



## Zadání bakalářské práce

<b>Název:</b>	Rozvoj aplikace pro dobrovolnickou brigádu SummerJob
<b>Student:</b>	Aneta Potomská
<b>Vedoucí:</b>	Ing. Ladislav Louka
<b>Studijní program:</b>	Informatika
<b>Obor / specializace:</b>	Softwarové inženýrství 2021
<b>Katedra:</b>	Katedra softwarového inženýrství
<b>Platnost zadání:</b>	do konce letního semestru 2024/2025

### Pokyny pro vypracování

SummerJob je letní dobrovolnická brigáda ve vesnicích a městečkách českého pohraničí. Každý rok při ní během jednoho týdne pomáhá 150 mladých dobrovolníků místním obyvatelům. Účastníci jsou rozděleni do skupin a vykonávají dobrovolnické práce, v některých případech ve vzdálenějších oblastech s vlastní dopravou a nástroji.

Cílem práce je rozvinout existující aplikaci především o funkce pro usnadnění administrativní práce organizátorů a komunikaci s účastníky a zlepšení celkového UI/UX aplikace.

- 1) Zanalyzujte současné řešení a požadavky organizátorů i účastníků na aplikaci
- 2) Navrhněte rozšíření stávající aplikace
- 3) Implementujte svůj návrh
- 4) Proveďte testování aplikace





Bakalářská práce

# Rozvoj aplikace pro dobrovolnickou brigádu SummerJob

Aneta Potomská

Fakulta informačních technologií  
Katedra softwarového inženýrství  
Vedoucí: Ing. Ladislav Louka  
13. května 2024

České vysoké učení technické v Praze  
Fakulta informačních technologií

© 2024 Aneta Potomská. 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, je nezbytný souhlas autora.*

Odkaz na tuto práci: Potomská Aneta. *Rozvoj aplikace pro dobrovolnickou brigádu SummerJob*.  
Bakalářská práce. České vysoké učení technické v Praze, Fakulta informačních technologií, 2024.

# Obsah

Poděkování	viii
Prohlášení	ix
Abstrakt	x
Seznam zkratek	xi
Úvod	1
<b>1 Analýza</b>	<b>3</b>
1.1 Východiska	3
1.1.1 Statistiky pracantů a jobů	3
1.1.2 Plánování a organizace	4
1.1.3 Historické řešení	4
1.2 Současné řešení	4
1.2.1 Architektura systému	5
1.2.2 Technologie	6
1.2.3 Struktura aplikace	8
1.2.4 Pohled na aplikaci	10
1.3 Sběr požadavků	16
1.3.1 Nevýhody současného řešení	16
1.3.2 Možnosti rozšíření ze současného řešení	18
1.3.3 Dotazník	18
1.3.4 GitHub issues	20
1.4 Analýza požadavků	21
1.4.1 Funkční požadavky	21
1.4.2 Nefunkční požadavky	22
1.4.3 Model případů užití	23
<b>2 Návrh</b>	<b>29</b>
2.1 Zvolené technologie	29
2.1.1 Mapa	29
2.1.2 Nahrávání fotografií	31
2.2 Přístupy	32
2.3 Návrh obrazovek a prototypy	32
2.3.1 Mapa	32
2.3.2 Fotografie	33
2.3.3 Výběr dostupnosti	34
2.3.4 Nástroje	35
2.3.5 Formuláře	35
2.3.6 Nástěnka	36
2.3.7 Můj plán	42
2.4 Konceptuální model	43

<b>3 Implementace</b>	<b>47</b>
3.1 Implementace klíčových částí	47
3.1.1 Formuláře	47
3.1.2 Fotografie	53
3.1.3 Nástěnka	55
3.2 Implementace drobných požadavků	60
3.2.1 Tabulky	60
3.2.2 Plány	61
3.2.3 Joby	61
3.2.4 Auta	62
3.2.5 Pracanti	62
3.2.6 Můj plán	62
<b>4 Testování</b>	<b>65</b>
4.1 Technologie	65
4.1.1 SuperTest	65
4.1.2 Mocha	65
4.1.3 Chai	65
4.1.4 CodeQL	66
4.2 Testování API	66
4.3 Statická analýza kódu	66
4.4 Testování uživateli	66
4.4.1 A/B testování	68
4.4.2 Testování použitelnosti	70
4.5 Ověření splnění požadavků	73
4.5.1 Funkční požadavky	73
4.5.2 Nefunkční požadavky	74
<b>5 Zhodnocení a návrh dalšího rozšíření</b>	<b>75</b>
<b>Závěr</b>	<b>77</b>
<b>A Dotazník pro administrátory</b>	<b>79</b>
<b>B Dotazník pro A/B testování</b>	<b>83</b>
<b>C Dotazník pro testování použitelnosti</b>	<b>89</b>
<b>Obsah příloh</b>	<b>97</b>

## Seznam obrázků

1.1	Vývoj počtu jobů a účastníků. Získáno z výroční zprávy 2019 akce SummerJob [8]	3
1.2	Architektura systému. Vytvořeno v programu Enterprise Architect [15]	5
1.3	Přihlašování	10
1.4	Vrchní panel s přesměrováním do „Můj plán“	10
1.5	Úvodní stránka pro správu pracantů	11
1.6	Úprava pracanta	12
1.7	Úprava auta	13
1.8	Úvodní stránka pro správu jobů	13
1.9	Úvodní stránka pro administraci	14
1.10	Detaily konkrétního ročníku	14
1.11	Detail jobu v plánu	15
1.12	Můj plán	16
1.13	Aktéři. Vytvořeno v programu Enterprise Architect [15]	23
1.14	Případy užití ve schématu. Vytvořeno v programu Enterprise Architect [15]	27
2.1	Návrh mapy v nástroji Figma [42]	33
2.2	Návrh uživatelského prostředí pro nahrání fotografií v nástroji Figma [42]	34
2.3	Návrh výběru dostupnosti v nástroji Figma [42]	34
2.4	Návrh nástrojů v nástroji Figma [42]	35
2.5	Návrh formuláře v nástroji Figma [42]	36
2.6	Návrh formuláře pro úpravu příspěvku v nástroji Figma [42]	37
2.7	Návrh příspěvku v nástroji Figma [42]	38
2.8	Návrh modálního okna příspěvku v nástroji Figma [42]	39
2.9	Návrh modálního okna pro řazení příspěvků v nástroji Figma [42]	40
2.10	Návrh modálního okna pro filtrování příspěvků v nástroji Figma [42]	40
2.11	Návrh úvodní stránky nástěnky v nástroji Figma [42]	41
2.12	Návrh obsahu pro stránku „Můj plán“ v nástroji Figma [42]	42
2.13	Konceptuální model vycházející ze současného stavu. Vytvořeno v modelovacím nástroji portálu dbs [43]	43
2.14	Návrh úpravy v konceptuálním modelu. Vytvořeno v modelovacím nástroji portálu dbs [43]	45
3.1	Implementovaný výběr dostupnosti	49
3.2	Implementovaný výběr buněk	50
3.3	Snímky obrazovek formuláře pro úpravu jobu z implementace.	51
3.4	Screenshot detailu příspěvku z nástěnky z implementace.	58
3.5	Snímek obrazovky mého plánu z implementace	63
A.1	Odpovědi na otázku „Kolika ročníků SummerJob jste se zúčastnil/a (respektive spravoval/a)?“	79

## Seznam tabulek

1.1	Role . . . . .	10
2.1	Nové role . . . . .	32
3.1	Rozšířené REST API o části k nástěnce . . . . .	60
A.1	Odpovědi na otázku „Jste si vědom/a nějakých chyb/bugů v systému? Případně uveďte.“ . . . . .	80
A.2	Odpovědi na otázku „Myslíte si, že existují oblasti, kde by uživatelské rozhraní mohlo být intuitivnější? Pokud ano, uveďte konkrétně.“ . . . . .	80
A.3	Odpovědi na otázku „Můžete uvést příklady situací (pokud takové existují), ve kterých bylo používání současného systému náročné nebo/a neefektivní?“ . . . . .	81
A.4	Odpovědi na otázku „Existují nějaké konkrétní nedostatky či obtíže, se kterými se setkáváte při používání současného systému? Případně napište jakýkoliv návrh na zlepšení dané problematiky.“ . . . . .	81
A.5	Odpovědi na otázku „Musíte používat nějaké externí aplikace/zdroje na řešení jistých problematik, které byste preferoval/a, aby se daly řešit přímo v systému?“ . . . . .	82
A.6	Odpovědi na otázku „Schází Vám v systému něco? Pokud ano, uveďte konkrétní návrhy na vylepšení nebo na nové funkce, které byste rád/a viděl/a implementované.“ . . . . .	82
B.1	Zhodnocení prvku „AB1 – Výběr dostupnosti“ . . . . .	83
B.2	Zhodnocení prvku „AB2 – Vyhodnocování formuláře“ . . . . .	84
B.3	Zhodnocení prvku „AB3 – Odeslání formuláře“ . . . . .	85
B.4	Zhodnocení prvku „AB4 – Obsahová stránka jobu v mém plánu“ . . . . .	86
B.5	Zhodnocení prvku „AB5 – Responzivita tabulky“ . . . . .	87
B.6	Zhodnocení prvku „AB6 – Statistiky“ . . . . .	88
C.1	Zhodnocení scénáře „T1 – Administrativní práce s nástěnkou“ . . . . .	90
C.2	Zhodnocení scénáře „T2 – Uživatelská práce s nástěnkou“ . . . . .	91
C.3	Zhodnocení scénáře „T3 – Orientace v mém plánu“ . . . . .	92



## Seznam výpisů kódu

3.1	Komponenta <code>TextInput.tsx</code> . . . . .	48
3.2	Zakázání SSR pro komponentu <code>Mapy</code> v souboru <code>index.ts</code> . . . . .	49
3.3	Nastavení vlastní ikony připínačku do <code>mapy</code> . . . . .	50
3.4	Zod schéma pro vytvářeného <code>pracanta</code> . . . . .	52
3.5	CSS pro změnu velikosti okna nahrané fotografie. Výpis je ze souboru <code>custom.css</code> . . . . .	53
3.6	Zobrazení fotografie. Výpis je ze souboru <code>ImageUploader.tsx</code> . . . . .	53
3.7	Zakázání parseru . . . . .	54
3.8	Rozřazení příspěvků. Výpis kódu vychází z komponenty <code>PostsClientPage.tsx</code> . . . . .	57
3.9	Query parametry. Výpis kódu vychází z komponenty <code>PostsClientPage.tsx</code> . . . . .	59
4.1	Testování vytváření příspěvku s fotografií. Výpis kódu vychází z <code>posts.spec.ts</code> . . . . .	67

## **Poděkování**

Ráda bych vyjádřila vděk Ing. Ladislavu Loukovi, vedoucímu této práce, za jeho cenné rady a užitečné připomínky. Dále bych pak ráda poděkovala týmu SummerJob za jejich ochotu se mnou konzultovat návrhové detaily, za poskytnuté informace ohledně fungování akce a vyplnění dotazníku.

## Prohlášení

Prohlašuji, že jsem předloženou práci vypracovala samostatně a že jsem uvedla veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při 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. § 2373 odst. 2 zákona č. 89/2012 Sb., občanský zákoník, ve znění pozdějších předpisů, 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 ne snižuje hodnotu Díla a za jakýmkoli účelem (včetně užití k výdělečným účelům). Toto oprávnění je časově, teritoriálně i množstevně neomezené.

V Praze dne 13. května 2024

.....

## Abstrakt

Tato bakalářská práce se zabývá rozvojem stávající webové aplikace pro dobrovolnickou brigádu SummerJob. Mezi hlavní cíle vytvořeného rozšíření patří usnadnění administrativní činnosti organizátorů akce, posílení komunikace mezi týmem SummerJob a účastníky akce a vylepšení uživatelské přívětivosti a intuitivnosti aplikace.

V práci je provedena analýza současného stavu aplikace. V rámci ní jsou zajištěny, zanalyzovány a popsány požadavky na rozšíření. Tyto požadavky jsou převedeny do návrhu, na který dále navazuje implementace. Implementace je provedena ve frameworku Next.js s využitím statického typování pomocí jazyku TypeScript. Na závěr bylo vytvořené rozšíření otestováno. Výsledkem této práce je funkční rozšířená webová aplikace umožňující efektivní chod akce SummerJob.

**Klíčová slova** webová aplikace, uživatelská přívětivost, SummerJob, dobrovolnická brigáda, Next.js, TypeScript

## Abstract

This bachelor thesis deals with the development of an existing web application for the SummerJob volunteer job. The main goals of the developed extension include simplifying the administrative activities of the event organizers, enhancing communication between the SummerJob team and the event participants, and improving the user-friendliness and intuitiveness of the application.

In this paper, an analysis of the current state of the application is performed. Within it, enhancement requirements are secured, analysed and described. These requirements are then translated into a design, which is further followed by implementation. The implementation is done in the Next.js framework using static typing using the TypeScript language. Finally, the developed extension was tested. The result of this work is a functional extended web application that enables the efficient running of the SummerJob event.

**Keywords** web application, user-friendliness, SummerJob, volunteer work, Next.js, TypeScript

## Seznam zkratek

AMQP	Advanced Message Queuing Protocol
API	Application Programming Interface
BDD	Behavior-driven development
CSS	Cascading Style Sheets
CSV	Comma-separated Values
DRY	Don't repeat yourself
ER	Entity-relationship
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
ID	Identification
JS	JavaScript
JSON	JavaScript Object Notation
ORM	Object Relational Mapping
REST	Representational State Transfer
TDD	Test-driven development
UI/UX	User Interface/User Experience
URL	Uniform Resource Locator



# Úvod

SummerJob je letní dobrovolnická brigáda, která se zpravidla koná na začátku července a trvá průměrně 8 dní. Náplní je dobrovolná činnost – pomoc obyvatelům vybraných oblastí pro daný ročník ve vesnicích a městečkách v České republice. Taková výpomoc má podobu všemožných prací, s kterou by místní potřebovali pomocnou ruku. Může se jednat o nasekání dříví, malování stěn, zahradničení, úklid v domácnosti, vyklízení sklepů a mnoho dalšího. Občasně některé oblasti vyžadují vlastní dopravu či nástroje a od účastníků speciálně i jistou sadu dovedností. Kromě práce nabízí SummerJob pro účastníky a místní obyvatele doprovodný program. V něm se nacházejí události jako je například fotbalový turnaj, koncert, adorace a mše. [1]

Důležitou poznámkou pro text této práce je vymezení terminologie akce SummerJob. Konkrétně se jedná o označení pro brigádníka/účastníka akce – **pracant**. Dále pak o termín **job**, který značí práci, které se pracanti účastní a je potřeba jej naplánovat.

Akci organizují mladí lidé z Hnutí fokoláre a jejich přátelé. Na akci se každoročně půl roku dopředu intenzivně připravuje třicetičlenný tým. V době konání akce se tento tým stará a vede další účastníky. [2] Tým SummerJob ve své přípravě na akci musí zařizovat několik věcí, mezi ty nejpodstatnější je zajištění pracantů a jobů. Každý rok se hledá kolem 120 pracantů ve věku od 18 do 35 let [3]. Dohromady se tedy jednoho ročníku účastní na 150 lidí.

SummerJob se koná už několik let, letos jde již o 15 ročník [4]. Každý ročník se koná v různých oblastech České republiky. Tyto oblasti se po několika letech obměňují. [5] Minulý rok, tedy 2023, se akce SummerJob přesunula do severních Čech a jejím novým působištěm se tak stala Krásná Lípa a okolí. Letos v roce 2024 zde zamíří podruhé. [6] Taktéž to bude druhý rok, kdy se bude využívat nová webová aplikace. Ta byla vyvinuta minulý rok v rámci diplomové práce Ing. Matyáše Ješiny [7].

Obecně tato aplikace zaznamenala pozitivní ohlasy a to jak od obyčejných uživatelů, tak také i od administrátorů, kteří ji využívají k efektivnější organizaci, plánování a sledování průběhu akce SummerJob. I přesto, že dosáhla významných úspěchů, existuje zde prostor na vylepšení. Taktéž se bohužel neobešla bez drobných chyb, jež potřebují ošetřit. S ohledem na tento fakt společně se zpětnou vazbu od uživatelů a požadavky administrátorů bylo rozhodnuto o rozšíření této aplikace. Tohoto úkolu jsem se zhostila já. Učinila jsem tak, protože mě zaujala myšlenka a dobročinný duch projektu SummerJob.

Hlavním cílem této bakalářské práce je rozvinout existující aplikaci. To tedy znamená odstranit nedostatky stávajícího řešení, ale zároveň přinést nové funkcionality. Tyto funkcionality pak přispějí k efektivnímu chodu akce SummerJob, neboť usnadní administrativní činnost organizátorů. V neposlední řadě posílí komunikaci mezi organizátory a účastníky akce. Finálně pak tyto změny přinesou zvýšení uživatelské přívětivosti a intuitivnosti aplikace, tedy celkově lepší UI/UX. Věřím, že mnou navržené rozšíření webové aplikace bude mít pozitivní přínos pro celkovou organizaci a ještě plynulejší průběh akce SummerJob.

K dosažení celkového cíle je potřeba splnění dílčích cílů. Tato práce je podle nich strukturovaná. Konkrétně se tato práce zabývá analýzou, návrhem, implementací a testováním rozšíření stávající aplikace. Kapitola „Analýza“ analyzuje projekt jako takový a také analyzuje současný stav aplikace. Z této analýzy jsou poté zjištěny chyby a nedodělky, které představují první požadavky na vylepšení stávající aplikace. Další požadavky jsou pak zajištěny i například prostřednictvím dotazníku a z GitHub issues projektu. Tyto požadavky jsou dále analyzovány a rozděleny do finálních funkčních a nefunkčních požadavků. Pro vybrané funkční požadavky byl vytvořen model případu užití.

Vlastním návrhem rozšíření aplikace na základě sesbíraných požadavků se věnuje kapitola „Návrh“. Součástí kapitoly je výběr technologií, návrh na nové role pro uživatele, vizuální návrhy obrazovek a funkční prototypy. Ve finále je v této kapitole vytvořen konceptuální model sloužící jako předloha pro databázové schéma rozšíření. Kapitola „Implementace“ je zaměřena na popis vlastní implementace požadavků dle vytvořených návrhů. V kapitole je přiblížena struktura projektu a popsán proces implementace jednotlivých požadavků. V rámci každého popisu jsou zmíněny vždy klíčové části při implementaci, jako je frontend, backend, databáze, překonané obtíže a obhájení jistých rozhodnutí.

Náplní kapitoly „Testování“ je testování aplikace, se silným důrazem na nově přidané funkcionality. Hlavním obsahem této kapitoly je popis procesu testování. Konkrétně je zde popsáno testování API, statická analýza kódu a testování s uživateli včetně A/B testování pro potvrzení zvolených vizuálních prvků. Na závěr tato kapitola zhodnocuje splnění funkčních a nefunkčních požadavků vydefinovaných v rámci analýzy. Finální kapitola „Zhodnocení a návrh dalšího rozšíření“ je zaměřena na všeobecné zhodnocení splnění celkového cíle. Kromě toho je součástí krátká diskuze o možnostech dalšího rozšíření po změnách, jež budou představeny v rámci této práce.



# Kapitola 1

## Analýza

Tato kapitola si klade za cíl základně seznámit čtenáře s průběhem akce SummerJob, přiblížit současné řešení aplikace, popsat získávané požadavky na rozšíření a tyto požadavky dále zanalyzovat. Výstupem této analýzy jsou popsány funkční a nefunkční požadavky, včetně modelu případu užití pro některé vybrané funkční požadavky.

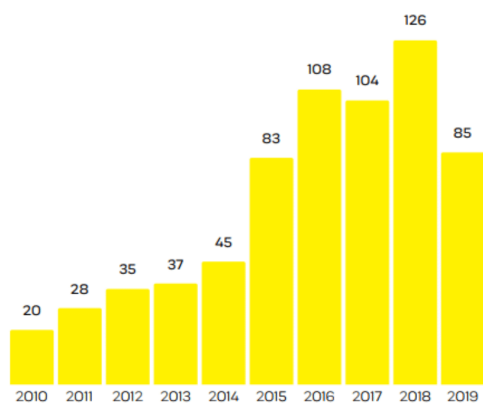
### 1.1 Východiska

Tato podkapitola se zaměřuje na klíčové prvky, pro porozumění problematice akce SummerJob.

#### 1.1.1 Statistiky pracantů a jobů

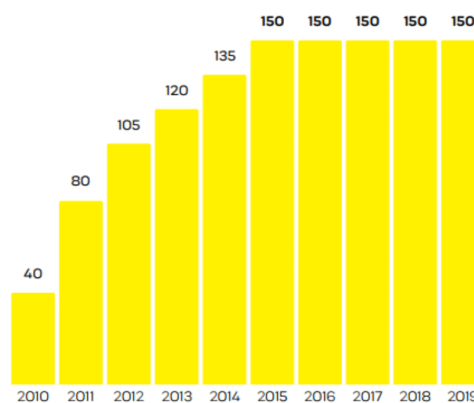
Na následujících obrázcích se můžete podívat na vývoj jobů a účastníků v rámci akce SummerJob za období 2010 – 2019.

VÝVOJ POČTU JOBŮ



(a) Vývoj počtu jobů

VÝVOJ POČTU ÚČASTNÍKŮ



(b) Vývoj počtu účastníků

■ **Obrázek 1.1** Vývoj počtu jobů a účastníků. Získáno z výroční zprávy 2019 akce SummerJob [8]

Statistiky ze současných let na stránkách SummerJobu dostupné nejsou. Částečně je tomu tak kvůli pandemii Covidu-19. Po zkontaktování organizátorů akce byly získány další údaje za roky

**2022:** 93 jobů a 160 účastníků,

**2023:** 127 jobů a 95 účastníků – „*Celkem jsme měli přihlášených asi 95 účastníků (trochu odhad, každý den se to měnilo) a bylo tam 29 členů týmu. Pracovali jsme na 86 unikátních místech, někde vícekrát, takže jsme celkem vyslali 127 skupin. Nejmenší skupina měla 1 člověka, největší 10. Celkem jsme za týden odpracovali 1,6 pracovního roku člověka.*“ [9].

### 1.1.2 Plánování a organizace

Práce organizačního týmu SummerJob spočívá v delikátní koordinaci zajištěných prací a vlastních zdrojů. Na každý den musí vytvořit celistvý plán pokrývající požadavky poskytovatelů jobů a požadavky a možnosti svých pracantů. Primárně se řeší sladění mezi časovou dostupností nabídnutých jobů a časovými možnostmi přihlášených účastníků akce. Dále se musí zkoordinovat možnosti dopravy do oblastí jobů a rozvrhnout dostupné nástroje. Kromě zmíněného je potřeba zharmonizovat rozdělení pracantů na joby, tak aby bylo vyhověno, i mimo zmíněné časové dostupnosti, jejich alergiím, schopnostem s nástroji a předešlé pracovní historii v aktuálním ročníku. [10]

Z toho vyplývá, že členové týmu SummerJob potřebují evidovat spoustu věcí, jako jsou například joby, pracanti a auta. Kromě toho potřebují pomocný nástroj, který jim pomůže naplánovat každý den, aby bylo vyhověno parametrům práce, ale také účastníkům. Účastníci proto potřebují své místo, kde by si specifikovali své požadavky, časové možnosti a podobně. Uživatelé potřebují prostor, kde by viděli jaké práce se daný den zúčastní a jaký je na ten den naplánovaný program. [10]

Ze jmenovaných důvodů je evidentní, že bylo potřeba přijít se systémem, pomocníkem pro organizačtorskou činnost, přehledným nástrojem pro evidování, plánování a zapisování.

### 1.1.3 Historické řešení

V minulosti se pro evidenci dat a plánování používala aplikace JobsPlanner využívající technologii .NET, která fungovala pouze na počítačích s nainstalovaným systémem Windows. Existovalo v ní jednoduché webové rozhraní pro evidenci pracantů, jobů a aut. Dále pak aplikace nabízela vygenerování plánů a jejich tisk. Byla využívána pouze týmem SummerJob, neboť zde nebyl pro účastníky způsob zobrazit si svůj denní plán a nebo si upravit preference. Mezi další nevýhody pak například patří nemožnost práce více uživateli současně. [7] O dalších informacích a popsáních nevýhodách tohoto řešení se můžete dočíst v diplomové práci Ing. Matyáše Ješiny v kapitole o současném řešení [7].

## 1.2 Současné řešení

Právě v rámci zmíněné diplomové práce byla vytvořená stávající podoba aplikace, která zcela nahradila aplikaci JobsPlanner. V kostce si nová aplikace od té staré vzala to dobré, to špatné opravila a představila nové funkcionality.

Po práci Ing. Ješiny na projektu ještě proběhly jisté změny implementované organizačním týmem SummerJob. O této skutečnosti se lze přesvědčit v GitHub historii projektu [11]. Poslední změny byly dodány dokonce 07.07.2023 [12], kdy SummerJob ročník 2023 probíhal v období 02.–09.07.2023 [13]. Některé změny byly tedy přidány ve spěchu a je i pravděpodobné, že v aplikaci budou z tohoto důvodu zaneseny chyby.

Každopádně z výše uvedených důvodů některé obrazovky a popisy, které se zde budou vyskytovat, nemusí zcela odpovídat konečnému stavu popsanému ve zmíněné diplomové práci. Konkrétně vše co zde bude popsáno vychází z commitu 332dd26 se jménem „Add advanced search“ [14]. Data jež lze vidět na některých obrazovkách nejsou reálné údaje, ale byla vygenerována pomocí knihovny Faker.

Náplní této podkapitoly bude představení a přiblížení současného stavu webové aplikace pro SummerJob. Cílem této podkapitoly je seznámit čtenáře s výchozím stavem, ze kterého vznikají změny představené v rámci této práce. Dále pak si klade za cíl podrobněji vysvětlit činnosti a procesy týmu SummerJob v korelaci s používáním stávající aplikace.

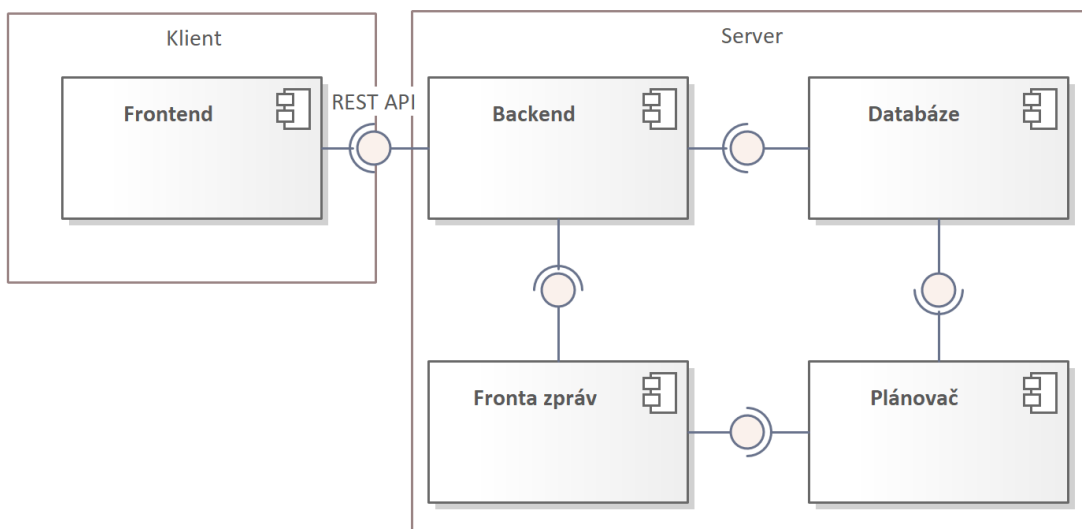
## 1.2.1 Architektura systému

Aplikace se dá rozdělit na dvě části – na tu klientskou a na serverovou. Klientská část představuje frontend webové aplikace. Serverová část je složena ze 4 stavebních bloků/komponent.

Prvním blokem serverové části je backend webové aplikace. Ten poskytuje API pro přístup k datům, jež jsou uložena v databázi. Právě databáze je druhým blokem serverové části. Třetím je pak automatický plánovač. Ten je takto vyčleněn jako samostatná služba, aby byla snazší jeho údržba a byla k dispozici možnost jeho jednoduché výměny. Díky snadné výměny plánovače je možné v budoucnu vybrat jakýkoliv programovacího jazyk a technologie pro jeho implementaci či snadno aktualizovat jeho algoritmus.

Čtvrtá komponenta představuje frontu zpráv využívající protokol AMQP. Ten funguje jako prostředník mezi backendem webové aplikace a plánovačem. Je zde k tomu, aby nedošlo k selhání komunikace mezi těmito dvěma částmi v případě dočasného výpadku plánovače nebo jeho nedostupnosti při plánování. [7]

Na obrázku 1.2 je tato architektura v jednoduchosti znázorněna. Z tohoto schématu jsou také vidět závislosti mezi jednotlivými komponentami.



■ **Obrázek 1.2** Architektura systému. Vytvořeno v programu Enterprise Architect [15]

Rozvoj aplikace s představenou architekturou systému se soustředí výhradně na frontend, backend webové aplikace a jeho databázi. Proto se následující částí textu budou věnovat převážně těmto zmíněným částem systému.

## 1.2.2 Technologie

Nejprve je však nutno se podívat na technologie, na nichž je vůbec současné řešení postaveno. Pro přehlednost budou technologie rozděleny do samostatných pododdílů. U každé z nich bude zmíněno, kde se daná technologie v aplikaci využívá a případně bude vysvětleno proč byla v minulosti zvolena právě ona. Na závěr proběhne krátké zhodnocení zvolených technologií.

Nebudou zde vyjmenovány všechny technologie, které se v aplikaci používají, ale bude se jednat o ty skalní a relevantní k této práci. Kompletní seznam se nachází v souborech pojmenovaných jako `package.json`, ve kterých jsou vypsány závislosti jednotlivých komponent projektu (viz oddíl 1.2.1). Tyto soubory se nacházejí v repozitáři projektu [16]. Dále pak je potřeba podotknout, že se v tomto oddílu nebudou vyskytovat použité technologie pro testování. O nich se můžete dozvědět více v kapitole o testování (viz kapitola 4).

### PostgreSQL

Obecně bylo pro databáze zvoleno řešení relačních databází. To bylo vybráno na základě pevně dané struktury ukládaných dat. Relační databáze jsou výhodné pro možnost definování vztahu mezi jednotlivými tabulkami, dále pak nabízejí možnost využití transakcí pro zajištění konzistenci dat. [7]

Konkrétním zvoleným řešením se stal systém PostgreSQL. PostgreSQL je výkonný, open source objektově–relační databázový systém s více než 35 lety aktivního vývoje. Získal si silnou reputaci spolehlivosti, robustnosti a výkonu. Díky tomu a také díky spolehlivé architektuře, integritě dat a otevřené komunitě se PostgreSQL stal volbou mnoha lidí a organizací včetně této aplikace. [17]

### Prisma

Pro přístup k databázi byl zvolen nástroj Prisma. Prisma [18] je ORM pro Node.js a TypeScript, který poskytuje výhody typové bezpečnosti s nulovými náklady díky automatickému generování typů z databázového schématu. Schéma Prisma používá modelovací jazyk Prisma k definování schématu databáze a k vygenerování odpovídajících typů TypeScript. Díky němu je modelování dat snadné a intuitivní, a to zejména pokud jde o modelování vztahů.

### TypeScript

Hlavním jazykem aplikace je TypeScript. V projektu je využit na drtivých 97,2% [14]. TypeScript je staticky typovaný jazyk, který vychází z jazyka JavaScript. Nabízí všechny funkce jazyka JavaScript a navíc umožňuje psát a ověřovat kód, což šetří čas. Díky němu jsou chyby odchyčeny ještě před samotným spuštěním kódu.

### Next.js

Next.js je React framework pro vytváření dynamických full–stack webových aplikací. React je JavaScriptová knihovna pro tvorbu interaktivního uživatelského rozhraní. Next.js poskytuje stavební bloky pro vytváření webových aplikací. Pod pojmem framework se rozumí, že Next.js zpracovává nástroje a konfiguraci potřebnou pro React a poskytuje další strukturu, funkce a optimalizace pro vytvářenou aplikaci. Next.js přichází s řešeními běžných požadavků na aplikaci, jako je směrování, načítání dat a ukládání do mezipaměti, a to vše při současném zlepšení počítka vývojářů i koncových uživatelů. [19]

Když byly vybírány technologie pro tento projekt, hlavními požadavky bylo, aby webová aplikace byla interaktivní (tzn., že nebylo možné všude generovat statické webové stránky). Dále hrála roli i oblíbenost mezi vývojáři. Ve výsledku byl tedy zvolen právě framework Next.js konkrétně ve verzi 13. [7]

## Zod

Zod je knihovna pro deklaraci a validaci schémat v jazyce TypeScript. Termín schéma zaštiťuje datového typu – od jednoduchého řetězce až po složitý vnořený objekt. Z těchto definic umožňuje vytvořit validátor, který ověří, zda vstupní data odpovídají definovaným typům, kterým automaticky odvodí statický typ jazyka TypeScript. Mezi jeho další výhody patří například nulová závislost na dalších knihovnách, stručné, přehledné řetězitelné rozhraní a parsování. [20]

V aplikaci se Zod využívá pro validaci dat jak na straně klienta tak i serveru. Pro vytvoření Zod validačních schémat pro tabulky v databázi byl využit automatický generátor pro Prisma, který při změně datových typů v databázi generuje nová validační schémata automatizovaně. [7]

## Bootstrap

Bootstrap [21] je framework, který nabízí kaskádové styly pro komponenty. Díky jeho použití je webová aplikace responzivní – je zajištěno správné zobrazení a funkcionality na mobilních zařízeních. V aplikaci je využito zkompileovaných a minimalizované CSS a JS (`bootstrap.min.*`).

## Docker

V rámci předešlého oddílu byla popsána architektura systému (viz oddíl 1.2.1), jednotlivé její stavební bloky se spouští prostřednictvím Dockeru. Docker je nástroj pro vývoj, testování, odesílání a spouštění aplikací. [22]

## Eslint

Eslint provádí statickou analýzu kódu, pomáhá najít a opravit problémy v kódu. Díky toho se zvyšuje kvalita kódu, jeho čitelnost a zrychlení hledání potenciálních chyb ještě před spuštěním programu [23]. V rámci projektu je součástí automatických procesů při práci s gitem.

## Prettier

Prettier [24] je nástroj pomáhající udržení přehledného a dobře čitelného kódu. Má za cíl udržet kód v jednotném stylu, čehož dosahuje smazáním původního stylu a vypsáním daného kódu znovu v jednotném stylu. Prettier je využíván v kombinaci s dříve zmíněným Eslintem.

