GYMNASIUM JANA KEPLERA

Parléřova 2/118, 169 00 Praha 6

Pochlebník - Systém pro distribuovaný offline sběr zpětných vazeb

Maturitní práce

Autor: David Nadrchal

Třída: R8.A

Školní rok: 2021/2022

Předmět: Informatika

Vedoucí práce: Šimon Schierreich

Praha, 2022

GYMNASIUM JANA KEPLERA
Kabinet informatiky

ZADÁNÍ MATURITNÍ PRÁCE

Student:	**David Nadrchal**
Třída:	**R8.A**
Školní rok:	**2021/2022**
Platnost zadání:	**30. 9. 2022**
Vedoucí práce:	**Šimon Schierreich**
Název práce:	**Systém pro distribuovaný offline sběr zpětných vazeb**

Pokyny pro vypracování:

Projekt si klade za cíl vytvořit nástroj pro sběr zpětných vazeb a to primárně ke skautským akcím. Získávání podrobné zpětné vazby k těmto událostem je často ztíženo, ne-li přímo znemožněno, hned několika faktory - neochotou lidí pravidelně vyplňovat dlouhé jednotvárné formuláře (což je problém především všednodenních schůzek) a jindy, obzvláště během táborů či různých kurzů v odlehlých základnách, zas nemožností se připojit na internet. Další velký problém pak představuje soustavné vyhodnocování těchto zpětných vazeb a hledání jejich dlouhodobých implikací.

Tento projekt se snaží s těmto problémy vyrovnat a nabídnout systém, který bude díky mobilní aplikaci využitelný i bez internetového připojení, který bude distribuovat mezi respondenty otázky tak, aby byla každá zodpovězená a nikdo nemusel vyplňováním trávit mnoho času, a který uživatelům umožní na grafech sledovat dlouhodobé trendy.

Doporučená literatura:

Zde bude *vedoucím práce* doplněna literatura, ze které je doporučeno při práci na tématu vycházet.

URL repozitáře:

https://github.com/Hobit2002/DOFBC

student	*vedoucí práce*

V Praze dne 5. 3. 2022

Prohlášení

Prohlašuji, že jsem svou práci vypracoval samostatně a použil jsem pouze prameny a literaturu uvedené v seznamu bibliografických záznamů. Nemám žádné námitky proti zpřístupňování této práce v souladu se zákonem č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších předpisů.

V Praze dne 24. března 2022 David Nadrchal

Poděkování

Rád bych tímto poděkoval svému vedoucímu Šimonu Schierreichovi za výzvy, ke kterým mě nasměroval. Dále bych rád vyjádřil vděk své rodině, která mi byla po celé středoškolské studium nepostradatelnou podporou.

Abstrakt

V rámci této práce představuji Pochlebníka - nástroj pro sběr zpětných vazeb a to primárně ke skautským akcím. Ty se při této činnosti často potýkají s následujícími problémy: neochota účastníků trávit vyplňováním formulářů více než několik málo minut, obtížná zpracovatelnost získaných výsledků a nedostupnost internetového připojení. Pochlebník se vyrovnává se všemi těmito komplikacemi - distribucí otázek šetří energii respondentů, v přehledných grafech uživatele seznamuje s dlouhodobými trendy a možností načíst formulář z QR kódu umožňuje vyplnění zpětných vazeb i v polních podmínkách.

Klíčová slova

zpětné vazby, offline mobilní komunikace, distribuované odpovědi, grafy s dlouhodobým hodnocením

Abstract

In this work I present the Pochlebník - a tool for collecting feedback, primarily for scouting events. These often face the following problems when trying to obtain feedback: the reluctance of participants to spend more than a few minutes filling out the forms, the difficulty of processing the results obtained, and the unavailability of Internet access. Pochlebnik copes with all these complications - by distributing the questions it saves respondents' energy, it familiarises users with long-term trends in clear graphs and the possibility to retrieve the form from a QR code enables feedback to be completed even in field conditions.

Keywords

feedbacks, offline mobile communication, distributed responses, graphs with long-term ratings

Obsah

1. Teoretická část

1.1 Proč Pochlebník?

Skutečnost, že pouze 17 % českých učitelů nepociťuje žádné známky syndromu vyhoření svědčí o tom, že pedagogická práce není zrovna nejvděčnější. Učitelé však s pocitem, že jejich úsilí není doceněno, nežijí sami. Podobně se cítí i bezpočet lektorů různých kroužků, skautských vedoucích a jiných lidí, jež mají tu smůlu, že s mládeží jednají z pozice autority, na níž nepříjemně velká část této věkové skupiny pohlíží s despektem. Tato vzdorovitost by však mohla být o poznání menší, pokud by mládež se svými vedoucími komunikovala nikoliv před zraky zbytku své skupiny nýbrž přes anonymní počítačovou platformu. Možná by pak pedagogům i pochlebovala a dopřála jim tedy něco oč určitě, byť třeba jen ve skrytu duše, stojí. Platformou, která právě toto umožní, je v této práci představený Pochlebník.

Jak již zadání práce z listopadu, abstrakty a mírně nadnesený prolog předestřely, bylo mým cílem vytvořit systém pro sběr zpětných vazeb k akcím pro mládež, především pak těm skautským. Tato volba vycházela z toho, že ze všech možných pedagogických činností jsem měl právě s touto nejvíce zkušeností, zároveň však stavěla na předpokladu, že při obrovské různorodosti skautských aktivit bude nástroj odpovídající všem jejich potřebám použitelný i v bezpočtu dalších prostředí.

