se-vzala-a-k-cemu-vsemu-je-vlastne-swot-analyza>.
27. NÁPLAVA, Pavel. *Tvorba informačních systémů: Procesní pohled, projekt, implementace IS* [online]. 2018 [cit. 2019-04-06]. Dostupné z: https://moodle.fit.cvut.cz/pluginfile.php/63282/mod_page/content/2/prednaska02.pdf [Soubor přístupný po přihlášení do sítě ČVUT].
28. MCCONNELL, Steve. *Odhadování softwarových projektů: Jak správně určit rozpočet, termíny, zdroje*. Brno: Computer Press, 2006. ISBN 80-251-1240-3.
29. ZOUBEK, Bohumír; HLAVATÝ, Martin. *Odhady, nabídky, měření a historie* [online]. 2018 [cit. 2019-04-16]. Dostupné z: https://moodle.fit.cvut.cz/pluginfile.php/172/course/section/11326/2_ProposalsEstimationsMeasurement.pdf [Soubor přístupný po přihlášení do sítě ČVUT].
30. MIKLÁŠ, Daniel. Analýza PERT v MS Project. In: *Efektivně* [online] [cit. 2019-04-17]. Dostupné z: <http://www.efektivne.eu/analiza-pert-v-ms-project.html>.
31. Odhad doby trvání úkolů pomocí analýzy PERT. In: *Office Support* [online]. Microsoft, 2019 [cit. 2019-04-15]. Dostupné z: <https://support.office.com/cs-cz/article/odhad-doby-trv%C3%A1n%C3%AD-%C3%BAko1%C5%AF-pomoc%C3%AD-anal%C3%BDzy-pert-864b5389-6ae2-40c6-aacc-0a6c6238e2eb>.

32. CHEMUTURI, Murali. *Software Estimation Best Practices, Tools & Techniques: A Complete Guide for Software Project Estimators: A Complete Guide for Software Project Estimators*. U.S.A.: J. Ross Publishing, Inc., 2009. ISBN 978-1-60427-024-2. Dostupné také z: <https://books.google.cz/books?id=IwE0B2Mfzx0C>.
33. Estimation Techniques - Use-Case Points. In: *TutorialsPoint* [online]. © 2019 [cit. 2019-04-18]. Dostupné z: https://www.tutorialspoint.com/estimation_techniques/estimation_techniques_use_case_points.htm.
34. Estimating With Use Case Points. In: *Methods & Tools* [online]. © 1995-2019 [cit. 2019-04-18]. Dostupné z: <http://www.methodsandtools.com/archive/archive.php?id=25>.
35. Percentage of mobile apps that have been used only once during the first six months of ownership from 2010 to 2018. In: *Statista* [online]. © 2019 [cit. 2019-04-18]. Dostupné z: <https://www.statista.com/statistics/271628/percentage-of-apps-used-once-in-the-us/>.
36. RODDE, Thomas. 21% of Users Abandon an App After One Use. In: *Localytics* [online]. 2018 [cit. 2019-04-18]. Dostupné z: <http://info.localytics.com/blog/21-percent-of-users-abandon-apps-after-one-use>.
37. PEREZ, Sarah. Users Have Low Tolerance For Buggy Apps – Only 16% Will Try A Failing App More Than Twice. In: *TechCrunch* [online]. 2013 [cit. 2019-04-18]. Dostupné z: https://techcrunch.com/2013/03/12/users-have-low-tolerance-for-buggy-apps-only-16-will-try-a-failing-app-more-than-twice/?utm_source=feedburner&utm_medium=feed&utm_campaign=Feed%3A+Techcrunch+%28TechCrunch%29.
38. 5 Interesting Facts About Millennials' Mobile App Usage from "The 2017 U.S. Mobile App Report". In: *Comscore* [online]. 2017 [cit. 2019-04-18]. Dostupné z: <https://www.comscore.com/lat/Prensa-y-Eventos/Blog/5-Interesting-Facts-About-Millennials-Mobile-App-Usage-from-The-2017-US-Mobile-App-Report>.
39. Rozdíl mezi UI a UX: definice pojmů. In: *iQuest blog* [online]. 2017 [cit. 2019-04-19]. Dostupné z: <http://blog.iquest.cz/2017/11/rozdil-mezi-ui-ux-definice-pojmu.html>.
40. Jakob Nielsen: Principal. In: *Nielsen Norman Group* [online]. © 1998-2019 [cit. 2019-04-19]. Dostupné z: <https://www.nngroup.com/people/jakob-nielsen/>.
41. NNGROUP. Jakob's Law of Internet User Experience. In: *Youtube* [online]. 2017 [cit. 2019-04-20]. Dostupné z: <https://www.youtube.com/watch?v=wzb4mK9DiHM>. Kanál společnosti Nielsen Norman Group.