## Swagger UI

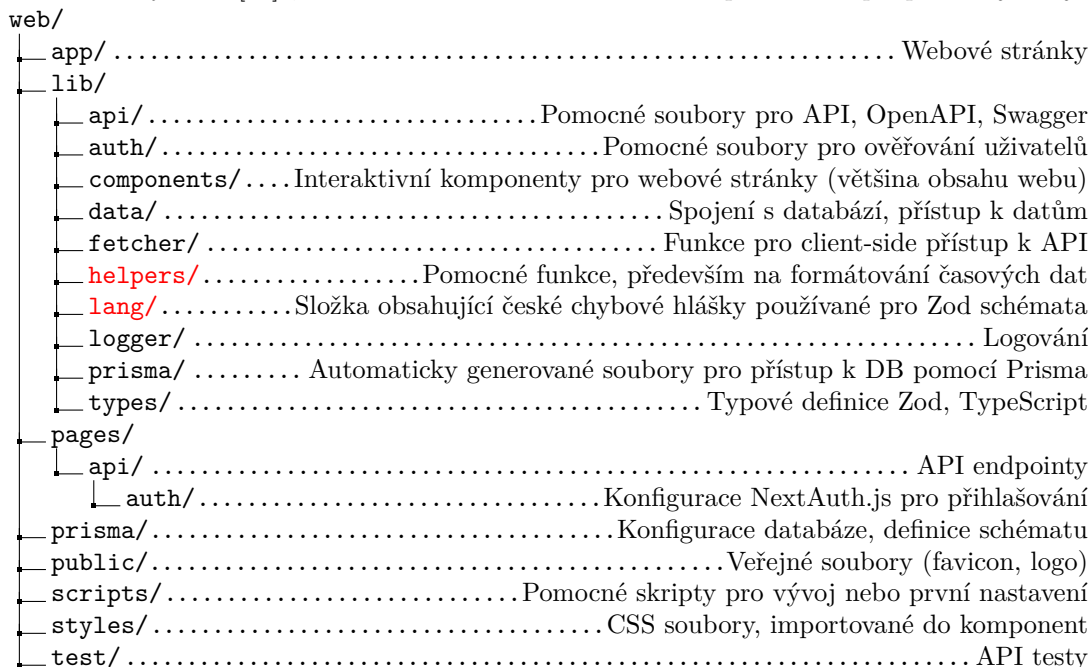
Swagger UI [25] je nástroj, který v rámci svého uživatelského rozhraní zobrazuje dokumentaci aplikace. Konkrétně vizualizuje API a umožňuje s ním interakci. Je automaticky generován z vytvořené specifikace OpenAPI v aplikaci.

## Závěr

Uvedené technologie byly zvoleny podle požadavků na v té době vytvářenou verzi aplikace. Pro rozvoj aplikace byly technologie vyhodnoceny jako stále dostačující. Proto nebylo nutné hledat jejich alternativu. Přípravované změny budou proto postaveny na představených technologiích. Může se však přesto stát, že pro některé změny nebudou tyto technologie dostačující a bude nutné je doplnit o další. Nad výběrem takové technologie proběhne diskuze v rámci návrhu (viz podkapitola 2.1).

## 1.2.3 Struktura aplikace

Struktura aplikace byla dobře vymyšlena, v době přebírání projektu. Pro přiblížení takto vypadá struktura složky `web/` [26]<sup>1</sup>, do které se bude v rámci tvořené implementace přispívat nejčastěji:



### Routing

Použitý framework Next.js ve verzi 13, využívá routing podle adresářů. Každá stránka má vlastní URL a routování mezi nimi je zajištěno automaticky. Stránky zodpovědné za routing se nachází v adresáři `app/`. Složky s názvem v kulatých závorkách (např. `(user)`) se v URL neprojeví a slouží pouze pro organizaci souborů. Složky s názvem v hranatých závorkách, jako je například `[id]`, slouží pro předání dynamických parametrů. [26] V rámci tohoto adresáře se nachází čtyři typy souborů:

**page.tsx:** Zprostředkovává unikátní URL cestu uživateli.

**layout.tsx:** Udává rozložení stránky, nebo i více stránek. Rozložení mohou být také vnořené. V rámci tohoto souboru se přijímá atribut `children`, který v rámci svého obsahu obklopuje.

**loading.tsx:** Jedná se o soubor, který zobrazuje prozatímní obsah stránky, než se načtou získávaná data a načte se skutečný obsah stránky.

**error.tsx:** Jedná se o klientskou komponentu, která zobrazuje stránku oznamující chybu. Reálně obaluje všechny její vnořené stránky a slouží jako záložní řešení, pokud se něco při zobrazení skutečné stránky pokazí.

Uživatelská navigace v rámci aplikace je z velké části řešena pomocí komponenty `Link` od Next.js, která zajišťuje snadnou navigaci bez nutnosti obnovení celé stránky, což přináší lepší a rychlejší požitek z používání aplikace.

<sup>1</sup>Označené složky červeně, `helpers` a `lang`, nebyly zmíněné v rámci použité citace. Složka `helpers` však v projektu existovala a složka `lang` bude do projektu přidána s novou implementací.

Složka `app/` využívá převážně server-side rendering (SSR), což znamená, že je příslušná stránka generována při každém požadavku. V rámci souborů `page.tsx` se získávají data z databáze bez nutnosti volání API. Tyto data jsou dále v atributech (props) předány příslušným klientským komponentám, které je využívají jako hlavní zdroj informací nebo jako záložní. Interaktivní stránky, které je potřeba vyrenderovat na klientovi, se nacházejí v adresáři `lib/components/` a jsou do serverových stránek importovány. [26]

## Komponenty

Tento adresář je dále členěný podle celků, kterým se dané podadresáře věnují. Obecně se komponenty dají pomyslně rozdělit podle Container/Presentational Patternu. Kontejnerové (Container) komponenty se starají o to, jaká data se uživateli zobrazují. Hlavní náplní prezentačních (Presentational) komponent je to, jak se data zobrazují uživateli. Takže například ve složce `lib/components/workers/` se nachází veškeré komponenty k pracantům, v ní je kontejnerová komponenta `WorkersClientPage.tsx` zobrazující hlavní stránku se všemi pracanty, které si získává z API. Tyto data poté zobrazuje přes prezentační komponentu `WorkersTable.tsx`.

V rámci komponent se stavy jejich proměnných řeší vždy jen v rámci dané komponenty (Component-Level State). Nejčastěji se stav zaznamenává pomocí hooks `useState` pro jednoduché proměnné a `useForm` pro pole formulářů.

## API

API se nachází v adresáři `pages/api`. Všechny API endpointy jsou vytvořeny pomocí Next.js API routes a jsou přístupné pouze pro přihlášené uživatele. Tyto cesty jsou použity pro získávání (resp. posílání) dat z backendu (resp. na backend). [26] V rámci nové verze Next.js je nyní možné mít API endpointy přímo ve složce `app/`, přesto je i současně použitá varianta dále podporována. Proto se API endpointy nebudou v rámci rozvoje aplikace přesouvat.

Pro komunikaci s API byl zvolen REST [27], což je architektonický styl používající se v rámci webových služeb pro výměnu dat mezi klientem a serverem. REST se vyznačuje těmito principy:

**Jednotné rozhraní:** REST definuje konzistentní a jednotné rozhraní pro interakce mezi klientem a serverem.

**Klient-server:** Klient a server jsou odděleni a mohou se tak vyvíjet nezávisle.

**Bezstavovost:** Bezstavovost vyžaduje, aby každý požadavek klienta na server obsahoval veškeré informace nezbytné k pochopení a dokončení požadavku, bez nutnosti zachování stavu mezi požadavky.

Konkrétně je v aplikaci použito REST API založené na protokolu HTTP. Nevyužívá všechny jeho standardní metody, neboť se v aplikaci nepoužívá metoda PUT. Další jsou však implementované:

**GET:** Používá se pro získání dat ze serveru. Jedná se o metodu určenou pouze pro čtení, tzn. neprovádí se žádné změny dat.

**POST:** Prostřednictvím této metody jsou odesílány nová data na server, což se používá pro vytvoření nové datové položky.

**PATCH:** Metoda PATCH se používá pro aktualizaci již existující datové položky na serveru. V těle požadavku jsou obsaženy pouze konkrétní změny nebo sada instrukcí pro změnu daného prostředku, ostatní části však nechává beze změny.

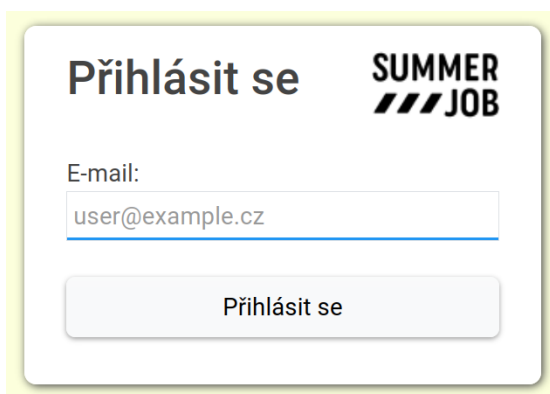
**DELETE:** Pomocí této metody lze odstranit konkrétní položku na serveru. Tato metoda nevratně odstraňuje zdroj a jeho obsah ze serveru.

## 1.2.4 Pohled na aplikaci

Aplikace již byla v rámci předchozích oddílů představena po formální stránce. Tento oddíl se však věnuje prezentaci z hlediska vizuálního provedení a funkčnosti. Budou zde přiblíženy procesy a činnost členů týmu SummerJob používající tuto aplikaci.

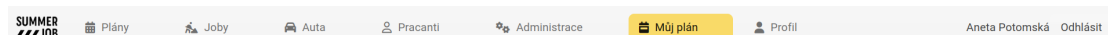
Současné řešení je responzivní webová aplikace. To tedy znamená, že uživatelské rozhraní je inkluzivní pro veškerá zařízení – od těch mobilních až po stolní počítače. Jednotlivé prvky aplikace se přizpůsobují velikosti displeje a utvářejí tak vizuálně pěkný celek.

První co uživatel uvidí po navštívení této aplikace je přihlašovací okno. Přihlašování do aplikace probíhá pouze přes e-mail. Po zadání e-mailové adresy se ověří zda se uživatel může přihlásit (tzn. jestli jeho k této e-mailové adrese evidovaný účet a tento účet není zablokovaný) a odešle se e-mail s odkazem pro přihlášení.



■ Obrázek 1.3 Přihlašování

Po přihlášení je možné si poprvé prohlédnout horní lištu. Zde se vyskytují jednotlivé dlaždice s odkazy na důležité stránky webové aplikace. První stránku, na kterou je uživatel bezprostředně přeměrován je „Můj plán“. Následující pododdíly se právě věnují jednotlivým stránkám z horní lišty.



■ Obrázek 1.4 Vrchní panel s přeměrováním do „Můj plán“

Ale ještě před tím je potřeba přiblížit něco o rolích. Každému uživateli mohou být přiřazeny role (o tomto procesu později – viz pododdíl 1.2.4), které je opravňují ke správě jistých částí aplikace. Jedná se o tyto role a umožňují správu těchto částí:

role	plány	joby	auta	pracanti	administrace
PLANS	✓				
JOBS		✓			
CARS			✓		
WORKERS				✓	
ADMIN	✓	✓	✓	✓	✓

■ Tabulka 1.1 Role

Každý uživatel potom bez ohledu na roli má přístup do dvou částí aplikace – „Můj plán“ 1.2.4 a „Profil“ 1.2.4.



## Pracanti

Do této části mají přístup uživatelé s rolí WORKERS nebo ADMIN. Vizuálně si lze stránku prohlédnout na obrázku 1.5. Hlavním prvkem této stránky je tabulka s přehledem uživatelů. Tato tabulka se dá základě filtrovat tlačítka nahoře (ale není možné si sloupce seřadit). Na stránce se vyskytuje možnost si tabulku i vytisknout (tisk však nereaguje na použité filtry). Dále se zde zobrazují jednoduché statistiky a náhledy fotografie pracanta, na kterého je zrovna nejeto myší. Jsou zde i tlačítka pro přidání nových pracantů.

The screenshot shows the 'Pracanti' management interface. At the top, there is a navigation bar with 'Pracanti' highlighted. Below it, there are buttons for 'Hromadný import', 'Přidat pracanta', and 'Tisknout'. A search bar is present with filters for 'Pouze silní' and 'Pouze s autem'. The main part of the page is a table with the following data:

Jméno	Příjmení	Telefonní číslo	E-mail	Vlastnosti	Akce
Aida	Pivoňková	575 955 296	aida_pivokov@centrum.cz		
Alberta	Macháčková	909 842 333	alberta.machkov77@seznam.cz		
Amos	Vondra	334 917 869	amos_vondra35@gmail.com		
Aneta	Potomská	000 552 225	aneta@gmail.com		
Angelika	Libalová	369 111 965	angelika_lbalov@seznam.cz		
Anselma	Urbancová	917 041 005	anselma_urbancov@atlas.cz		
Antonie	Vysloužilová	324 960 646	antonie37@volny.cz		
Aram	Nový	750 368 397	aram47@gmail.com		
Arzen	Cisař	624 713 443	arzen.csa93@seznam.cz		
Blažena	Buchtová	554 688 799	blaena_buchtov@gmail.com		
Bohdana	Václavková	260 924 682	bohdana_vclavkov63@centrum.cz		
Božidara	Králová	480 221 308	boidara.krlov@centrum.cz		
Břetislava	Řehořová	882 861 858	betislava2@atlas.cz		
Budislav	Čížek	068 614 680	budislav11@gmail.com		

On the right side, there is a 'Statistiky' panel showing 'Pracantů 101' and a 'Foto' panel with a placeholder icon.

■ **Obrázek 1.5** Úvodní stránka pro správu pracantů

Tlačítko „Hromadný import“ umožňuje nahrání vícero pracantů najednou. Data s informacemi o pracantech se nahrávají ve formátu CSV oddělené středníkem. Hodnoty, které je potřeba vyplnit, jsou prakticky stejné jako při editaci pracanta (viz obrázek 1.6) až na hodnoty od „Další vlastnosti“ a níž.

Administrátoři zaměřeni na část s evidencí pracantů zpravidla využívají hromadný import. Je tomu tak kvůli tomu, že se přihlašování na akci dělá externě přes jinou webovou stránku [3]. Z této stránky se později hromadně přidávají všichni zájemci o aktuální ročník. Další detaily (např. poznámky) se pak provádí později přes editaci pracanta.

## Auta

K této stránce mohou přistupovat uživatelé s rolí CARS nebo ADMIN. Hlavní stránka zobrazující přehled aut je velice podobná té s pracanty. Obsahuje však jenom tabulku, základní filtrování a tlačítko pro přidání nového vozidla.

Vytváření nového vozidla a jeho editace je podobná až na některá pole. V editaci např. nelze již měnit majitele auta a při vytváření chybí „Konečný stav kilometrů“, „Částka k proplacení“ a „Proplaceno“. Tohle je dle týmu SummerJob chtěné chování. Formulář s editací vozidla si lze prohlédnout na obrázku 1.7.

Auta se evidují až po zaregistrování pracantů. Od týmu SummerJob není vyžadováno přidání možnosti pro vytváření auta rovnou s vytvářením pracanta [10].

**SUMMER JOBS**

**Aida Pivoňková**

**Jméno**  
Aida

**Příjmení**  
Pivoňková

**Telefonní číslo**  
575 955 296

**E-mail**  
aida\_pivokov@centrum.cz  
Změnou e-mailu dojde k odhlášení uživatele z aplikace.

**Může pracovat v následujících dny**

Čt 3. 7.	Pá 4. 7.	So 5. 7.	Ne 6. 7.	Po 7. 7.	Út 8. 7.
St 9. 7.					

**Chce adorat v následujících dny**

Čt 3. 7.	Pá 4. 7.	So 5. 7.	Ne 6. 7.	Po 7. 7.	Út 8. 7.
St 9. 7.					

**Alergie**

Prach Zvířata Seno Pyl Roztoči

**Alergie**

Prach Zvířata Seno Pyl Roztoči

**Další vlastnosti**

Silák ↔

**Fotografie**

No file selected.

**Auta**  
Žádná auta

**Poznámka**  
Poznámka

■ **Obrázek 1.6** Úprava pracanta

## Joby

Joby mohou spravovat uživatelé s rolí JOBS nebo ADMIN. Evidence jobů (viz obrázek 1.8) se nápadně podobá předešlým evidencím. Ze zmíněného obrázku je patrné jaké informace se přibližně pro job evidují.

Tabulka je však o něco složitější, neboť se zde objevují sbalené sekce a možnost filtrování podle sloupců. Toto filtrování se nedá kombinovat, v jeden čas se tedy dá řadit jenom podle jednoho sloupce. Navíc má pak v akcích v tabulce možnosti si job označit jako:

- **dokončený**: job byl zcela vykonaný napříč všemi dny, kdy byl naplánovaný a nebude se dále zobrazovat při plánování,
- **skrytý**: job se nebude zobrazovat při plánování.

Takto označený job se poté v tabulce přesune do jiné sekce. Tyto sekce jsou v tabulce úplně vespod a v případě mnoha jobů může být zdlouhavé se k nim dostat.

### Polestar Explorer, OZ06UMH - Blažena Buchtová

**Název**  
Polestar Explorer, OZ06UMH

**Poznámka pro organizátory**

olive

**Počet sedadel**  
5

**Počáteční stav kilometrů**  
16254

**Konečný stav kilometrů**  
17169

**Částka k proplacení**  
0

Proplaceno

Zpět
Uložit

■ Obrázek 1.7 Úprava auta

**SUMMER** Plány Joby Auta Pracanti Administrace Můj plán Profil Aneta Potomská Odhlásit

**Dostupné joby** Nový job

Vyhledat... Vyberte oblast Vyberte den

Název	Lokalita	Kontaktní osoba	Adresa	Naplánované dny	Dostupné dny	Pracantů	Akce
> Hrabání listů	Hradec Králové	Alfréd Andrie, 769 361 906	9. Května 227	1 / 3	6	3 - 4	✓ 🗑️ 📄 🗑️
> Práce: Awesome Fresh Ball	Nová Paka	Emanuela Mikulová, 388 604 941	Špotzova 587	0 / 3	3	2 - 6	✓ 🗑️ 📄 🗑️
▼ Práce: Licensed Wooden Soap	Česká Kamenice	Lumíra Sochorová, 114 947 454	U Nemocenské Pojišťovny 44	0 / 1	4	3 - 4	✓ 🗑️ 📄 🗑️

**Popis**  
Laudantium laboriosam dolorem. Eaque perferendis magni doloribus ad impedit quos. Consequuntur perspiciatis deserunt eius. Dicta accusamus quo ratione. Ipsam incidunt adipisci facilis perspiciatis.

**Poznámka pro organizátory**  
Nostrum id enim harum illum est repudiandae quidem recusandae ipsa. Officis incidunt quisquam fuga aut temporibus voluptate vero hic. Quos a delectus. Necessitatibus quibusdam laboriosam ratione labore unde repudiandae natus culpa quaerat.

**Počet pracantů:** 3 - 4 (1 siláků)

**Doprava do oblasti požadována:** Ano

**Alergeny:** HAY

**Dostupné:** Čt 3. 7., Ne 6. 7., Po 7. 7., Út 8. 7.

**Naplánované dny:** 0 / 1

> Práce: Luxurious Bronze Bacon	Nová Paka	Rodan Jiránek, 011 881 327	Měsíční 3	1 / 3	1	2 - 4	✓ 🗑️ 📄 🗑️
> Práce: Modern Cotton Chair	Cheb	Ing. Cyntie Šípová Phd., 037 192 581	Radhošťská 54	0 / 1	1	2 - 4	✓ 🗑️ 📄 🗑️
> Práce: Tasty Fresh Towels	Cheb	Slavibor Rada, 263 419 532	Nevanova 4	0 / 2	4	3 - 5	✓ 🗑️ 📄 🗑️
> Práce: Unbranded Steel Bike	Sázava	Alina Slaninová, 365 518 714	Úvozová 5	0 / 1	5	2 - 5	✓ 🗑️ 📄 🗑️

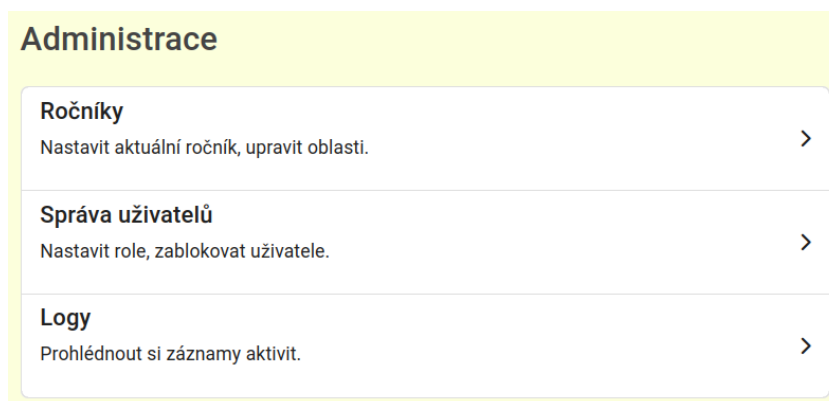
> **Dokončené (0)** - Joby označené jako dokončené se nebudou zobrazovat při plánování

> **Skrýté (0)** - Joby označené jako skryté se nebudou zobrazovat při plánování

■ Obrázek 1.8 Úvodní stránka pro správu jobů

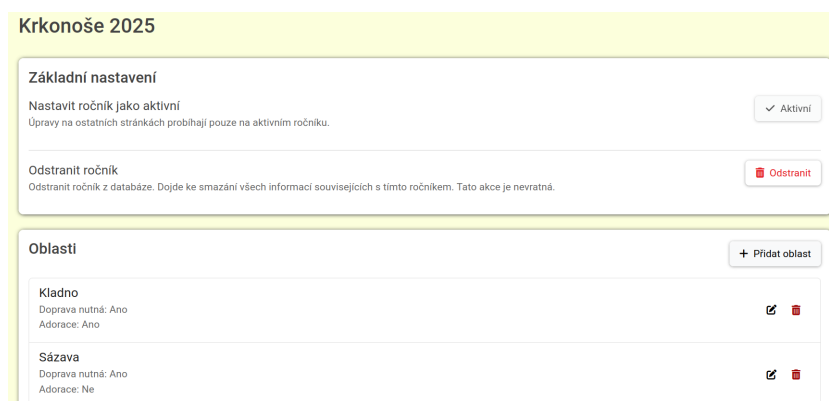
## Administrace

Tato webová stránka je dostupná pouze uživatelům s rolí ADMIN. Hlavní stránka obsahuje pouze navigaci s odkazy na další stránky. Na obrázku 1.9 je výmluvně popsáno, co daná stránka obsahuje:



■ **Obrázek 1.9** Úvodní stránka pro administraci

- **Ročníky:** Zobrazuje seznam všech proběhlých ročníků, je zde možnost si vytvořit nový ročník. Kliknutím na konkrétní ročník v seznamu otevře stránku s editací informací o daném ročníku, kde si lze daný ročník nastavit jako aktivní<sup>2</sup>. V neposlední řadě je zde možné ročník celý smazat a s tím smazat i příslušné informace k němu spojené, to znamená veškerou evidenci o pracantech, autech, jobech, plánech a podobně. Dále se pak zde zobrazuje seznam oblastí pro daný ročník a jejich správa. Popsané informace jsou viditelné na obrázku 1.10.



■ **Obrázek 1.10** Detaily konkrétního ročníku

- **Správa uživatelů:** V této správě se dají nastavit role uživatelům. Dále pak lze zablokovat uživatelům jejich účet, to jim zamezí se přihlásit znovu do účtu. Jejich účet se ale tím nesmaže a s uživatelem lze dále manipulovat – například v plánech.
- **Logy:** V rámci této stránky se zde vypisují všechny změny, co byly v aplikaci provedeny. Mezi ně patří vytvoření, editace, smazání libovolné entity, např. pracanta. Logování se provádí na backendu po úspěšně provedeném požadavku.

<sup>2</sup>V jeden čas může být aktivní pouze jeden ročník

## Plány

Plány jsou k dispozici uživatelům s rolí PLANS nebo ADMIN. Plány představují asi ten nejhlavnější důvod k existenci tohoto systému. Úvodní stránka obsahuje seznam všech plánů na jednotlivé dny akce a možnost přidání si nového plánu.

Tabulka plánu je podobně „složitá“ jako v případě evidence jobů. Obsahuje znovu sbalené sekce, kdy jejich obsah je patrný z obrázku 1.11.

Pracant	Kontakt	Vlastnosti	Alergie	Doprava	Akce
Aneta Potomská (?)	000 552 225			1) model (2/4)	✕
Alberta Macháčková (?)	909 842 333			1) model (2/4)	✕

Pracant	Kontakt	Vlastnosti	Alergie	Akce
Aida Pivořiková (?)	575 955 296	Adoruje		✕
Amos Vondra (?)	334 917 869	Auto, Silák	DUST	✕
Angelika Libalová (?)	369 111 965		DUST, ANIMALS	✕
Anselma Urbancová (?)	917 041 005	Silák		✕
Aram Nový (?)	750 368 397	Auto		✕

■ **Obrázek 1.11** Detail jobu v plánu

Po njetí myši na libovolného pracanta se zobrazuje náhled jejich fotografie ve stejném duchu jako u pracantů. Joby, a to jak ty v evidenci tak v plánu, neobsahují v současné verzi aplikace fotografie. Uživatel zodpovídající za plány může do plánu přiřadit pouze ty joby, které mají nastavenou dostupnost na daný den plánu. Na začátku se pracanti, se zvolenou časovou dostupností v den plánu, přiřazují do plánu automaticky a to do sekce „Bez práce“.

Nabídka nahoře stránky obsahuje speciálně tlačítka pro vygenerování plánu a od/zveřejnění plánu. Vygenerování plánu pošle na backend žádost, která je pak prostřednictvím AMQP poslána na plánovač. Plánovač vygeneruje plán s poskytnutými daty z databáze a odešle zpět výsledek (znovu přes AMQP). Zveřejnění plánu má za důsledek to, že se pracantům přiřazeným na práci v tomto plánu bude tato skutečnost zobrazovat v „Můj plán“.

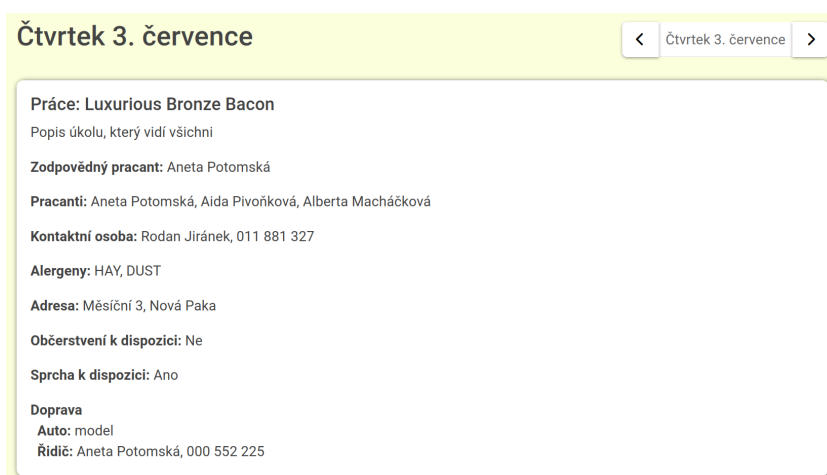
Proces samotného plánování představuje kombinatorický problém, který je v současnosti řešen pomocí naivního algoritmu. Plánovač se snaží vygenerovat optimální plán. Ten například zahrnuje:

- Aby byl uspokojený počet minimálních, maximálních a silných pracantů.
- Práce vyžadující dopravu měla svého řidiče, nebo aby pracanti z takové práce měli zařízení odvoz autem jedoucí na jinou práci, který je vysadí po cestě.
- Na práci s jistými alergeny nebyl zařazen pracant, který je na ně právě alergický.
- Aby pracant s požadavkem na možnost adorování ten den, byl přiřazen do oblasti, kde je toto umožněno.

Ne vždy plánovač je schopen zajistit splnění všech požadavků vypsanych výše. Proto jsou často nutné manuální úpravy od organizátorského týmu SummerJob. Jedna z takových úprav je například přeřazení pracanta na jinou práci. Pracanti se dají přetahovat jednotlivě buďto myší a nebo po kliknutí na symbol dvou šipek, který se zobrazuje u každého pracanta. Po kliknutí se zobrazí modální okno s výběrem, na kterou práci v plánu má být pracant přeřazen.

## Můj plán

Jak již bylo zmíněno, tato stránka je dostupná všem uživatelům. Je jim zde zobrazen jejich plán, job na který se mají daný den dostavit, a jednoduchý kalendář s výběrem data. Ukázkou takového „mého plánu“ si lze prohlédnout na obrázku 1.12.



■ Obrázek 1.12 Můj plán

## Profil

Poslední stránkou zbývající na popsání je profil. Svůj profil si může aktualizovat každý uživatel. Oproti správci pracantů si uživatel nemůže upravit cokoli od „Další vlastnosti“ (viz. obrázek 1.6).

### 1.3 Sběr požadavků

Cílem této podkapitoly je popsat sběr požadavků a tyto požadavky dále definovat. Jednotlivé oddíly se věnují vždy jednomu zdroji, ze kterého požadavky pocházejí.

#### 1.3.1 Nevýhody současného řešení

Možná již z předcházejícího textu o současném řešení mohly být patrné jisté problémy. Zde se však bude vyskytovat finální a celkový přehled, co třeba na aktuálním řešení není ideální.

## Fotografie

První problematiku představují fotografie. Fotografie je jedna z částí aplikace jež byly přidány až po práci Ing. Ješiny. Fotografie byly implementované pouze pro formuláře s editací pracanta. Bylo by proto vhodné, toto rozšířit i pro formuláře s vytvářením pracanta. Fotografie by byly k užítku i v jiných místech aplikace a to například u jobů, kde by bylo možno nahrávat fotografií i více než jednu.

Dalším problémem je samotné zobrazení fotografie a nedostatek funkcí pro interakci. Nahraná fotografie se zobrazuje s fixním poměrem stran, tzn. nedodrhuje skutečný poměr stran fotografie. Dále pak již jednou nahraná fotografie nelze smazat, což je pro takovou evidenci velmi problematické. Ve finále si nelze fotografii rozkliknout a zobrazit si je tak ve větším formátu.

Po odeslání formuláře se fotografie posílá v požadavku na server zvlášť a ne se společně s ostatními daty z formuláře, což porušuje dodržení standardů na konzistenci dat. V rámci development módu spuštění aplikace se fotografie nenahrávají vůbec. Když se fotografie povede nahrát, tak se ukládají přímo do obecné složky `/web-storage` v projektu. Bylo by vhodné zvolit lepší strukturu, aby bylo případně umožněno odebrat fotografie k neaktuálním ročníkům a uvolnit tak místo na disku. Kromě toho se do databáze k jednotlivým pracantům ukládají cesty k jejich fotografiím v podobě absolutní cesty, což zabraňuje přenositelnosti aplikace.

## Formuláře

Formuláře jsou v aplikaci využity prakticky na každém místě aplikace, přesto je kód k nim velice nepřehledný a porušuje DRY princip. Z toho důvodu bude v rámci změn důležité kód refaktorovat pro udržitelnější a snadněji spravovanou podobu. Některá pole ve formulářích obsahují vizuální chyby. Mezi ně patří obecně nedodržování stejného rozestupu (margin) a nejednotné barvy.

Mimo to, by některá pole zasloužila přepracování úplně od začátku – například výběr dostupnosti. Ten se zobrazuje v aplikaci při vytváření/upravování pracanta nebo jobu. Vizuálně by bylo vhodné tento výběr vylepšit, konkrétně například nastává problém se zobrazováním na menších zařízeních, kde se jednotlivé dny zobrazují v nevzhledných kachlích (viz obrázek 1.6).

Co se týče interaktivních prvků, tak se vespod formulářů vyskytují tlačítka na vrácení se zpět a uložení, ty jsou kvůli své poloze špatně přístupné a vyžadují scrollování až na konec stránky. Další prvek zpomalující workflow uživatele jsou blokující modální okna, jež se zobrazují po uložení formuláře, kdy takové okno zobrazuje pouze informace o ne/úspěchu odeslání.

Posledním zde zmíněným problémem je malá míra validace některých polí, přičemž stávající validace nedisponuje v mnoha místech českým popisem chyb. To může být pro mnoho uživatelů problematické, kromě toho se jedná o nekonzistenci. Dále ve formulářích chybí vyznačení, která pole jsou povinná, což zpomaluje uživatele, kteří nejsou ještě tolik s daným typem formuláře seznámeni. Obecně je pak validace a kontrola formulářů nedostačující. V implementaci se na mnoha místech validace nespolehá čistě na Zod, což vede k nekonzistentnímu stylu hlášení chyb – některé chyby jsou pak vypisovány přímo prohlížečem.

## Tabulky

Z oddílu 1.2.4 je patrné, že aplikace při svých evidencích především spoléhá na tabulky. I z toho důvodu by bylo vhodné se zaměřit na jejich vylepšení. Například filtrování podle sloupců není k dispozici u všech sloupců v aplikaci, které by to svou povahou (např. text podle abecedy apod.) umožňovaly. Dále se pak tabulky špatně zobrazují a to i na velkých zařízeních, kdy je nutno horizontálně scrollovat. Na závěr mohou být tabulky obsahující sekce (u jobů a plánů) nepřehledné, neboť jsou tyto sekce často až vespod stránky.

## Přihlašování

Jak již jednou bylo zmíněno přihlašování je řešeno prostřednictvím e-mailových adres. Řešení je implementováno pomocí knihovny NextAuth.js. „Zvolená knihovna podporuje přihlašování pomocí e-mailu, služeb Google, Facebook, GitHub, Twitter, Apple a dalších. Přihlašování pomocí uživatelského jména a hesla je autory knihovny považováno za nebezpečné a pro podporu je nutné implementovat vlastní ověření uživatelů. Na základě analýzy bylo proto zvoleno přihlašování pomocí e-mailu.“ [7].

Do budoucna by však bylo vhodné přihlašování rozšířit právě například o možnost se přihlašovat pomocí uživatelského jména a hesla s vlastním ověřením uživatelů, případně pak přes služby jako je Google a podobně.

### 1.3.2 Možnosti rozšíření ze současného řešení

V rámci diplomové práce [7] byla zmíněna rozšíření tehdejšího výstupního stavu aplikace, která jsou platná i v současnosti:

**Evidence typů prací:** „Typy prací by mohly být využity pro lepší přehlednost a organizaci práce. Plánovací systém by následně mohl využívat typy prací tak, aby nepřirazoval brigádníky opakovaně na stejné práce. Tato funkcionality by mohla být využita i pro vyhledávání brigádníků, kteří mají s danou prací zkušenosti. Pracovníci by také mohli v profilu vyjádřit preference k určitým typům prací.“

**Evidence vybavení a nástrojů:** „Evidence by zaznamenávala počty potřebného vybavení na práci a docházelo by k párování prací s pracantů, kteří mají dané nástroje k dispozici.“

**Systému pro zasílání upozornění:** „Tento systém by mohl být využit pro zasílání upozornění na nové práce, změny v plánu, změny v profilech pracantů, spuštění přihlašování a další. Hromadné zprávy by také mohly být využity administrátory pro zasílání zpráv všem účastníkům akce.“

**Elektronická nástěnka:** „Vhodným dodatkem by dále mohla být elektronická nástěnka, kam by organizátoři mohli přidávat obecné informace pro účastníky akce. Tato nástěnka by byla dostupná všem účastníkům po přihlášení do systému.“

### 1.3.3 Dotazník

Pro lepší pochopení toho, co si organizátoři akce SummerJob slibují od následujícího rozšíření, byl pro ně vytvořen dotazník. Ten mohli vyplňovat po celou dobu psaní této práce, tzn. od října 2023 do dubna 2024. Konkrétně byl dotazník rozeslán organizátorům, kteří se účastnili loňského ročníku SummerJob.