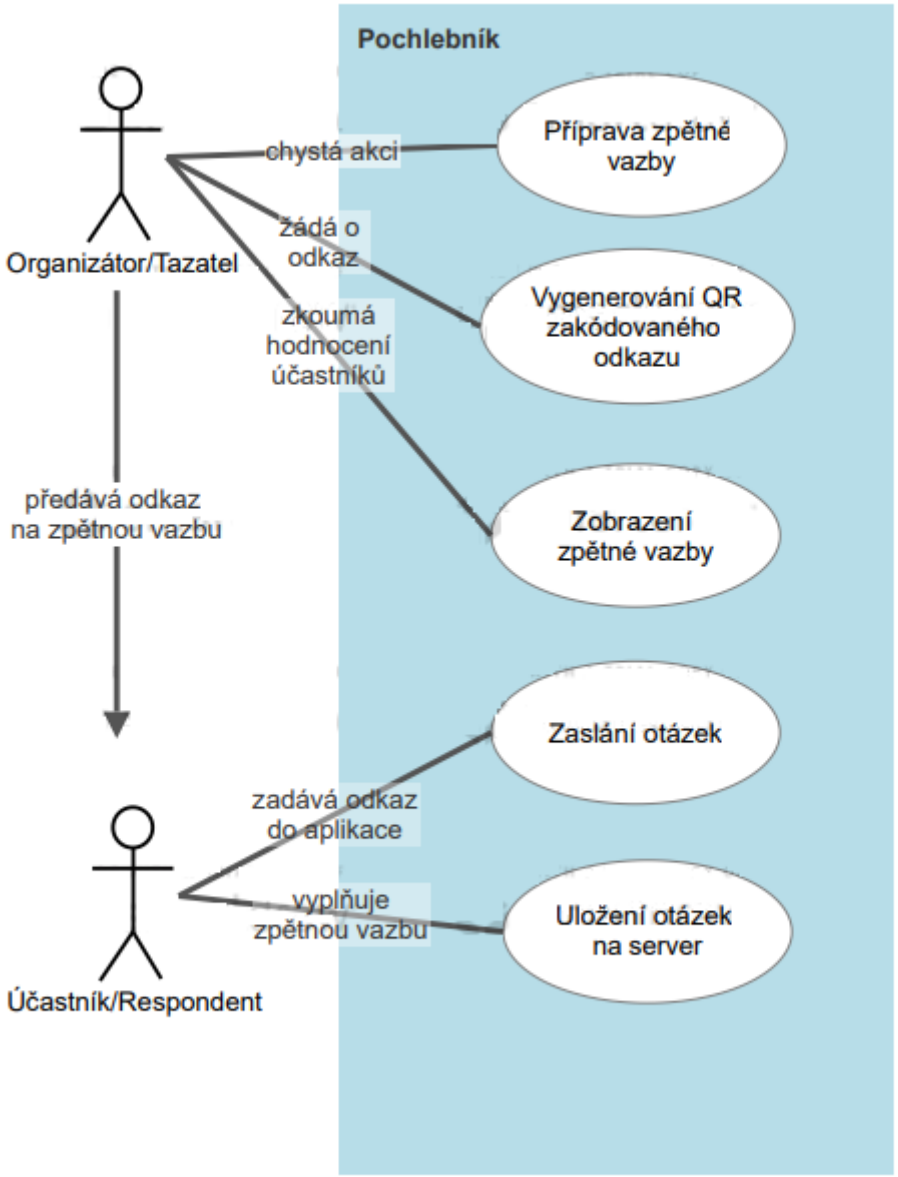
Nástroj, který skutečně vyhovuje skautským potřebám totiž musí:

- Být naprosto intuitivní, aby nad ním vedoucí mající věčný pocit, že dobrovolničení i beztak věnují energie až moc, neohrnuli nos.
- Minimalizovat veškeré požadavky na respondenty zpětných vazeb - účastníky akcí.
- Plnohodnotně fungovat na mobilu, jinak by bylo sotva možné získat na konci schůzky zpětnou vazbu k jejímu průběhu.
- Nabízet jak otevřené, tak uzavřené otázky.
- Posbíraná data také analyzovat, neboť dobrovolničtí vedoucí by na to jen sotva měli kapacitu.
- Fungovat i bez připojení k internetu, a být tedy k dispozici i na táborech či puťácích.

Po specifikaci z toho pro mou práci vyplývaly následující požadavky na konkrétní funkce systému i požadavky nefunkční, tedy ty, které specifikují rychlost, robustnost či platformu, na které aplikace běží [11] (vnější pohled na systém a jeho interakce s uživateli poskytuje diagram případů užití na Obrázku 1.1 [3]):

- Webové rozhraní, díky němuž se k tazateli dostanou odpovědi respondentů na jeho zpětnou vazbu.
- Mobilní aplikace, která umožní vyplňování zpětných vazeb i v podmínkách bez internetu a bude schopná se po jeho připojení synchronizovat se serverem.
- Umožnit vyplnění zpětné vazby pomocí QR kódu.
- Uživatelské webové rozhraní, díky němuž budou moct zpětné vazby připravit či vyplnit i lidé, kteří mobilní aplikaci (například kvůli nekompatibilní platformě) nemají.
- Každému respondentovi položit jen malé množství otázek, aby ho jejich vyplňování neotrávilo.
- Zobrazovat v aplikaci na grafech, jak se postupně vyvíjely odpovědi na jednotlivé otázky.

Obrázek 1.1: Diagram případů užití



1.2 Alternativy

Aplikací pro sběr zpětných vazeb k akcím pro mládež mnoho neexistuje. V českém prostředí stojí za zmínku poměrně všeobjímající aplikace Levitio[1], která digitalizuje administrativu obnášenou skautskými schůzkami. Sběr zpětných vazeb v této aplikaci sice dosud není, ale měl by přibýt v budoucnu [12]. I kdyby Levitio nabízelo distribuci otázek a offline sběr zpětných vazeb, bude oproti němu Pochlebník ve výhodě jednak díky tomu, že je zdarma a, vzhledem k menšímu množství funkcí, intuitivnější. Těžko však srovnávat produkt existující s produktem pouze ohlášeným.

Nejběžnějším způsobem, alespoň z mé zkušenosti, jak sbírat zpětné vazby k akcím mládeže jsou Formuláře Google[2] či jiné dotazníkové aplikace [17]. Oproti těm má Pochlebník při hodnocení standardních akcí výhod celou řadu - distribucí otázek šetří respondentům čas (na druhou stranu ho tato distribuce činí nepoužitelným pro nějaké rozsáhlé komplexní dotazníky), funguje i v podmínkách, kam se s aplikacemi Google dostat nelze, a přehledněji zobrazuje dlouhodobé trendy.

[1] https://www.levitio.cz/funkce.html
[2] https://www.google.cz/intl/cs/forms/about/

2. Implementace

V této kapitole představím nejdůležitější část systémové architektury a popíši, jaké technologie jsem využil k zajištění té či oné funkcionality. Nebudu uvádět verze použitých nástrojů, vždy jsem volil nejnovější variantu.

2.1 Webová aplikace

Webové rozhraní jak pro mobilní aplikace tak pro uživatele zajišťuje stránka www.pochlebnik.eu.

K uživatelskému webovému rozhraní se mi zdá ještě vhodné podotknout, že se větví do dvou částí - *respondentské* sloužící jen a pouze k vyplňování zpětných vazeb a *organizační*, kde se vytváří zpětné vazby a zobrazují posbírané odpovědi.