42. NIELSEN, Jakob. 10 Usability Heuristics for User Interface Design. In: *Nielsen Norman Group*. 1994. Dostupné také z: <https://www.nngroup.com/articles/ten-usability-heuristics/>.
43. KOSTELECKÝ, Josef [osobní komunikace]. 2019.
44. AXFN. *Productivity Challenge Timer. version 1.9.4* [software]. 2019 [cit. 2019-04-20]. Dostupné z: <https://play.google.com/store/apps/details?id=com.wlxd.pomochallenge&hl=cs>.
45. SIX TO START LSD. *Zombies, Run! version 7.2.0* [software]. 2019 [cit. 2019-04-20]. Dostupné z: <https://play.google.com/store/apps/details?id=com.sixtostart.zombies5ktraining>.
46. HABITRPG, INC. *Habitica: Gamify Your Tasks. version 1.8.1* [software]. 2019 [cit. 2019-04-20]. Dostupné z: <https://play.google.com/store/apps/details?id=com.habitrpg.android.habitica>.
47. THEFABULOUS. *Fabulous – Self Care. version 3.55* [software]. 2019 [cit. 2019-04-20]. Dostupné z: <https://play.google.com/store/apps/details?id=co.thefabulous.app>.
48. INTRASOFT. *Mé cíle, návyky a předsevzetí. version 2.1.1* [software]. 2019 [cit. 2019-04-20]. Dostupné z: <https://play.google.com/store/apps/details?id=info.intrasoft.habitgoaltracker>.
49. APP HOLDINGS. *Habit Tracker. version 1.5.11* [software]. 2019 [cit. 2019-04-20]. Dostupné z: <https://play.google.com/store/apps/details?id=com.oristats.habitbull&hl=cs>.
50. VACLAV BALAK. *Návykovač. version 0.9.6* [software]. 2019 [cit. 2019-04-20]. Dostupné z: <https://play.google.com/store/apps/details?id=cz.shmoula.habbeat&hl=cs>.
51. RANDEME STUDIOS. *HabitHub – Habit and Goal Tracker. version 9.5.37* [software]. 2019 [cit. 2019-04-20]. Dostupné z: <https://play.google.com/store/apps/details?id=com.rstudioz.habits&hl=cs>.
52. PEČENKA, Jakub. *Testování prototypu* [online]. 2019 [cit. 2019-05-08]. Dostupné z: <http://jakubovabc.borec.cz/>.
53. *How Much to Make an App* [online]. Z1 Digital Studio, © 2019 [cit. 2019-04-25]. Dostupné z: <http://howmuchtomakeanapp.com>.
54. *Estimate My App* [online]. Oozou, © 2018 [cit. 2019-04-25]. Dostupné z: <https://estimatemyapp.com/>.
55. *How Much Does It Cost To Make An App?* [online]. YeePLY, © 2019 [cit. 2019-04-25]. Dostupné z: <https://www.howmuchcostanapp.co.uk/>.
56. Android programátor. In: *Platy.cz* [online]. Profesia, © 1997-2019 [cit. 2019-04-27]. Dostupné z: <https://www.platy.cz/platy/informacni-technologie/android-programator>.