Dotazník slouží k podchycení dalších problematik nebo návrhů, o kterých se doposud ani neuvažovalo. V případě duplicitních odpovědí mohou tyto odpovědi dát jisté oblasti větší váhu a zdůraznit tak, kterým problémům by se spíše měla věnovat pozornost.

## Otázky

Možná by se mohlo předpokládat, že budou administrátoři využívat repozitář projektu [16] a to jak pro nahlašování chyb, tak i pro vytváření požadavků na vylepšení systému. Vzhledem, ale k současnému stavu repozitáře tedy, že přispívají maximálně 3 uživatelé [16], a celkovému počtu administrátorů, kterých je každoročně kolem 30 [2], nepředpokládá se, že jich mnoho tuto možnost využívá.



Proto byly pro ně vymyšleny následující dotazy:

1. Kolika ročníků SummerJob jste se zúčastnil/a (respektive spravoval/a)?
2. Jste si vědom/a nějakých chyb/bugů v systému? Případně uveďte.
3. Myslíte si, že existují oblasti, kde by uživatelské rozhraní mohlo být intuitivnější? Pokud ano, uveďte konkrétně.
4. Můžete uvést příklady situací (pokud takové existují), ve kterých bylo používání současného systému náročné nebo/a neefektivní?
5. Existují nějaké konkrétní nedostatky či obtíže, se kterými se setkáváte při používání současného systému? Případně napište jakýkoliv návrh na zlepšení dané problematiky.
6. Musíte používat nějaké externí aplikace/zdroje na řešení jistých problematik, které byste preferoval/a, aby se daly řešit přímo v systému?
7. Schází Vám v systému něco? Pokud ano, uveďte konkrétní návrhy na vylepšení nebo na nové funkce, které byste rád/a viděl/a implementované.
8. Máte nějaké další poznámky? Pokud ano, podělte se. (nepovinná)

## Vyhodnocení dotazníku

Na dotazník ve výsledku odpovědělo 10 lidí z týmu SummerJob. Což je vzhledem k proměnlivému počtu administrátorů, kterých je ale průměrně 30 na ročník, třetina. Toto číslo je i přesto dostačující vzhledem k velikosti projektu.

Dotazník byl relevantní pro všechny, co mají alespoň rok zkušeností, což byli všichni respondenti (viz příloha A). Nejvhodnější by však bývalo, kdyby všichni respondenti měli nad dva roky zkušeností a to zejména kvůli tomu, aby znali bývalý systém administrace před zavedením stávající aplikace. To ale neznamená, že by lidé s jedním rokem zkušeností neměli určené dotazy, na které by nemohli odpovědět. Z výsledků ale vyplynulo, že lidé s jedním rokem zkušeností na zbývající dotazy odpověděli jednoslovně záporně a nepřinesli žádné nové podněty.

Odpovědi na otázky jsou zpracovány do přehledných tabulek, které se nacházejí v příloze A. Bodově jsou tohle všechny návrhy na vylepšení systému sesbírané od respondentů:

- Přidat více funkcí do budoucna (z A.1).
- Nefungují kilometry při evidenci (z A.1).
- Občas nefunguje přihlášení do účtu na první pokus (z A.1, A.4).
- Přihlašování ideálně neřešit jako odkaz v emailu (z A.3).
- Při plánování jobů chybí upozornění na to, že dva lidé již spolu předtím pracovali a že člověk už jednou byl na dané práci (z A.1, A.4).
- Změnit pozadí stránek (z A.2).
- Design by mohl být více intuitivní (z A.2).
- Zajistit větší důslednost ve vyplňování informací (z A.3).
- Přidat notifikace například na nabádání účastníků k vyplnění svých časových možností (z A.3).
- Více propojit s databází (z A.3, A.5).
- Přidat offline režim aplikace s automatickou synchronizací po připojení k síti (z A.3).

- Přidat možnost vytváření si vlastních tabulek a provádět exporty (z A.4, A.5, A.6).
- Zmodernizovat grafiku (z A.4).
- Přidat konkrétní časy na adorování (z A.4, A.5).
- Přidat nástěnku, kde se dají zveřejňovat aktuální informace a časový harmonogram (z A.6).
- Přidat důležité kontakty, jako je například zdravotník (z A.6).
- Přidat tlačítko na prokliknutí adresy do mapy (z A.6).

### 1.3.4 GitHub issues

Další vlna požadavků byla hned od začátku převzetí projektu zřetelná z jeho GitHub Issues. Dohromady zde existovalo 19 nahlášených issues čekajících na vyřešení. [28]

Jmenovitě se jednalo o následující zaznamenané issues (jsou řazeny od data přidání, od nejstaršího k nejnovějšímu):

- **#14 – Problém s dropdown menu:** „Prosím o vyřešení problému s dropdown menu na prohlížečích na Iphone. Klidně to můžete nechat jako jednoduchý defaultní drop-down, který má prohlížeč, pokud by to byl problém. Plus prosím přidat výběr obce do editace práce, kde v tuto chvíli chybí.“
- **#15 – Přidat prioritu jobu:** „Prosím o přidání priority jobu. Pouze 1/0 - je či není prioritní. Pouze pro interní čely JobTýmu, není potřeba reflektovat v job planner algoritmu.“
- **#16 – Přidat fotku k práci - nízká PRIO:** „Asi by občas bylo dobré připojit k jobu i nějakou fotku, kterou místňáci pošlou. Ale jde o detail, není potřeba dávat prioritu.“
- **#17 – Rozšířit textbox s poznámkou k jobu:** „Prosím rozšířit výšku textboxu tak, aby se vždy vešly 4 řádky poznámek. Někdy se tam píše poznámek hodně a pak se to schová.“
- **#18 – Chování tabulek na desktopu:** „Bylo by super, aby na desktopu nikdy nebylo potřeba horizontálně scrollovat. To bych nechal jen na mobil/tablet. Na desktopu bych raději zalamoval / schovával část textu tak, aby šířka tabulky nepřesahovala screen width.“
- **#21 – Přidat nářadí!:** „Přidat k jobu potřebné nářadí, které je třeba vzít. Ideálně přidat k plánu statistiky potřebných nářadí. Přidat k jobu dva seznamy: Jeden nářadí na místě, s čím má alespoň někdo z jobu umět, druhý seznam, co vzít s sebou. U pracantů pak přidat seznam kompetencí.“
- **#41 – Přidat pořadové číslo u práce v mém plánu.**
- **#42 – Přidat zaškrtnutí - hotovo:** „V plánu přidejte zaškrtnutí k označení jobu jako hotového. Možná ať se při tom otevře modální okno k úpravě jobu - nebo poznámky.“
- **#43 – Použít popis navrženého jobu jako popis aktivního jobu:** „Zbavte se těchto polí a použijte navrženou práci při editaci aktivního jobu Přidejte také upravování jména.“
- **#45 – Zobrazte ikonu adorace v plánu v sekci bez práce.**
- **#52 – Vytvoření trvalejšího filtru:** „Když se vrací z editace - zajistěte, aby filtr zůstal. Uložit do query parametrů asi nejlepší varianta.“
- **#53 – Předělat připínání, aby bylo jedinečné pro každého uživatele:** „Ne globální – stejné pro všechny.“

- #54 – Vyhledávat v plánu podle kontaktní osoby.
- #57 – Přidat tlačítko „zavolat job týmu“: „Po 18:00 deaktivujte tlačítka dostupnosti na další den a při pokusu o jejich stisknutí otevřít modální okno se zprávou – Prosím, zavolejte job týmu.“
- #58 – V mém plánu zobrazte nejbližší plán: „Teď se zobrazuje první den ročníku. Ukázat další den ročníku“.
- #64 – Při přidání jobu do plánu zobrazit varování: „Když má být job naplánován na další dva dny a zbývají další dva dny jeho dostupnosti, zobrazte tuto skutečnost červeně nebo nějak podobně“.
- #148 – Mapa / souřadnice k jobu: „Přidat možnost zadat místo k jobu pomocí souřadnic/ vybrání na mapě, rozkliknutí mapy u jobu na denním plánu a možnost otevřít navigaci (smart link pro mobily) na denním plánu“.
- #149 – Najeté km se neukládají: „Změna najetých km se u aut neukládá do databáze, pouze do logu.“
- #150 – Navázání aut a dalších věcí k ročníku.

## 1.4 Analýza požadavků

V předcházející podkapitole 1.3 bylo sesbíraných spoustu požadavků na opravu a vylepšení aplikace. Do práce byly zařazeny požadavky dle priority, která vzešla z konzultací s organizátory akce. Za požadavky s největší prioritou byly označeny požadavky vyplývající z GitHub Issues (viz oddíl 1.3.4). Jsou přednostní především díky svého obsahu a doby evidování. Jistá váha byla přisouzena i některým požadavkům, které vzešly z dotazníku (viz oddíl 1.3.3) a z nalezených problémů ze současného řešení (viz oddíl 1.3.1).

Některé požadavky jsou menších rozměrů (např. nefunguje evidence najetých kilometrů, změnit pozadí stránek) a z toho důvodu jsou sloučeny do obecnějších funkčních a nefunkčních požadavků. Jiné požadavky jsou pak těžko ověřitelné, správný požadavek by totiž měl splňovat tyto vlastnosti: ověřitelnost, jednoznačnost a splnitelnost, a proto byly z dalších úvah vynechány. Jedná se o například o požadavek na větší propojenost s databází a nebo o požadavky představující obecné cíle (např. přidat více funkcí do budoucna). Ostatní požadavky, jimiž se tato práce nebude zabývat, budou doporučeny na budoucí vývoj v kapitole o možnostech rozšíření (viz kapitola 5).

### 1.4.1 Funkční požadavky

- **F1 – Nástěnka:** Aplikace bude obsahovat nástěnku, kde se budou moct zveřejňovat příspěvky s aktuálními informacemi.
  - **F1.1 – Příspěvek:** Aplikace bude umožňovat evidenci příspěvků. Příspěvek má název a zkrácený popis, dále pak může obsahovat delší popis, informace o místě konání, času konání, fotografii a typ (např. kulturní apod.). Místo konání může být doplněno mapou a souřadnicemi. Čas konání zahrnuje ve kterých dnech a v jakém času se odkdy dokdy koná.
  - **F1.2 – Časová osa příspěvků:** Příspěvky se rozdělují na časové a obecné tzn. ty které mají vyplněný čas a které nikoliv. Ty časové pak tvoří harmonogram.
  - **F1.3 – Připínání:** Uživatelé zodpovídající za nástěnku mohou příspěvky připínat. Připnutý je globální a vidí ho tak všichni uživatelé. Připnuté příspěvky se zobrazují vždy a nevztahují se na ně filtrování, řazení ani vyhledávání.

- **F1.4 – Vyhledávání:** V rámci zobrazených příspěvků lze vyhledávat.
- **F1.5 – Řazení:** Příspěvky lze řadit vždy podle jednoho kritéria.
- **F1.6 – Filtrování:** Příspěvky lze filtrovat podle různých kritérií.
- **F1.7 – Povinnost a účast:** Příspěvky mohou být administrátory spravující nástěnku speciálně označené – zda jsou povinné pro všechny uživatele a nebo jestli je událost otevřená pro zapsání. Všichni uživatelé se mohou vyjádřit k příspěvku s otevřeným zápisem, zda se události zúčastní. Takové příspěvky a také příspěvky, které mají nastavenou povinnou účast, se budou zobrazovat uživatelům v jejich plánu (tzn. „Můj plán“).
- **F2 – Náradí a dovednosti:** U každého jobu bude přidána možnost zaevidovat, jaké náradí (a v jakém množství) je na místě a jaké náradí bude potřeba vzít s sebou. U pracantů by se evidovaly jejich možnosti – to znamená, s jakými nástroji jsou ochotni pracovat. V plánu by se pak rozšířily statistiky o položky, co bude potřeba vzít s sebou. U pracantů by zase byly statistiky rozšířeny o dovednosti.
- **F3 – Tabulky:**
  - **F3.1 – Řazení dle sloupců:** U všech sloupců v tabulkách napříč aplikací bude možnost řazení právě podle nich, pokud to dovoluje jejich povaha.
  - **F3.2 – Stálost filtrů:** Filtry u tabulek budou stálé a při znovu-vrácení se na stránku budou tyto filtry aplikovány.
  - **F3.3 – Reakce statistik:** Statistika budou reagovat na aktuálně zvolené filtry.
- **F4 – Fotografie:**
  - **F4.1 – Podpora nahrávání více fotografií:** Nahrávač fotografií bude podporovat nahrávání vícero fotografií.
  - **F4.2 – Mazání:** Nahrané fotografie bude možné smazat.
- **F5 – Mapa:** Aplikace bude podporovat v částech, kde se vyskytuje vyplňování adresy, možnost zvolení si adresy/souřadnic na mapě. Napsaná adresa bude snadno převeditelná do souřadnic na mapě. Naopak vybrané souřadnice na mapě budou převeditelné na adresu.
- **F6 – Jednotný požadavek s daty z formuláře:** Vyplněné údaje z formuláře se budou na backend posílat v jediném požadavku.
- **F7 – Rozšířená kontrola plánů:** Kontrola naplánovaných jobů bude obohacena o kontrolu faktů, zda dva lidé již spolu předtím pracovali a že člověk už jednou byl na dané práci. Dále pak bude kontrolovat dostatečnou zručnost s nástroji přihlášených pracantů.

## 1.4.2 Nefunkční požadavky

- **N1 – Responzivita:** Aplikace by měla být responzivní. Tedy by měla umožňovat snadné prohlížení na různě velikých zařízeních – od mobilních zařízení až po počítače.
- **N2 – Vzhled:** Aplikace bude mít vylepšenou a zmodernizovanou svou vizuální stránku, která pomůže uživateli s pochopením aplikace. U formulářů se toto vylepšení projeví vyšší důsledností ve vyplňování informací.
- **N3 – Udržovatelnost a rozšiřitelnost:** Kód by měl být přehledný a logicky členěný do komponent. Také by měl dodržovat DRY princip, tzn., že v případě výskytu stejné funkcionality by měla být vytvořena nová komponenta. Dále bude obtížně pochopitelný kód okomentovaný a bude k dispozici Swagger dokumentace rozšířeného API.

- **N4 – Oprávnění pro psaní na nástěnku:** Pouze administrátoři (tzn. lidé s rolí ADMIN) a lidé s rolí POSTS budou mít oprávnění psát příspěvky na nástěnku.
- **N5 – Fotografie**
  - **N5.1 – Množství a velikost nahrávaných fotografií:** Maximální velikost libovolné nahrané fotografie je omezena na 10 MB. Množství nahrávaných fotografií se bude lišit pro různé části aplikace. Pro pracanta a příspěvek na nástěnku to bude 1 fotografie a pro práci maximálně 10.
  - **N5.2 – Zobrazení nahrané fotografie:** Nahrané fotografie se budou zobrazovat se svým skutečným poměrem stran.
  - **N5.3 – Místo nahrávání:** Fotografie se budou nahrávat do příslušných složek k dané entitě a ročníku. Tato struktura bude dovolovat v případě potřeby uvolnění disku pro neaktivní ročníky.
  - **N5.4 – Způsob zaznamenávání cesty k souboru:** Cesty k souborům budou u příslušných entit ukládány pomocí relativní cesty k nim.
- **N6 – Vyhodnocování formulářů:** Vyhodnocování formulářů bude probíhat čistě prostřednictvím Zod a ne pomocí validace vycházející z HTML a podobně.

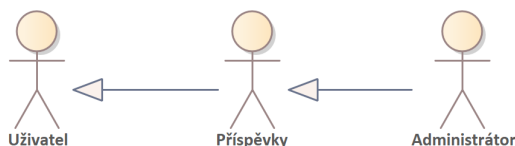
### 1.4.3 Model případů užití

Případy užití představují detailní specifikace funkčních požadavků. V tomto oddílu se budou vyskytovat netriviální případy užití týkající se nástěnky. Vzhledem k rozsahu práce bylo shledáno zaměření se v tomto oddílu jenom na nástěnku za dostačující.

#### Aktéři

Aktéři jsou entity, které interagují s aplikací. Může se jednat o další systém nebo čas, ale spíše častěji se jedná o uživatele s různými rolami. V následujícím seznamu jsou vyjmenováni aktéři, kterých se týkají popsané případy užití:

- **Administrátor:** uživatel s rolí ADMIN, který má přístup do všech částí aplikace,
- **Příspěvky:** uživatel s rolí POSTS, který může spravovat nástěnku,
- **Uživatel:** obyčejný uživatel bez přiřazené role.



■ **Obrázek 1.13** Aktéři. Vytvořeno v programu Enterprise Architect [15]

## Případy užití

**UC1 – Vytvoření příspěvku:** Případ užití umožňuje uživateli s příslušnou rolí vytvořit příspěvek, který se následně bude zobrazovat na hlavní stránce nástěnky.

**Aktér:** Administrátor nebo Příspěvky

**Výsledek:** Příspěvek je součástí všech vytvořených příspěvků a zobrazuje se na nástěnce, pokud není vyfiltrováný.

**Hlavní scénář:**

1. Aktér vstoupí na stránku s nástěnkou.
2. Systém mu zobrazí možnost přidat příspěvek, na kterou klikne.
3. Systém mu zobrazí formulář na vyplnění a aktér jej vyplní.
4. Aktér má možnost jít zpět na nástěnku (přesunuje se do 1. bodu) nebo uložit formulář (pokračuje se dalším bodem).
5. Systém vyhodnotí formulář. V případě chyb zobrazí u příslušných polí chybové hlášky a pokračuje se tak na 3. bod. Pokud je však formulář v pořádku pokračuje se dalším bodem.
6. Systém zpracuje vyplněné informace a v případě úspěšného zaregistrování zobrazí modální okno s úspěchem. V případě neúspěchu zobrazí modální okno s chybou.
7. a) Systém zobrazil aktérovi úspěch. Ten po potvrzení je přesměrován na nástěnku.  
b) Systém zobrazil aktérovi chybu. Ten po potvrzení je přesměrován na 3. bod.

**UC2 – Upravení příspěvku:** Případ užití umožňuje uživateli s příslušnou rolí upravit příspěvek, který se následně bude zobrazovat na hlavní stránce nástěnky.

**Aktér:** Administrátor nebo Příspěvky

**Výsledek:** Příspěvek je součástí všech vytvořených příspěvků a zobrazuje se na nástěnce, pokud není vyfiltrováný.

**Hlavní scénář:**

1. Aktér vstoupí na stránku s nástěnkou.
2. Systém mu zobrazí příspěvky.
3. Uživatel si zvolí libovolný příspěvek, a klikne na možnost úpravy.
4. Systém mu zobrazí formulář s již dříve vyplněnými informacemi a aktér jej upraví.
5. Aktér má možnost jít zpět na nástěnku (přesunuje se do 1. bodu) nebo uložit formulář (pokračuje se dalším bodem).
6. Systém vyhodnotí formulář. V případě chyb zobrazí u příslušných polí chybové hlášky a pokračuje se tak na 3. bod. Pokud je však formulář v pořádku pokračuje se dalším bodem.
7. Systém zpracuje vyplněné informace a v případě úspěšného zaregistrování zobrazí modální okno s úspěchem. V případě neúspěchu zobrazí modální okno s chybou.
8. a) Systém zobrazil aktérovi úspěch. Ten po potvrzení je přesměrován na nástěnku.  
b) Systém zobrazil aktérovi chybu. Ten po potvrzení je přesměrován na 3. bod.

**UC3 – Filtrování příspěvků:** Uživatel může příspěvky na nástěnce filtrovat podle různých kritérií.

**Aktér:** Uživatel

**Výsledek:** Zvolené filtry jsou zaregistrovány do url adresy a příspěvky jsou vyfiltrovány.

**Hlavní scénář:**

1. Uživatel vstoupí na stránku s nástěnkou.
2. Systém mu zobrazí všechny příspěvky vyfiltrované dle defaultních filtrů a tlačítko s filtry. Defaultní filtry jsou odvozeny z url stránky.
3. Uživatel klikne na filtry.
4. Systém mu zobrazí modální okno s možnými filtry, kde jsou vyplněné defaultní filtry zmíněné v předchozím bodě.
5. Uživatel si navolí filtry.
6. Uživatel může svou volbu filtrů potvrdit přes tlačítko uložit, v tom případě se pokračuje dalším bodem. Nebo může modální okno zavřít a to buď přes křížek, tlačítko zpět nebo kliknutím mimo modální okno. V tom případě se pokračuje na bod 1.
7. Systém aplikuje zvolené filtry na všechny příspěvky vyjma připnutých a u tlačítka s filtry zobrazí kolik filtrů je aktuálně aktivních.

**UC4 – Řazení příspěvků:** Uživatel může příspěvky základně řadit vždy podle jednoho kritéria.

**Aktér:** Uživatel

**Výsledek:** Zvolené řazení je zaregistrováno do url adresy a příspěvky jsou seřazené.

**Hlavní scénář:**

1. Uživatel vstoupí na stránku s nástěnkou.
2. Systém mu zobrazí možnost příspěvky seřadit. Defaultně je zvolena možnost řazení dle času začátku události – od nejnovějšího k nejstaršímu, nebo možnost, která je odvozena z url stránky.
3. Uživatel na tuto možnost klikne.
4. Systém mu zobrazí modální okno, jehož obsahem jsou všechny možnosti řazení. Standardně je zvolená defaultní možnost zmíněna v předešlém bodě.
5. Uživatel si zvolí jednu z možností.
6. Systém příspěvky (kromě připnutých) seřadí dle zvolené možnosti.
7. Uživatel může kliknutím na křížek nebo mimo modální okno skončit s řazením. Alternativně může zvolit jinou možnost (bod 5).

**UC5 – Vyhledávání v příspěvcích:** Uživatel může základně vyhledávat v rámci příspěvků. Vyhledává se podle napsaného textu, kdy se zobrazí příspěvky, ve kterých je napsaný text uvedený.

**Aktér:** Uživatel

**Výsledek:** Vyhledávací fráze je zaregistrována do url adresy a příspěvky jsou vyfiltrovány.

**Hlavní scénář:**

1. Uživatel vstoupí na stránku s nástěnkou.
2. Systém mu zobrazí pole s vyhledáváním. Defaultně je toto prázdné nebo se odvozuje z url stránky.
3. Uživatel napíše text podle, kterého vyhledává.
4. Co uživatel píše, systém zobrazuje relevantní příspěvky. To znamená ty, které obsahují hledanou frázi. Připnuté příspěvky jsou zobrazeny vždy a vyhledávání na ně nemá vliv.

**UC6 – Účast na události:** Některé příspěvky mohou být označeny jako povinné pro všechny nebo otevřené pro účast. S povinnými uživatel nemůže interagovat, ale vidí, že se jich nutně musí zúčastnit. S příspěvky s otevřenou účastí uživatelé mohou interagovat potvrzením účasti.

**Aktér:** Uživatel

**Předpoklad:** Uvažovaný příspěvek je označený jako otevřený pro účast.

**Výsledek:** Uživatel je zapsán na událost z příspěvku a tento příspěvek se mu bude zobrazovat v „Můj plán“.

**Hlavní scénář:**

1. Uživatel vstoupí na stránku s nástěnkou.
2. Systém mu zobrazí příspěvky.
3. Uživatel klikne u příspěvku označené s otevřenou účastí na tlačítko zápisu.
4. Systém uživatele na tuto událost zapíše.

**Alternativní scénář:** Tento scénář začíná po 2. bodu hlavního scénáře.

1. Uživatel klikne na příspěvek, který je označený jako s otevřenou účastí.
2. Systém mu zobrazí modální okno s detaily o daném příspěvku a tlačítko pro zápis.
3. Uživatel klikne na tlačítko zápisu.
4. Systém uživatele na tuto událost zapíše.

**UC7 – Připínání příspěvků:** Uživatel s rolí POSTS nebo ADMIN může připnout příspěvky. To má za důsledek, že se tyto příspěvky zobrazují na nástěnce úplně nahoře a nevztahují se na ně filtry, řazení ani vyhledávání.

**Aktér:** Administrátor nebo Příspěvky

**Předpoklad:** Uvažovaný příspěvek není připnutý.

**Výsledek:** Příspěvek se zobrazuje jako připnutý všem uživatelům a nevztahují se na ně filtry, řazení ani vyhledávání.

**Hlavní scénář:**

1. Aktér vstoupí na stránku s nástěnkou.
2. Systém mu zobrazí příspěvky, kde u každého je možnost připnutí.
3. Aktér si zvolí ještě nepřipnutý příspěvek a klikne na možnost připnutí.
4. Systém tento příspěvek zobrazí na vrcholu stránky.

**UC8 – Zobrazení si podrobností o příspěvku:** Každý uživatel si může u jakéhokoliv příspěvku zobrazit jeho podrobné údaje.

**Aktér:** Uživatel

**Hlavní scénář:**

1. Uživatel vstoupí na stránku s nástěnkou.
2. Systém mu zobrazí příspěvky.
3. Uživatel si zvolí libovolný příspěvek, na který klikne.
4. Systém mu zobrazí modální okno s detaily o daném příspěvku.



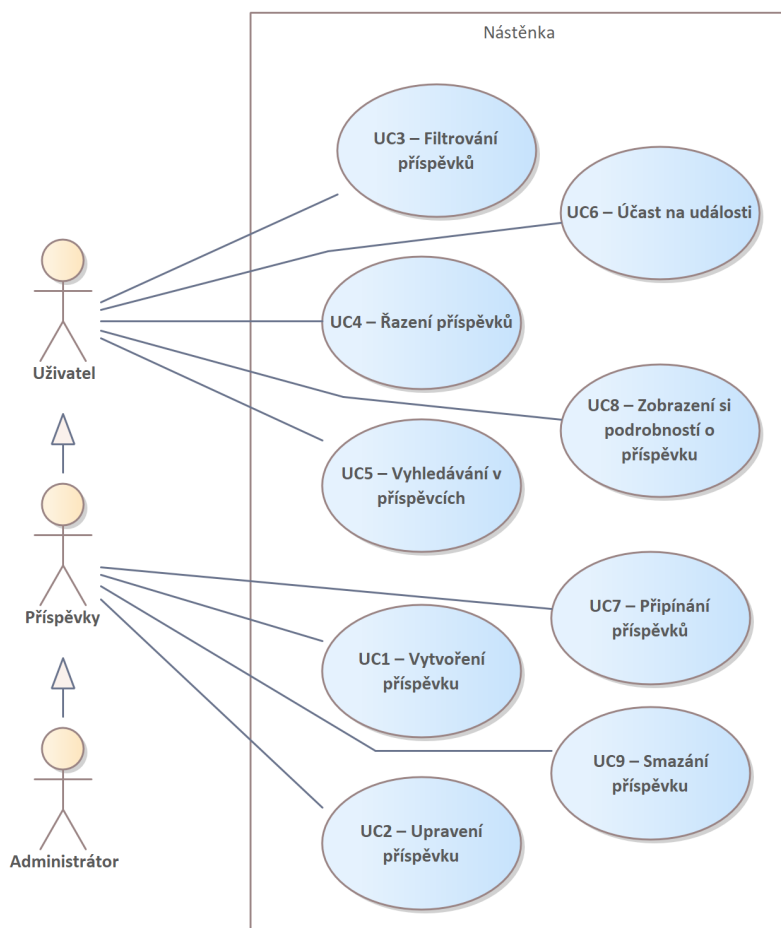
**UC9 – Smazání příspěvku:** Administrátor nebo uživatel s rolí POSTS mohou smazat libovolný příspěvek.

**Aktér:** Administrátor nebo Příspěvky

**Výsledek:** Příspěvek bude smazaný z nástěnky pro všechny uživatele. Pokud příspěvek měl nastavenou povinnou účast nebo byl otevřený na zápis, uživatelé budou z události příspěvku odhlášeni. To má za důsledek, že tuto událost neuvidí v „Můj plán“.

**Hlavní scénář:**

1. Aktér vstoupí na stránku s nástěnkou.
2. Systém mu zobrazí příspěvky.
3. Uživatel si zvolí libovolný příspěvek, a klikne na možnost smazání.
4. Systém mu zobrazí modální okno s potvrzením o smazání.
5. Uživatel potvrdí smazání a pokračuje se dalším bodem. Nebo z volby odejde, v tom případě se příspěvek nesmaže a vrací se do 1. bodu.
6. Systém zobrazí všechny příspěvky, kromě smazaného.



■ **Obrázek 1.14** Případy užití ve schématu. Vytvořeno v programu Enterprise Architect [15]



Tato kapitola představí návrh popisující, jak sesbírané a zanalyzované požadavky z předcházející kapitoly budou zpracovány v aplikaci. Kromě očividného návrhu týkajícího se vzhledu, se bude tato kapitola věnovat výběru některých technologií na uspokojení vybraných požadavků. Dále se jedna podkapitola bude věnovat návrhu konceptuálního modelu vytvářeného rozšíření v souvislosti s konceptuálním modelem vycházejícího z databázového modelu současného řešení.

## 2.1 Zvolené technologie

Vzhledem k tomu, že se tato práce věnuje rozvoji již existující aplikace, nebudou součástí této podkapitoly rozvahy o výběru základních technologií, ve kterých bude výsledná aplikace napsaná. Tyto technologie byly představeny v rámci podkapitoly o současném řešení (viz podkapitola 1.2), kde byla ostatně i zmíněna spokojenost se zvolenými technologiemi.

Avšak je zde stále malý prostor pro výběr aspoň dílčích technologií pro řešení jistých požadavků. Přesto drtivá většina požadavků je možné napsat již ve stávajících technologiích jako je Next.js (Typescript) a Zod.

### 2.1.1 Mapa

#### Požadavky

Před výběrem řešení je na začátek vhodné si nastínit požadavky na řešení mapy. Zvolená knihovna musí splňovat následující:

- **Cena:** Vzhledem k velikosti tohoto projektu a jeho samotné podstaty řešení musí být nenákladné, ideálně nezpлатněné.
- **API:** Řešení musí mít dobře zdokumentované API.
- **Responzivita:** Mapa musí dobře fungovat a zobrazovat se i na mobilních zařízeních.
- **Aktuálnost:** Data v mapě musí být aktuální a často aktualizované.
- **Funkcionality:**
  - Zobrazení dostatečně detailní mapy České republiky.
  - Podpora výběru místa na mapě.
  - Převod adresy na souřadnice a naopak (tzn. geokódování a reverzní geokódování).

Kromě těchto požadavků je potřeba zmínit, kde se budou mapy v aplikaci používat. Prvním místem jsou joby, kde je momentálně možno napsat pouze adresu. Zde budou na mapu vyžadovat pokročilejší funkcionality specifikované výše. Dalším místem vyžadující podobnou interaktivitu bude nástěnka s jednotlivými příspěvky. Poslední místo, kde se mapa bude zobrazovat je v „Můj plán“, kde se však bude jednat jen o mapu statickou bez dalších vlastností. Bude tedy dovolovat jenom očekávané funkce jako je přibližování/oddalování, pohybování se po mapě a jednoduché zobrazování informací (špendlík).

## Uvažovaná řešení

V rámci průzkumu bylo zaregistrováno několik populárních řešení splňujících požadavek na aktuálnost. Budou zde vyjmenovány jedny z těch nejvíce zmiňovaných:

- **Google Maps:** Google nabízí na svých stránkách několik služeb, ale použitelné pro tuto aplikaci jsou Maps JavaScript API, splňující požadavky na funkcionality, a Maps Embed API, která umožňuje zobrazení mapy bez dalších pokročilých funkcionalit. Maps JavaScript API je zpoplatněná a cenově vychází na 7 USD (přibližně 160 Kč) za 1 000 požadavků. Maps Embed API je zadarmo. [29] Jeden požadavek Google specifikuje zobrazení mapy webovou stránkou nebo aplikací [30].

Největší výhodou Google Maps je, že jsou známé širokou veřejností a toto řešení je zcela kompletní – od samotného zobrazování mapy, tak po samotná data. Nevýhodou je nízká přizpůsobitelnost designu.
- **Leaflet:** Z oficiálního webu [31] zaznívá následující: „*Leaflet je přední open-source knihovna JavaScriptu pro interaktivní mapy pro mobilní zařízení. Má jen asi 42 KB JS a má všechny mapové funkce, které většina vývojářů potřebuje. Leaflet je navržen s ohledem na jednoduchost, výkon a použitelnost. Funguje efektivně na všech hlavních desktopových i mobilních platformách, lze ji rozšířit pomocí mnoha zásuvných modulů, má krásné, snadno použitelné a dobře zdokumentované rozhraní API a jednoduchý, čitelný zdrojový kód.*“. Dále pak z webu npmjs je zřetelná popularita tohoto řešení – kolem 964 600 stažení za týden [32].

Z toho všeho vyplývá, že Leaflet vychází jako ideální kandidát a splňuje všechny důležité požadavky. Je nutné však podotknout, že samotný Leaflet je pouze knihovna na zobrazování map a sám o sobě nenabízí data a další funkce. Musí být tedy použit v kombinaci s další službou. Na jejich oficiálním webu je použit v kombinaci s OpenStreetMap, ale je možné jej použít i v kombinaci se Seznam mapami. V případě výběru tohoto řešení musí proběhnout diskuze, kterou službu na obsah mapy použít.
- **Mapbox GL JS:** Z npmjs vyháází, že i toto řešení je velice populární se svými přibližně 988 300 staženími za týden [33]. Z jejich oficiálního webu je pak patrné, že toto řešení je velmi přizpůsobitelné a nabízí vyžadované funkce [34].

Právě v nabízených funkcionalitách se skrývá hlavní nevýhoda tohoto řešení. Zabírá velikostně hodně místa a může tak být náročný na klienta. I přes optimalizace, rozebírané například v diskusi na GitHubu [35], toto řešení pořád zabírá více místa než dříve zmíněný Leaflet. Toto řešení také není zcela zdarma, ale vzhledem k velikosti tohoto projektu, by bylo možné řešení i tak používat zadarmo [36]. Ještě je potřeba zmínit, že stejně jako výše zmiňovaný leaflet sám za sebe nenabízí data do map a musí být použit v kombinaci s další službou.

## Závěr

Z předchozího textu vychází nejlépe Leaflet. Existuje však ještě jeho nástavba, která by se do této aplikace hodila o něco více. Tou je react-leaflet, který je ideální do React aplikací jako je právě tato. Sice z npmjs není tak stahovaný (se svými asi 329 000 staženími za týden [37]), jako Leaflet, přesto je pořád dostatečně populární. Nabízí příjemně předpřipravené komponenty a zpřístupňuje všechny pro tuto aplikaci podstatné funkce z Leafletu.

Jak již bylo zmíněno u představení Leafletu, je potřeba ještě zvolit službu s rozumným API pro poskytnutí samotných dat a obracené/geokódování. Bylo zmíněno řešení od Seznam.cz a.s. a OpenStreetMaps. Seznam nabízí neplacený tarif na 250 000 kreditů a za dalších 1 000 kreditů si účtuje 1,60 Kč. Jeden kredit například odpovídá jedné zobrazené dlaždici. [38] Tento zdánlivě vysoký tarif je tedy velmi snadné rychle vyčerpat. Oproti tomu OpenStreetMaps je řešení zcela zadarmo a proto bylo zvoleno právě to.