2.1.1 Webhosting

Webovou aplikaci jsem umístil na virtuální počítač Kancelářských strojů. s. r. o.[1] s unixovým operačním systémem *CentOS*[2] a zajistil jeho obsluhu pomocí webového serveru *Nginx*[3] (vzhledem k tomu, že na hostiteli běží i jiné služby, nebylo možné Pochlebníka spustit přímo na některém z otevřených portů). Https certifikát zajišťující šifrovanou komunikaci mezi klientem a serverem jsem obstaral pomocí služby *Let's Encrypt*[4]. Volba těchto technologií byla dána čistě tím, že pro mě byly díky různým předchozím projektům dostupné i známé.

2.1.2 Webový backend

Backend webové aplikace jsem napsal v jazyce Python s využitím frameworku *Django*[5]. Tuto technoloigii jsem si vybral, protože mi již byla částečně známá a mé dosavadní zkušenosti s ní byly dobré. Python je přehledný jazyk, což v kombinaci s high-level funkcemi, které celý framework nabízí, činí vývoj velice rychlým. V průběhu projektu jsem však začal o této volbě pochybovat a zvažovat, zda by Node.js server nebyl vhodnější. Na rozdíl od Python využívajícího Djanga totiž s Node.js člověk webový backend píše v JavaScriptu, díky čemuž bych při vývoji offline módu mobilní aplikace, nemusel mnohé backendové funkce přepisovat z Pythonu do JavaScriptu, ale stačilo by mi je zkopírovat, popřípadě vytvořit pro webovou i mobilní aplikaci jednu společnou knihovnu těchto funkcí.

[1] https://www.kancelarskestroje.cz/
[2] https://www.centos.org/
[3] https://www.nginx.com/
[4] https://letsencrypt.org/
[5] https://www.djangoproject.com/

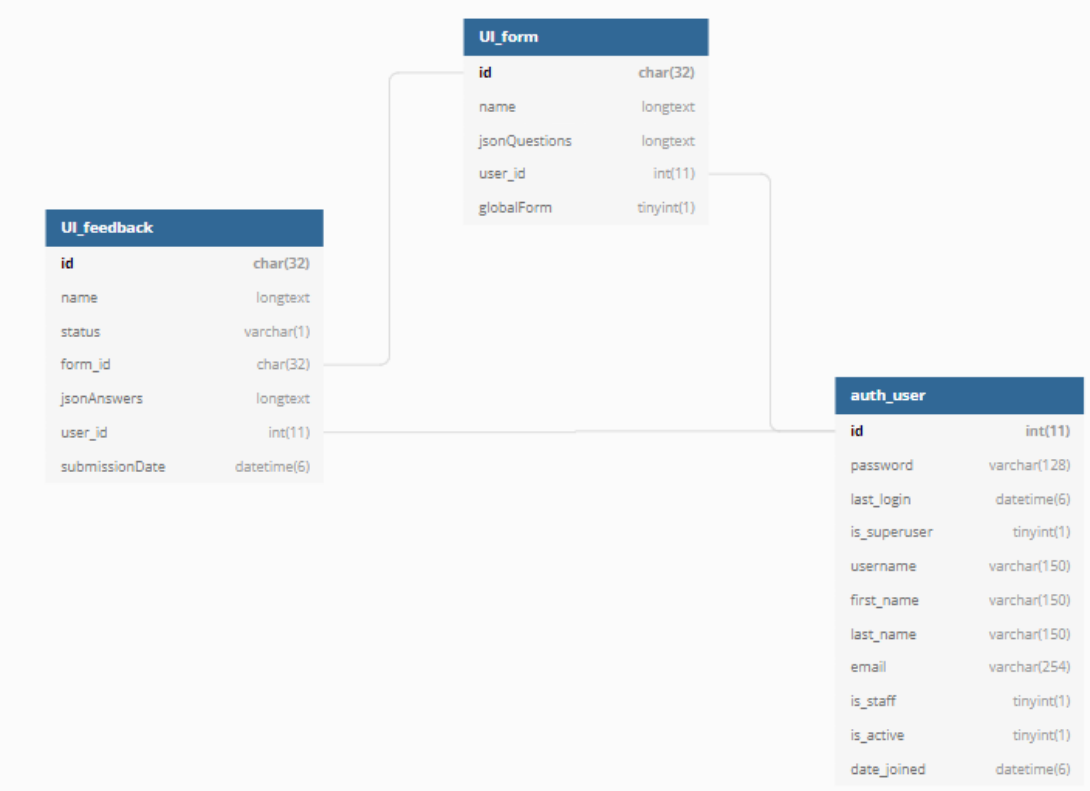
2.1.2.1 Databáze

Pochlebník používá *MySQL*[6] databázi (vybranou na základě dobré předchozí zkušenosti). Můj kód používá pouze následující tři entity (jejich vzájemné vztahy znázorňuje diagram na Obrázku 2.1):

- *auth_user* - Djangem vygenerovaná a spravovaná tabulka. Uchovává údaje o registrovaných uživatelích.
- *UI_feedback* - zpětná vazba ke konkrétní události. Obsahuje přehled odpovědí a je provázaná s uživatelem, který ji vytvořil, a formulářem obsahujícím otázky, které budou položeny respondentům.
- *UI_form* - S každou zpětnou vazbou je vytvořen formulář, který uchovává její otázky. Pokud uživatel chce stejnou, či velmi podobnou sadu otázek použít i v jiné zpětné vazbě, může její formulář uložit jakožto globální. V takovém případě ho bude moct použít jako šablonu pro jiné zpětné vazby.

Při zpětném zhodnocení se mi zdá, že jsem databázi nenavrhnul úplně dokonale - celá databáze by byla menší a přehlednější, kdyby tabulka UI_forms zahrnovala pouze globální formuláře a ty lokální se ukládaly do tabulky UI_feedback.

Obrázek 2.1: Databázový diagram



[6]https://www.mysql.com/

2.1.2.2 Backend zpětných vazeb

Byť ne zcela triviální, je algoritmus rozdělující mezi respondenty otázky k vyplnění, poměrně přímočarý. Spíše než prostorově náročný diagram, popíše ho seznam kroků, které jsou potřeba k tomu, aby se dojmy z hodnocené události přenesly z hlav účastníků do databáze Pochlebníka.