BIBLIOGRAFIE

57. Cloud Server. In: *Snackhost* [online]. CE Online Systems s.r.o., © 2019 [cit. 2019-04-30]. Dostupné z: <https://www.snackhost.com/cs/cloud-server/>.
58. Fakta a čísla. In: *Skaut* [online]. Junák – český skaut, z.s., © 2019 [cit. 2019-04-30]. Dostupné z: <https://www.skaut.cz/skauting/fakta-a-cisla/>.

Seznam použitých zkratk

HTML Hypertext Markup Language

PDF Portable Document Format

GUI Graphical user interface

UI Graphical interface

UML Unified Modelling Language

UX User experience

UP Unified Process

UC Use Case

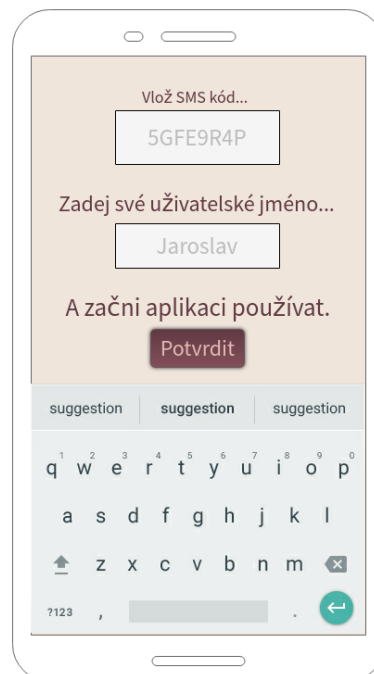
MD Man-day

;