OpenStreetMaps pochopitelně vyžaduje dodržování autorských práv a licencí. Právě jejich zásady používání rozhraní API [39] je něco co je třeba mít na paměti. Například v rámci Geocoding Policy se zmiňuje, že počet požadavků musí být v rozumné míře a to maximálně jeden požadavek za sekundu, jinak hrozí zamezení přístupu. Proto návrh musí být v počtu možných požadavků restriktivní.

## 2.1.2 Nahrávání fotografií

V aplikaci již existuje základní nahrávání fotografií, které bylo napsáno čistě s použitím současných technologií. V rámci tohoto oddílu proběhne diskuze nad případnou doplňující technologií pro naplnění rozšiřujících změn.

### Požadavky

V následujícím seznam jsou vyjmenované požadavky, co by nahrávač fotografií měl splňovat:

- Možnost nahrávat více fotografií.
- Zobrazení nahrané fotografie musí dodržovat skutečný poměr stran.
- Nahraná fotografie půjde smazat.
- Nahranou fotografii si půjde rozkliknout a zobrazit si je tak ve větším formátu.

### Uvažovaná řešení

Byl proveden průzkum, které nahrávače se často používají v React aplikacích. Jedny z nejčastěji zmiňovaných byly tyto:

- **React-dropzone:** React-dropzone poskytuje komponentu pro snadné nahrávání souborů. Mezi jeho vlastnosti patří možnost nahrávat více souborů, funkce drag-and-drop, možnost úpravy rozhraní, zobrazování si nahraných souborů (v základu, ale fotografie nedodrží svůj poměr stran) a je responzivní. Nedovoluje však si fotografie rozkliknout a mazat je. [40]
- **Filepond:** Filepond nabízí široké možnosti konfigurace, je znám svou snadnou použitelností a responzivitou, kdy se dobře přizpůsobuje například displejům menších zařízení. Podporuje nahrávání více souborů, zachovává poměru stran obrázků, zobrazení obrázku s jeho možnou úpravou (oříznutí, barvy apod.), dovoluje mazání a spoustu dalšího. [41]

### Závěr

V rámci uvažovaných řešení nebylo nalezeno žádné, které by zcela splňovalo funkční a vizuální nároky. Spoustu z těchto knihoven nabízí v základu, alespoň částečné pokrytí zmíněných požadavků. To by ale znamenalo další kombinace s jinými knihovnami nebo vlastní doplňující kód s nutností hlubšího pochopení daného API. Přesto však nejvíce zaujal Filepond a kdyby funkce drag-and-drop a upravování fotografie byly ty hlavní, tak by bylo zvoleno právě toto řešení. Ve finále bylo rozhodnuto nahrávání fotografií udělat vlastní s použitím nativního HTML inputu a TypeScriptu. To tedy znamená, že se naváže na již existující řešení nahrávače, které bude rozšířeno o požadované funkce.

## 2.2 Přístupy

Kvůli nově chystané nástěnce musí být přidán nový druh oprávnění – POSTS. Oproti tabulce 1.1 přibude ještě právě tento přístup. Nové role si je možno prohlédnout v tabulce 2.1.

role	nástěnka	plány	joby	auta	pracanti	administrace
POSTS	✓					
PLANS		✓				
JOBS			✓			
CARS				✓		
WORKERS					✓	
ADMIN	✓	✓	✓	✓	✓	✓

■ Tabulka 2.1 Nové role

## 2.3 Návrh obrazovek a prototypy

Některé požadavky, které byly v rámci analýzy sesbírané a byly vyhodnoceny, že se jim tato práce bude věnovat, jsou svým obsahem velmi malé a často nevyžadují zvláštní péči v podobě návrhu obrazovky. Těm, kterým ale ano, byly vytvořeny vizuální návrhy. V rámci této podkapitoly budou představeny a budou vysvětlena jistá návrhářská rozhodnutí, která ovlivnily jejich vytváření.

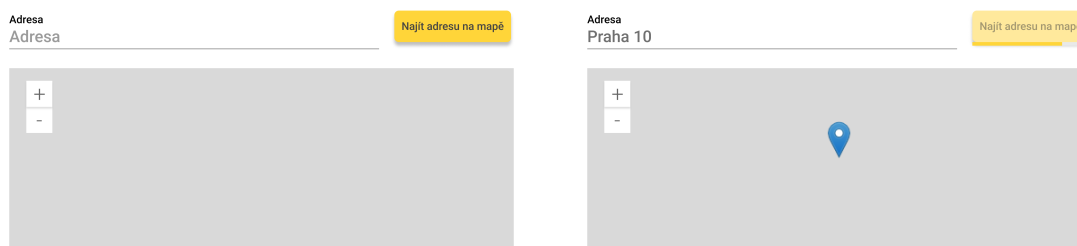
Cílem těchto návrhů je zlepšení vzhledu aplikace, ale i také samotné její používání. To tedy znamená vylepšení UI/UX. User interface (UI, česky „uživatelské rozhraní“) je převážně to, na co se tato podkapitola zaměří. Uživatelské rozhraní označuje celkově vzhled aplikace (např. barevné palety, typografii), konkrétní vizuální prvky (např. tlačítka, nadpisy apod.) a obecně představuje estetickou stránku aplikace. User experience (UX, česky přibližně „uživatelský prožitek“) na druhou stranu označuje celou interakci s vizuálními prvky. Dále pak zahrnuje i to, jaký pocit z takové interakce uživatel má. Zde v návrzích bude zmíněn také v rámci prototypů.

Pro vytváření návrhů byl využit nástroj Figma. Nástroj byl zvolen především kvůli předešlé zkušenosti autorky textu a také kvůli toho, že byl již v minulosti použit při navrhování současné aplikace. Figma nabízí příjemné rozhraní pro vytváření návrhů a interaktivitu v podobě prototypů [42]. Prototypy slouží jako ukázka a tudíž ne všechny jinak interaktivní prvky jsou v prototypu interaktivní. Prototypy si můžete vyzkoušet a v případě, že nebudete vědět s jakým konkrétně prvkem se dá navázat kontakt, zkuste kliknout kdekoliv v prototypu, Figma by měla modře zvýraznit místa interaktivitu. Prototypy jsou vytvořené pro nástěnku (viz pododdíl 2.3.6) a pro můj plán (viz oddíl 2.3.7).

### 2.3.1 Mapa

Vzhledem k vybranému technologickému řešení pro mapu, musí být návrh částečně restriktivní 2.1.1. Například, když by se mělo volat geokódování na převod adresy na souřadnice musí jít o cílenou a vědomou akci uživatele. To znamená, že zde bude existovat tlačítko na převod a nebude se tak volat geokódování při samotném psaní adresy. Geokódování je zde potřeba využít, aby se na mapě mohl zobrazit špendlík s určitou polohou a k tomu jsou potřeba právě souřadnice, které jsou výstupem geokódování.

Při výběru konkrétní polohy na mapě, tedy souřadnic, bude voláno reverzní geokódování. Při obou požadavcích, geokódování a reverzní geokódování, pak bude nastaveno zpoždění zamezující vybrání souřadnic na mapě, automatického převodu na adresu a tlačítko na převod adresy na souřadnice



(a) Mapa na začátku

(b) Mapa po vybrání

### ■ Obrázek 2.1 Návrh mapy v nástroji Figma [42]

Na obrázku 2.1a si lze prohlédnout jak vypadá komponenta na začátku, kdy ještě není zvolena adresa ani souřadnice. Do stavu na obrázku 2.1b se lze dostat buď:

- **Po kliknutí na konkrétní místo na mapě:** To má za důsledek převedení souřadnic na adresu, která se vypíše do pole pro adresu a zobrazení špendlíku v kliknutém místě.
- **Po napsání adresy a následném kliknutí na tlačítko „Najít adresu na mapě“:** Toto má za důsledek převod adresy na souřadnice a následném zobrazení špendlíku na mapě.

Na obrázku 2.1b lze také vidět, jak vypadá samotné omezení v podobě prodlevy před dalším požadavkem.

Návrh je v základu přívětivý i ke zobrazení na telefonech. Mapa se responzivně zmenší a tlačítko pro převod adresy se také lehce zmenší a posune se pod adresu.

## 2.3.2 Fotografie

Nahrávač fotografií bude podporovat nahrávání více fotografií. Pro nahrání bude muset uživatel kliknout na velké tlačítko plus. Toto tlačítko zmizí po dosáhnutí limitu nahrávaných fotografií. Nahrávat může uživatel i více fotografií najednou.

Kritéria akceptovaného souboru jsou tyto:

- jedná se o obrázek,
- nedosahuje velikosti či není větší než 10 MB,
- nebyl ještě dosazen limit pro maximální počet fotografií.

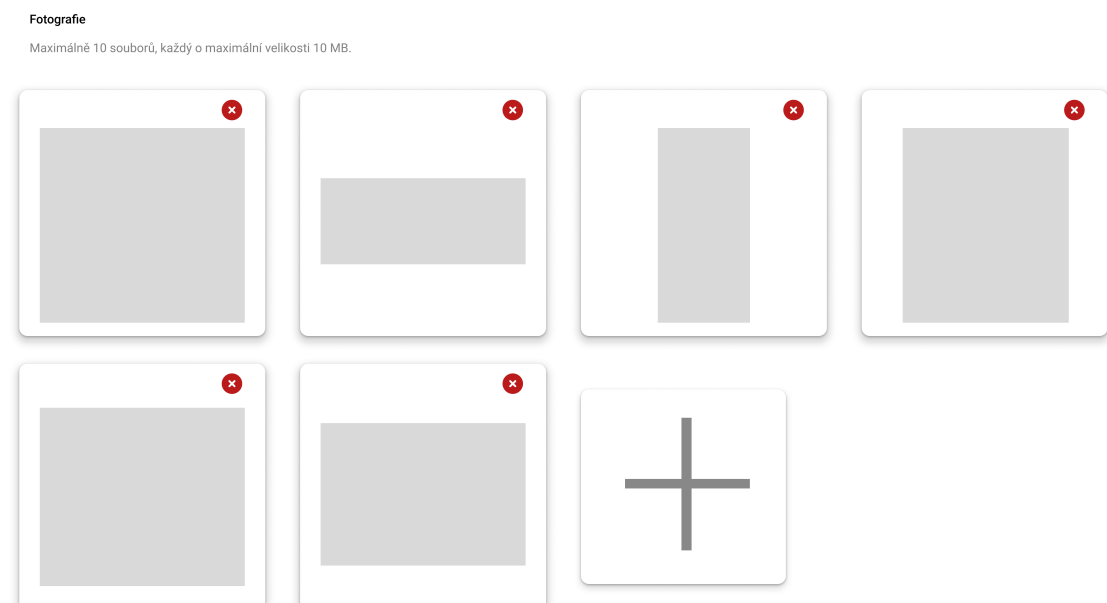
Pokud v rámci nahrávání fotografií jakýkoliv soubor nebude v souladu s kritérii, tak se tento soubor z nahrávání přeskočí.

Nahrané fotografie bude možné mazat, kdy se tyto soubory smažou i z místa uložení na serveru, po úspěšném odeslání formuláře, kde tento nahrávač bude součástí. Nahrávač fotografií nebude dělat rozdíly mezi fotografiemi, které byly již nahrány dříve (tedy těmi, které byly již jednou poslány na server). Tyto dřívější nahrané fotografie zobrazuje vedle nově nahraných s žádnými vizuálními rozdíly.

Na obrázku 2.2 si lze prohlédnout nahrávač fotografií, kde je nahráno 6 fotografií s různým poměrem stran a maximální počet fotografií nebyl ještě překonán.

Nahrané fotografie se v rámci svého pole zobrazují se správnými poměry stran. Po rozkliknutí libovolné fotografie se zobrazí modální okno obsahující tuto fotografii, také se správnými poměry stran. Po kliknutí na fotografii v rámci modálního okna se otevře fotografie na nové kartě prohlížeče. Modální okno je možné zavřít pomocí křížku nebo kliknutím kdekoli mimo modální okno.

Celkově je návrh responzivní a bude se zobrazovat pěkně i na telefonech. Jednotlivé dlaždice s fotografiemi se jednoduše posunou pod sebe a případně se lehce zmenší.



■ **Obrázek 2.2** Návrh uživatelského prostředí pro nahrání fotografií v nástroji Figma [42]

### 2.3.3 Výběr dostupnosti

Výběr dostupnosti představuje menší komponentu, která nahradí stávající výběr dat dostupnosti v rámci formulářů (viz obrázek 1.6). Vizually více připomíná kalendář, obsahově však obsahuje stejné informace (tzn. den, den v týdnu, měsíc) jako staré řešení jenom v trochu jiné podobě.

Návrh je k vidění na obrázku 2.3. Měl by být dobře viditelný i na telefonech, kdy se musí změnit trochu rozestup tlačítek, aby byly správně zachovány řady představující dny v týdnu.



■ **Obrázek 2.3** Návrh výběru dostupnosti v nástroji Figma [42]

Ve výběru dnů je možno kliknout pouze na ta tlačítka se dny, kdy se koná aktuální ročník SummerJob. Ostatní tlačítka se dny mimo akci jsou vypnuté a přesto viditelné, kdy slouží pouze estetickému účelu pro doplnění týdnů. Uživatelem vybraná tlačítka jsou podbarvena žlutou barvou.

Komponenta bude umožňovat i možnost omezení stisku některých tlačítek v době akce. Například pracant, dle jednoho požadavky z GitHub Issues, bude mít zamezený přístup ke změně své dostupnosti na další den vždy po 18:00.



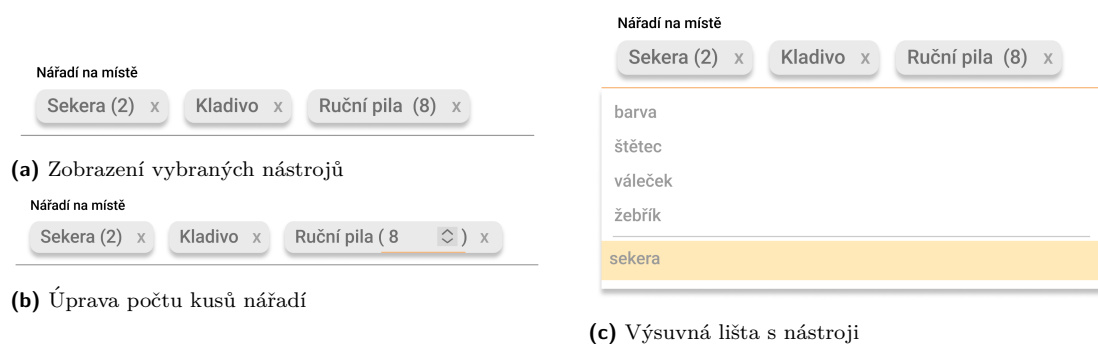
## 2.3.4 Nástroje

Nástroje je potřeba evidovat u jednotlivých jobů. Požadavek na nástroje nevyžaduje samostatnou evidenci. Předpokládá se tedy existence předvytvořeného seznamu možných nástrojů.

Tento seznam reaguje na „Typ práce“, což je pole obsahující jediný typ, pod kterým je job klasifikován. Každý nástroj je vázán i na více typů práce. Seznam nástrojů vytvářené komponenty bude mít dvojitou část seřazenou abecedně oddělenou oddělovačem. Vrchní část seznamu bude obsahovat ty nástroje, které jsou vázány k vybranému typu práce a spodní bude zobrazovat zbytek nástrojů. V seznamu je možné vyhledávat podle názvu nástroje.

Při vybrání nástroje ze seznamu se tento nástroj zobrazí v buňce se vstupem pro napsání počtu (základně je počet nastaven na 1) a křížkem pro smazání. Pokud je počet nastavený na jeden, tak se v buňce s neaktivním vstupem nebude vedle názvu toto číslo zobrazovat. Pokud je počet nastavený na nulu, tak se buňka s nástrojem smaže (tzn., že tato možnost je alternativou kliknutí na křížek). V ostatních případech se bude počet nástrojů v buňce zobrazovat vedle názvu nástroje.

Editace počtu u nástroje je možná prostřednictvím kliknutí na buňku. Dále se tato editace spustí i při znovu vybrání nástroje ze seznamu. V seznamu se již vybrané nástroje podbarví žlutou barvou. Pro doplnění textového popisu si je možno prohlédnout konkrétní návrhy na následujících obrázcích:



■ **Obrázek 2.4** Návrh nástrojů v nástroji Figma [42]

## 2.3.5 Formuláře

Všechny předešlé oddíly se věnovaly komponentám vyskytujících se ve formulářích. Samotný formulář si však také zaslouží alespoň malé vizuální vylepšení 2.5. Konkrétně se jedná o následující:

- **Nadpis:** Nadpis není v současnosti příliš výrazný a zapadá se zbytkem formuláře. Návrhem je obrácení barev, tzn. tmavé pozadí a světlý text. Obsah nadpisu bude v aplikaci sjednocen a bude se vždy zobrazovat název typu formuláře.
- **Stín:** Stín je aktuálně nevýrazný, změna tkví ve změně barvy a velikosti stínu. Tento návrh se týká také i jiných stínů použitých v aplikaci.
- **Tlačítka „Zpět“ a „Uložit“:** Některé formuláře (např. editace jobu) bývají velmi dlouhé a tato tlačítka se vyskytují až na úplném konci stránky. Návrhem je tyto tlačítka mít stále na očích vespod stránky.
- **Povinnost polí:** Běžný uživatel, který dobře nezná formulář, který vyplňuje, nemá možnost poznat, která pole jsou povinná a která nikoliv. Tuto informaci se dozví až u pokusu o odeslání formuláře, kdy se u polí napíše chybové hlášky ohlašující jejich povinnost. Návrh by využíval všeobecně známého designu s používáním hvězdiček u polí která jsou povinná.

■ **Obrázek 2.5** Návrh formuláře v nástroji Figma [42]

## 2.3.6 Nástěnka

Nástěnka je část aplikace, která je celá od začátku vytvářena v rámci této práce. Při návrhu nástěnky bylo možno se inspirovat již existujícím návrhem hlavní stránky jobů 1.8, která má jednotlivé joby v tabulce. Každý řádek s jobem je možno rozkliknout pro získání dalších informací, které se zobrazí v podobě rozbalovací nabídky. Tento návrh je zajisté funkční a hodí se pro správu a administraci. Nástěnka však má být otevřená pro všechny uživatele a má mít responzivní design, který je u tabulek obecně problematický, proto bylo rozhodnuto vytvořit návrh celý od začátku a řešit podobnou problematiku jiným způsobem.

V následujícím seznamu jsou zmíněné dílčí komponenty, které je potřeba vytvořit pro nástěnku a její příspěvky:

- **formulář** pro editaci / vytvoření příspěvků,
- **příspěvek** s důležitými informacemi,
- **modální okno** se všemi **informacemi** k příspěvku,
- možnost **zapsání se** na událost z příspěvku,
- **řazení** s modálním oknem pro úpravu,
- **filtry** s modálním oknem pro úpravu.

### Formulář

Nástěnka je sice otevřená všem uživatelům, ale uživatelé s rolí ADMIN a POSTS mají možnost správy jednotlivých příspěvků. To zahrnuje vytvoření, editaci, mazání a připínání příspěvků. Tyto pokročilé funkce se tedy nezobrazují běžným uživatelům.

Vytvoření příspěvku a jeho editace probíhá všude přítomným formulářem. Na obrázku 2.6 je možno shlédnout návrh formuláře pro úpravu příspěvku, ze kterého je patrný i jeho obsah.

**SUMMER  
JOB**

Nástěnka Plány Joby Auta Pracanti Administrace Můj plán Profil

## Upravit příspěvek

**Název \***  
Fotbálek s místními

**Zkrácený popis \***  
Vemte si sportovní oblečení. Pokud máte s sebou fotbalový míč vemte jej taky.

**Doplňující popis**  
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam ante. Itaque eorum rerum hic tenetur a sapiente, aut reiciendis voluptatibus maiores alias consequatur aut perferendis doloribus asperiores repellat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Nullam eget nisi. Aliquam erat volutpat. Mauris suscipit, ligula sit amet pharetra semper, nibh ante cursus purus, vel sagittis velit mauris vel metus. Fusce suscipit libero eget elit.

**Platné pro dny**  
Červen / Červenec 2024

Po	Út	St	Čt	Pá	So	Ne
24	25	26	27	28	29	30
1	2	3	4	5	6	7
8	9	10	11	12	13	14

**Čas**  
15:40 - 16:55

**Adresa**  
Na Lučínách u Vodojemu [Najít adresu na mapě](#)

**Fotografie**

**Tagy**  
sportovní

**Další vlastnosti**  
 povinné pro všechny  
 otevřeno pro zápis

Zpet Uložit

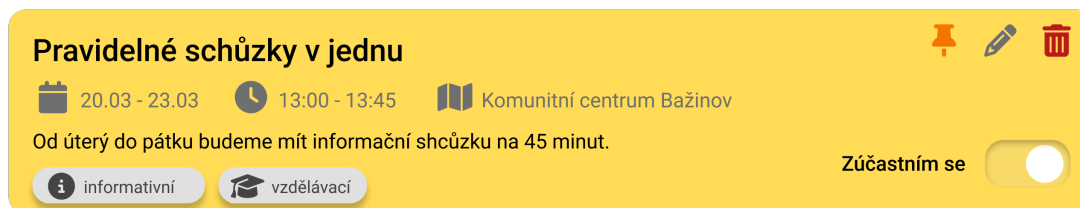
■ **Obrázek 2.6** Návrh formuláře pro úpravu příspěvku v nástroji Figma [42]

Pro úplnost zde budou jednotlivá pole (tzn. obsah formuláře) vyjmenována a případně dále doplněna textovým popisem:

- **Název:** Povinné pole, zobrazuje se u příspěvků.
- **Zkrácený popis:** Povinné pole, zobrazuje se u příspěvků.
- **Doplňující popis:** Zobrazuje se jenom v modálním okně příspěvků.
- **Platné pro dny:** Zobrazuje se u příspěvků, je možné zvolit více dnů, kdy se tato událost opakuje.
- **Čas:** Zobrazuje se u příspěvků, podle tohoto údaje se rozdělují příspěvky na obecné a časové. U časových je tento čas vytažený ven z příspěvku. U připnutých příspěvků se zobrazuje vždy. Pole sice není povinné, ale v případě vyplněného času od, musí být vyplněný i čas do a naopak.
- **Adresa a mapa:** Adresa se zobrazuje u příspěvků, mapa se zobrazuje jenom v modálním okně příspěvků.
- **Fotografie:** Zobrazuje se jenom v modálním okně příspěvků. Příspěvek může mít jenom jednu 10 MB fotografii.
- **Tagy:** Zobrazuje se u příspěvků, je možno vybrat více tagů.
- **Další vlastnosti:** V případě povinné účasti pro všechny se všem uživatelům zobrazuje zaškrťovací pole bez možnosti odhlášení. V případě nastavené otevřené účasti se u příspěvku zobrazuje interaktivní zaškrťovací pole. Zapsání znamená propsání události z příspěvku do „Můj plán“.

## Příspěvek

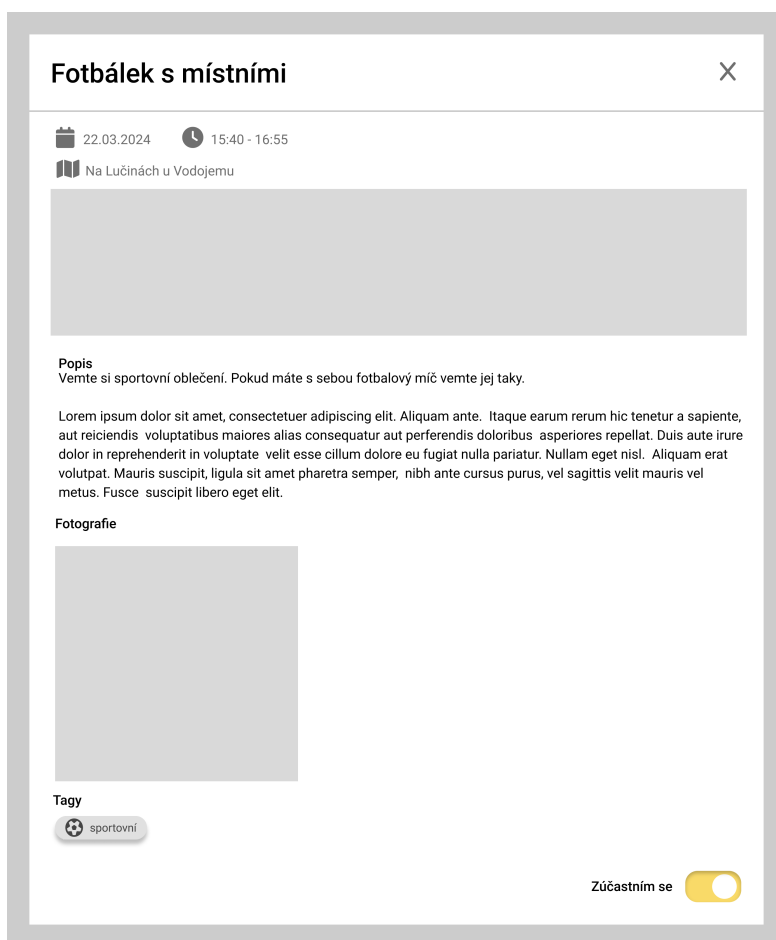
Na obrázku 2.7 je zobrazena varianta příspěvku zobrazovaná lidem s rolí ADMIN nebo POSTS, který je připnutý. To, že je příspěvek připnutý, je znázorněno tmavším podbarvením příspěvku a barevným připínáčkem.



- **Obrázek 2.7** Návrh příspěvku v nástroji Figma [42]

## Modální okno příspěvku

Všechny příspěvky lze rozkliknout pro zhlédnutí doplňujících informací (viz obrázek 2.8). U části „Popis“ je potřeba podotknout, že se skládá ze zkráceného popisu a dlouhého popisu.



■ **Obrázek 2.8** Návrh modálního okna příspěvku v nástroji Figma [42]

## Řazení

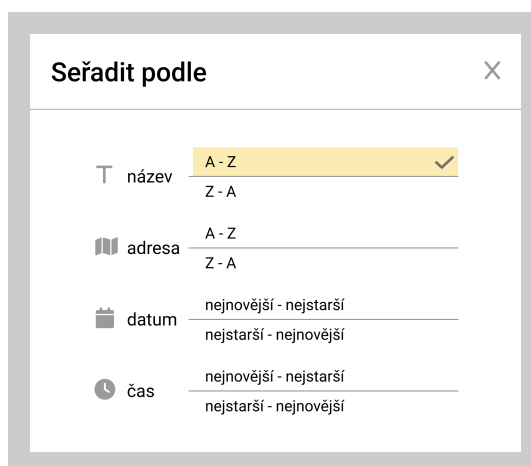
Příspěvky lze alespoň základně řadit vždy podle jedné vlastnosti, základně je vybraná možnost „čas: nejnovější – nejstarší“. Řazení je jedna z věcí, která by byla mnohem snadněji proveditelná a i lépe vypadající v případě řešení pomocí tabulky. Proto může být tento konkrétní návrh vnímán jako nevýhoda celkového návrhu.

Na pole, které zobrazuje aktuální vybrané řazení, lze kliknout a tím si zobrazit modální okno (viz obrázek 2.9). V něm si lze vybrat požadované řazení, po kliknutí se příspěvky na hlavní stránce nástěnky okamžitě seřadí.

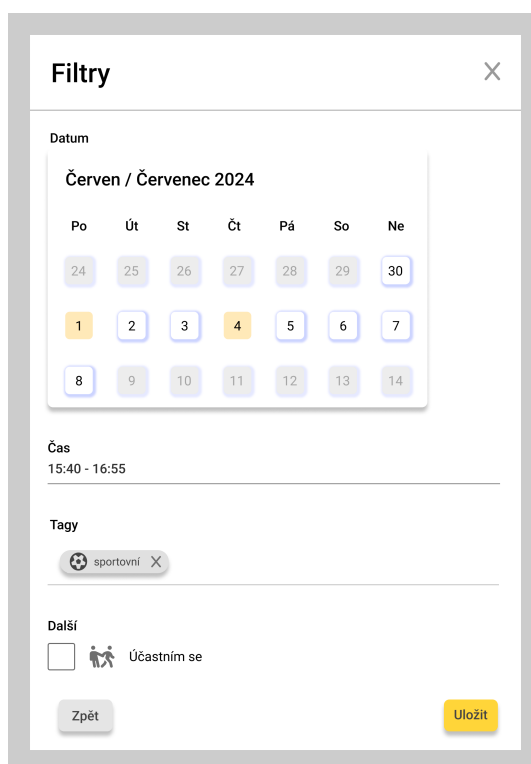
## Filtry

Řazení bylo v přecházejícím oddílu označeno jako potenciální slabina návrhu, kdežto filtry by se daly považovat naopak za jeho silnou stránku. Oproti původním filtrům u tabulek (viz obrázek 1.8), dovoluje tento návrh filtrů mnohem větší přehlednost a prostor pro výběr.

Filtry jsou zobrazovány velice podobně jako řazení. Navíc však mají u sebe zvýrazněné číslo s počtem aktivních filtrů. Po kliknutí na tlačítko s filtry se zobrazí modální okno 2.10 pro výběr filtrů. Oproti řazení musí být volba ztvrzena tlačítkem „Uložit“. Je to především z důvodu, že neustálé filtrování na pozadí modálního okna, může působit na uživatele rušivě. Dalším důvodem je zamezení zvýšené výpočetní náročnosti na klientovi.



■ **Obrázek 2.9** Návrh modálního okna pro řazení příspěvků v nástroji Figma [42]

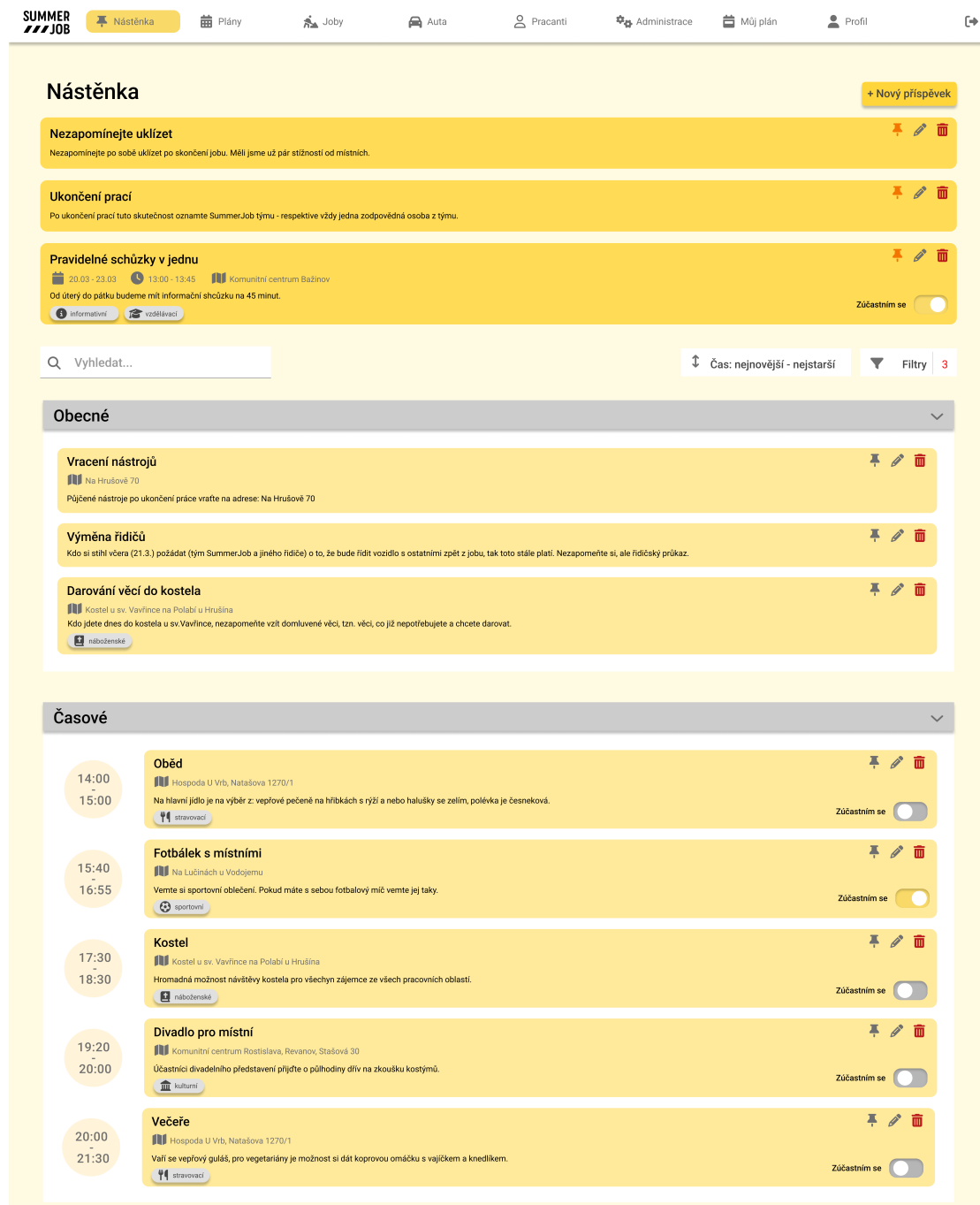


■ **Obrázek 2.10** Návrh modálního okna pro filtrování příspěvků v nástroji Figma [42]

## Úvodní stránka

Na závěr některé výše zmíněné prvky budou uvedeny do souvislosti úvodní stránky nástěnky, kde bude možno si prohlédnout jak tyto prvky budou spolu vypadat a interagovat. Na návrh stránky nástěnky je možno se podívat na obrázku 2.11. Většina obsahu byla již vysvětlena výše, zbytek by měl být dobře pochopitelný. Dále byl vytvořen interaktivní prototyp nástěnky, který je dostupný na této adrese<sup>1</sup>.

<sup>1</sup>Vzhledem k velikosti odkazu, není odkaz viditelně uveden. Je však dostupný v online verzi této práce.