1. Vedoucí akce přenastaví její status z výchozího „Připravovaná“ na „Otevřená“ a umožní tak lidem majícím patřičnou url adresu vyplnit zpětnou vazbu.
2. Do až doposud prázdného atributu zpětné vazby, „Odpovědí“, se uloží seznam otázek k zodpovězení. U každé bude seznam do něhož se později uloží odpovědi.
3. Vedoucí předá účastníkům adresu zpětné vazby a ti ji otevřou buď ve webovém prohlížeči, nebo přímo v mobilní aplikaci Pochlebníka.
4. Když dorazí na server žádost o otázky k vyplnění, načte server „Odpovědi“ dané zpětné vazby a najde 5 otázek s nejkratším seznam odpovědí. (Pokud je otázek dohromady méně než 5, vybere všechny). Do seznamu odpovědí na dané otázky se uloží None.
5. Vybrané otázky jsou poslány respondentovi, který na ně odpoví a pošle na server JSON obsahující odpovědi.
6. Server načte „Odpovědi“ dané zpětné vazby a u v JSONu obsažených otázek nahradí jedno None ze seznamu odpovědí na danou otázku odpovědí uživatelem zaslanou.
7. Když vedoucí usoudí, že má již odpovědí dost, změní status „Otevřená“ na „Uzavřená“, čímž ukončí možnost zpětnou vazbu vyplnit.

2.1.3 Webový frontend

Pochlebník je single page aplikace. Frontendové funkce zajišťuje *vanilla JavaScript*, který je rozdělený do třech souborů:

general.js - obsahuje funkce nezbytné pro vykreslování stránek a komunikaci se serverem (a tedy vlastně vše, co je potřeba pro přípravu zpětných vazeb a prohlížení zaslaných odpovědí). Tento soubor se stahuje jak pro respondentskou tak pro organizační část.

feedback.js - tento soubor se stahuje pouze pro respondentskou část a kromě funkcí pro správné vykreslení otázek obsahuje paměťovou strukturu, do níž se během vyplňování ukládají odpovědi na jednotlivé otázky.

chart.js - externí knihovna pro vykreslení grafu s přehledem dlouhodobých trendů v odpovědích. Tento soubor se stahuje pouze pro organizační část.

2.2 Design

Při designování mobilní i webové aplikace jsem kombinoval *vlastnoručně psané CSS s Bootstrapem* [7]. Předpřipravené CSS šablony Bootstrapu jsem využíval především k základnímu pozicování HTML prvků do jedné řádky či do rohu stránky, vlastní CSS jsem pak psal například tlačítka, kde jsem chtěl nastavit zaoblení, barvu textu, barvu pozadí a řadu dalších parametrů.

[7] https://getbootstrap.com/

2.3 Testování

Testování mělo dvě fáze. První sestávala z manuálních testů[10] následujících vyvinutí každé funkce. V rámci nich jsem zkoušel do funkcí vkládat data, která by svou délkou či obsahem mohla být problematická, a opravoval případné chyby. V rámci unit testů jsem se párkrát dostal i k mockování a to především u vývoje mobilní aplikace, který jsem z důvodu vyšší rychlosti a snadnějšího debugování z větší části prováděl v prohlížeči. To ovšem vyžadovalo, abych se při testování obešel bez skutečného QR scanneru a bez schopnosti zařízení poznat, zda je online či nikoliv.

V rámci druhé, závěrečné, fáze jsem si dal dohromady seznam operací, k nimž v Pochlebníku dochází a postupně je zkoušel (na produkční verzi aplikace). Když jsem narazil na chybu, diagnostikoval jsem ji a na vývojovém serveru ji opravil. Když jsem takto prošel všechny operace, nahrál jsem změny na vývojový server a do mobilního zařízení a celý proces opakoval do té doby, než jsem na žádné chyby nenarazil. Typově mezi testované operace patřilo následující:

- Probíhá v pořádku registrace a autentizační mechanismy?
- Zvládá Pochlebník znaky české abecedy na všech místech, kam je uživatel může vložit?
- Komunikují spolu mobilní a webová aplikace hladce?
- Zvládá mobilní aplikace fungovat offline?

2.4 Mobilní aplikace

Pro vývoj mobilní aplikace jsem použil *Apache Cordovu*[8], která umožňuje vytvářet mobilní aplikace pomocí webových technologií. To se mi zdálo pro nástroj, jenž měl být pro aplikaci s webovou i mobilní verzí ideální, neboť jsem mohl velké množství svých HTML, CSS a JS kódů pouze překlopit do nového prostředí a nemusel se s každou změnou věnovat pracnému přepisování do XML a Javy/Kotlinu/Swiftu.

Výhodu multiplatformního vývoje v JavaScriptu nenabízela pouze Cordova ale také například React Native[9] popřípadě Ionic[10]. Dočetl jsem se však, že tyto nástroje dosud nejsou zcela stabilní a rozhodl se proto pro Cordovu. V úvahu přicházel i Flutter[11], který je údajně v současnosti nejpoužívanějším nástrojem pro multiplatformní vývoj, ovšem vzhledem k tomu, že má v jádru Dart a nikoliv mně známý JavaScript, jsem se pro něj nerozhodl. Nechtěl jsem riskovat, že bych si vzal příliš pomalý (ať už pro svou nízkoúrovňovost, nekompatibilitou s mně dostupnými zařízeními či pouhou neintuitivností) nástroj a kvůli tomu pak měl problém projekt dokončit[14].

Aplikace je zatím připravená pouze pro Android, ale Cordova by měla umožnit i snadné vygenerování její verze pro iOS.

2.4.1 Fungování offline

Mobilní aplikace je naprogramovaná tak, aby pokud možno plnohodnotně fungovala, i když je zrovna bez internetu. Dokud má přístup k internetu, ukládá si aplikace všechna data o zpětných

[8] https://cordova.apache.org/
[9] https://reactnative.dev/
[10] https://ionicframework.com/
[11] https://flutter.dev/

vazbách také v sobě samotné a když zjistí, že jí připojení chybí, začne data místo ze serveru načítat ze své lokální paměti.

Veškeré změny, jako například přidávání či mazání otázek ve zpětných vazbách, v tomto módu probíhají tak, že JavaScript zanese změny do lokálních datových struktur a zároveň do zvláštní datové struktury uloží popis zprávy, kterou má mobil poslat na server po té, co se znovu připojí k internetu.

2.4.1.1 Zpětné vazby offline

Pro vyplňování zpětných vazeb za nepřítomnosti internetového připojení však výše popsaný koncept použít nešel, neboť neumožňoval přenos dat mezi jednotlivými mobilními zařízeními.