Obrazovky prototypu

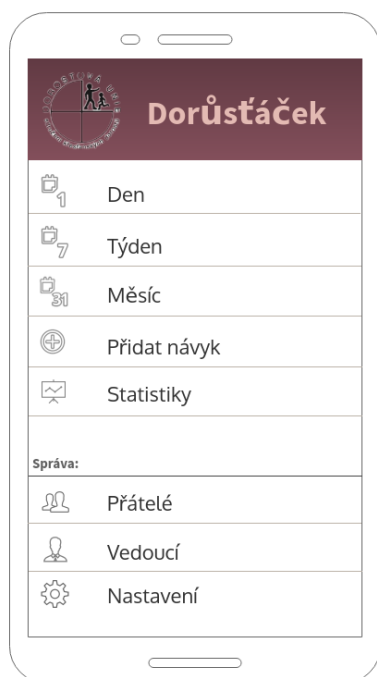


Obrázek B.1: Registrace

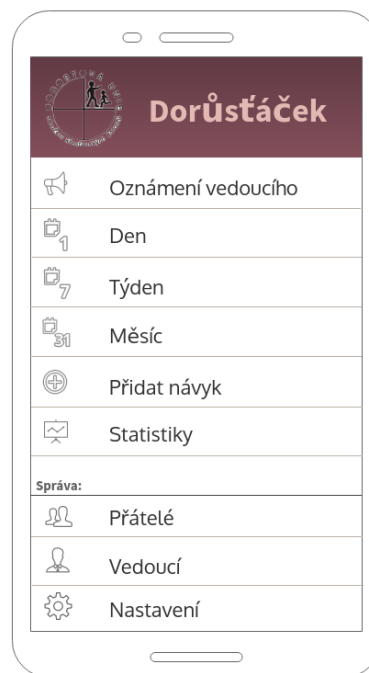


Obrázek B.2: Dokončení

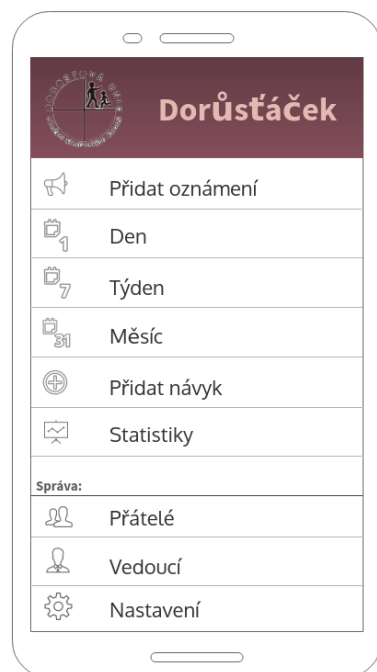
B. OBRAZOVKY PROTOTYPU



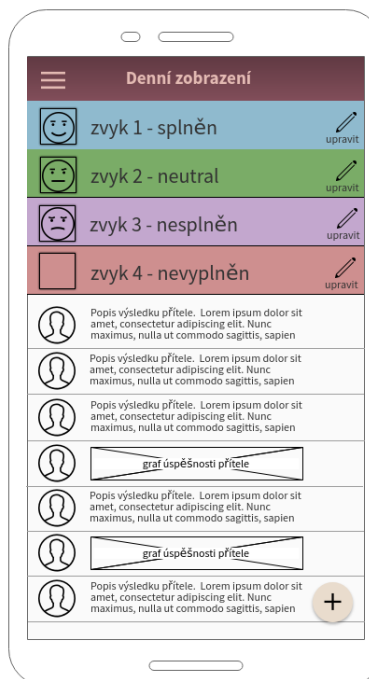
Obrázek B.3: Standardní menu



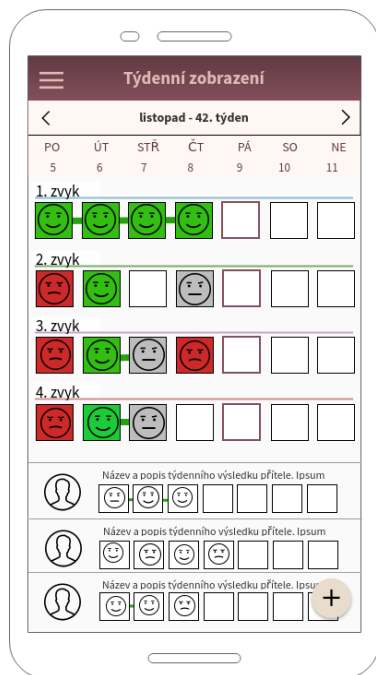
Obrázek B.4: Menu následovníka



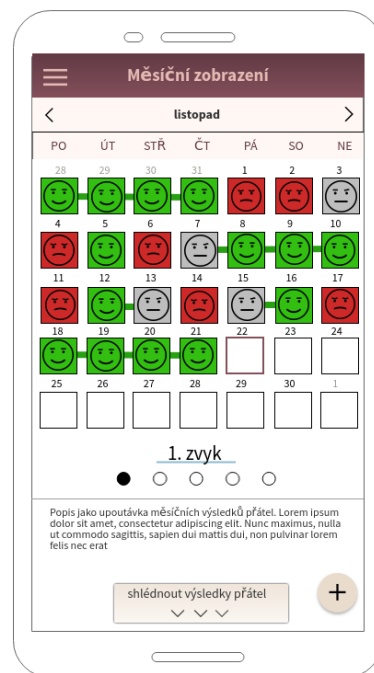
Obrázek B.5: Menu vedoucího



Obrázek B.6: Denní zobrazení



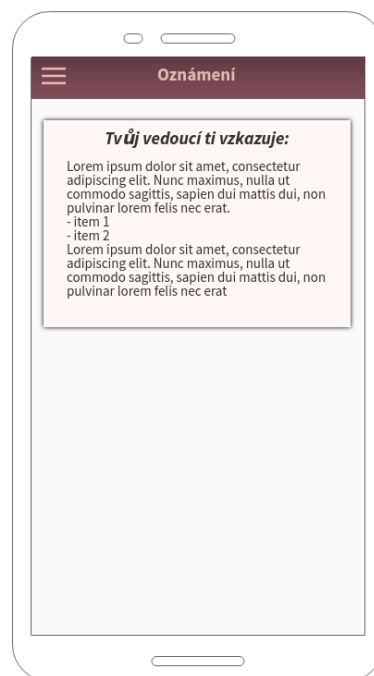
Obrázek B.7: Týdenní zobrazení



Obrázek B.8: Měsíční zobrazení

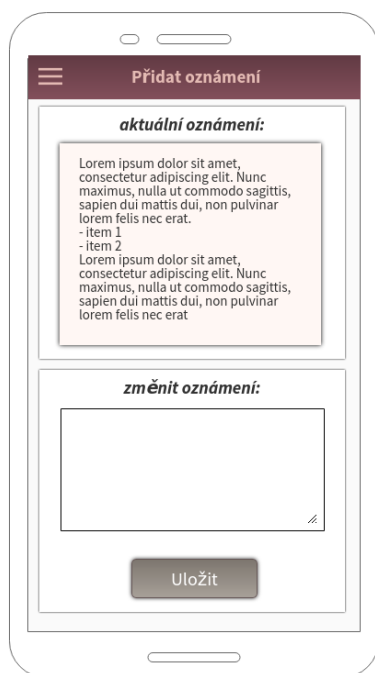


Obrázek B.9: Přidat návyk