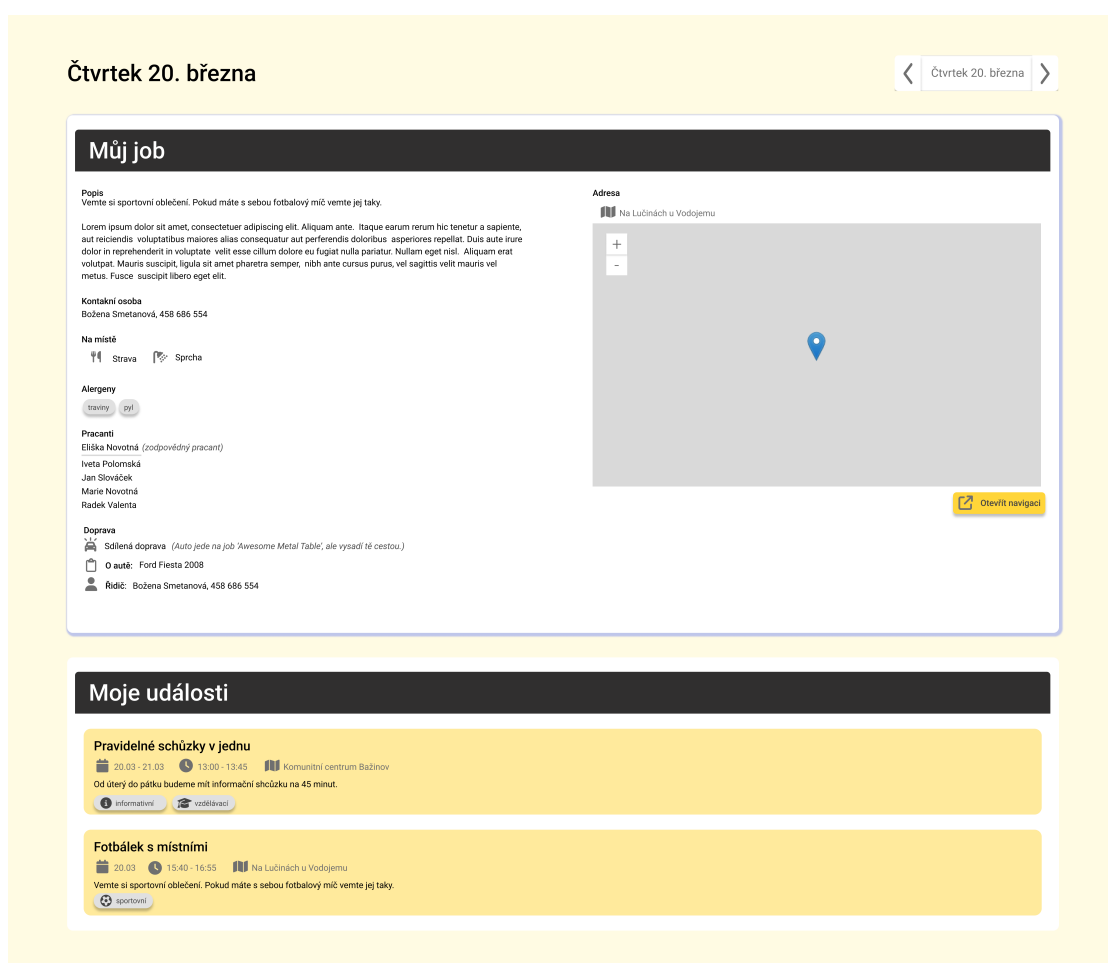
■ Obrázek 2.11 Návrh úvodní stránky nástěnky v nástroji Figma [42]

## 2.3.7 Můj plán

Stránka Můj plán, podobně jako Nástěnka, bude navštěvována obyčejnými uživateli. Proto by měla být prioritní, co se vizuálního vylepšení týče. Aktuální stav si lze prohlédnout na obrázku 1.12. Je z něj patrné, že samotný obsah je vzhledově velice strohý. Vytvořený návrh byl zkonstruován, aby vizuálně připomínal dříve zmíněný návrh obsahu modálního okna příspěvku (viz obrázek 2.8).

Kromě informací o jobu, na který daný den uživatel jde, bude stránka „Můj plán“ obohacena o události, kterých se uživatel účastní. Bude se jednat o události z nástěnky, které mají povinnou účast a také o události, u kterých uživatel označil, že se jich zúčastní. Události se budou zobrazovat podobně jako příspěvky z nástěnky s možností rozkliknutí pro zobrazení modálního okna s podrobnostmi.

Obsah plánu na určitý den by pak mohl například vypadat, tak jako na obrázku 2.12. Tento návrh si můžete i vyzkoušet v rámci této stránky<sup>2</sup>.



■ Obrázek 2.12 Návrh obsahu pro stránku „Můj plán“ v nástroji Figma [42]

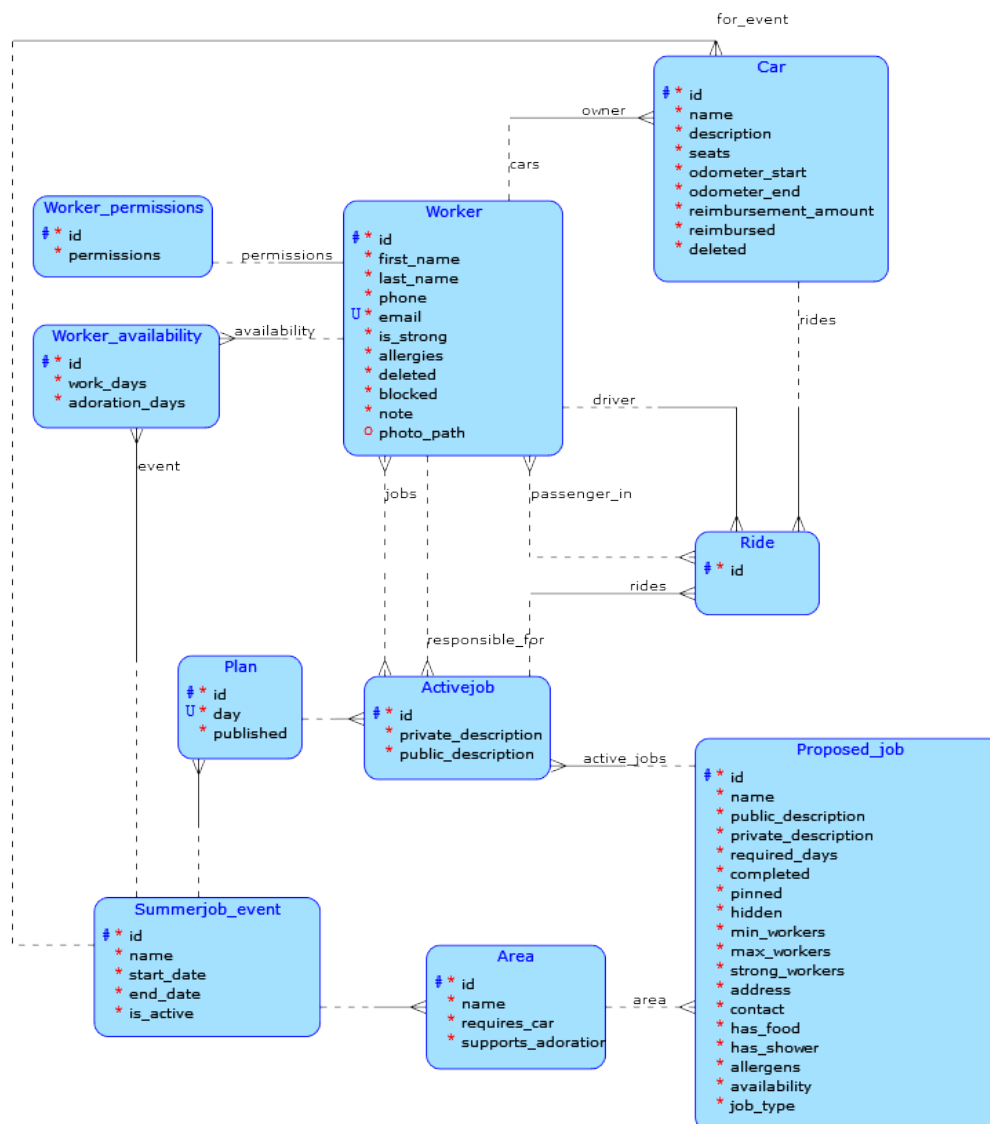
<sup>2</sup>Vzhledem k velikosti odkazu, není odkaz viditelně uveden. Je však dostupný v online verzi této práce.



## 2.4 Konceptuální model

Obrázky v této podkapitole jsou reprezentací schématu databáze zachycené pomocí ER diagramu. Jde o konceptuální model a proto nejsou uváděny datové typy. Při vytváření byl použit nástroj Data modeller od ČVUT FIT [43]<sup>3</sup>.

Schéma na obrázku 2.13 vychází z databázového Prisma schématu ve fázi převzetí projektu [44]. Na obrázku jsou znázorněny všechny modely, na kterých bude provedeno rozšíření nebo jsou potřebné k pochopení celkového schématu.



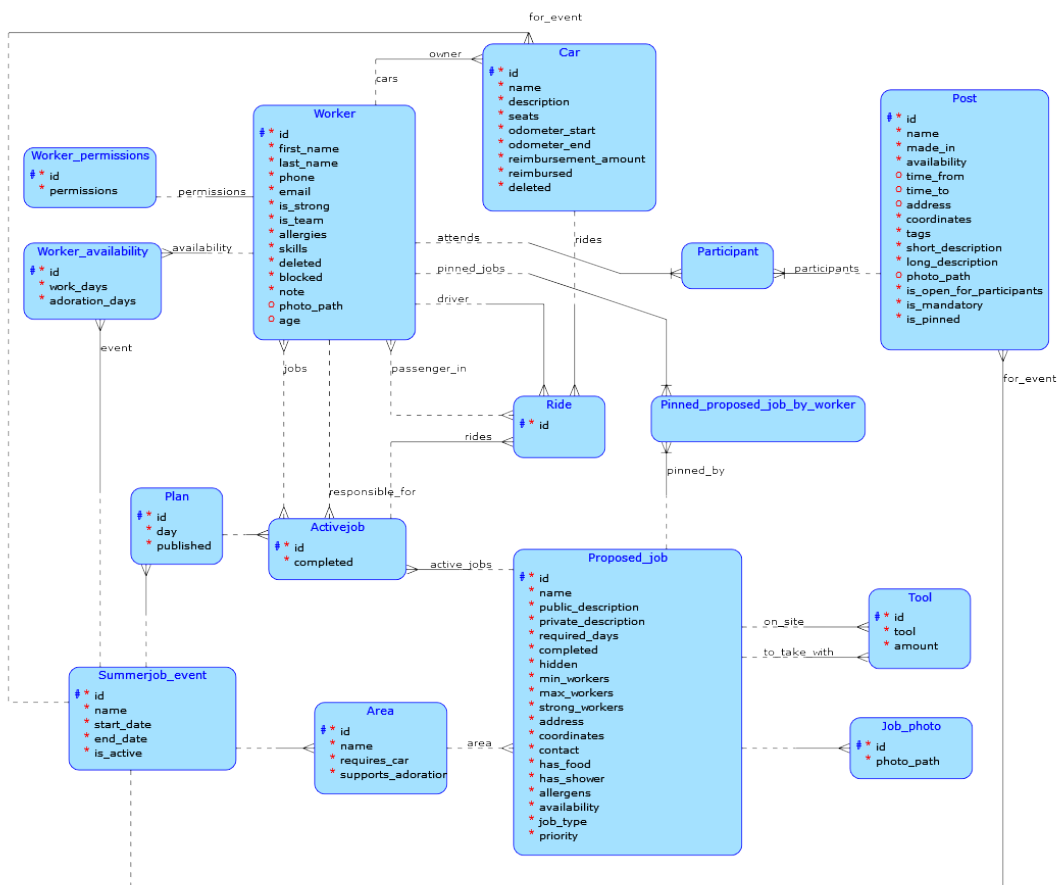
■ **Obrázek 2.13** Konceptuální model vycházející ze současného stavu. Vytvořeno v modelovacím nástroji portálu dbs [43]

<sup>3</sup>Tento nástroj nepodporuje camelCase ale pouze snake\_case, proto některé názvy neodpovídají stylizaci schématu v Prismě.

Návrh upraveného konceptuálního modelu zahrnuje následující:

- **Nástroje** (model `Tool`) patřící k navrhované práci (model `Proposed_job`): U práce se evidují nástroje, které jsou k dispozici na místě a ty, které je potřeba vzít s sebou. U nástroje evidujeme o jaký nástroj jde (`tool`) a kolik ho je (`amount`).
- **Účastník** (model `Participant`) události z příspěvku (model `Post`), zahrnující pracanta (model `Worker`): Účastník je identifikován kombinací identifikátorů pracanta a příspěvku.
- **Příspěvek** (model `Post`): U příspěvku se eviduje jeho jméno (`name`), kdy byl vytvořen (`made_in`), kdy se koná (`availability`), odkdy (`time_from`), dokdy (`time_to`), souřadnice pro mapu (`coordinates`), adresa (`address`), tagy (`tags`), krátký popis (`short_description`), dlouhý popis (`long_description`), cesta k fotografii (`photo_path`), je-li otevřen pro zápis (`is_open_for_participants`), je-li povinný (`is_mandatory`) a je-li připnutý (`is_pinned`). Příspěvek se vždy vztahuje k SummerJob ročníku (model `Summer_job_event`). Mohou se u něj evidovat účastníci (model `Participant`), pokud je označený jako otevřený pro účast.
- **Fotografie práce** (model `Job_photo`): Tento model slouží k evidenci více fotografií, které je možné nahrávat u navrhované práce (model `Proposed_job`).
- Drobné úpravy u **pracanta** (model `Worker`): Úpravy reagují na požadavky z GitHub Issues. Nově se eviduje je-li součástí týmu SummerJob (`is_team`), schopnosti (`skills`), věk (`age`), jakých událostí se účastní (model `Participant`) a které joby (model `Proposed_job`) si připnul (model `Pinned_proposed_job_by_worker`).
- Drobné úpravy u **navrhované práce** (model `Proposed Job`): Byla odstraněná informace, zda je globálně připnutý (`pinned`) a místo toho se eviduje, kterými pracanty byl připnutý (model `Pinned_proposed_job_by_worker`). Dále byly přidány položky jako jsou souřadnice pro mapu (`coordinates`), nástroje na místě (model `Tool`), nástroje potřebné na vzetí s sebou (model `Tool`), fotografie (model `Job_photo`) a priorita (`priority`).
- **Připnutí navrhované práce** (model `Pinned_proposed_job_by_worker`): Globálního připínání práce (model `Proposed_job`) bylo nahrazeno připínáním jednotlivými pracanty (model `Worker`).
- Drobné úpravy u **aktivní práce** (model `Active Job`): Byly odstraněny položky s popisy (`private_description` a `public_description`), které budou v aplikaci nahrazeny z modelu `Proposed_job`. Byla přidána informace o tom, zda je práce za daný den hotova (`completed`).

Změny si lze prohlédnout i na obrázku 2.14.



■ Obrázek 2.14 Návrh úpravy v konceptuálním modelu. Vytvořeno v modelovacím nástroji portálu dbs [43]



# Implementace

Tato kapitola se zabývá popisem vlastní implementace požadovaných změn (viz podkapitola 1.3 a podkapitola 1.4). Budou zde zmíněny i obtíže, které byly při implementaci potřeba překonat.

## 3.1 Implementace klíčových částí

V rámci této podkapitoly budou popsány klíčové části implementace. Následující oddíly nabídnou ucelený pohled na řešenou problematiku, to znamená popis frontendu, backendu, databáze, obhájení jistých rozhodnutí a řešené obtíže.

### 3.1.1 Formuláře

V převzatém řešení se vyskytovalo spoustu duplicitního kódu a to převážně na frontendové části aplikace. Především se tak dělo u formulářů. Proto bylo rozhodnuto začít s implementací právě tady.

#### Refaktorizace polí

Jednotlivá pole formulářů se dala rozdělit podle jejich očekávaného vyplněného obsahu a přesně podle tohoto kritéria bylo rozhodnuto je rozdělit do vlastních komponent. Takto bylo vytvořeno na 6 komponent. Toto rozdělení, díky své centralizaci do jediného místa, umožnilo snazší manipulaci a efektivní aplikování dalších funkčních a vizuálních změn.

Pro příklad ve výpisu kódu 3.1 si můžete prohlédnout komponentu reprezentující nejčastěji využívaný typ pole – `input`. Je složena ze tří částí tzn. Nadpisu (`Label`), hlavní části (`input`) a hlášení chyby (`FormWarning`). Díky této komponentě bylo umožněno sjednotit celkový vzhled (především odsazení), zpřehlednit a na některých místech zkrátit kód.

Některé takto přepracované komponenty přinesly ještě o něco razantnější změny. Například komponenta `DateSelectionInput.tsx` změnila celkový vzhled zobrazování výběru dostupnosti na základě návrhu vytvořeného v podkapitole 2.3. Tato komponenta dále umožňuje zamezení přístupu na následující den, podle jednoho z požadavků z GitHub Issues. Na obrázku 3.1 je k vidění snímek obrazovky vytvořené komponenty `DateSelectionInput.tsx`.

Kromě přepracování polí formuláře bylo vytvořeno na dalších tří nových typů polí. Jedná se o pole pro výběr priority, mapa a pole pro výběr buněk (např. pro výběr nástrojů pro job nebo tagů pro nástěnku).

```

1 import { type DetailedHTMLProps, type InputHTMLAttributes } from 'react'
2 import { FieldErrors, Path, UseFormRegisterReturn } from 'react-hook-
  form'
3 import FormWarning from '../FormWarning'
4 import { Label } from '../Label'
5
6 interface TextInputProps
7   extends DetailedHTMLProps<
8     InputHTMLAttributes<HTMLInputElement>,
9     HTMLInputElement
10  > {
11   id: string
12   label: string
13   register: () => UseFormRegisterReturn
14   errors: FieldErrors<FormData>
15   labelClassName?: string
16   margin?: boolean
17   mandatory?: boolean
18 }
19
20 export const TextInput = ({
21   id,
22   label,
23   register,
24   errors,
25   labelClassName = '',
26   margin = true,
27   mandatory = false,
28   ...rest
29 }: TextInputProps) => {
30   const error = errors?.[id as Path<FormData>]?.message
31
32   return (
33     <>
34       <Label
35         id={id}
36         label={label}
37         className={labelClassName}
38         margin={margin}
39         mandatory={mandatory}
40       />
41       <input
42         className="form-control smj-input p-0 fs-5"
43         {...register()}
44         {...rest}
45       />
46       <FormWarning message={error} />
47     </>
48   )
49 }

```

■ Výpis kódu 3.1 Komponenta TextInput.tsx

Pracovní dostupnost

Červen / Červenec 2025

Po	Út	St	Čt	Pá	So	Ne
30	1	2	3	4	5	6
7	8	9	10	11	12	13

■ **Obrázek 3.1** Implementovaný výběr dostupnosti

## Výběr priority

Komponenta `ScaleInput.tsx` vyhověla předloženému požadavku pro výběr priority u evidovaného jobu. Ta využívá výchozí HTML posuvník pro výběr čísla. Priorita je nastavitelná od 1 do 5. Bylo přidáno ještě mapování tohoto čísla na textovou reprezentaci pro větší pochopení toho, zda priorita s číslem 1 je nejvyšší nebo nejmenší.<sup>1</sup>

## Mapa

Komponenta `MapInput.tsx` zaštiťuje, jak výběr adresy, tak i mapy. Ve své logice se zabývá i převodem adresy na souřadnice a naopak. Pro to využívá funkcí z `GeocodingData.ts`, ve které se odesílají požadavky na Nominatim od OpenStreetMap. Jak bylo zmíněno u návrhu (viz podkapitola 2.1), výběr adresy a souřadnic musí být restriktivní a neumožňovat odesílání požadavků častěji než 1 vteřinu. Uživateli je proto dovoleno odeslat požadavek co 2 vteřiny. Čekání je znázorněno pomocí komponenty `ProgressBar.tsx`, ukazatelem průběhu, a deaktivováním tlačítka pro převod adresy na souřadnice.

Samotná mapa je vydefinovaná do komponenty `Map.tsx`, která využívá knihovny `react-leaflet`, která byla vybrána v návrhu. Komponenta `Map.tsx` je napsána tím způsobem, že je vhodná na použití jak do formulářů s interaktivním výběrem přípínáčku, tak i například na stránku „Můj plán“, kde je uživateli odebrána možnost vybírat místa na mapě.

U implementace se naskytl problém, že se tato komponenta pokoušela o SSR a při kompilaci se zobrazovala chyba `Reference error: window is not defined`. To bylo vyřešeno zakázáním SSR [45] (viz výpis kódu ze souboru `index.ts` 3.2). Komponenta mapy se důsledkem toho musí importovat jako odkaz na složku `Map/`. Struktura a popis této složky je popsána zde:

```
Map/
├─ index.ts ..... Soubor zakazující SSR pro komponentu mapy
├─ Map.tsx ..... Samotná komponenta mapy využívající react-leaflet
```

```
1 import dynamic from 'next/dynamic'
2 import { MapProps } from './Map'
3 import { ComponentType } from 'react'
4
5 const Map: ComponentType<MapProps> = dynamic(
6   () => import('./Map').then(module => module.Map),
7   {
8     ssr: false,
9   }
10 )
11 export default Map
```

■ **Výpis kódu 3.2** Zakázání SSR pro komponentu Mapy v souboru `index.ts`

<sup>1</sup>Číslo 1 je pro prioritu u jobu bráno jako nejmenší.

Dalším problémem související s mapou byl fakt, že se nezobrazovala správně ikona pro výběr na mapě. Tento problém byl způsoben tím, že react-leaflet neměl k dispozici obrázky připínáčku a jeho stínu. To bylo opraveno prostřednictvím vlastního nastavení ikony používající oficiální obrázky z Leafletu (viz 3.3).

```

1 import L from 'leaflet'
2
3 // Fix for Marker Icon that is not defaultly showing for react-leaflet
4 const DefaultIcon = L.icon({
5   iconUrl: 'https://unpkg.com/leaflet/dist/images/marker-icon.png',
6   shadowUrl: 'https://unpkg.com/leaflet/dist/images/marker-shadow.png',
7   iconAnchor: [11, 46], // fix icon position
8   popupAnchor: [2, -46], // fix popup window position
9 })
10 L.Marker.prototype.options.icon = DefaultIcon

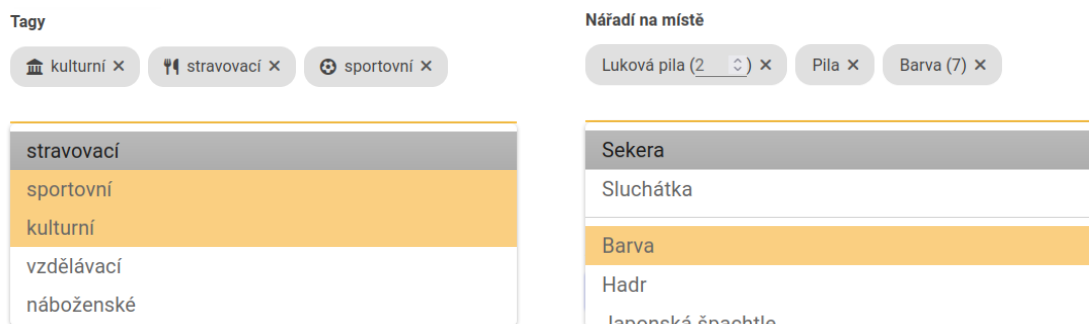
```

■ **Výpis kódu 3.3** Nastavení vlastní ikony připínáčku do mapy

## Výběr buněk

Komponenta na výběr buněk vesměs rozšiřuje již existující komponentu `FilterSelect.tsx`. Předtím však kód v komponentě `FilterSelect.tsx` byl opraven dle jednoho z požadavků z GitHubu, aby fungoval i na zařízeních od Apple. Na Apple zařízeních a konkrétně na jejich prohlížečích Safari nefungují správně `onBlur` eventy. Z konverzace ze StackOverflow [46], ve které se právě tento problém řeší, bylo zvoleno navržené řešení od uživatele lavzyd3v. Konkrétně se jedná o řešení s detekováním kliknutí mimo vybraný objekt.

Komponenta na výběr buněk pojmenovaná jako `PillSelect.tsx` je napsaná univerzálně, aby byla vhodná pro výběr jednoho nebo více prvků, s možností jejich smazání. Dále podporuje přidání množství k dané buňce, což se hodí právě pro výběr nástrojů. Kromě toho se tato komponenta využívá pro tagy na nástěnku, kdy nabízí možnost zobrazování ikon u jednotlivých tagů. Využití této komponenty v aplikaci si můžete prohlédnout na obrázku 3.2.



(a) Implementovaný výběr tagů příspěvku

(b) Implementovaný výběr nástrojů jobu

■ **Obrázek 3.2** Implementovaný výběr buněk

## Celkový vzhled

Na všechny formuláře byl aplikován nový vzhled dle návrhu 2.3.5. Aby změny nemusely být aplikovány manuálně do všech výskytů formulářů byla znovu nutná refaktORIZACE KÓDU. Proto byla vytvořena komponenta `Form.tsx` obalující obsah jednotlivých formulářů. Vedlejším účinkem této změny byla ještě vyšší čitelnost kódu jednotlivých formulářů, které už tak zaznamenaly velké vylepšení díky refaktORIZACI jejich obsahu – polí.



Zmiňovaná komponenta dodává formulářům jejich vzhled, který byl právě v rámci implementace lehce pozměněn. Obsahuje například další nově vytvořenou komponentu spravující nadpis – `FormHeader.tsx`. Kromě toho jsou součástí i tlačítka „Zpět“ a „Uložit“, která díky CSS (konkrétně díky `position: sticky` a `bottom: 0`) zůstávají vždy viditelná vespod formuláře.

Konkrétně se lze na vytvořenou implementaci na obrázku 3.3.

The image shows two side-by-side screenshots of a mobile application interface for editing a job. Both screenshots have a header with the text 'SUMMER' and 'JOB' and a hamburger menu icon.

The left screenshot is titled 'Upravit job'. It contains the following sections:
 

- Název jobu \***: Input field with the text 'Hrabání listí'.
- Popis navrhované práce**: Text area containing the Latin text 'Eos id molestiae vitae facilis sapiente asperiores accusamus. Et recusandae at incidunt accusamus facere unde. Expedita expedita'.
- Poznámka pro organizátory**: Text area containing the Latin text 'Rem illum consecetur laudantium deserunt. Velit voluptatum eum quos velit consequatur. Dolorem quis corporis quasi possimus occaecati'.
- Oblast jobu \***: Input field with the text 'Krupka'.
- Adresa \***: Input field with the text 'U Vesny 85'. Below it is a yellow button with a magnifying glass icon and the text 'Najít adresu na mapě'.
- At the bottom are two buttons: 'Zpět' (grey) and 'Uložit' (yellow).

The right screenshot is titled 'Adresa \*'. It contains the following sections:
 

- Adresa \***: Input field with the text 'U Vesny 85'. Below it is a yellow button with a magnifying glass icon and the text 'Najít adresu na mapě'.
- Mapa**: A map showing the location of 'U Vesny 85' in the Czech Republic. The map includes labels for various regions and cities like Praha, Plzeň, and Ostrava.
- Fotografie**: Text indicating 'Maximálně 10 souborů, každý o maximální velikosti 10 MB.' Below this is a photo upload area with a red 'x' icon, a small thumbnail of a job listing, and a large blue square with a white crosshair.
- Kontakt \***: Input field with the text '011 478 006 011'.
- At the bottom are two buttons: 'Zpět' (grey) and 'Uložit' (yellow).

■ Obrázek 3.3 Snímky obrazovek formuláře pro úpravu jobu z implementace.

## Vyhodnocování formulářů

Pro primární vyhodnocování formulářů je používán Zod. V době převzetí projektu však existovaly oblasti, kdy se tak nedělo a kontrola polí probíhala částečně i prostřednictvím HTML validace (např. pattern u input apod.). V rámci nové implementace byla validace zcela delegována na Zod.

U Zod validace nebyly prakticky nikde nastaveny české chybové hlášky. Ty tedy byly přidány. Všechny tyto hlášky lze v projektu nalézt v souboru `error-messages.ts`. Chybové hlášky se pak v Zod schémata vyskytují jako `message`, `required_error` nebo `invalid_type_error`. Pro představu si můžete prohlédnout v ukázce 3.4 celé Zod schéma ze souboru `workers.ts` ze složky `types/`.

```

1 import { z } from 'zod'
2 import { customErrorMessages as err } from 'lib/lang/error-messages'
3
4 export const WorkerCreateSchema = z
5   .object({
6     firstName: z
7       .string({ required_error: err.emptyFirstName })
8       .min(1, { message: err.emptyFirstName })
9       .trim(),
10    lastName: z
11      .string({ required_error: err.emptyLastName })
12      .min(1, { message: err.emptyLastName })
13      .trim(),
14    email: z
15      .string({ required_error: err.emptyEmail })
16      .min(1, { message: err.emptyEmail })
17      .email({ message: err.invalidEmail }),
18    phone: z
19      .string({ required_error: err.emptyPhone })
20      .min(1, { message: err.emptyPhone })
21      .refine(
22        phone =>
23          /^(?:\+|00)[0-9]{1,3})?[ ]?[0-9]{3}[ ]?[0-9]{3}[ ]?[0-9]{3}$
24          /.test(
25            phone
26          ),
27        {
28          message: err.invalidRegexPhone,
29        }
30      ),
31    ...
32  }).strict()

```

#### ■ Výpis kódu 3.4 Zod schéma pro vytváření pracovníka

Dále pak validace byla rozsáhle rozšířena. To zahrnuje rozšíření validace samostatných polí, kdy byl prakticky u všech zvětšen a zkvalitněn řetězec validování. Kromě toho se mezi toto rozšíření zařazuje i dříve neexistující validování mezi jednotlivými poli.

Validování mezi více poli bylo problematické sjednotit. Jde totiž o to, že schémata pro vytvoření (`create`) a upravení (`update`) jsou vždy oddělená. Avšak schéma pro upravení se často odkazuje na schéma vytvoření, do kterého přidává další pole pomocí metody `.merge()`. Prvotní nápad pro validaci mezi více poli zahrnovalo přidání metody `.refine()` nebo `.superRefine()` nakonec schématu vytvoření. To však vedlo na neočekávané chování, kdy na takové schéma nešly volat metody typu `.merge()` a podobně. Tato problematika je například rozebírána v rámci GitHub repozitáře pro Zod [47].

Tento problém byl vyřešen tak, že se přidalo ještě jedno základní (`basic`) schéma, které zahrnuje pole ze schématu vytvoření. Toto schéma se pak využije do schémat vytvoření a upravení. Na schémata vytvoření a upravení se pak aplikují metody `.refine()` nebo `.superRefine()` podle potřeby. To tedy bohužel vede na duplicitní kód, kterému se nelze snadno vyhnout, jak bylo popsáno výše. Vzniklé duplicity jsou proto v aplikaci nuceně tolerovány.

## 3.1.2 Fotografie

Nahrávání fotografií bylo vyhotoveno podle dříve popsaného návrhu 2.3. Samotná implementace však čelila několika problémům, které byly nutné při vývoji překonat. Fotografie se dříve používaly pouze u pracantů, kdy se každému při jeho úpravě dala nahrát jedna jediná. Dle požadavků bylo potřeba opravit stávající řešení a přidat například možnost nahrávání více fotografií u jobů.

### Vizuální stránka

Nahrávač fotografií nově zobrazuje nahrané fotografie se správným poměrem stran. Celková velikost se upravuje podle aktuální velikosti okna. Děje se tak díky CSS (viz 3.5) a nastavených stylů (viz 3.6).

```
1 .smj-photo-size {
2   height: 200px;
3   width: 200px;
4 }
5
6 @media (max-width: 768px) {
7   .smj-photo-size {
8     height: 150px;
9     width: 150px;
10  }
11 }
```

■ **Výpis kódu 3.5** CSS pro změnu velikosti okna nahrané fotografie. Výpis je ze souboru `custom.css`

```
1 <div
2   className="smj-photo-size"
3   style={{
4     position: 'relative',
5     cursor: 'zoom-in',
6   }}
7 >
8   <Image
9     style={{ objectFit: 'contain' }}
10    alt={`Fotografie ${index + 1}`}
11    src={url.url}
12    fill
13    sizes="(max-width: 768px) 100vw, (max-width: 1200px) 50vw, 33vw"
14    loading="eager"
15    priority
16    onClick={() => openPhotoModal(index)}
17    onMouseDown={e => {
18      // open image in new tab with middle mouse click
19      if (e.button === 1) {
20        window.open(url.url)
21      }
22    }}
23  />
24 </div>
```

■ **Výpis kódu 3.6** Zobrazení fotografie. Výpis je ze souboru `ImageUploader.tsx`

Na jednotlivé fotografie je možné kliknout, kdy se zobrazí modální okno s fotografií. Toto modální okno se dynamicky přizpůsobuje velikosti zařízení a fotografie uvnitř udržuje správný poměr stran. Řešení této skutečnosti je o něco složitější, než jak bylo popsáno výše, především kvůli neznámé nutné velikosti modálního okna, jinak je v principu problém řešený stejně. V rámci řešení se využívá odposlouchávání okna na událost `resize` získávající velikost okna prohlížeče. Tento údaj společně se získanými skutečnými velikostmi obrázku z komponenty `Image` je využit při výpočtu velikosti modálního okna.

Ze snímku obrazovky implementace je nahrání fotografie viditelné na obrázku 3.3.

## Validace fotografií na frontendu

Validace na frontendu se nově provádí přes `Zod`, tak jako u ostatních polí formuláře. Kontroluje se maximální velikost všech nahraných souborů, maximální počet a fakt zda se vůbec jedná o fotografii.

`Zod` validace se dá prohlásit až za druhou vlnu validace, prvním místem při validování je samotný nahrávač fotografií. Ten nevhodné fotografie vyřazuje ze samotného procesu nahrání. To tedy znamená, že pokud jakýkoliv soubor nevyhovuje jakémukoliv kritériu, tak tento soubor jednoduše přeskóčí a nebude nikdy evidován jako nahraný.

## Odesílání fotografií

Největší řešený problém představoval požadavek na to, aby se nahrané fotografie odesílaly se zbytkem formuláře. V době převzetí se dala nahrávat fotografie jenom při editaci pracanta. Tato fotografie se posílala zvlášť ve vlastním požadavku na koncový bod `/api/workers/worker_id/image`.

Data z formuláře se posílala přes komponentu `fetcher.ts` a data se odesílala v těle požadavku jako JSONvský string s nastavenou hlavičkou jako `application/json`, což k povaze dat z té doby bylo dostačující. Na druhou stranu požadavek s nahranou fotografií se odesílal v těle požadavku jako `FormData`. Posílání dvou různých požadavků bylo vyhodnoceno jako nevhodné převážně kvůli problematice sladění výsledků obou těchto požadavků a potenciální nekonzistenci dat.

Zvažovány byla dvě řešení, jak odeslat fotografie se zbytkem formuláře:

1. Data se budou odesílat jako `FormData`, který dokáže odesílat data typu `Blob`, `File` a `string`. Obrázky by se tedy odesílaly jako `File`. Zbytek polí z formuláře, by se odeslalo jako jeden `string`, který by obsahoval `JSON data`.
2. Data se budou odesílat jako `JSON string` s tím, že obrázky se budou muset převézt do `Base64`.

Po konzultaci s vedoucím této práce bylo rozhodnuto postupovat, tak jak je popsáno v 1. řešení.

Tato změna znamenala významné zásahy do všech koncových bodů na backendu. Dříve bylo snadné získat data přímo z požadavku (`req`) jako tělo tohoto požadavku (`req.body`), neboť `Next.js API Routes` provádí automatické parsování podle typu hlavičky požadavku. To s `FormData` nebylo možné. Proto se na všech koncových bodech musel zakázat automatický parser (viz 3.7) a bylo potřeba napsat si vlastní.

```

1 export const config = {
2   api: {
3     bodyParser: false,
4   },
5 }

```

### ■ Výpis kódu 3.7 Zakázání parseru

Vzhledem k tomu, že už nějaké nahrávání bylo zprovozněno u editovaného pracanta, byl již parser v nějaké podobě napsaný. V něm se využívalo řešení přes Formidable. To má nevýhodu v tom, že soubory rovnou nahrává na disk, ale vzhledem k použití nahrávání souborů v aplikaci, kdy na to mají právo jenom někteří lidé z týmu SummerJob nehrozí zahlcení disku a nebylo nutno tuto technologii nahrazovat. Mezi uvažované alternativní řešení patřil například Busboy neukládající soubory okamžitě na disk.