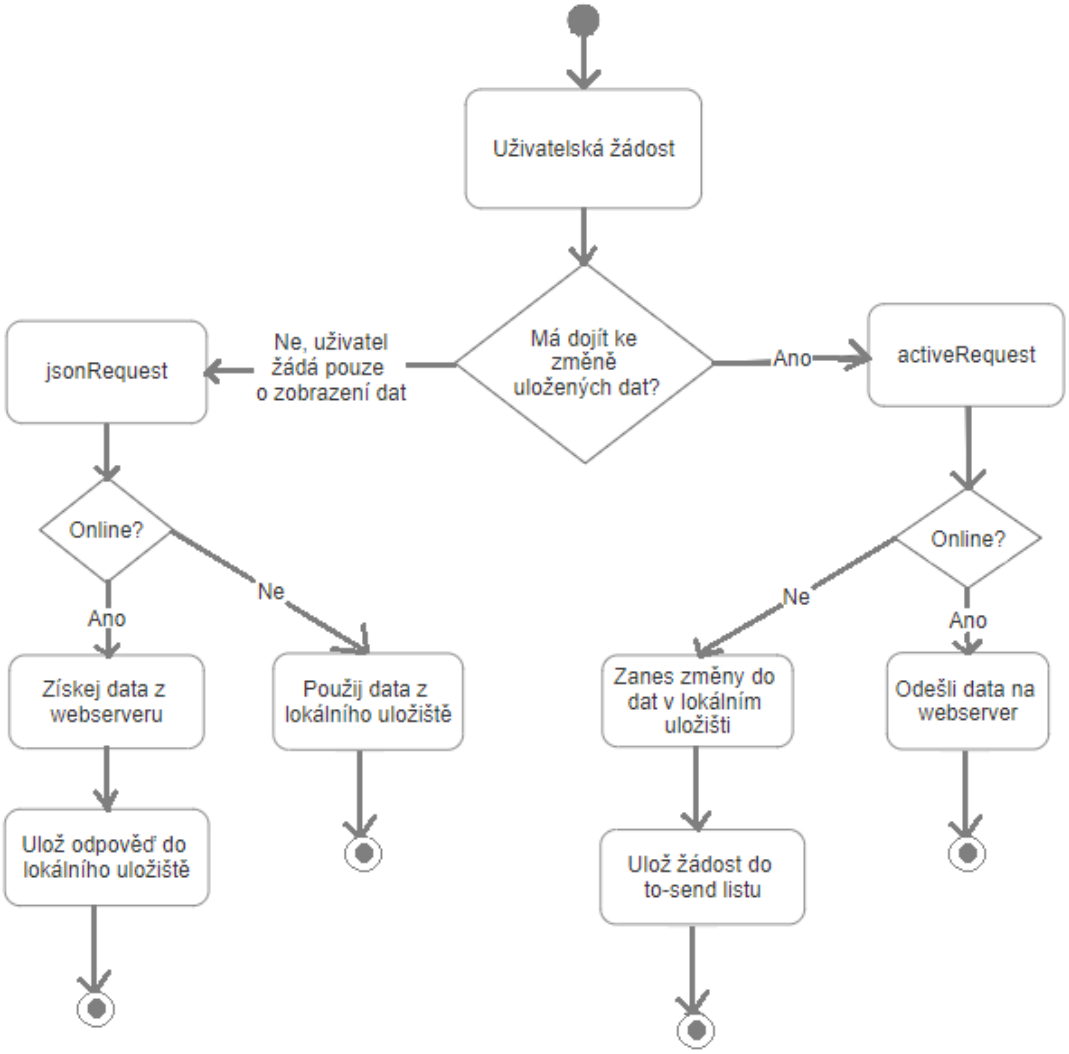
Původní vize přitom byla taková, že by zařízení organizátora zvládlo v případě potřeby zastoupit webový server, poslat respondentům výběr otázek a následně uložit jejich odpovědi. To se bohužel ukázalo jako nereálné. Zkusil jsem nicméně následující způsoby:

- *Komunikace přes hotspot* - Nejprve jsem chtěl, aby byl Pochlebník na telefonu organizátora schopen zapnout hotspot a rozjet na některém z portů webový server, který by byl schopen vyřídit žádosti o otázky a jejich odevzdání v podstatě stejně jako server standardní. Respondenti by se pouze připojili na organizátorovu WiFi a jejich aplikace by, po zjištění že nemají přístup k internetu, posílaly své žádosti na poskytovatele lokální sítě.
  Vzhledem k tomu, že Cordova nabízí webserverové pluginy[12], jsem se domníval, že je možné na mobilu tazatele spustit server, otevřít port, na kterém by se nacházel, a s tímto portem následně z mobilů respondentů komunikovat. Ukázalo se však, že pro otevření portu jsou třeba rootovská privilegia[16] a rozhodně nepřichází v úvahu, aby standardní součástí instalace bylo právě rootování telefonu[4]. Sledovat z aplikace, jaké žádosti přes hotspot chodí a některé z nich odchytávat by pak bylo ještě krkolomnější[13].
- *Bluetooth* - Když nefungoval hotspot, pokusil jsem se využít Bluetooth, technologii pro komunikaci elektronických zařízení na krátkou vzdálenost používanou většinou jako alternativu ke drátovému připojení[2].
  Vykomunikovat potřebná data přes bluetooth by teoreticky bylo možné, bohužel však Cordova nenabízí plugin, který by to umožňoval. Našel jsem a prozkoumal jsem hned tři. Nejdoporučovanější z nich však byl stavěn na komunikaci se zařízeními vysílajícími Bluetooth LE [6], další zas vyžadoval komunikaci přes sériový port a již v dokumentaci varoval, že není schopen zajistit komunikaci Androidu s Androidem[5]. Poslední pak byl již velice starý a neměl dostupnou dokumentaci.

Rozhodl jsem se proto do vyplňování zpětné vazby mimo internetové připojení více angažovat organizátora. Pokud si je organizátor vědom toho, že jeho zpětná vazba bude vyplňována mimo připojení k internetu, může si nechat vygenerovat QR kód pro offline vyplnění. Tento QR kód obsahuje JSON s přehledem jednotlivých otázek. Když účastníci tento QR kód v aplikaci načtou, vybere jejich aplikace až pět náhodných otázek a nechá je na ně odpovědět, nakonec uloží data o jejich odpovědí k dalším requestům "k odeslání". Zároveň mobilní aplikace po vyplnění zpětné vazby vygeneruje QR kód obsahující JSON s odpověďmi, ty může zadavatel zpětné vazby načíst do své aplikace a hodnocení si zobrazit.

[12] Účelem těchto pluginů je obvykle zprostředkování interní komunikace.