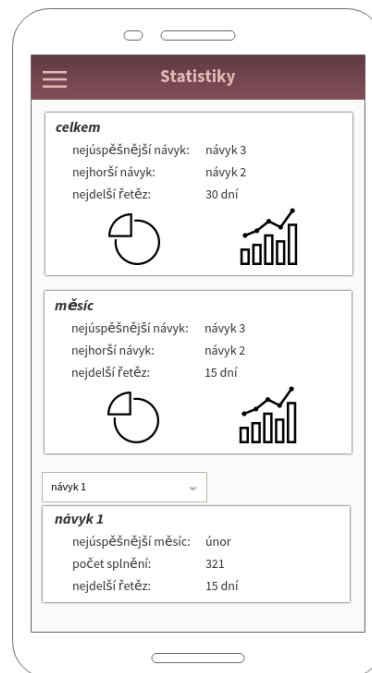


Obrázek B.10: Oznámení

B. OBRAZOVKY PROTOTYPU



Obrázek B.11: Přidat oznámení



Obrázek B.12: Statistika



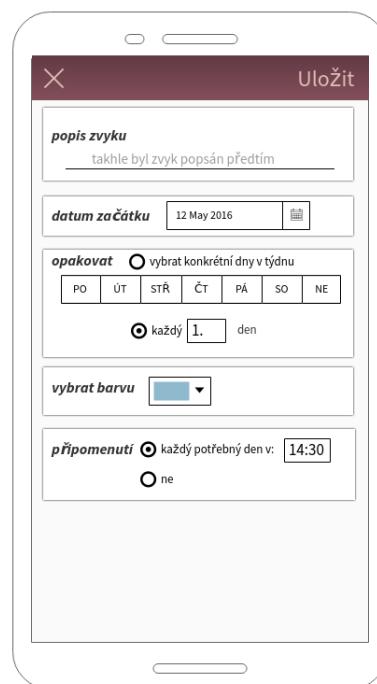
Obrázek B.13: Přátelé



Obrázek B.14: Vedoucí



Obrázek B.15: Nastavení



Obrázek B.16: Upravit návyk

Obsah přiloženého CD

| | |
|-------------------------------------|---|
| readme.txt | stručný popis obsahu CD |
| prototype | |
| ├── HTML_source..... | HTML stránka s prototypem |
| ├── pictures..... | obrázky obrazovek prototypu |
| wireframe..... | obrázky wireframe |
| thesis | zdrojová forma práce ve formátu \LaTeX |
| └── BP_Pecenka_Jakub_2019.pdf | text práce ve formátu PDF |