Každopádně tento parser byl přepsán, aby vyhovoval novým nárokům. Bylo zprovozněno nahrávání souborů i v development verzi běhu aplikace. Dále proběhly změny řešící pojmenování více nahraných souborů a k tomu přibyla také další validace. Konkrétně aby se hlídal maximální počet fotografií a celkový objem nahraných dat. Kromě toho byla přidána podpora pro převod JSON dat ve stringu z požadavku na samotný JSON, s kterým dále už koncové body mohly pracovat jako dříve.

## Místo nahrávání

Jak již bylo zmíněno, Formidable nahrává fotografie ihned. Je však možné určit místo nahrání a jak taková fotografie bude pojmenovaná. Pro Formidable se místo nahrávání získává z proměnné `UPLOAD_DIR` ze souboru s proměnnými prostředí `.env` nebo jako náhradní místo se použije složka `web-storage/` v projektu. Dále se pak fotografie nahraje do složky pojmenované jako identifikátor (`id`) příslušného ročníku SummerJob. Podle toho, pro jakou entitu se fotografie nahrává se určuje název další složky. Pokud je u dané entity možnost nahrávání více fotek další složka se pojmenovává jako `id` dané entity. Takže například nahraná fotografie pracanta se ukládá do složky `web-storage/{event_id}/workers/`. Pro job pak do složky `web-storage/{event_id}/jobs/{job_id}/`.

Název fotografie by měl být ve finále stanoven jako `id` této fotografie, pokud pro danou entitu je možné nahrávat více fotografií, nebo jako `id` samotné entity. Koncovka souboru, pak zůstává stejná jako u původního nahraného souboru. U parseru se název fotografie vytváří dočasný v případě, že není možno stanovit jméno fotografie kvůli neznalosti příslušného `id`, například v případě požadavku `POST`. U více fotografií je u dočasného názvu přidáno ještě pořadové číslo fotografie. Jednou nahrané fotografie od Formidable se pak podle potřeby přejmenují nebo posunou do jiných složek.

## Databáze

Pokud se k dané entitě dá nahrát jediná fotografie, tak se u ní ukládá prostě jenom string s relativní cestou k souboru. V případě nahrávání více fotografií, například u jobu, bylo nutné přidat entitu `JobPhoto`, která obsahuje `id` a relativní cestu k danému souboru.

## Mazání fotografií

Nahrávač fotografií na frontendu umí rozpoznat, které fotografie jsou čerstvě nahrané a které jsou získány z předešlých nahrání. Ty čerstvě nahrané jednoduše odstraní z připravovaného požadavku na koncový bod. Ty předešle nahrané musí označit jako smazané. V případě jediné fotografie tak může učinit podle jednoduchého booleanu. V případě více možných fotografií všechny takto smazané fotografie označí podle jejich `id`.

V koncovém bodě se tyto fotografie smažou ze svých uložených míst a v požadavku na databázi se odstraní ze záznamů jejich existence.

### 3.1.3 Nástěnka

Nástěnka představuje asi největší pro uživatele viditelný zásah do stávající aplikace. Přináší uživatelům aplikace možnost jednoduché komunikace prostřednictvím příspěvků.

## Oprávnění

Podle návrhu 2.2 musí mít přístup na správu nástěnky a psaní příspěvku pouze uživatelé s rolí ADMIN nebo POSTS. Role POSTS byla zavedena nově a její zaregistrování bylo jednoduše provedeno zapsáním do souboru `auth.ts` ke zbytku rolí.

U frontendové části aplikace se u přihlášeného uživatele kontroluje jeho role, pokud má povolený pokročilý přístup k nástěnce, může vytvářet a upravovat příspěvky, jinak mu je přístup na tyto stránky zamítnut. Kromě toho jsou mu k dispozici na úvodní stránce u jednotlivých příspěvků pokročilé funkce pro editaci, smazání a připnutí příspěvku. Bez pokročilého přístupu se mu zobrazuje jenom neinteraktivní ikona připnutí u připnutých příspěvků. O popisované zobrazení pokročilých funkcí u příspěvku se stará komponenta `PostBubbleActions.tsx`.

Po obdržení požadavku na koncový bod je pro jistotu znovu zkontrolována kredibilita uživatele odesílajícího tento požadavek. Tato druhá kontrola je nutná kvůli snaze o neporušení oddělitelnosti částí aplikace mezi frontendem a backendem. To do budoucna umožňuje snadnou výměnu libovolné této části. Kontrola rolí využívá již dříve vytvořené funkce ve stávající aplikaci.

## Vytvoření a úprava příspěvku

Vytvoření a úprava příspěvku probíhá přes formuláře. Tyto formuláře jsou lehce rozdílné a jsou od sebe oddělené. V aplikaci jsou obecně formuláře pro vytváření a úpravu vždy od sebe oddělené. Důvodem oddělení je převážně přehlednost a občasně drobné rozdíly, které by bylo složité pozorovat při snaze spojit tyto formuláře.

Pro vytváření (`CreatePost.tsx`) a úpravu (`EditPost.tsx`) byla použita nově vytvořená komponenta formuláře `Form.tsx` rozebíraná dříve (viz oddíl 3.1.1). Součástí vytvářeného a upravovaného obsahu jsou všechny pole zmíněné v rámci návrhu 2.3.6. V následujícím seznamu budou znovu připomenuty a bude u nich zmíněno pomocí jaké komponenty pro pole formuláře, vytvořené v rámci 3.1.1, byly vytvořeny:

- **Název:** `TextInput`
- **Krátký popis:** `TextAreaInput`
- **Dlouhý popis:** `TextAreaInput`
- **Platné pro dny:** `DateSelectionInput`
- **Čas:** `TimeInput`
- **Adresa a Souřadnice:** `MapInput`
- **Fotografie:** `ImageUploader`
- **Tagy:** `PillSelectInput`
- **Další vlastnosti:** `OtherAttributesInput`

Zmíněné názvy polí sice vycházejí z původního návrhu, ale některé byly později přejmenovány, aby lépe zapadaly se zbytkem aplikace.

Validace probíhá zcela přes Zod schéma. Po odeslání formuláře se vyplněná pole zkontrolují oproti Zod schématu a případně se odešle požadavek na koncový bod. Pro vytvářený příspěvek se jedná o požadavek typu `POST` na adresu `/api/posts/` a pro upravený pak o požadavek typu `PATCH` na adresu `/api/posts/{post_id}`. Tyto koncové body komunikují s databází, do které se přidá respektive upraví záznam do nově vytvořené respektive upravené entity příspěvku – `Post`.

## Hlavní stránka

Hlavní stránka nástěnky (komponenta `PostsClientPage.tsx`) obsahuje jednotlivé příspěvky, které z API získá přes metodu `GET` na adrese `/api/posts/`. Příspěvky jsou vizuálně rozřazeny do tří kategorií. Do první kategorie (`pinnedPosts`) spadají připnuté příspěvky, ty se na stránce zobrazují úplně nahoře a jsou vyjmuty z řazení a filtrování. Do druhé (`regularPosts`) a třetí (`timePosts`) kategorie spadají ostatní příspěvky (`otherPosts`). V druhé kategorii jsou zařazeny příspěvky, které nemají nastavený čas konání, ve třetí ty které mají. Časové příspěvky vytváří na stránce s příspěvků harmonogram.

Druhou a třetí kategorii zaštiťuje komponenta `PostType.tsx`, která představuje rozbalovací nabídku. Způsob rozřazení si lze prohlédnout ve výpisu kódu 3.8. Dříve nezmiňovaná proměnná `filteredData` představuje seřazené a vyfiltrované všechny nepřipnuté příspěvky.

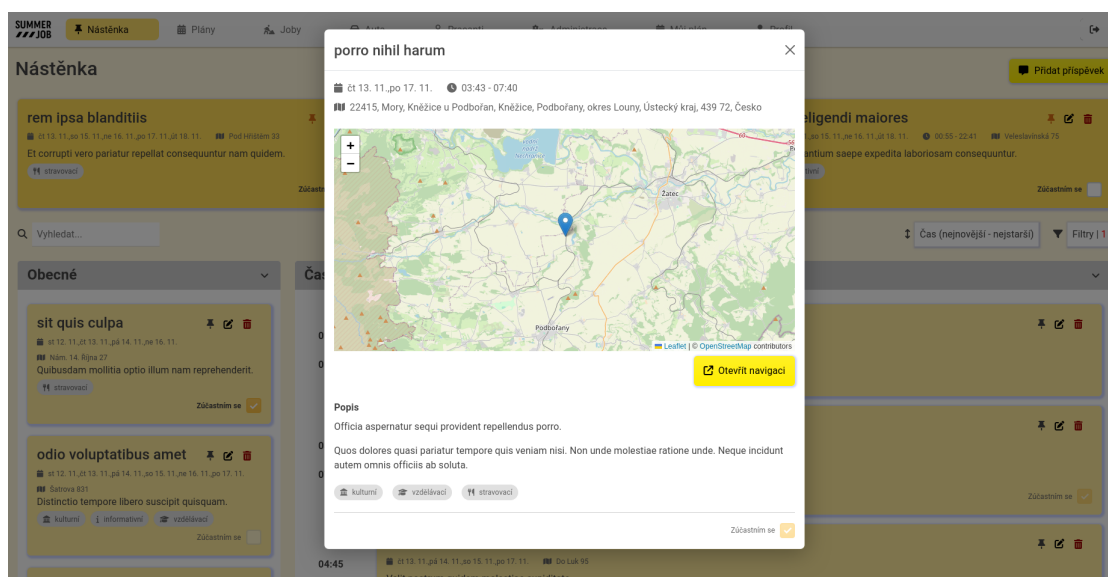
```
1 import React, { useEffect, useMemo, useState } from 'react'
2 const [pinnedPosts, otherPosts] = useMemo(() => {
3   const { pinned, other } = (data ?? [])
4     .map(item => deserializePostsDates(item))
5     .reduce(
6       (acc, post) => {
7         if (post.isPinned) {
8           acc.pinned.push(post)
9         } else {
10          acc.other.push(post)
11        }
12        return acc
13      },
14      { pinned: [], other: [] } as {
15        pinned: Array<PostComplete>
16        other: Array<PostComplete>
17      }
18    )
19   return [pinned, other]
20 }, [data])
21 ...
22 const [regularPosts, timePosts] = useMemo(() => {
23   const { regular, time } = filteredData.reduce(
24     (acc, post) => {
25       if (post.timeFrom) {
26         acc.time.push(post)
27       } else {
28         acc.regular.push(post)
29       }
30       return acc
31     },
32     { regular: [], time: [] } as {
33       regular: Array<PostComplete>
34       time: Array<PostComplete>
35     }
36   )
37   return [regular, time]
38 }, [filteredData])
```

■ **Výpis kódu 3.8** Rozřazení příspěvků. Výpis kódu vychází z komponenty `PostsClientPage.tsx`

Příspěvky jsou zobrazovány pomocí komponenty `PostBubble.tsx`, která u příspěvku zobrazuje jeho nejpodstatnější informace. Těmi jsou jeho název, datum, čas, adresa, krátký popis a tagy. Dále pak může zobrazovat dva prvky interaktivity, první představuje dříve zmíněné položky pro úpravu, smazání a připnutí příspěvku (`PostBubbleActions.tsx`), druhou pak možnost zápisu v případě speciálního označení příspěvku (`Participate.tsx`). Datum, čas a adresu se zobrazuje pomocí komponenty `PostAddressAndDateTime.tsx`, která se používá i v modálním okně příspěvku.

Modální okno (komponenta `PostModal.tsx`) příspěvku se zobrazuje po kliknutí na příspěvek. V něm se už vyskytují všechny evidované informace k příspěvku. Kromě toho může speciálně obsahovat i komponentu pro zápis – `Participate.tsx`.

Na obrázku 3.4 je možno si prohlédnout screenshot z implementace nástěnky. Konkrétně je na něm vyobrazen detail vybraného příspěvku.



■ **Obrázek 3.4** Screenshot detailu příspěvku z nástěnky z implementace.

## Připnutí příspěvku

Uživatel se speciální rolí může jakýkoliv příspěvek připnout. Po připnutí se pošle PATCH požadavek s boolean informací o připnutí, který tento příspěvek aktualizuje. Změna vyvolá znovu-vyžádání všech příspěvků a tím je tato změna pro uživatele ihned viditelná.

## Řazení, filtrování a vyhledávání

Řazení, filtrování a vyhledávání představují velkou část funkcionality na hlavní stránce nástěnky. Vyhledávání a filtrování se využívá prakticky u všech tabulek aplikace. Proto byla vytvořena komponenta `Filters.tsx` v rámci refaktORIZACE za účelem zpřehlednění kódu. Tato komponenta v sobě skrývá možnost vyhledávání a základního filtrování. U tabulek byla už v době převzetí jenom malá možnost tabulky podrobněji filtrovat a vylepšení této skutečnosti nebylo součástí požadavků, proto se u tabulek ponechalo toto řešení úpravně zmíněnou komponentou. U nástěnky však bylo potřeba vymyslet podrobnější filtrování podle více kritérií. Z komponenty `Filters.tsx` se proto u nástěnky používá jenom vyhledávání a filtry pro nástěnku byly napsány zvlášť.

Informace ohledně filtrování zobrazuje komponenta `FilterPostsBy.tsx`, která zobrazuje počet aktivních filtrů a po kliknutí na ní otevírá modální okno s filtry (`FilterPostsModal.tsx`). Modální okno filtrů funguje stejně jako formuláře jinde v aplikaci včetně validace oproti Zod schématu. Rozdílem je však to, že se po uložení filtrů neodesílá požadavek na koncový bod, ale nastavují se filtry do hlavní komponenty nástěnky – `PostsClientPage.tsx`.

Aktuální zvolené řazení zobrazuje komponenta `SortPostsBy.tsx`. Po kliknutí na tuto komponentu se zobrazí modální okno `SortPostsModal.tsx` s vlastním výběrem řazení. Vzhledem k nekomplikovanému způsobu výběru řazení, kdy je umožněno vybrat pouze jedno řazení, se výběr automaticky zpropaguje na nepřipnuté příspěvky. Oproti filtrům se tedy nemusí tato informace ztvrzovat. Další důvody byly vysvětleny v rámci návrhu (viz oddíl 2.3.6).



Na nepřipnuté příspěvky (`otherPosts`) na nástěnce se aplikuje vyhledávací text, zvolené řazení a filtry. Z toho vzniknou vyfiltrované příspěvky (`filteredData`), které se pak dále rozdělují na obecné (`regularPosts`) a časové (`timePosts`), jak bylo popsáno dříve.

Pro podstatu nástěnky bylo potřeba, aby její dřívější nastavení zůstávala i po navigaci na jiné stránky a zpět. Tohoto je dosaženo prostřednictvím query parametrů. Ty představují způsob zaznamenání textové informace do url adresy jako její nepovinné parametry. Řešení získávání a zápisu informací do url adresy si lze prohlédnout ve výpisu kódu 3.9.

```
1 import { useRouter, useSearchParams } from 'next/navigation'
2 import React, { useEffect, useMemo, useState } from 'react'
3
4 // get query parameters
5 const searchParams = useSearchParams()
6
7 // #region Search
8
9 const searchQ = searchParams?.get('search')
10 const [search, setSearch] = useState(searchQ ?? '')
11
12 // #endregion
13 ...
14 // replace url with new query parameters
15 const router = useRouter()
16 useEffect(() => {
17   router.replace(
18     `?${new URLSearchParams({
19       search: search,
20       days: selectedDays.map(d => d.id).join(';') ?? '',
21       sort: selectedSort.id,
22       participate: `${participate}`,
23       timeFrom: timeFrom === null ? '' : timeFrom,
24       timeTo: timeTo === null ? '' : timeTo,
25       tags: tags?.join(';') ?? '',
26       showAll: `${showAll}`,
27     })}`,
28     {
29       scroll: false,
30     }
31   )
32 }, [search, selectedSort, selectedDays, router, participate, timeFrom,
   timeTo, tags, showAll])
```

■ **Výpis kódu 3.9** Query parametry. Výpis kódu vychází z komponenty `PostsClientPage.tsx`

Konkrétně se do url adresy zaznamenává vyhledávací text (`search`), zvolené řazení (`sort`) všechny možnosti filtrování – zvolené dny (`days`), zobrazovat příspěvky bez ohledu na zvolené dny (`showAll`), zobrazovat jenom příspěvky, kterých se účastní (`participate`), odkdy (`timeFrom`) dokdy (`timeTo`) se dané příspěvky konají a zvolené tagy (`tags`).

Pro získání parametrů url adresy se využívá funkce `useSearchParams()`, pomocí níž se pak dá odvodit, které parametry jsou v url vyplněny. Kromě kontroly parametru, že je vyplněn, musí proběhnout ještě jeho validace a případný převod na požadovaný typ, vzhledem k faktu, že se do url adresy dají zaznamenávat pouze textové informace.

## Zapisování se na událost

Možnost zápisu u příspěvků je otevřena u všech příspěvků, které mají zvoleno pole „Otevřeno pro zapsání účastníky“. U příspěvků, které mají zvoleno (navíc) pole „Povinná účast pro všechny“, se možnost zápisu zobrazuje vždy zaškrtnutá a neinteraktivní. Ostatním příspěvkům se možnost zápisu nijak neukazuje. O zápis se v implementaci stará komponenta `Participate.tsx`.

Zmíněná komponenta v případě interaktivity s možností zápisu, odesílá požadavek typu `PATCH` na adresu `/api/posts/{post_id}` s informací o tom, který uživatel klikl na možnost zápisu a zda se jedná o zápis nebo odepsání z události příspěvku. Tento koncový bod při komunikaci s databází vytvoří nebo odebere záznam z nově vytvořeného schématu `Participant`. Záznam v této tabulce je identifikován pomocí id příspěvku (`postId`) a id pracanta (`workerId`).

## API

Nástěnka rozšířila stávající REST API. V rámci dřívějšího textu byly už některé koncové body zmíněny, tento pododdlíl však přinese shrnutí a přehledně tyto nové body vypíše do tabulky. Byly implementovány následující metody:

HTTP metoda	Adresa	Popis
GET	<code>/api/posts</code>	vrátí všechny příspěvky
POST	<code>/api/posts</code>	vytvoří nový příspěvek podle dat v těle požadavku
PATCH	<code>/api/posts/{post_id}</code>	upraví existující příspěvek podle dat v těle požadavku
DELETE	<code>/api/posts/{post_id}</code>	smaže existující příspěvek
GET	<code>/api/posts/{post_id}/photo</code>	vrátí uložený obrázek příspěvku <code>post_id</code>

■ **Tabulka 3.1** Rozšířené REST API o části k nástěnce

## 3.2 Implementace drobných požadavků

Oproti předcházející podkapitole jsou zde řešené požadavky svou implementací ne tolik rozsáhlé. Nebude zde kladen takový důraz na detaily, přesto zde budou popsány ty nejpodstatnější charakteristiky v řešení daného požadavku.

### 3.2.1 Tabulky

Největší problém ohledně tabulek bylo jejich zobrazení na počítači, kdy občas bylo nutné horizontálně scrollovat. Řešením bylo nastavení možnosti zalamování textu v těle tabulek. Na jednotlivé sloupce, pak byla individuálně aplikována jejich minimální velikost, aby sloupce nemohly být nikdy zalomené až příliš. Navíc bylo na poslední sloupce tabulek, obsahující akce k dané položce, aplikován styl, aby se vždy zobrazoval vpravo.

V převzatém projektu existovala komponenta `SortableTable.tsx` reprezentující tabulku s možným řazením dle sloupců. Tato komponenta nebyla použita na všechny tabulky v aplikaci, například u pracanta, aut a logů. U evidence logů byla jeho neexistence pochopitelná, vzhledem k podstatě jeho záznamů, ale u ostatních nikoliv. Proto tato komponenta byla použita univerzálně u všech tabulek vyjma logů. U všech sloupců tabulek využívající tuto komponentu, které to svou povahou dovolovaly, byla nastavená možnost řazení.

Dalším požadavkem týkající se tabulek byla stálost filtrů. Předtím však bylo potřeba, mít filtrování tabulek sjednocené. Jak již bylo zmíněno u popisu implementace nástěnky (viz oddíl 3.1.3), v rámci nové implementace byla vytvořena komponenta `Filters.tsx`. Ta zredukovala duplicitní kód k filtrování u každé tabulky, které začaly používat zmíněnou komponentu. Filtry pak byly pomocí textové reprezentace zaznamenány jako query parametry do url adresy. Tento proces byl popsán u zmíněného popisu implementace nástěnky.

Vyhledávání nově normalizuje uživatelský vstup a vyhledávané sekce – jsou odstraněny přebytečné mezery, diakritika a text je převeden na malá písmena. Filtry měly vždy efekt pouze na samotnou tabulku, ale nikdy ne na statistiky. To s novou implementací bylo změněno. Kromě toho byly statistiky napříč různými záložkami aplikace rozšířeny a zobrazují tak další informace navíc.

### 3.2.2 Plány

Jedním z požadavků týkající se plánu byl požadavek k zobrazení varování při přidávání jobů, kterému zbývá posledních pár dní k naplánování a je kritické jej ten den naplánovat. Toto varování, prostřednictvím nové komponenty `Issue.tsx`, bylo tedy přidáno do existující komponenty `AddJobToPlanForm.tsx`.

Komponenta `AddJobToPlanForm.tsx` využívá knihovnu `react-select` a konkrétně její komponentu `Select`. V rámci úprav byl upraven vzhled pro tento výběr, aby více odpovídal barvám akce. Navíc byla lehce upravena funkcionalita, aby se po výběru konkrétní položky nezavírala nabídka s položkami.

Dle dalšího z požadavků bylo u jednotlivých položek tabulky plánu přidáno tlačítko zaškrtnutí, které značí, zda konkrétní naplánovaná práce na daný den byla dokončena (tzn. pracanti se vrátili z jobu). Tlačítko slouží k manuálnímu zaškrťování, po kliknutí na něj se otevře popis pro administrátory k danému jobu, který si správce může dále upravit a až poté odeslat požadavek o splnění/nesplnění práce. Tlačítko je vydefinováno jako další sloupec tabulky, kde se tento sloupec může řadit podle hodnoty zaškrtnutí. Tohle je rozdílný přístup oproti možnosti zaškrtnutí u evidence celkové práce, kde je toto tlačítko součástí akcí a hotová práce je přesunuta do části hotových. Ani jedna z možností není zcela ideální a toto je konkrétně část, která bude doporučena na revizi v doporučených rozšíření (viz 5).

Změnu zaznamenal i obsah formuláře pro naplánované práce (`ActiveJobs`). Z něj byly odstraněny jeho vlastní popisy – „Popis“ a „Poznámky pro organizátory“, které byly nahrazeny popisy od navrhované práce (`proposedJob`) – „Popis navrhované práce“ a „Poznámka pro organizátory“. Kromě toho byla do něj přidána možnost změny názvu navrhovaného jobu.

V tabulce jobů pro plán se vyskytují tabulky pro pracanty, kteří jsou buď zařazení na práci, nebo čekají na přiřazení. Tyto tabulky byly lehce zpřehledněny, například přidáním patřičných ikon, pokud to bylo možné. Kromě toho byla u pracantů přidána informace o jejich dovednostech.

Pro jednotlivé práce v plánu byly přidány doplňující kontroly, které kontrolují, jestli někteří nahlášení pracanti na job již v minulosti spolu někdy pracovali, případně kde. Další kontrola sleduje, jestli pracanti na jobu někdy v minulosti pracovali na tomto konkrétním jobu a případně kdy. Obě tyto kontroly se dívají na data plánů předešlých dnů.

### 3.2.3 Joby

K jobům přibyla evidence náradí. Náradí jobu se dělí na ty na místě a ty, které je potřeba vzít s sebou. Na frontendu byl výběr vyřešen pomocí komponenty `PillSelect.tsx`, která byla zmíněna již dříve v rámci oddíl 3.1.1. Náradí je v rámci požadavku na backend pak zaevidováno do databáze do nově vytvořeného modelu `Tool`, který obsahuje id, množství a název nástroje. Název nástroje je typu `enum` se jménem `ToolName`. Tento název je na klientské části převeden na český název pomocí funkce `toolNameMapping`. Co se týče mazání nástrojů, tak to funguje podobně jako nahrávání fotografií pro práci, jak je bylo popsáno v oddílu 3.1.2.

Tabulka jobů již dříve obsahovala možnost připínání, ale toto připnutí bylo globální a viděli jej tak všichni organizátoři akce SummerJob s příslušnou rolí. Bylo rozhodnuto připínání udělat individuální. To se provádí pomocí databázového modelu `PinnedProposedJobByWorker`, ve kterém jsou vždy uloženy záznamy jako dvojice – id připnuté práce a id pracanta, který si tuto práci připnul. Tato informace je poté interpretovaná na frontendu a uživatelům se zobrazují správně jejich připnutí.

Kvůli nové evidenci fotografií byl přidán endpoint na získání fotografie – metoda `GET` na adresu `/api/proposed-jobs/{proposedJob_id}/photos/{photo_id}`.

### 3.2.4 Auta

Drobným požadavkem na opravu byla evidence najetých kilometrů. Chyba vznikala při zaevidování do databáze, kde byla tato položka vynechána. To tedy znamená, že informace se správně evidovala na klientovi, správně se poslala na koncový bod, ale při zapsání do databáze byla vypuštěna.

### 3.2.5 Pracanti

U pracantů byla přidána evidence jejich dovedností. Tato evidence je sice podobná té s nářadím, ale je ještě o něco jednodušší tím, že se eviduje jenom jméno této dovednosti. V databázi se tato skutečnost eviduje díky nově přidané položce u tabulky pracanta (`Worker`) – `skills`, která je typu enum jménem `Skill`. Takže podobně jako u nářadí, je tato informace na frontend převedena pomocí mapování, zde konkrétně pomocí `skillMapping`.

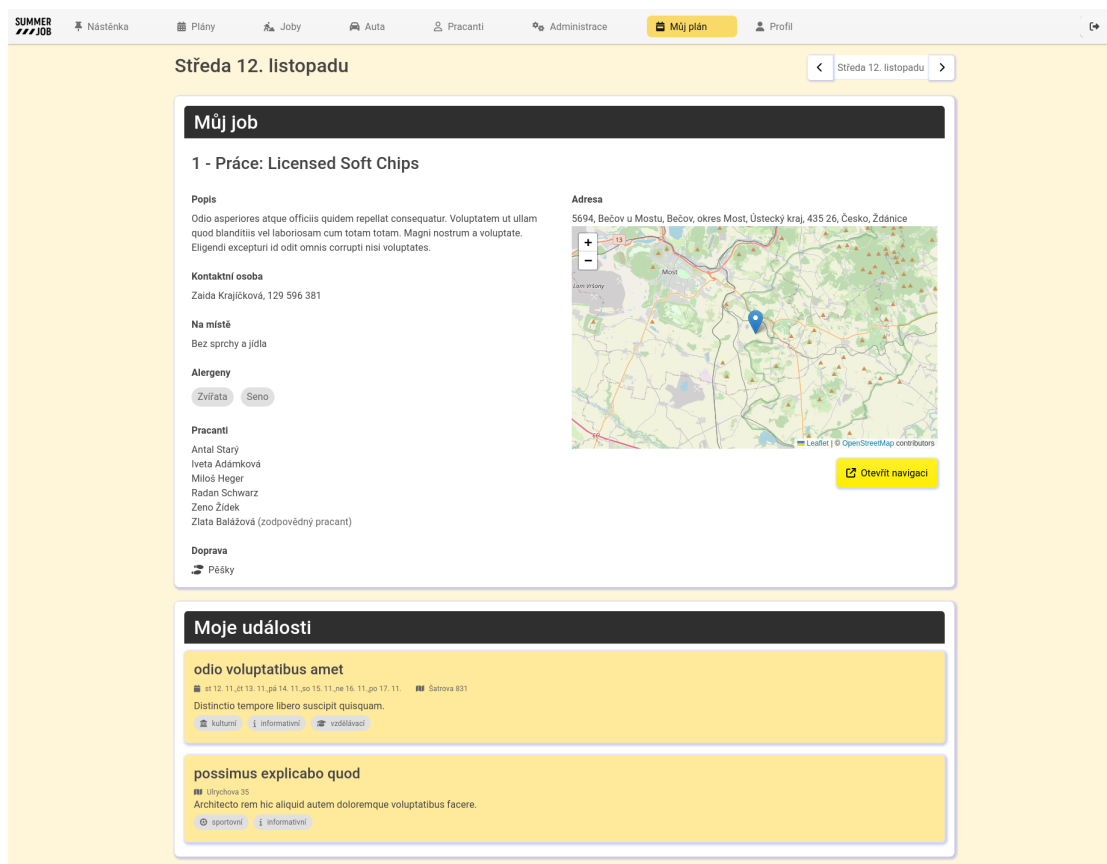
Dále byla u pracantů vypracovaná možnost evidovat jejich věk. Ten společně s dovednostmi je nyní možné nahrávat i při hromadném importu dat pracantů. Také se zobrazuje v informacích o pracantech v plánu.

Kvůli nahrávání fotografií, musely být u pracantů předělány koncové body. Hlavním problémem bylo spojené rozhraní pro vytvoření jednoho a více pracantů – data se posílala (`POST`) na adresu `/api/workers/`. Tyto endpointy byly odděleny. Na zde zmíněný endpoint se nyní posílají jenom data pro vytvoření více pracantů a na adresu `/api/workers/new` data vytvořeného (jediného) pracanta. Dále pak byla předělána adresa ohledně fotografií pracanta – `/api/workers/{worker_id}/photo`, z které byla odstraněna metoda `POST` kvůli toho, že se fotografie nyní eviduje se zbytkem dat.

### 3.2.6 Můj plán

Můj plán představuje úvodní stránku, úvodní místo, na které se každý uživatel po přihlášení do aplikace dostane. Jak bylo již zmíněno, stránka zobrazuje po dnech joby, kterých se má účastník zúčastnit, vzhledově je však stránka velice strohá. Proto byl upraven její vzhled, podle vytvořeného návrhu (viz oddíl 2.3.7).

Přepracovaný návrh respektuje všechny dříve existující informace a navíc dle jednoho z požadavků je zobrazenému jobu přidáno jeho pořadové číslo. Kromě toho byla dle návrhu přidaná sekce se všemi událostmi, kterých se pracant ten den účastní. Na závěr byla přepracována i navigace mezi jednotlivými dny, aby se vždy ukazoval aktuální den. Navíc je umožněno se navigovat napříč všemi dny ročníku SummerJob. Jak vypadá stránka „Můj plán“ po implementovaných změnách je k vidění na obrázku 3.5.



■ Obrázek 3.5 Snímek obrazovky mého plánu z implementace



## Kapitola 4

# Testování

Tato kapitola se zaměřuje na testování připraveného rozšíření aplikace. Bude zde popsáno testování rozšířeného API a statická analýza kódu. Kromě toho bylo provedeno uživatelské testování s malou skupinou lidí ověřující uživatelskou přívětivost při používání nové verze aplikace a ověření spokojenosti s proběhlými vizuálními a některými funkčními změnami v aplikaci. Na závěr této kapitoly proběhne diskuze o ověření splnění požadavků, jež byly stanoveny v rámci textu o analýze požadavků – podkapitola 1.4.

### 4.1 Technologie

V rámci analýzy současného řešení (viz podkapitola 1.2) byly představeny klíčové technologie používané v aplikaci. Tato podkapitola tento přehled lehce rozšíří a zaměří se na představení zvolených technologií na testování.

#### 4.1.1 SuperTest

SuperTest je Node.js knihovna na testování API. Je možno ji využívat samostatně nebo v kombinaci s dalšími testovacími frameworky. [48] V aplikaci je Supertest používán pouze pro posílání požadavků na příslušné endpointy a není využíván pro validování. Validování probíhá pomocí frameworku Mocha a knihovny Chai zmiňovaných dále.

#### 4.1.2 Mocha

Mocha je testovací framework na testování JavaScriptových aplikací. Podporuje asynchronní testování a umožňuje vytvářet testovací sady a testy. [49] V aplikaci je využit pro svůj přehledný zápis testů a odchyťávání vzniklých chyb, které správně mapuje k příslušným testům. O samotnou kontrolu se však stará až dále popsaný Chai.

#### 4.1.3 Chai

Chai [50] je testovací knihovna, která nabízí více stylů zápisů validování dat. Testy je možno psát pomocí řetězců ve stylu BDD nebo TDD. V aplikaci bylo zvoleno psaní stylem BDD, který umožňuje použití přirozeného jazyka, což má za výsledek zvýšení čitelnosti testů. Konkrétně se v kódu převážně využívá jeho varianta `should`.

### 4.1.4 CodeQL

CodeQL slouží k statické analýze kódů. Hledá v kódu známky bezpečnostních zranitelností, chyb a jiných problematických míst [51]. Je součástí automatizovaných kontrol v GitHub repozitáři projektu.

## 4.2 Testování API

V rámci projektu již existovaly testy na tehdejší podobu API. Tyto testy však musely být vinou vzniklých úprav všechny revidovány a případně celé přepsány. Velkou změnou byla výměna způsobu posílání dat na endpointy – místo jednoduchého JSON se nyní posílají data jako `FormData`. Změna tedy byla snadno promítnuta i do testů pomocí knihovny `Supertest`. Kromě těchto očividných změn byla většina testů zpřehledněna pomocí návodných komentářů a zlepšení jejich struktury. Pro jejich popis byl zvolen styl Given-When-Then, který se používá právě pro styl BDD.

Při rozvoji stávající aplikace byly přepsány některé endpointy, největší změny zaznamenaly endpointy týkající se pracantů (`workers`) a jobů (`proposed-jobs`). Jednalo se převážně o novou podporu nahrávání fotografií a evidence dovedností, respektive nástrojů. Nově napsané testy na toto téma objevily několik problémů s vytvořeným řešením nahrávání fotografií, které byly díky toho opraveny. Například se jednalo o chybějící mazání fotografií v případě smazání dané entity a špatná kontrola maximálního počtu nahrávaných fotografií.

Zcela nové endpointy byly především přidány kvůli nově vzniklé nástěnky a jejích příspěvků (`posts`). V rámci testů byla správa příspěvků celá zkontrolována. To tedy znamená endpointy byly zkontrolovány podle různých scénářů jejich očekávaného i neočekávaného použití.

V rámci výpisu kódu 4.1 je možno si prohlédnout příklad jednoho z vytvořených testů. Konkrétně tento test zkoumá správné vytvoření příspěvku s fotografií.