Obrázek 2.2: Mobilní aplikace, diagram aktivit



2.5 Uplatněné návrhové vzory

Při práci jsem se velmi snažil dodržovat princip *DRY* (Don't repeat yourself). To se projevilo především na frontendu, kde jsem téměř všechny operace skládal z atomických funkcí jako „pošli request" či „vykresli blok". I na serveru jsem se však snažil všechny opakované procesy (například načítání zpětné vazby) zapouzdřit do samostatných funkcí.

Dále jsem uplatnil *singleton* a to především ve způsobu, jakým se v mobilní aplikaci a na webovém frontendu zachází s daty o poslední načtené zpětné vazbě, která se ukládají ve formě objektu do jedné globální proměnné „fbData", kam k nim pak přistupují funkce pro jejich aktualizaci, generování offline zpětné vazby či zobrazení odpovědí.

Svou roli v mém navrhování softwaru sehrála i snaha dodržet *chain of responsibility* - kdykoliv je nějaký request či jiný objekt zpracováván několika různými metodami, má v každou chvíli jen jedna metoda či objekt prostor data zpracovávat. To se v praxi projevuji například tím, že program čeká na úspěšné vyřízení požadavku s nějakou změnou(například změnu statusu zpětné vazby, její smazání anebo odeslání vyplněných odpovědí), i když by teoreticky nemusel.

3. Technická dokumentace

3.1 Návod k modifikaci

Pochlebník je „chráněný“ MIT licencí a je proto zcela v pořádku si ho naklonovat a novu instanci libovolně modifikovat. K plnohodnotnému vývoji je však třeba nejprve udělat následující kroky:

1. Instalovat Git [1].
2. Instalovat Apache Cordovu [2]
3. Instalovat Python (verzi alespoň 3.7) [3]
4. Instalovat MySQL a vytvořit si lokální databázový prostor [4]
5. Příkazem „git clone https://github.com/Hobit2002/DOFBC“ naklonovat repozitář.
6. Příkazem „pip install django django-cors-headers qrcode“ instalovat potřebné knihovny Pythonu.
7. Záleží-li na zabezpečení nově vznikající instance, je vhodné změnit v settings.py heslo k databázi. Stejně tak je vhodné přepsat SECRET_KEY a DEBUG nastavit na False(taktéž v settings.py)
8. Vytvořit novou databázi jménem DOFBC a vytvořit databázového uživatele, který má pravomoc v rámci této databáze vytvářet, mazat i upravovat tabulky. Přihlašovací údaje tohoto uživatele by se měly shodovat s těmi, jež jsou uvedeny v server/server/settings.py.
9. Pomocí příkazů „python manage.py makemigrations“ a „python manage.py migrate“ spuštěných v cestě /server inicializovat databázi
10. Příkazem „python manage.py runserver 8001“ spustit vývojový webový server.
11. V cestě /DOFBCmobile pomocí příkazů „cordova platform add browser/android/ios...“ instalovat platformy, pro které budete chtít vyvíjet.
12. Příkazem „cordova run browser/android...“ spustit aplikace na příslušné platformě. Pokud testujete zařízení s Androidem, je vhodné mít do počítače pomocí USB kabelu připojené zařízení s povoleným USB debugováním [5].

3.2 Návod k použití

3.2.1 Instalace

3.2.1.1 Webová aplikace

Pro používání webové aplikace stačí zadat do prohlížeče adresu www.pochlebnik.eu a je to.

[1] https://git-scm.com/downloads
[2] https://evothings.com/doc/build/cordova-install-windows.html
[3] https://phoenixnap.com/kb/how-to-install-python-3-windows
[4] https://dev.mysql.com/doc/mysql-windows-excerpt/8.0/en/windows-installation.html
[5] https://www.microfocus.com/documentation/silk-test/210/en/silktestworkbench-help-en/GUID-BE1EA2BA-EFF2-4B2D-8F09-4BEE0947DFB2.html

3.2.1.2 Mobilní aplikace

Časem se snad Pochlebník objeví na Google Play, ovšem vzhledem k tomu, že schvalovací procedura trvá klidně i několik týdnů, jsou případní zájemci zatím nuceni si aplikaci instalovat méně konvenčním způsobem.

V následujícím seznamu uvádím kroky potřebné k instalaci aplikace na zařízení s Androidem:

1. Stáhnout si na počítač app.apk soubor obsahující zkompilovanou aplikaci. To je možné na GitHubu[6]. Pokud je k dispozici klon repozitáře, lze app.apk nalézt v podadresáři DOFBCmobile
2. Připojit USB kabelem k počítači zařízení s Androidem.
3. Povolit na zařízení USB debugování [7].
4. Instalovat si na počítač Minimal ADB and Fastboot [8].
5. Zkopírovat app.apk do adresáře s Minimal ADB and Fastboot
6. Otevřít v adresáři Minimal ADB and Fastboot příkazový řádek.
7. Příkazem „adb install app.apk“ instalovat aplikaci na zařízení.
8. Smazat adb.apk z adresáře Minimal ADB and Fastboot

3.2.2 Použití

3.2.2.1 Zadávání zpětných vazeb

Pro vytváření zpětných vazeb je třeba si založit účet, to je možné pouze pomocí webového rozhraní. Jakmile je však již účet vytvořený, lze se do aplikace přihlásit jak přes webové rozhraní tak z mobilní aplikace.