## 4.3 Statická analýza kódu

Celý projekt, včetně nově předělaných a vytvořených změn, byl v rámci kontrol na GitHubu zanalyzován pomocí CodeQL. Statická analýza poukázala ve výsledku na jediný problém, který se týkal přístupu k souborům. Problém (`Uncontrolled data used in path expression`) spočíval v používání neošetřeného uživatelského vstupu při sestavení adresy pro přístup k souborům nebo složce. Konkrétně se jednalo o id dané entity. Problém byl vyřešen s pomocí ověření pravosti daného id, respektive existence záznamu s daným id v databázi. Databáze tedy vrátila nalezený záznam včetně samotného id, kdy právě to bylo použito namísto původního id dodaného přímo od uživatele.

## 4.4 Testování uživateli

Testování uživateli bylo nasazeno až v pozdních fázích vývoje. Do té doby byly vytvářené změny konzultovány s týmem SummerJob, aby bylo zajištěno v první řadě splnění jejich požadavků, představ a nároků na finální podobu aplikace. Po splnění těchto představ organizátorů akce, bylo nutné získat další pohled na zvolené řešení.

Vzhledem k tomu, že většina běžných uživatelů využívající aplikaci nemá speciální role a mají tudíž přístup k minimu toho, co aplikace nabízí, nebylo potřeba využívat reálné účastníky, kteří by byli součástí předešlého ročníku, kde tato aplikace byla představena. Testeři jsou proto často lidé, kteří nejsou s akcí SummerJob obeznámeni. Proto jim tato akce a aplikace byla na začátku testování přiblížena, aby měli představu s čím je po nich vyžadováno pracovat. Celkově bylo zajištěno na 8 testerů.



```
1 describe('Posts', function () {
2   ...
3   describe('#photo', () => {
4     it('creates post with valid photo', async function () {
5       // given
6       const body = createPostData()
7       const filePath = `${__dirname}/resources/favicon.ico`
8       const numOfWorkFilesBef = await api.numberOfFilesInsideDirectory(
9         api.getUploadDirForImagesForCurrentEvent() + '/posts'
10      )
11      numOfWorkFilesBef.should.equal(0)
12      // when
13      const resp = await api.post('/api/posts', Id.POSTS, body, [
14        filePath])
15      // then
16      resp.status.should.equal(201)
17      resp.body.should.be.an('object')
18      // verify existence of photo path
19      resp.body.should.have.property('photoPath')
20      resp.body.photoPath.should.not.be.empty
21      const absolutePath = api.getAbsolutePath(resp.body.photoPath)
22      api.pathExists(absolutePath).should.equal(true)
23      // verify content by reading the image file
24      const fileStat = statSync(filePath)
25      const expectedSize = fileStat.size
26      const fileStatUploaded = statSync(absolutePath)
27      const uploadedSize = fileStatUploaded.size
28      expectedSize.should.equal(uploadedSize)
29      // verify naming of file
30      const { fileName, fileType } = getFileNameAndType(resp.body.
31        photoPath)
32      fileName.should.equal(resp.body.id)
33      fileType.should.equal('.ico')
34      // verify number of files in /posts folder
35      const numOfWorkFiles = await api.numberOfFilesInsideDirectory(
36        api.getUploadDirForImagesForCurrentEvent() + '/posts'
37      )
38      numOfWorkFiles.should.equal(1)
39    })
40    ...
41  })
42 }
```

■ **Výpis kódu 4.1** Testování vytváření příspěvku s fotografií. Výpis kódu vychází z `posts.spec.ts`

### 4.4.1 A/B testování

A/B testování je metoda založená na datech, při které se porovnávají dvě varianty prvku designu s cílem určit, která z nich funguje lépe. Každý A/B test se soustředí na určitý aspekt/prvek, který je středem optimalizace na webové stránce. Uživateli je představen buďto prvek v původní verzi A a nebo v upravené variantě B.

Zpravidla je A/B testování prováděno v reálném provozu, kde jsou uživatelé aplikace náhodně rozděleni a jsou zkoumány metriky jako je míra prokliku nebo konverzní poměr pro každou variantu. Náhodně vybrané skupiny zajišťují, že uživatelé reprezentují skutečné publikum, čímž se snižuje zkreslení výběru vzorku. Po uplynutí předem stanovené doby se dále analyzují sesbírané metriky pro variantu A a B, aby se posoudilo, která z nich si vedla lépe. Vítězná varianta je poté zpravidla rozšířena mezi všechny uživatele. [52]

Vzhledem k specifické době spuštění této webové aplikace a posléze malého vzorku testerů nebylo možné provést klasické A/B testování. Proto bylo přistoupeno k zjednodušenému modelu testování, které zahrnuje sbírání spíše menšího množství kvalitativních dat, než typicky velkého počtu metrických dat. To tedy znamená, že bylo k A/B testování částečně přistupováno jako k testování použitelnosti, které bude představeno v dalším oddílu.

Testeři byli rozděleni na poloviny do dvou testovacích skupin – skupiny A a B. Skupině A, kontrolní skupině, byla zobrazena stará verze aplikace, respektive ta verze, ze které proběhly změny vznikaly. Druhé skupině byla zobrazena nová verze aplikace.

Celkovým cílem testování bylo změření dopadu změn, a ověření toho, že vzniklé změny obecně nesou pozitivní výsledky. Každý test se zaměřuje na jinou oblast, která byla v rámci rozvoje přepracována. Je mu vytyčen cíl, podle toho čeho měla daná změna dosáhnout.

Každý uživatel měl po každém testu vyjádřit svou spokojenost s prvkem jež mu byl v testu představen prostřednictvím dotazníku. Uživatel mohl hodnotit od 1 do 5, kde jedna je nejmenší spokojenost a pět největší. Hodnotilo se v rámci 3 kategorií:

1. **Intuitivnost:** Zhodnoťte, jestli je pro vás design pochopitelný. *(škála od 1 – málo pochopitelný – do 5 – hodně pochopitelný)*
2. **Náročnost na používání:** Zhodnoťte náročnost používání daného prvku. *(škála od 1 – složité – do 5 – jednoduché)*
3. **Vzhled:** Zhodnoťte vzhled prvku. *(škála od 1 – nevzhledný – do 5 – hezký)*

Kromě číselného hodnocení měl každý uživatel prostor vyjádřit svůj názor textovým projevem. Dotazníky byly vytvořeny pro každou skupinu zvlášť a jejich podrobné výsledky je možno si prohlédnout v rámci přílohy B. Mimo samotné hodnocení uživatelů v rámci dotazníků bylo vždy vyhodnoceno splnění celkového cíle daného zlepšení prostřednictvím pozorování testerů.

### Testy

**AB1 – Výběr dostupnosti:** V rámci implementovaných změn byl přepracován výběr dostupnosti. Obsah vybíraných dnů byl zachován a byl přepracován do podoby kalendáře.

**Cíl:** Zrychlení orientace v rámci výběru.

**Postup:** Oběma skupinám byl ukázán výběr dostupnosti v rámci úpravy pracanta. Každý byl vyzván k interakci s tímto prvkem.

**Závěr:** Největší vylepšení komponenta výběru dostupnosti zaznamenala u svého vzhledu, kde uživatelé ze skupiny A průměrně vzhled hodnotili 1,5 body, kdežto uživatelé skupiny B 4,5 body. Několik testerů však poznamenalo, že by v tomto místě spíše čekali přímo použití kalendáře než jeho napodobeniny. Pozorováním testerů bylo prokázáno mírné zlepšení, co se týče rychlosti orientace v rámci výběru dostupnosti.

**AB2 – Vyhodnocování formuláře:** U povinných polí formulářů se nyní zobrazuje červená hvězdička, vyhodnocování je důslednější a v případě chyby se vypisují české hlášky.

**Cíl:** Zlepšení přehlednosti vyplňování formulářů.

**Postup:** Každému z obou skupin byl ukázán formulář pro vytvoření jobu. Testeři byli vyzváni k interakci, respektive vyplnění formuláře jobu s cílem jej schválně nevyplnit dobře.

**Závěr:** Ve výsledku bylo pozorováno zlepšení přehlednosti při práci s formulářem. Testerům skupiny B nečinilo vyplňování formuláře žádné problémy oproti těm ze skupiny A. V rámci dotazníku bylo ponecháno pole ohledně vzhledu, které je však ke zkoumané problematice spíše irelevantní.

**AB3 – Odeslání formuláře:** Tlačítka Zpět a Uložit byla vizuálně podbarvena a vždy se zobrazují vespod formuláře i při scrollování, dále bylo obecně pro modální okna přidána možnost je zavřít kliknutím mimo toto okno.

**Cíl:** Zrychlení vyplňování formulářů.

**Postup:** Uživatelům byl zobrazen jeden z nejdelších formulářů v aplikaci – úprava jobu. Měli za úkol si tento formulář prohlédnout a případně jej upravit a vyjádřit svůj názor na umístění spodních tlačítek.

**Závěr:** Testování prokázalo spokojenost uživatelů s novou polohou tlačítek a také bylo pozorováno signifikantní zrychlení vyplňování zkoumaného formuláře. Uživatelům se však nelíbilo použití modálního okna. Tato problematika je doporučena k dalšímu rozšíření v rámci diskuze 5.

**AB4 – Obsahová stránka jobu v mém plánu:** Náplň jobu v mého plánu byla v rámci změn pozměněna. Informace byly zachovány, ale byly graficky jinak zpracovány.

**Cíl:** Zpřehlednění předávaných informací uživateli.

**Postup:** Uživatelům byl zobrazen plán s detailem jobu, který poté zhodnotili.

**Závěr:** Z udělených bodů uživateli v dotaznících je jasné, že nový vzhled části zkoumané stránky zaznamenal velký úspěch. Uživatelé skupiny B se také v předaných informacích byli schopni snadněji orientovat než ti z A.

**AB5 – Responzivita tabulky:** Sloupce se nyní více zalomují. Poslední sloupec s akcemi je pro uživatele vždy viditelný i při vertikálním scrollování.

**Cíl:** Zjednodušená práce s tabulkami.

**Postup:** Doteď testování probíhalo na velkých obrazovkách, pro tento test byla velikost obrazovky upravena na velikost odpovídající středně velké obrazovce. Každému testerovi byla ukázána tabulka přehledu jobů.

**Závěr:** I přes relativně dobré hodnocení v dotazníku od skupiny B oproti skupině A, byla u skupiny B pozorována prakticky stejná nespokojenost jako u skupiny A. Hlavní příčina byla nemožnost si některé sloupce skrývat, tato funkcionality byla doporučena a v rámci kapitoly 5. Dodané změny byly B testery vnímány jako nutné minimum, jen z tohoto faktu se dá prohlásit, že jsou tyto změny pozitivní, ale mohly být i více signifikantní.

**AB6 – Statistiky:** Statistiky nyní reagují na vybrané filtry a vyhledávání.

**Cíl:** Zvýšení zájmu o statistiky.

**Postup:** Na stránce detailu plánu byly uživatelům ukázány statistiky. Měli poté hodnotit práci s těmito statistikami.

**Závěr:** Statistiky byly vnímány pozitivně již od začátku, přidaná funkcionality se však v rámci hodnocení projevila trochu vyšším skóre u části o intuitivnosti.

## Závěr

Testování se zúčastnilo celkově 8 lidí, kteří byli rovným dílem náhodně rozděleni do skupin A a B. Výsledky poukázaly na zlepšení ve všech zkoumaných aspektech. Avšak vzhledem k malému vzorku uživatelů a testování převážně jen na desktopové verzi aplikace se nedá s jistotou prohlásit, že jsou výsledky 100% průkazné. Minimálně však v pohledu na testovací skupiny bylo ověřeno, že vzniklé změny byly vnímány kladně.

### 4.4.2 Testování použitelnosti

Testování použitelnosti hodnotí, jak je produkt pro uživatele intuitivní. Poskytuje přímé údaje z pozorování, jak uživatelé s produktem nebo službou pracují. V rámci testování by měly být vytvořeny scénáře realisticky zvládnutelných úkolů, které se týkají klíčových otázek zkoumané problematiky. Úkoly by měly pokrývat kritické pracovní postupy a funkce.

Při moderovaném testu účastníci plní úkoly uvedené ve scénáři, zatímco pozorovatelé si dělají poznámky o výkonu, komentářích, nejasnostech a chybách. V rámci analýzy testování je cílem identifikování problémů s použitelností, jako jsou spleť pracovní postupy, nejasná sdělení nebo neintuitivní interakce. [52]

Testování se zúčastnili všichni účastníci z A/B testování. Všem testerům byly vysvětleny nutné skutečnosti vázané k zaměření jednotlivých testů. Každému testu byl vytvořen scénář na specifický průchod aplikací. Scénáře byly pro všechny stejné a testování probíhalo nad stejnými daty z databáze, tedy nad stejnou podobou aplikace. Uživatelé byly při jejich průchodech napříč scénáři pozorováni. Testování probíhalo vždy po jednom členovi testovací skupiny. Pozorované průchody měly za cíl ověřit design aplikace, převážně pak její přehlednost a jednoznačnost.

### Scénáře

Scénáře především cílily na ověření nově vytvořených změn, než na komplexní testování celkové aplikace jako takové. Scénáře se zaměřily převážně na vyplňování formulářů, orientaci v rámci jednotlivých stránek a interakci s interaktivními prvky aplikace. Scénáře jsou psány vágně, aby byl prostor pro uživatele vymyslet svůj způsob řešení. V případě, kdy by si tester nevěděl rady, tak je mu řešení naznačeno a tento zádrhel poznamenán. Uživatel může obecně po celou dobu testování komunikovat a vyjadřovat svůj názor.

Se všemi testery byl v průběhu testování vypracován doprovodný dotazník, který ke každému scénáři obsahuje tyto otázky:

1. Vyskytl se v rámci plnění scénáře jakýkoliv zádrhel?
2. Je  $x$  přehledná a jednoznačná? Rozveďte. ( $x$  zde značí konkrétní zaměření daného scénáře)
3. Máte nějaké návrhy na zlepšení?
4. Pokud máte ještě nějaké poznámky k tomuto scénáři, uveďte.

Odpovědi na dotazník jsou součástí přílohy C a budou sloužit pro stanovení závěrů k jednotlivým testům a celkovému závěru testování použitelnosti.

Před spuštěním testování byl každý scénář důkladně vyzkoušen i autorkou tohoto textu. V rámci toho bylo zjištěno několik nedostatků, které byly před testováním opraveny. U každého takového scénáře je tento fakt zmíněn v části o „Přípravě“. V textu této práce byly tyto změny již zahrnuty (např. v kapitole o implementaci 3).

Testovací scénáře byly použity tyto:

**T1 – Administrativní práce s nástěnkou:** Scénář ověřuje pochopení administrativní práce s příspěvkem na nástěnce. Tento komplexní scénář v sobě částečně zahrnuje popsané případy užití z analýzy požadavků – podkapitola 1.4. Konkrétně ty, kde aktérem byl Administrátor nebo Příspěvký. Jednotlivé případy užití jsou u daných bodů vyznačeny v závorce.

**Scénář:**

1. Navigujte se na stránku nástěnky.
2. Vytvořte nový příspěvek, nezapomeňte k němu přidat obrázek. (UC1)
3. Příspěvek upravte, přidejte/změňte čas konání. (UC2)
4. Tento příspěvek připněte. (UC7)
5. Příspěvek smažte. (UC9)

**Závěr:** Testerů byli obecně schopní i v případě jakékoliv, ve většině případů lidské, chyby se opravit s použitím nástrojů nástěnky. U jednoho z uživatelů nastalo to, že omylem smazal jiný příspěvek než bylo úkolem. U mazání se totiž zobrazuje potvrzující dialog jenom v případě, že příspěvek obsahuje účastníky. Toto chování bylo opraveno. Dalším problémem u jiného uživatele byla jeho rychlost, neboť se mu podařilo uložit formulář, ještě před nahráním fotografie. Vizuálně bylo patrné, že se fotografie nemohla ještě nahrát – nezobrazovala se v náhledu, proto tento fakt nebude součástí dalších úvah. Obecně si však uživatelé vedli velice dobře a byli se na stránce nástěnky schopni dobře zorientovat a využívat poskytnuté nástroje. Někteří však zmiňovali, že se jim místy nelíbí vizuální návrh, konkrétně například použité barvy a velikost příspěvků. Tato skutečnost však bude muset být ještě ověřena v rámci běhu tohoto ročníku.

**T2 – Uživatelská práce s nástěnkou:** Scénář ověřuje pochopení běžné uživatelské práce s příspěvkem na nástěnce. Tento komplexní scénář v sobě částečně zahrnuje popsané případy užití z analýzy požadavků – podkapitola 1.4. Konkrétně ty, kde aktérem byl Uživatel. Jednotlivé případy užití jsou u daných bodů vyznačeny v závorce.

**Předpoklad:** Stránka obsahuje různorodé příspěvky všech typů (připnuté, časové, obecné) na všechny dny akce, aby byl scénář rozumně proveditelný v každém svém kroku.

**Scénář:**

1. Navigujte se na stránku nástěnky.
2. Zobrazte si příspěvky týkající se 5. dne akce, obsahující tag „stravovací“. (UC3)
3. Příspěvky si seřaďte podle názvu abecedně. (UC4)
4. Vyhledejte s použitím nástrojů dodané stránkou příspěvky, který v názvu obsahuje řetězec „večeře“. (UC5)
5. O prvním výsledku v rámci časového harmonogramu si zobrazte podrobnosti. (UC8)
6. Zúčastněte se tohoto příspěvku. (UC6)

**Příprava:** U filtrování byly přidány možnosti nulovat všechny zvolené dny a čas, u dnů je nyní i možné stiskem tlačítka zvolit všechny možné dny. Byl také přidán filtr pro zobrazení všech příspěvků bez ohledu na zvolené dny, což má za důsledek zobrazení současně příspěvky bez i s vyplněným dnem konání. Kromě toho byla u filtrů opravena možnost mazání tagů, kdy toto mazání mělo dříve za účinek zavření modálního okna filtrování. Drobná úprava byla provedena i u modálního okna s řazením, kdy se nyní po vybrání možnosti toto okno automaticky zavře. Poslední důležitá oprava se týkala zapisování se na událost, kdy se u příspěvků občas špatně zobrazovalo jejich zapsání – uživatelem nebo s nastavenou povinnou účastí.

**Závěr:** Stejně jako u předešlého scénáře, někteří uživatelé dělali lidské chyby, které však byly schopni vyřešit. Nastaly, ale i situace, kdy návrh nebyl uživatelům zcela jasný. Konkrétně u modálního okna filtrování, uživatelé předpokládali, že se jimi zvolené filtry uloží, pokud opustí modální okno libovolným způsobem. To však není výchozí chování a po úvahách bylo rozhodnuto původní způsob ukládání ponechat. Další problém s návrhem měl uživatel, který nevěděl, jak si zobrazit podrobnosti o daném příspěvku. Jeho návrhem je přidání dalšího tlačítka, které na tuto možnost explicitně upozorňuje. Vzhledem však k už tak velké velikosti příspěvků bylo rozhodnuto toto nepřidávat. Ostatně jsou uživatelé na možnost kliknutí na příspěvek upozorněni změnou kurzoru. Kromě toho měli někteří uživatelé pěkné návrhy na vylepšení, konkrétně se například jednalo o zobrazování zvolených filtrů přímo na nástěnce. Některé návrhy byly proto navrženy na dodělání v kapitole 5.

**T3 – Orientace v mém plánu:** Scénář ověřuje orientaci v rámci stránky „Můj plán“.

**Předpoklad:** Uživateli se v jeho plánu zobrazují rozmanité plány a je tedy na většinu dní součástí jobů a sleduje mnoho událostí z příspěvků, aby bylo vyhověno všem krokům scénáře.

**Scénář:**

1. Navigujte se na stránku mého plánu.
2. Zjistěte jaké práce se první den akce účastníte, s kým budete pracovat, jak se na dané místo dopravíte a kde se konkrétně práce koná.
3. Podívejte se na svůj plán posledního dne akce.
4. Nalezněte událost, které jste se zúčastnil/a v rámci minulého scénáře a zobrazte si o ní doplňující informace.

**Příprava:** U výběru dnů byla nastavena jeho minimální fixní šířka, aby se šipky pro přesouvání mezi dny zobrazovaly vždy na stejné úrovni.

**Závěr:** Uživatelé neměli vesměs vůbec problém se orientovat napříč jednotlivými dny plánu ani v konkrétních informacích naplánovaného jobu. Informace byli schopní pochopit a nečinilo jim výraznější potíží najít událost, které se účastnili v předešlém scénáři. Nebylo jim jednoznačné, proč na stránce chyběli některé údaje, jako je například v kolik hodin a kde je místo vyzvednutí autem. Tohle konkrétně je však záležitost organizace akce, která nebyla součástí požadavků organizátorů na aplikaci. Celkově se dá nově přepracovaná stránka hodnotit pozitivně.

## Závěr

Testy neobjevily výraznější slabiny designu UI/UX či funkcionalit. Stejně jako u A/B testování, je však potřeba podotknout, že se jednalo o velice malý testovací vzorek, konkrétně o osm lidí. I když se v rámci testování neprojeví výraznější problémy, je potřeba mít na paměti ostrý provoz v rámci letošního ročníku. Až při něm se ukáže, jestli je design pro většinu účastníků dostatečně čitelný a pochopitelný. Přesto však nebyly provedené testy nazmar a díky nim byl design a funkcionalita ještě lehce upraveny a zdokonaleny.

## 4.5 Ověření splnění požadavků

V rámci podkapitoly 1.4 byly stanoveny funkční a nefunkční požadavky na rozšíření aplikace. V rámci této podkapitoly bude zde u každého požadavku uvedeno ověření a zhodnocení jeho splnění.

### 4.5.1 Funkční požadavky

- **F1 – Nástěnka:** Aplikace obsahuje nástěnku, kde mohou uživatelé se speciální rolí zveřejňovat příspěvky s aktuálními informacemi. Nástěnka je jinak přístupná všem uživatelům aplikace.
  - **F1.1 – Příspěvek:** Aplikace umožňuje evidenci příspěvků. U příspěvku se eviduje vše zmíněné v požadavku.
  - **F1.2 – Časová osa příspěvků:** Příspěvky se na frontendu rozdělují na časové a obecné.
  - **F1.3 – Připínání:** Připínání všech příspěvků je zpřístupněno uživatelům s rolí ADMIN nebo POSTS. Všichni uživatelé tyto příspěvky vidí vizuálně oddělené od ostatních. Na příspěvky se nevztahuje filtrování, řazení ani vyhledávání.
  - **F1.4 – Vyhledávání:** V rámci zobrazených příspěvků lze vyhledávat.
  - **F1.5 – Řazení:** Příspěvky lze řadit vždy podle jednoho kritéria. Řazení je vizuálně zpřístupněno pomocí modálního okna. Lze řadit podle jména (a – z, z – a), adresy (a – z, z – a), data (nový – starý, starý – nový) a času (nový – starý, starý – nový).
  - **F1.6 – Filtrování:** Příspěvky lze filtrovat podle různých kritérií. Filtrování je vizuálně zpřístupněno pomocí modálního okna. Lze filtrovat podle více filtrů zároveň. Mezi filtry patří časová dostupnost, čas od, čas do, tagy a informace o účasti a zobrazení si příspěvků bez ohledu na časovou dostupnost.
  - **F1.7 – Povinnost a účast:** U příspěvků byla rozšířená evidence, která dovoluje uživatelům starajícím se o nástěnku příspěvky speciálně označit – zda jsou povinné pro všechny uživatele a nebo jestli je událost otevřená pro zapsání. Příspěvky s výhradně otevřenou účastí dovolují uživatelům zápis. Zapsané příspěvky se správně evidují a společně s těmi povinnými se zobrazují uživatelům v jejich plánu.
- **F2 – Nářadí a dovednosti:** U každého jobu byla přidána možnost zaevidovat, jaké nářadí (a v jakém množství) je na místě a jaké nářadí bude potřeba vzít s sebou. U pracantů je evidence rozšířena o jejich dovednosti ohledně práce s nástroji. Tato rozšířená evidence obohacuje příslušné statistiky v aplikaci.
- **F3 – Tabulky:**
  - **F3.1 – Řazení dle sloupců:** Ve všech tabulkách napříč aplikací je nyní možnost řadit podle všech jejích sloupců s výjimkou posledního sloupce k akcím.
  - **F3.2 – Stálost filtrů:** Filtry u tabulek jsou stále a implementačně jsou ukládány jako query parametry url adresy dané stránky.
  - **F3.3 – Reakce statistik:** Statistiky reagují na aktuálně zvolené filtry.
- **F4 – Fotografie:**
  - **F4.1 – Podpora nahrávání více fotografií:** Nahrávač fotografií podporuje nahrávání vícero fotografií. Toho se v aplikaci využívá na jediném místě – u jobů.
  - **F4.2 – Mazání:** Nahrané fotografie je možné smazat a to jak ze záznamu dané entity, tak z disku.

- **F5 – Mapa:** Aplikace obsahuje komponenty pro správu mapy. V místě použití komponent je možnost si mapu zobrazit, případně na ní zvolit bod, který je převeden na adresu nebo naopak navolit adresu a podle ní najít místo na mapě.
- **F6 – Jednotný požadavek s daty z formuláře:** Vyplněné údaje z formuláře se odesílají na backend v rámci jediného požadavku.
- **F7 – Rozšířená kontrola plánů:** Na frontendu byly přidány kontroly podle požadavků.

## 4.5.2 Nefunkční požadavky

- **N1 – Responzivita:** U aplikace byl dodržen responzivní design.
- **N2 – Vzhled:** Grafika byla vylepšena a zmodernizována. Výsledkem je zvýšená přehlednost aplikace a vyšší míra důslednosti ve vyplňování informací.
- **N3 – Udržovatelnost a rozšiřitelnost:** Kód byl částečně refaktorován a jednotlivé části logicky rozčleněné do znovu použitelných komponent. Nové části kódu dodržují DRY princip. Složitější části aplikace byly okomentované. Dále pak byla rozšířena Swagger dokumentace.
- **N4 – Oprávnění pro psaní na nástěnku:** Pouze administrátoři (tzn. lidé s rolí ADMIN) a lidé s rolí POSTS mají oprávnění psát a spravovat příspěvky na nástěnce.
- **N5 – Fotografie**
  - **N5.1 – Množství a velikost nahrávaných fotografií:** Maximální velikost libovolné nahrané fotografie je omezena na 10MB. Množství nahrávaných fotografií je pro pracanta a příspěvek omezena na 1 a pro job na 10. Validace probíhá na frontendu i na backendu.
  - **N5.2 – Zobrazení nahrané fotografie:** Nahrané fotografie se zobrazují se svým skutečným poměrem stran.
  - **N5.3 – Místo nahrávání:** Fotografie se nahrávají do příslušných složek k dané entitě a ročníku. Tato struktura dovoluje v případě potřeby uvolnění disku pro neaktivní ročníky.
  - **N5.4 – Způsob zaznamenávání cesty k souboru:** Cesty k souborům se u příslušných entit ukládají pomocí relativní cesty k nim. Do aplikace byly dodány funkce pro získání absolutní cesty.
- **N6 – Vyhodnocování formulářů:** Vyhodnocování formulářů probíhá čistě prostřednictvím Zod.



# Zhodnocení a návrh dalšího rozšíření

V rámci analýzy byly zajištěny požadavky z mnoha zdrojů – z analýzy současného řešení, dotazníku GitHub Issues a návrhů na rozšíření z dříve zmiňované diplomové práce. Ne všechny požadavky byly zařazeny do celkové řešené problematiky, na kterou se tato práce zaměřila. Ty které však ano, byly zpracovány do funkčních a nefunkčních požadavků. K některým z těchto požadavků byl vytvořen doplňující návrh, ale jinak byly všechny v rámci implementace vypracovány. Jejich splnění bylo ověřeno v rámci testování.

Bylo zmíněno, že některé sesbírané požadavky nebyly zahrnuty do této práce. V rámci této kapitoly tyto požadavky budou připomenuty, přiblíženy a případně doplněny o krátký návrh na jejich splnění. Mimo tyto požadavky zde zazní i některé ještě dříve nezmiňované vlastní podněty a návrhy, které vyplývaly z uživatelského testování.

- **Nahradiť blokující okna:** Například po uložení formuláře se zobrazuje blokující modální okno s informací o ne/úspěchu odeslání. Toto modální okno zpomaluje workflow uživatele. Možným řešením by mohla být náhrada za toast zprávy.
- **Upozornění:** Aplikace by mohla obsahovat systém pro zaslání upozornění uživatelům. Zaslány by byly například upozornění ohledně přiřazení uživatele na novou práci, změny v plánu, změny v profilech pracantů a další. Notifikace by mohly také pomoci k důslednosti vyplňování informací účastníky, kdy by účastníci byli nabádáni k vyplnění svých časových možností na další den akce. Hromadné zprávy by také mohly být využity administrátory pro zaslání zpráv všem účastníkům akce.
- **Přihlašování:** S aktuální verzí přihlašování jsou vážnými problémy s občasnou nemožností se přihlásit na první pokus a obtížnost navigování se mezi emailovou schránkou a webovou aplikací. Přihlašování by bylo vhodné do budoucna rozšířit o možnost se přihlásit pomocí uživatelského jména a hesla případně pak přes služby jako je Google.
- **Rozšíření komunikace nástěnky:** Nástěnka představila možnost (převážně jednostranné) komunikace administrátorů s účastníky. Nástěnka by mohla být obohacena o možnost přidávat emoji reakce či krátké textové zprávy k jednotlivým příspěvkům uživateli.
- **Offline verze:** Možným rozšířením by představovalo přidání offline režimu aplikace s automatickou synchronizací po připojení k síti.
- **Rozšíření adorací:** Bylo by vhodné přidat konkrétní časy na adorování. Aktuálně je možno si zvolit adoraci pouze po dnech a další správa je proto řešena externě.

- **Kontakty:** V aplikaci momentálně chybí důležité kontakty, jako je například zdravotník.
- **Přístupnost webu:** Mohla by být vylepšena přístupnost webu („Web Accessibility“).
- **CSS:** Bylo by potřeba refaktorovat a sjednotit použité CSS styly v aplikaci. Aktuálně jsou CSS velice nepřehledně zpracovány v jediném souboru (`custom.ts`). Kromě toho bylo zasaženo přímo do souboru s bootstrap styly, který bude nutné celý aktualizovat a upravené styly přenést či vytvořit nanovo. Obecně je také potřeba vytvořit udržitelnou barevnou paletu a tím i připravit půdu pro případnou podporu světlého a tmavého režimu aplikace.
- **Vlastní tabulky a exporty:** Vhodné rozšíření představuje i přidání možnosti vytvářet si vlastní tabulky a provádět exporty. Minimálně by bylo vhodné dovolit uživateli přizpůsobit si zobrazovaný obsah tabulek pomocí vypínání/zapínání si některých sloupců. Aktuálně existuje možnost tisku tabulky pracantů a plánu, ale tento tisk nereaguje na zvolené filtry, to by bylo potřeba změnit. Tento tisk by bylo vhodné rozšířit i do dalších oblastí zájmu (např. u časového harmonogramu nástěnky), ale kromě toho by se hodilo přidat možnost si data stáhnout jako CSV soubor. S tím si pak správci budou moct provádět pokročilejší úpravy externě.
- **Rozšíření fotografií pro práce:** Nyní je možno sice nahrávat fotografie jobům, ale tyto fotografie nejsou dále využívány mimo interní účely týmu. Potenciálním rozšířením by proto bylo přidání možnosti výběru jedné prioritní fotografie, která by se poté zobrazovala uživatelům v rámci jejich denního plánu, pokud byli přiřazeni právě na tento job. Tento výběr by mohl být upřesněn i v rámci definovaného jobu z plánu (tzn. `active-job`). To by mělo za důsledek, že by se na každý den, co je daná práce vypsána, zobrazovala jiná fotografie.
- **Stálost řazení:** Podobně jako filtrování by bylo vhodné aby i zvolené řazení bylo stálé, například prostřednictvím query parametrů.
- **Filtry:** Nástěnka představila pokročilejší filtry, podobně by bylo vhodné filtry vypracovat i pro další tabulky. U nástěnky by se daly však filtry dále upravit a to například, tak aby se na hlavní stránce zobrazovaly v bublinách již navolené filtry.
- **Sbalovací sekce:** Sbalovací sekce v některých tabulkách (konkrétně např. u jobů) jsou občas nepřehledné a až úplně vespod stránky. Řešením může být jistá forma navigace po stránce, která by uživateli byla vždy viditelná.
- **Hotovo u jobů:** U `active-jobs` z plánu a `proposed-jobs` z přehledu jobů byl zvolen jiný postup pro označení si jobu jako hotový. Požadavkem je tuto možnost vizuálně sjednotit.
- **Navázání věcí k ročníku:** Jedná se o požadavek z GitHub Issues (issue 150), který byl v rámci implementace částečně splněný. Stále však některé části aplikace nejsou správně navázány na ročník. Například při znalosti url adresy na formulář s entitou navázanou na ročník `x` je možnost tuto entitu upravovat i když je právě zvolen jako aktivní ročník `y`.
- **Enum typy:** Aktuálně se typy pro alergie, nástroje a dovednosti musí upravit přímo v Prisma schématu a poté i v rámci jejich mapování na české názvy. Rozšíření by umožnilo tyto typy spravovat přímo v aplikaci v místě administrace uživatelů s rolí ADMIN.

# Závěr

Hlavním cílem této bakalářské práce bylo rozvinout aplikaci pro akci SummerJob. Z této aplikace byly odstraněny některé její neduhy, ale zároveň byly představené nové prvky a funkcionality. Vytvářené rozšíření především cílilo na usnadnění administrativní činnosti organizátorů, posílení komunikace mezi organizátory a účastníky. Kromě toho bylo cílem také zvýšení uživatelské přívětivosti a intuitivnosti aplikace, tedy celkového zlepšení UI/UX.

V rámci analýzy byla zanalyzovaná samotná akce, ale především současný stav aplikace, ze kterého tato práce vychází. Z analýzy současného řešení byly zjištěny první požadavky na nové rozšíření. Kromě toho byly požadavky zajištěny i z GitHub issues projektu a odpovědí na dotazník mířený na administrátory akce. Tyto požadavky byly dále analyzovány a rozděleny do funkčních a nefunkčních požadavků. Funkční požadavky týkající se nové stránky webové aplikace – nástěnky, byly zachyceny do modelu případu užití.

V návaznosti na analýzu byl proveden návrh realizace sesbíraných požadavků. V rámci návrhu byly zvoleny technologie pro nově přidanou komponentu mapy a předělanou komponentu nahrávání fotografií. Kvůli nově vytvářené sekce aplikace – nástěnky, byl proveden návrh na rozšíření uživatelských rolí. Součástí návrhů byly dále i návrhy obrazovek a prototypy vytvářené pomocí nástroje Figma. Ve finále byl proveden návrh konceptuálního modelu, který dále sloužil jako předloha při rozšiřování databázového schématu.

Navržené změny byly dále implementovány s použitím stávajících i nově zvolených technologií. Mezi hlavní implementované části patří již dříve zmíněná nástěnka, nahrávání fotografií a vylepšení formulářů. Kromě toho bylo provedeno mnoho dalších změn v rámci celé aplikace – frontendu, backendu a databáze. Mezi ně například patří částečná refaktorizace stávajícího kódu, aby byl obecně lépe udržovatelný, přehledný a dodržoval se DRY princip.

V neposlední řadě bylo provedeno testování aplikace, především zaměřené na nově přidané funkcionality a změny. Byly rozšířeny API testy, některé stávající byly přepsány a byla zlepšena jejich štábní kultura. Nad aplikací byla provedena statická analýza kódu pomocí nástroje CodeQL. Dále byla aplikace otestována uživateli. V první řadě bylo provedeno A/B testování zaměřené na vybrané přidané či upravené prvky aplikace. V druhé řadě byly testerům předloženy scénáře na základě kterých, byla zhodnocena úspěšnost změn jako celku.

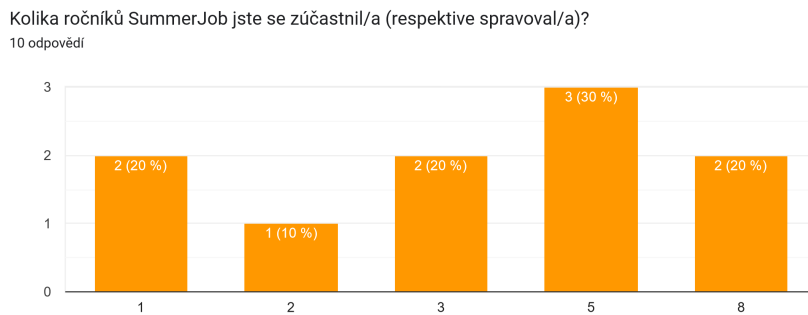
Celkově byly splněny všechny definované funkční a nefunkční požadavky, což bylo ověřeno v rámci testování. Finální verze aplikace se neodklonila od popsaného modelu případu užití a vytvořeného návrhu. Do budoucna však stále existuje velký prostor pro další vylepšení aplikace. Mezi ně samozřejmě patří, některé z požadavků zajištěné z dotazníku, Github Issues a vlastních pozorování současného stavu, které nebyly zařazeny do dílčích cílů této práce. Jmenovitě se jedná o přepracování přihlašování za pomoci emailu a hesla, přidání upozornění, rozšíření filtrování tabulek, přidání podpory si tabulky více upravovat a provádět exporty a dalších.

Ve finále byl splněn hlavní cíl této práce. Vytvořené změny usnadní administrativní činnost organizátorům akce a byl jim dán i prostředek pro komunikaci s účastníky v podobě nástěnky a obecně bylo zlepšeno UI/UX v rámci celé aplikace. Novinky věřím, přispějí nejen k efektivnímu chodu příštího, ale i následujících ročníků akce SummerJob.

# Dotazník pro administrátory

Drtivá většina otázek je položena formou volné odpovědi a všechny až na poslední jsou povinné. Poslední otázka dává prostor pro vyjádření jakýchkoliv dalších poznámek, které nebylo možno jinde v dotazníku vyjádřit. V odpovědích se očekává i jistá míra kreativity.

První otázka byla spíše na rozehrání a měla za cíl zjistit kolik ročníků se jako administrátoři akce zúčastnili. Výsledky jsou patrné z obrázku A.1.



■ **Obrázek A.1** Odpovědi na otázku „Kolika ročníků SummerJob jste se zúčastnil/a (respektive spravoval/a)?“

Další odpovědi na otázky jsou zpracovány do přehledných tabulek. Jsou zde uvedeny všechny odpovědi, které přinesly nový podnět a nebo byly svou náplní zajímavé. To znamená, že se zde vynechávají odpovědi typu: „ne“, „nemyslím si“, „nevybavuji si“ a podobně.

Na dotaz „Máte nějaké další poznámky? Pokud ano, podělte se.“ jsem nedostala už žádné návrhy na rozšíření stávajícího systému. Ale dostala jsem od některých respondentů pěkné zprávy. Tímto všem děkuji za vyplnění.

Otázka	Odpovědi
Jste si vědom/a nějakých chyb/bugů v systému? Případně uveďte.	<p>Chtělo by to časem více funkcí.</p> <p>Nefungovali nám kilometry, když jsme evidovali jízdy účastníků.</p> <p>Na iPhone se mi nedařilo přihlásit. Odkázalo mě to na můj mail, tam byl mail s odkazem na přihlášení a to mě znovu hodilo na napsání mailu a takto se to cyklilo, nepustilo mě to na můj účet.</p> <p>Při plánování jobů by to mělo fungovat nejlépe tak, aby dva lidé, kteří již spolu jeden den pracovali, se již žádný následující den nejlépe znovu na jobu nesetkali, ale na to bohužel systém neupozorňoval. Také by bylo fajn, kdyby systém hlídal, aby jeden člověk nešel na jedno místo pracovat dvakrát.</p>

■ **Tabulka A.1** Odpovědi na otázku „Jste si vědom/a nějakých chyb/bugů v systému? Případně uveďte.“

Otázka	Odpovědi
Myslíte si, že existují oblasti, kde by uživatelské rozhraní mohlo být intuitivnější? Pokud ano, uveďte konkrétně.	<p>Za mě je současná podoba vyhovující, rozdělení do záložek je přehledné. Líbila by se mi tmavší žlutá na pozadí, ale to se netýká intuitivnosti. :)</p> <p>Myslím, že design by mohl být více blbuvzdorný. :-)</p>

■ **Tabulka A.2** Odpovědi na otázku „Myslíte si, že existují oblasti, kde by uživatelské rozhraní mohlo být intuitivnější? Pokud ano, uveďte konkrétně.“

Otázka	Odpovědi
Můžete uvést příklady situací (pokud takové existují), ve kterých bylo používání současného systému náročné nebo/a neefektivní?	<p>Asi je to spíš o zvyku, přechody na nový systém jsou vždy náročné v takovém počtu lidí (například zajistit, aby všichni využívali současné funkce a byli důslední ve vyplňování informací).</p> <p>Z počátku při přihlášení. Dostal jsem zprávu na email a přes link se přihlásit, to nefungovalo a bylo složité překlikávat mezi odkazem a přihlašovací stránkou.</p> <p>Účastníci si občas zapomínali zapisovat nebo aktualizovat svoje možnosti práce a adorování. Nevím, zda je to technicky možné, ale pomohly by to zlepšit notifikace, kdy by účastníkovi každý den před deadline úprav přišlo upozornění, ať si zkontroluje, jestli má všechny údaje aktuální.</p> <p>Chtělo by to push notifikace, které upozorní, že „zbývá 5 minut na vyplnění formuláře práce na další den“ apod.</p> <p>Nedostatečná propojenost s databází.</p> <p>Někdy účastníci neměli přístup k internetu (neměli data) a jediná možnost byla kouknout se přímo na nástěnce nebo v notebooku na recepci.</p>

■ **Tabulka A.3** Odpovědi na otázku „Můžete uvést příklady situací (pokud takové existují), ve kterých bylo používání současného systému náročné nebo/a neefektivní?“

Otázka	Odpovědi
Existují nějaké konkrétní nedostatky či obtíže, se kterými se setkáváte při používání současného systému? Případně napište jakýkoliv návrh na zlepšení dané problematiky.	<p>Na iPhone jsem zadal mail na přihlášení, poté v mailu otevřel odkaz na přihlášení a neotevřel se mi jobsplanner, ale hodilo mě to zase na přihlášení se = zadání mailu.</p> <p>Možnost vytvářet si vlastní tabulky a provádět exporty (nejsem si jistá, jestli a komu ve správě to jde).</p> <p>Je super mít rozdělené bloky, ve kterých se dá jednoduše pracovat a to se celkem děje. Spíš bych ocenil modernější grafiku.</p> <p>U přihlašování na adorace by bylo super přidat i konkrétní časy adorování. Teď se musely zapisovat fyzicky na papír.</p> <p>Aby systém upozorňoval na to, pokud nějaký člověk již na daném místě pracoval nebo již s někým pracoval. (Aby každý den pracoval s někým jiným a na novém místě.)</p> <p>Napadá mě, že i jako člen týmu nevidím, kde kdo pracuje s kým, to asi vidí jen Jobteam..</p>

■ **Tabulka A.4** Odpovědi na otázku „Existují nějaké konkrétní nedostatky či obtíže, se kterými se setkáváte při používání současného systému? Případně napište jakýkoliv návrh na zlepšení dané problematiky.“

Otázka	Odpovědi
Musíte používat nějaké externí aplikace/zdroje na řešení jistých problematik, které byste preferoval/a, aby se daly řešit přímo v systému?	<p>Excelové tabulky, každá na jinou oblast zájmu (auta, pracovní náčiní, alergie, přítomnost na akci apod.).</p> <p>Zmiňovala jsem výše zapisování konkrétních časů adorací.</p> <p>Propojit lépe databázi.</p> <p>Nevím, jestli členové jedné skupinky na jobu mohou vidět možnosti ostatních členů týmu - to se asi řeší spíše ústně.</p>

■ **Tabulka A.5** Odpovědi na otázku „Musíte používat nějaké externí aplikace/zdroje na řešení jistých problematik, které byste preferoval/a, aby se daly řešit přímo v systému?“

Otázka	Odpovědi
Schází Vám v systému něco? Pokud ano, uveďte konkrétní návrhy na vylepšení nebo na nové funkce, které byste rád/a viděl/a implementované.	<p>Možnost společného chatu/nástěnky, kde by se zveřejňovaly aktuální informace, možnost vytvářet si soukromé tabulky.</p> <p>Hodila by se nástěnka, kde by byl uveden časový harmonogram akce a informace pro účastníky.</p> <p>Důležité kontakty – zdravotník, apod.</p> <p>Možná by bylo fajn, kdyby se adresy jednotlivých pracovišť daly prokliknout do map (google mapy, mapy.cz).</p> <p>Možná zadávání pomoci na recepci či baru.</p>

■ **Tabulka A.6** Odpovědi na otázku „Schází Vám v systému něco? Pokud ano, uveďte konkrétní návrhy na vylepšení nebo na nové funkce, které byste rád/a viděl/a implementované.“



## Dotazník pro A/B testování

V rámci A/B testování byly vytvořeny doprovodné dotazníky pro testery. Budou zde přehledně zpracovány jednotlivé odpovědi na všechny testované prvky, o nichž se dá více dozvědět v rámci uživatelského testování – 4.4.

Dotazníky byly vytvořeny dva odděleně pro každou testovací skupinu. Otázky jsou však v každém bodu totožné, proto i v rámci tabulek budou odpovědi zobrazeny vedle sebe s přihlédnutím ke které skupině patřily.

Otázka	Odpovědi – skupina A	Odpovědi – skupina B
Intuitivnost	3 3 4 5 <b>Průměr: 3,75</b>	2 4 5 5 <b>Průměr: 4</b>
Náročnost na používání	2 3 4 4 <b>Průměr: 3,25</b>	3 4 5 5 <b>Průměr: 4,25</b>
Vzhled	1 1 2 2 <b>Průměr: 1,5</b>	4 4 5 5 <b>Průměr: 4,5</b>
Poznámky	Čekal bych spíše kalendář. Asi je to drobnost, ale tlačítka nemají stejné podbarvení jako ta s alergiemi. Oproti ostatním polím je tohle vizuálně moc odlišné.	Vypadá to v pohodě, ale místo tohoto pseudo kalendáře bych tam dala skutečný. Přijde mi zbytečné tam mít tlačítka na dny, kdy se ta akce ani nekoná.

■ **Tabulka B.1** Zhodnocení prvku „AB1 – Výběr dostupnosti“

Otázka	Odpovědi – skupina A	Odpovědi – skupina B
Intuitivnost	2 2 3 3 <b>Průměr: 2,5</b>	3 4 4 5 <b>Průměr: 4</b>
Náročnost na používání	2 2 3 3 <b>Průměr: 2,5</b>	3 4 4 5 <b>Průměr: 4</b>
Vzhled	3 3 3 5 <b>Průměr: 3,25</b>	4 4 5 5 <b>Průměr: 4,5</b>
Poznámky	<p>Povinná pole by měla být asi nějak označená a nebo ta nepovinná.</p> <hr/> <p>Zkoušela jsem zadat záporné hodnoty u počtu dní na splnění a po odeslání mi to vypisovalo vzhledově odlišnou chybovou hlášku, než když jsem třeba třeba zkoušela tento údaj vůbec nevyplnit a to mi zobrazilo červenou chybu.</p> <hr/> <p>I když mi ta stránka nepsala, že by pole oblasti bylo nutné vyplnit tak mi to bez jeho zvolení nešlo uložit.</p>	<p>Ten formulář je takový moc roztahaný.</p> <hr/> <p>Nevím, jestli se mi úplně líbí, že jsou předvyplněná ta povinná pole s pracanty a počtem dní na splnění.</p>

■ **Tabulka B.2** Zhodnocení prvku „AB2 – Vyhodnocování formuláře“

Otázka	Odpovědi – skupina A	Odpovědi – skupina B
Intuitivnost	1 2 2 3 <b>Průměr: 2</b>	3 5 5 5 <b>Průměr: 4,5</b>
Náročnost na používání	1 1 3 3 <b>Průměr: 2</b>	3 5 5 5 <b>Průměr: 4,5</b>
Vzhled	1 2 2 4 <b>Průměr: 2,25</b>	3 3 4 5 <b>Průměr: 3,75</b>
Poznámky	<p>Je blbě scrollovat až dolů, aby to člověk mohl odeslat.</p> <p>Nelíbí se mi, jak jsou ta tlačítka úplně vespod, dala bych je spíše do záhlaví toho formuláře a možná by i mohly vypadat jinak, že by to například byly ikony místo textu, takže uložit by byla disketa a zpět šipka doleva.</p> <p>Nelíbilo se mi to okno po odeslání formuláře, čekala jsem že po úspěchu mě to pošle zpět na předešlou stránku rovnou, protože i když jsem to zavřela jakkoliv (křížek, pokračovat), tak mě to pak stejně na tu stránku poslalo, čekala bych že třeba ten křížek znamená že na té stránce zůstanu.</p>	<p>Nelíbí se mi to modální okno.</p> <p>Nechápu k čemu je tam to okno, když to stejně přesměrovává po kliknutí kdekoliv na hlavní stránku prací.</p> <p>Nelíbí se mi ty kulaté rohy u pozadí spodních tlačítek.</p>

■ **Tabulka B.3** Zhodnocení prvku „AB3 – Odeslání formuláře“

Otázka	Odpovědi – skupina A	Odpovědi – skupina B
Intuitivnost	2 4 4 5 <b>Průměr: 3,75</b>	4 5 5 5 <b>Průměr: 4,75</b>
Náročnost na používání	2 5 5 5 <b>Průměr: 4,25</b>	3 5 5 5 <b>Průměr: 4,5</b>
Vzhled	1 1 2 2 <b>Průměr: 1,5</b>	5 5 5 5 <b>Průměr: 5</b>
Poznámky	Jako dá se v tom vyznat, ale je to takové nemastné neslané. Obsahově nemůžu úplně soudit, co tam má být, ale vizuálně to nepůsobilo moc dobře, ale k tomu obsahu bych možná přidala telefony na ostatní členy ze skupiny a možná místo vyzvednutí k tomu autu. Je to takové o ničem.	Líbí se mi, hlavně ta mapa. Možná ten zodpovědný pracant by měl být první v tom seznamu? Vadilo mi, že se dalo v mapě oddalovat kolečkem, protože se pak hůř scrollovalo na stránce.

■ **Tabulka B.4** Zhodnocení prvku „AB4 – Obsahová stránka jobu v mém plánu“

Otázka	Odpovědi – skupina A	Odpovědi – skupina B
Intuitivnost	1 2 2 3 <b>Průměr: 2</b>	3 4 4 5 <b>Průměr: 4</b>
Náročnost na používání	1 1 1 1 <b>Průměr: 1</b>	3 3 3 4 <b>Průměr: 3,25</b>
Vzhled	1 2 3 4 <b>Průměr: 2,5</b>	4 5 5 5 <b>Průměr: 4,75</b>
Poznámky	Na telefonu si to nedokážu představit používat. Asi by se mi líbilo, kdyby se ty sloupce daly skrývat nebo aby se zalomovaly.	Chybí mi skrývání sloupců. Čekal bych, že organizátoři akce budou stejně spíše používat počítače, takže je to asi ok. Těch sloupců je strašně moc na používání třeba na tabletu nebo telefonu.

■ **Tabulka B.5** Zhodnocení prvku „AB5 – Responzivita tabulky“

Otázka	Odpovědi – skupina A	Odpovědi – skupina B
Intuitivnost	2 3 4 4 <b>Průměr: 3,25</b>	4 5 5 5 <b>Průměr: 4,75</b>
Náročnost na používání	4 5 5 5 <b>Průměr: 4,75</b>	5 5 5 5 <b>Průměr: 5</b>
Vzhled	5 5 5 5 <b>Průměr: 5</b>	5 5 5 5 <b>Průměr: 5</b>
Poznámky	Asi bych čekal, že to bude nějak reagovat na to filtrování. Asi detail, ale moc nerozumím proč je tam statistika k tomu, kolik účastníků je ještě bez práce, když ta informace je i dole v tabulce.	

■ **Tabulka B.6** Zhodnocení prvku „AB6 – Statistiky“

## Dotazník pro testování použitelnosti

Pro testování použitelnosti aplikace byl vytvořen doprovodný dotazník, který byl vyplněn s každým testerem. O tomto testování je možno se více dozvědět v rámci uživatelského testování – 4.4.

Z tabulek níže bude vypuštěna otázka „Pokud máte ještě nějaké poznámky k tomuto scénáři, uveďte.“, neboť na ni nebyly zaznamenány žádné odpovědi. Taktéž jsou vypuštěny některé odpovědi, které nepřinášejí nic konstruktivního (např. jednoslovné „ne“ apod.).

Vyplňování každé části formuláře probíhalo vždy po absolvování příslušného scénáře. První otázku vyplňovala zpravidla zadavatelka testu v průběhu testování. Ostatní otázky si častěji pak vyplňovali testeři sami.

Otázky	Odpovědi
Vyskytl se v rámci plnění scénáře jakýkoliv zádrhel?	<p>Tester byl ve vyplňování informací velice rychlý a povedlo se mu odeslat formulář, ještě před úplným nahráním fotografie.</p> <p>Uživatelka u vytvořeného příspěvku nevyplnila povinné pole „Krátký popis“, což pak sama opravila. Dále pak ne zvolila den konání, ale přes vyhledávání na stránce a vyplněného názvu, co si pamatovala příspěvek znovu našla.</p> <p>Při vytváření příspěvku byly vyplněny jenom povinné údaje a obrázek (tzn. název a krátký popis), to pak mělo za důsledek, že se příspěvek nezobrazoval na stránce při tehdy zapnutých filtrech. Bylo proto nutné použít filtrování s čímž si ale uživatel poradil bez obtíží.</p> <p>Při odstraňování příspěvku byl omylem smazán jiný.</p> <p>Uživatelka dlouho hledala, kde je její vytvořený příspěvek, ale přes hledání jej pak našla.</p> <p>Byl chvíli problém nalézt tlačítko na vytvoření příspěvku. Když se pak vyplňoval čas konání, tak byl vyplněn špatně (větší čas od než do), na to byl tester systémem upozorněn a byl schopný jej opravit.</p>
Je administrativní práce s nástěnkou přehledná a jednoznačná? Rozved'te.	<p>Ano, ale chyběl mi potvrzující dialog před smazáním příspěvku.</p> <p>Ty připnuté příspěvky zabírají dost místa a moc se mi nelíbí použité barvy, ale dá se v tom orientovat.</p> <p>Jo asi jo, možná se mi úplně nelíbí, jak jsou ty příspěvky rozdělené na obecné a časové, že ty sloupce nejsou stejně velké.</p> <p>Je to jednoznačné a přehledné, ale nelíbilo se mi, že na stránce formuláře při rolování jsem pak začal oddalovat místo toho mapu. Tu mapu bych zmenšil nebo zúžil.</p> <p>Při mazání přidat dialog.</p> <p>Po vytvoření, upravení, připnutí příspěvku by bylo dobré udělat to, že se stránka posune na daný prvek a ten se jakoby zvýrazní.</p> <p>Podle testera by měly být příspěvky na stránce o něco menší, aby více vyniklo tlačítko příspěvku.</p>
Máte nějaké návrhy na zlepšení?	<p>Aby tlačítko na mazání časů bylo blíž těm polím.</p> <p>K té mapě bych ještě dodal, že bych asi zmenšil to vyhledat na mapě, nebo ho nějak upravil, aby nebylo větší a výraznější než „uložit“.</p>

■ **Tabulka C.1** Zhodnocení scénáře „T1 – Administrativní práce s nástěnkou“



Otázky	Odpovědi
Vyskytl se v rámci plnění scénáře jakýkoliv zádrhel?	<p>Tester zavřel modální okno před tím než jej uložil, dle jeho slov bylo očekávaným chováním to, že se filtry automaticky uloží.</p> <p>Chvilku trvalo než se tester zorientoval a našel možnost se zúčastnit.</p> <p>Uživatel při řazení klikl na první A-Z položku, co viděl, což byla ale adresa. Chyby si všiml, tím že se příspěvky neseřadily, jak očekával.</p> <p>Uživatelka zavřela modální okno s filtrováním před uložením změn, takže je musela vyplňovat znovu.</p> <p>Při filtrování byl zvolený špatný tag, ale uživatelka se sama opravila.</p> <p>Uživatel byl chvíli zmaten, jak si zobrazit podrobnosti o příspěvku, což mohlo být i špatnou interpretací příslušného bodu scénáře. Uživateli bylo vysvětleno, že se nemyslí úprava tohoto příspěvku. Uživatel měl tendence hledat tlačítko s „i“, ale po nějaké době mu došlo, že na příspěvek může přímo kliknout.</p>
Je uživatelská práce s nástěnkou přehledná a jednoznačná? Rozved'te.	<p>Ano je, ale čekala jsem jiné chování po zavření okna s filtry.</p> <p>Asi bych čekala jiné nastavení počátečních filtrů. <i>(Pozn.: Základně byly zobrazeny všechny dny, neboť zrovna nebyl den v době konání akce.)</i></p> <p>Nelíbilo se mi, že to vyskakovací okno překrývalo ten vrchní panel.</p> <p>Bylo by vhodné u příspěvku přidat speciální tlačítko s „i“ jako informace pro zobrazení podrobností.</p>
Máte nějaké návrhy na zlepšení?	<p>Každý zvolený filtr by bylo fajn zobrazovat rovnou na hlavní stránce v bublinách s křížkem, pro rychlé mazání filtrů.</p> <p>Přijde mi, že chybí mezery u datumů u příspěvků.</p> <p>Možná by bylo pěkné kdyby ten časový harmonogram šel vytisknout. <i>(Pozn.: Uživateli byla před testováním aplikace ukázána a byla i zmíněna možnost tisku u pracantů.)</i></p> <p>Tlačítka u výběru dnů připadaly testerovi málo responzivní.</p>

■ **Tabulka C.2** Zhodnocení scénáře „T2 – Uživatelská práce s nástěnkou“

Otázky	Odpovědi
Vyskytl se v rámci plnění scénáře jakýkoliv zádrhel?	Ne úplně zádrhel, ale uživatel se ptal, kdy konkrétně je poslední den akce, ale přišel sám na to, že ho navigace sama zastaví. Uživatelce nebylo jasné, který den je poslední, bylo jí vysvětleno, že jí komponenta pro výběr data sama zastaví. Testerovi chvíli trvalo než se zorientoval se v informacích o jobu, ale dle jeho slov šlo spíše o malou znalost této akce než nepřehledný design.
Je orientace na stránce „Můj plán“ přehledná a jednoznačná? Rozved'te.	Je to přehledné, ale nechápu proč by se na stránce měly zobrazovat události, když ten den se nejde do práce. Ano, ale nechybí tam údaj ohledně toho, v kolik hodin a kde bude místo vyzvednutí? Všimla jsme si, že po kliknutí na připínáček na mapě se zobrazují informace navíc, možná by bylo fajn aby se ty informace zobrazovaly vždycky.
Máte nějaké návrhy na zlepšení?	Možná přidat i zde nějaké lehké filtrování, aby se ty události daly najít rychleji. Možná přidat kontakty na další pracanty?

■ **Tabulka C.3** Zhodnocení scénáře „T3 – Orientace v mém plánu“

# Bibliografie

1. *Začátek akce SummerJob 2022* [online]. Pontes z.ú., 2024 [cit. 2024-01-24]. Dostupné z: [https://summerjob.eu/wp-content/uploads/2023/07/TZ-uvodni\\_SummerJob-2022.docx.pdf](https://summerjob.eu/wp-content/uploads/2023/07/TZ-uvodni_SummerJob-2022.docx.pdf).
2. *SummerJob – tým* [online]. Pontes z.ú., 2024 [cit. 2024-04-03]. Dostupné z: <https://summerjob.eu/tym/>.
3. *Přihláška na SummerJob 2024* [online]. Pontes z.ú., 2024 [cit. 2024-04-05]. Dostupné z: <https://prihlaska.summerjob.eu/2024>.
4. *Začátek akce SummerJob 2019* [online]. Pontes z.ú., 2024 [cit. 2024-01-23]. Dostupné z: [https://summerjob.eu/wp-content/uploads/2022/05/TZ-uvodni\\_SummerJob-2019.pdf](https://summerjob.eu/wp-content/uploads/2022/05/TZ-uvodni_SummerJob-2019.pdf).
5. *SummerJob – média* [online]. Pontes z.ú., 2024 [cit. 2024-01-24]. Dostupné z: <https://summerjob.eu/media/>.
6. *SummerJob* [online]. Pontes z.ú., 2024 [cit. 2024-04-01]. Dostupné z: <https://summerjob.eu/>.
7. JEŠINA, Matyáš. *Návrh a implementace aplikace pro dobrovolnickou brigádu SummerJob*. 2024. Diplomová práce. České vysoké učení technické v Praze, Fakulta informačních technologií. Dostupný také z WWW: <https://github.com/jesinmat/diplomova-prace/>; Alternativně dostupný z WWW: <https://dspace.cvut.cz/bitstream/handle/10467/108925/F8-DP-2023-Jesina-Matyas-thesis.pdf?sequence=-1&isAllowed=y>.
8. *Výroční zpráva SummerJob 2019* [online]. Pontes z.ú., 2024 [cit. 2024-01-23]. Dostupné z: [https://summerjob.eu/wp-content/uploads/2022/05/V%C3%BDro%C4%8Dn%C3%AD-zpr%C3%A1va\\_2019\\_v02.pdf](https://summerjob.eu/wp-content/uploads/2022/05/V%C3%BDro%C4%8Dn%C3%AD-zpr%C3%A1va_2019_v02.pdf).
9. *Doplňující statistiky - 2023, 2024*. Tým SummerJob, 2024. Osobní komunikace, 24. ledna 2024.
10. *Informace o chodu akce*. Tým SummerJob, 2024. Osobní komunikace, 24. ledna 2024.
11. *Historie GitHub repozitáře pro SummerJob* [online]. GitHub, Inc., 2024 [cit. 2024-01-23]. Dostupné z: <https://github.com/ladal1/summerjob/network>.
12. *GitHub commit Add advanced search* [online]. GitHub, Inc., 2024 [cit. 2024-04-02]. Dostupné z: <https://github.com/AnetaPotomska/summerjob/commit/332dd26ef14214fae6022473998355faa26ab491>.
13. *Začátek akce SummerJob 2023* [online]. Pontes z.ú., 2024 [cit. 2024-01-24]. Dostupné z: <https://summerjob.eu/wp-content/uploads/2023/07/SummerJob-2023-uvodni%C3%AD-tiskov%C3%A1-zpr%C3%A1va.pdf>.

14. *GitHub repozitář v době commitu 'Add advanced search'* [online]. GitHub, Inc., 2024 [cit. 2024-04-02]. Dostupné z: <https://github.com/AnetaPotomska/summerjob/tree/332dd26ef14214fae6022473998355faa26ab491>.
15. SPARX SYSTEMS PTY LTD. *Enterprise Architect* [soft.]. 2024. Ver. 15.2 [cit. 2024-04-08]. Dostupné z: <https://sparxsystems.com/>.
16. *GitHub repozitář summerjob* [online]. GitHub, Inc., 2024 [cit. 2024-01-23]. Dostupné z: <https://github.com/ladali/summerjob/>.
17. THE POSTGRESQL GLOBAL DEVELOPMENT GROUP. *PostgreSQL* [soft.]. 2024. Ver. 15 [cit. 2024-04-05]. Dostupné z: <https://www.postgresql.org/>.
18. PRISMA DATA, INC. *Prisma* [soft.]. 2024. Ver. 5.10.2 [cit. 2024-04-24]. Dostupné z: <https://www.prisma.io/typescript>.
19. VERCEL. *Next.js* [soft.]. 2024. Ver. 13.5.6 [cit. 2024-04-24]. Dostupné z: <https://nextjs.org/>.
20. MCDONNELL, Colin. *Zod Documentation* [soft.]. 2024. Ver. 3.22.3 [cit. 2024-04-24]. Dostupné z: <https://zod.dev/>.
21. BOOTSTRAP. *Bootstrap* [soft.]. 2024. Ver. 5.2.3 [cit. 2024-04-24]. Dostupné z: <https://getbootstrap.com/>.
22. DOCKER INC. *Docker* [soft.]. 2024. [cit. 2024-04-24]. Dostupné z: <https://www.docker.com/>.
23. OPENJS FOUNDATION AND ESLINT CONTRIBUTORS. *Eslint* [soft.]. 2024. Ver. 8.30 [cit. 2024-04-24]. Dostupné z: <https://eslint.org/>.
24. OPENJS FOUNDATION AND ESLINT CONTRIBUTORS. *Prettier* [soft.]. 2024. Ver. 2.8.8 [cit. 2024-04-24]. Dostupné z: <https://prettier.io/>.
25. SMARTBEAR SOFTWARE. *Swagger UI React* [soft.]. 2024. Ver. 4.18 [cit. 2024-04-24]. Dostupné z: <https://swagger.io/>.
26. *README.md* [online]. GitHub, Inc., 2024 [cit. 2024-04-11]. Dostupné z: <https://github.com/AnetaPotomska/summerjob/blob/332dd26ef14214fae6022473998355faa26ab491/web/README.md>.
27. *What is REST?* [online]. Sitemap, 2024 [cit. 2024-05-07]. Dostupné z: <https://restfulapi.net/>.
28. *SummerJob issues* [online]. GitHub, Inc., 2024 [cit. 2024-02-07]. Dostupné z: <https://github.com/ladali/summerjob/issues/>.
29. *Maps JavaScript API cena* [online]. Google LLC, 2024 [cit. 2024-04-08]. Dostupné z: <https://mapsplatform.google.com/pricing/>.
30. *Jak se počítá načtení mapy na platformě Google Maps?* [online]. Google LLC, 2024 [cit. 2024-04-08]. Dostupné z: [https://developers.google.com/maps/faq#usage\\_mapload](https://developers.google.com/maps/faq#usage_mapload).
31. VOLODYMYR AGAFONKIN. *leaflet* [online]. 2024. [cit. 2024-04-09]. Dostupné z: <https://leafletjs.com/>.
32. NPM, INC. *leaflet* [soft.]. 2024. Ver. 1.9.4 [cit. 2024-04-09]. Dostupné z: <https://www.npmjs.com/package/leaflet>.
33. NPM, INC. *mapbox-gl* [soft.]. 2024. Ver. 3.2.0 [cit. 2024-04-09]. Dostupné z: <https://www.npmjs.com/package/mapbox-gl>.
34. MAPBOX. *Mapbox GL JS* [online]. 2024. [cit. 2024-04-09]. Dostupné z: <https://www.mapbox.com/mapbox-gljs>.
35. GITHUB, INC. *How to reduce bundle size of mapbox-gl* [online]. 2024. [cit. 2024-04-09]. Dostupné z: <https://github.com/mapbox/mapbox-gl-js/issues/6320>.

36. MAPBOX. *Mapbox GL JS pricing* [online]. 2024. [cit. 2024-04-09]. Dostupné z: <https://www.mapbox.com/pricing#map-loads-for-web>.
37. NPM, INC. *react-leaflet* [soft.]. 2024. Ver. 4.2.1 [cit. 2024-04-09]. Dostupné z: <https://www.npmjs.com/package/react-leaflet>.
38. SEZNAM.CZ, A.S. *Mapy.cz cena* [online]. 2024. [cit. 2024-04-09]. Dostupné z: <https://developer.mapy.cz/cena/>.
39. OPEN-SOURCE. *API Usage policy* [online]. 2024. [cit. 2024-04-09]. Dostupné z: <https://operations.osmfoundation.org/policies/>.
40. NETLIFY. *react-dropzone* [online]. 2024. [cit. 2024-04-09]. Dostupné z: <https://react-dropzone.js.org>.
41. RIK SCHENNINK PQINA. *filepond* [online]. 2024. [cit. 2024-04-09]. Dostupné z: <https://pqina.nl/filepond/#features>.
42. FIGMA. *Figma* [online]. 2024. [cit. 2024-04-09]. Dostupné z: <https://www.figma.com/>.
43. ČVUT FIT. *Data modeller* [soft.]. 2024. [cit. 2024-04-11]. Dostupné z: <https://dbs.fit.cvut.cz/dbsdm/>. Odkaz je chráněný ČVUT heslem.
44. *schema.prisma* [online]. GitHub, Inc., 2024 [cit. 2024-04-11]. Dostupné z: <https://github.com/AnetaPotomska/summerjob/blob/332dd26ef14214fae6022473998355faa26ab491/web/prisma/schema.prisma>.
45. *Lazy loading with no SSR* [online]. Vercel, Inc., 2024 [cit. 2024-04-11]. Dostupné z: <https://nextjs.org/docs/pages/building-your-application/optimizing/lazy-loading#with-no-ssr>.
46. *Why blur and focus doesn't work on Safari?* [online]. Stack Exchange Inc., 2024 [cit. 2024-04-11]. Dostupné z: <https://stackoverflow.com/questions/61245883/why-blur-and-focus-doesnt-work-on-safari>.
47. *ZodObject that are .refine or .superRefine returns a ZodEffects type* [online]. GitHub, Inc., 2024 [cit. 2024-04-15]. Dostupné z: <https://github.com/colinhacks/zod/issues/2474>.
48. NPM, INC. *supertest* [soft.]. 2024. Ver. 6.3.3 [cit. 2024-04-23]. Dostupné z: <https://www.npmjs.com/package/mocha>.
49. OPENJS FOUNDATION. *Mocha* [soft.]. 2024. Ver. 10.2.0 [cit. 2024-04-23]. Dostupné z: <https://mochajs.org/>.
50. CHAI ASSERTION LIBRARY. *Chai Assertion Library* [soft.]. 2024. Ver. 4.3.7 [cit. 2024-04-23]. Dostupné z: <https://www.chaijs.com/>.
51. GITHUB, INC. *CodeQL* [soft.]. 2024. [cit. 2024-04-23]. Dostupné z: <https://codeql.github.com/>.
52. *Usability Testing vs AB Testing: Which Is Right For You?* [online]. Reynoldston Business Services t/a AWA digital, 2024 [cit. 2024-05-07]. Dostupné z: <https://www.awa-digital.com/blog/usability-testing-vs-ab-testing/>.



# Obsah příloh

	readme.txt.....	stručný popis obsahu média
	src	
	impl.....	zdrojové kódy implementace
	thesis.....	zdrojová forma práce ve formátu L <sup>A</sup> T <sub>E</sub> X
	text.....	text práce
	thesis.pdf.....	text práce ve formátu PDF