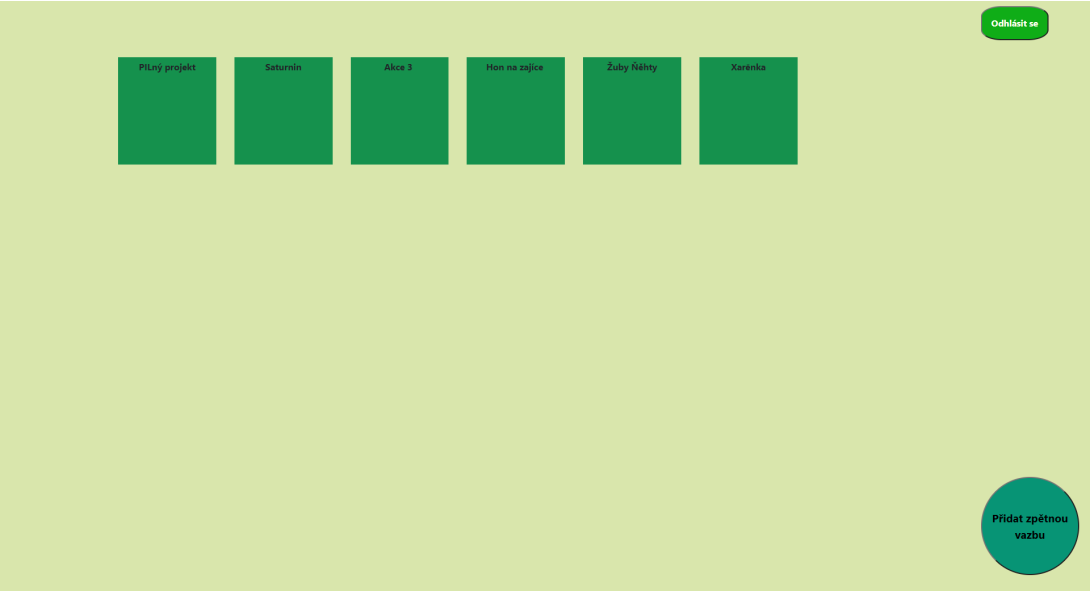
Po přihlášení se uživateli otevře obrazovka s tlačítkem pro každou již vytvořenou zpětnou vazbu (vizte Obrázek 3.1.). Když na některé z těchto tlačítek uživatel klikne, bude přesměrován na stránku dané zpětné vazby. V pravém dolním rohu je pak tlačítko, které po kliknutí vytvoří novou zpětnou vazbu.

[6] https://github.com/Hobit2002/DOFBC/blob/master/DOFBCmobile/app.apk

[7] https://www.microfocus.com/documentation/silk-test/210/en/silktestworkbench-help-en/GUID-BE1EA2BA-EFF2-4B2D-8F09-4BEE0947DFB2.html

[8] https://forum.xda-developers.com/t/tool-minimal-adb-and-fastboot-2-9-18.2317790/

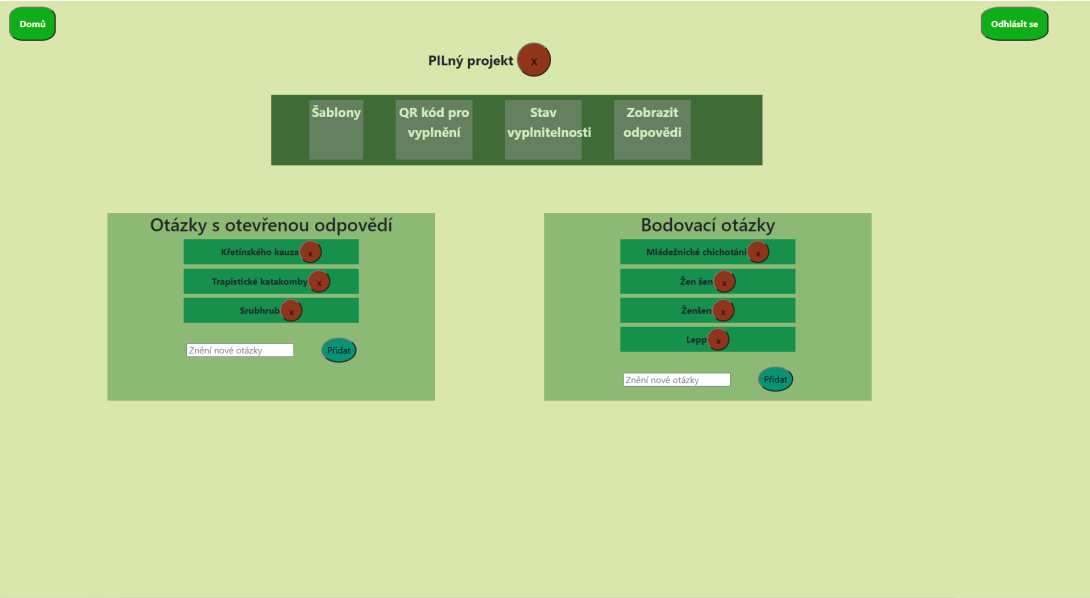
Obrázek 3.1: Domovská stránka



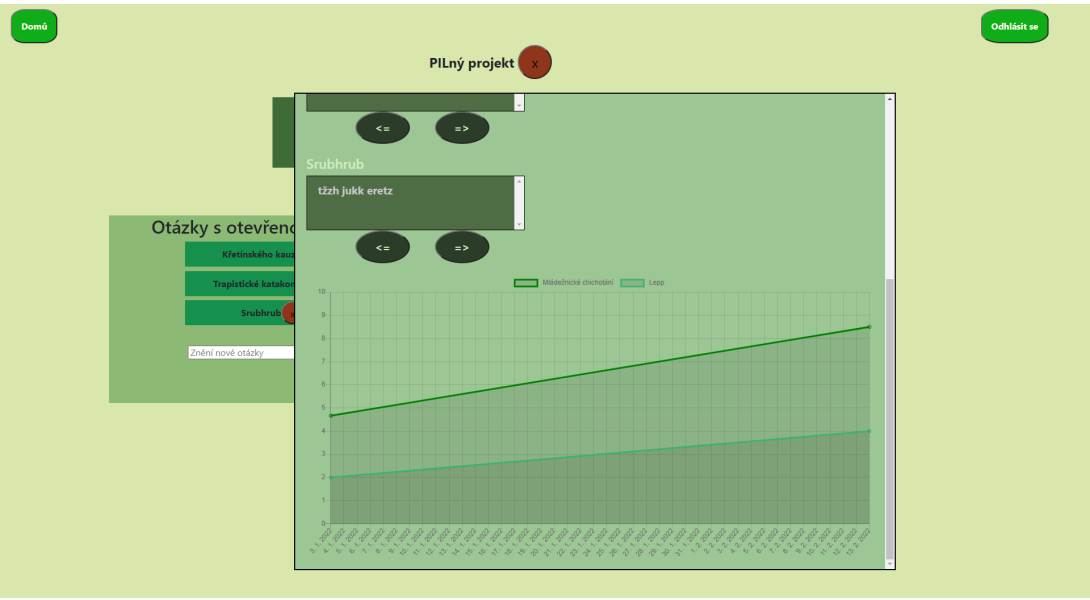
Na stránce pro správu zpětné vazby (vizte Obrázek 3.2), je možné:

- Kliknutím na rezavý křížek u názvu zpětné vazby danou zpětnou vazbu smazat.
- Kliknutím na název zpětné vazby zpětnou vazbu přejmenovat.
- Rozkliknutím výběru „Použít šablonu“ zvolit dříve uloženou sadu otázek, která bude aplikována i na tuto zpětnou vazbu.
- Vyplněním políčka „Uložit jako šablonu“ a kliknutím na tlačítko „Přidat“ uložit sadu otázek této zpětné vazby jako šablonu, která bude moct být použita v budoucnou.
- Vyplněním políčka „Znění nové otázky“ a kliknutím na tlačítko "Přidat", přidat novou otevřenou nebo uzavřenou otázku.
- Kliknutím na položku s otázkou tuto otázku přeformulovat.
- Kliknutím na rezavý křížek po pravé straně položky s otázkou danou otázku smazat.
- Kliknutím na tlačítko „QR kód pro vyplnění“ v horní liště si zobrazit údaje nutné pro vyplnění zpětné vazby. Pochlebník nabízí údaje pro vyplňování za přítomnosti i nepřítomnosti internetu.
- Kliknutím na tlačítko Stav vyplnitelnosti lze přenastavit stav zpětné vazby. Stavy existují tři: Připravovaná (výchozí), Otevřená a Uzavřená. Pokud je zpětná vazba otevřená, je možné ji vyplnit.
- Pokud je zpětná vazba otevřená, nebo uzavřená, nabízí se na horní liště tlačítko „Zobrazit odpovědi“. Po jeho rozkliknutí se uživatelům zobrazí přehled odpovědí na jednotlivé otázky a zároveň graf, kde jsou průměrné odpovědi na bodovací otázky porovnány s předchozími akcemi (vizte Obrázek 3.3).
- Pouze pro mobilní aplikaci: pokud je mobil offline a zpětná vazba je otevřená, nabízí se na horní liště tlačítko „Naskenovat odpověď“. Pokud na něj uživatel klikne, otevře se QR scanner připravený k načtení QR kódu s odpověď'mi na offline vyplňovanou zpětnou vazbu.

Obrázek 3.2: Stránka zpětné vazby



Obrázek 3.3: Shrnutí zpětné vazby



3.2.2.2 Vyplňování zpětných vazeb

Pro vyplnění zpětných vazeb stačí zadat příslušnou adresu do prohlížeče nebo mobilní aplikace popřípadě v mobilní aplikaci naskenovat příslušný QR kód (vlastní účet k vyplňování není třeba). Zobrazí se stránka (vizte Obrázek 3.4), na které respondent buď zakliká hvězdičky či nastaví správnou hodnotu „teploměru“ pro otázky uzavřené či do textových polí uvede odpovědi na otázky otevřené.

Pokud vyplňování probíhá offline, zobrazí se po vyplnění zpětné vazby QR kód. Pokud zadavatel tento QR kód načte do své mobilní aplikace, bude si moct odpovědi zobrazit i bez připojení k internetu.

Obrázek 3.4: Vyplňování zpětné vazby

Závěr

Z pozice studenta považuji projekt za úspěšný. Zadaných cílů, z nichž některé rozhodně nebyly triviální, jsem dosáhl v ze svého pohledu uspokojivé míře. Jsem si vědom, že aplikace stále není zcela bezvadná a že by se dal ještě jak vylepšit design, tak vyladit několik funkčních maličkostí, ale věřím, že je bez významnějších problémů použitelná v reálném světě.

V rámci projektu jsem výrazně prohloubil své znalosti Djanga a především vývoje mobilních aplikací (s Apache Cordovou jsem do té doby nepracoval). V rámci tvorby Pochlebníka jsem také vůbec poprvé pracoval s QR kódy. Za velmi užitečné považuji studium toho, jak v aplikaci využít Bluetooth (a to i když mně samotnému se nakonec nehodil), stejně tak jako znalost bezpečnostních mechanismů webových frontendů (až při tomto projektu jsem zjistil, co jsou to CORS) a mobilních telefonů.

Bibliografie

[1] Evothings AB. *Installing Cordova on Windows*. URL: https://evothings.com/doc/build/cordova-install-windows.html.

[2] kolektiv anonymních autorů. *Bluetooth*. URL: https://en.wikipedia.org/wiki/Bluetooth.

[3] kolektiv anonymních autorů. *Use case diagram*. URL: https://en.wikipedia.org/wiki/Use_case_diagram.

[4] Paula Beaton. *How to root Android phones and tablets (and unroot them)*. URL: https://www.digitaltrends.com/mobile/how-to-root-android/.

[5] Maxim Belov. *Bluetooth Classic Serial Plugin for Cordova*. URL: https://www.npmjs.com/package/cordova-plugin-bluetooth-classic-serial-port#list.

[6] Rand Dusing. *Cordova Bluetooth LE Plugin*. URL: https://github.com/randdusing/cordova-plugin-bluetoothle/blob/master/readme.md.

[7] Django Software Foundation. *Cross Site Request Forgery protection*. URL: https://docs.djangoproject.com/en/4.0/ref/csrf/.

[8] Django Software Foundation. *Django documentation*. URL: https://docs.djangoproject.com/en/4.0/.

[9] The Apache Software Foundation. *Apache Cordova Documentation*. URL: https://cordova.apache.org/docs/en/11.x/.

[10] David Gloag. *Manual vs Automated Unit Testing*. URL: https://study.com/academy/lesson/manual-vs-automated-unit-testing.html.

[11] Pavel Gorbachenko. *What are Functional and Non-Functional Requirements and How to Document These*. URL: https://enkonix.com/blog/functional-requirements-vs-non-functional/.

[12] Lukáš Haraga. *Jediný komplexní nástroj pro správu Tvého oddílu*. URL: https://www.levitio.cz/.

[13] 123wee & Irfan Latif. *How to redirect all web requests from hosts connected to hotspot to a local web server?* URL: https://android.stackexchange.com/questions/162093/how-to-redirect-all-web-requests-from-hosts-connected-to-hotspot-to-a-local-web.

[14] Yuriy Luchaninov. *The Sunset of Apache Cordova: Alternatives For Cross Platform Mobile Development in 2022*. URL: https://mobidev.biz/blog/apache-cordova-alternatives-cross-platform-mobile-app-development.

[15] MDN přispěvatelé. *Cross-Origin Resource Sharing (CORS)*. URL: https://developer.mozilla.org/en-US/docs/Web/HTTP/CORS.

[16] Deepak Mittal & Sunny. *How to run a server on port 80 as a normal user on Linux?* URL: https://serverfault.com/questions/112795/how-to-run-a-server-on-port-80-as-a-normal-user-on-linux.

[17] Athira Unnikrishnan. *10 Best Online Form Builder Apps to look for in 2021*. URL: https://surveysparrow.com/blog/form-builder-apps/.

Seznam obrázků