



Zadání bakalářské práce

Název:	Webový provozní deník kompresoru
Student:	Martin Fried
Vedoucí:	doc. RNDr. Dušan Knop, Ph.D.
Studijní program:	Informatika
Obor / specializace:	Webové inženýrství 2021
Katedra:	Katedra softwarového inženýrství
Platnost zadání:	do konce letního semestru 2024/2025

Pokyny pro vypracování

Cílem práce je popsat problematiku vedení provozního deníku a implementace open-source webové aplikace pro vedení provozního deníku kompresoru, včetně nasazení pro potřeby JSDH Praha - Řepy.

V rámci deníku je nutné evidovat tlakové lahve včetně jejich technických údajů, jednotky SDH a k nim přiřazené lahve, záznamy o plnění včetně přesného času.

Cíle práce:

1. Proveďte rešerši aktuálního stavu.
2. Vypracujte analýzu a návrh webové aplikace řešící tuto problematiku.
3. Navrhněte jednoduché uživatelské rozhraní pro jednotlivé druhy uživatelů.
4. Implementujte výslednou webovou aplikaci a nasadte ve spolupráci se zadavatelem.
5. Aplikaci vhodně otestujte.

Funkční požadavky

1. Po instalaci bude aplikace nabízet jednoduchý a intuitivní formulář s jednotlivými kroky, které vytvoří a nastaví systém pro danou jednotku DH.
2. Funkčností se bude inspirovat, pokud to bude možné, systémem IKIS pro profesionální hasiče.

Nefunkční požadavky

1. Aplikace bude jednoduše přístupná a použitelná na mobilním zařízení.



2. Citlivé informace jako data doplňování lahví a informace o kompresoru bude možné zobrazit pouze po přihlášení do systému.
3. Nasazení nové instance aplikace bude přehledné. Postup bude dostupný v dokumentaci, popř. i v instruktážním videu.
4. Pro hostování bude potřeba minimální znalost informačních technologií jakou je například verzovací systém Git. Nebude potřeba schopnost číst a rozumět kódu, rozumět virtuálnímu hostování, AWS, Google Cloud atd.
5. Hostovat aplikaci bude finančně dostupné pro jednotky dobrovolných hasičů, tz. do 1000 Kč za rok.
6. Aplikace bude dostatečně flexibilní tak, že nebude tzv. šitá na míru nějaké společnosti nabízející hostování webových aplikací v ČR.



Bakalářská práce

WEBOVÝ PROVOZNÍ DENÍK KOMPRESORU

Martin Fried

Fakulta informačních technologií
Katedra softwarového inženýrství
Vedoucí: doc. RNDr. n Knop, Ph.D.
16. května 2024

České vysoké učení technické v Praze

Fakulta informačních technologií

© 2024 Martin Fried. 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: Fried Martin. *Webový provozní deník kompresoru*. Bakalářská práce. České vysoké učení technické v Praze, Fakulta informačních technologií, 2024.

Obsah

Poděkování	vii
Prohlášení	viii
Abstrakt	ix
Seznam zkratk	x
Úvod	1
1 Teoretická část	3
1.1 Dobrovolné hasičské sbory obecně	3
1.1.1 Kategorie Jednotek dobrovolných hasičských sborů . . .	5
1.1.2 Základní položky hasičského inventáře	5
1.1.3 Pražské sbory dobrovolných hasičů	7
1.1.4 Zdroje financí JPO	9
1.2 Povinnosti evidence předmětů	11
1.2.1 Důsledky na implementaci aplikace	12
1.2.2 Vzduchové kompresory	13
1.2.3 Tlakové lahve	14
1.2.4 Masky, nosiče, ostatní předměty	15
1.3 Plnění lahví	15
1.3.1 Průběh plnění lahve	16
1.3.2 Znalostní požadavky pro hasiče	17
1.4 Design uživatelského rozhraní	17
1.4.1 Požadavky na uživatelské rozhraní	17
1.4.2 Typický uživatel aplikace	19
1.4.3 Možnosti zobrazení	20
1.5 Existující alternativy aplikace	20
1.5.1 Internetový portál PORT.ALL	20
1.5.2 Aplikace profesionálních hasičů IKIS II	21
1.6 Open-source požadavky	23
1.6.1 Licence bakalářské práce	23
1.6.2 Open-source licence	24

2 Praktická část	25
2.1 Funkční a nefunkční požadavky	25
2.1.1 Funkční požadavky	25
2.1.2 Nefunkční požadavky	26
2.2 Případy užití	27
2.2.1 Role uživatelů	27
2.2.2 Případy užití	28
2.2.2.1 UC1: Úvodní přihlášení	28
2.2.2.2 UC2: Přihlášení	28
2.2.2.3 UC3: Úvodní formulář	29
2.2.2.4 UC4: Úvodní formulář v aplikaci bez záznamů uživatelů a prostředků	29
2.2.2.5 UC5: Přidání prostředku libovolného typu	29
2.2.2.6 UC6: Úprava prostředku	30
2.2.2.7 UC7: Smazání prostředku	31
2.2.2.8 UC8: Přidání záznamu o plnění lahve	31
2.2.2.9 UC9: Úprava záznamu o plnění lahve	32
2.2.2.10 UC10: Smazání plnění	32
2.2.2.11 UC11: Přidání uživatele	32
2.2.2.12 UC12: Úprava záznamu uživatele	33
2.2.2.13 UC13: Přepnutí tmavého a světlého režimu	33
2.2.2.14 UC14: Přidání jednotky	33
2.2.2.15 UC15: Úprava jednotky	34
2.2.2.16 UC16: Smazání jednotky	34
2.2.3 Diagramy aktivit	35
2.3 Volba technologií	35
2.3.1 Volba technologie pro perzistentní vrstvu	35
2.3.2 Důvody pro PHP	38
2.3.3 Framework Symfony	38
2.3.4 Knihovna Api-platform	39
2.3.5 JS framework	39
2.4 Diagram nasazení	40
2.5 Architektura perzistentní vrstvy	40
2.5.1 Doménový model	42
2.6 Logická vrstva	42
2.6.1 Definice entit	44
2.6.2 Role uživatelů	45
2.6.3 Povolené metody	45
2.6.4 Testovací data	46
2.7 Prezentační vrstva	46
2.7.1 Volba fontu	47
2.7.2 Použité knihovny	47
2.7.3 Struktura kódu aplikace	48
2.7.4 Struktura stránky aplikace	48

2.8	Testování aplikace	49
2.8.1	Testy logické vrstvy	49
2.8.2	Testy prezentační vrstvy	49
2.9	Uživatelské rozhraní	50
2.9.1	Základní rozdělení stránky	50
2.9.2	Vzdušnost stránky	51
2.9.3	Vyskakovací formulář	52
2.9.4	Postranní lišta	53
2.9.5	Použití barev a ikon	54
2.9.6	Optimalizace pro různé šířky displejů	55
2.9.7	Tmavý režim	55
2.9.8	Přístupnost aplikace	57
	Závěr	58
	A Detail oprav a kontrol TL	60
	B Roční provozní kontrola	61
	C Plánované kontroly	62
	D Současný způsob evidování stavů tlakových lahví	63
	E Snímek z finální verze aplikace	64
	Obsah příloh	69

Seznam obrázků

1.1	Zápisový sešit záznamů o plnění lahví, SDH Řepy, doc. RNDr. Dušan Knop, Ph.D.	18
1.2	PORT.ALL přidání prostředku, Modul Chemická služba [17]	21
1.3	Snímek z aplikace IKIS II, Petr Koleňák	22
2.1	Diagram případů užití, Enterprise architect, vlastní tvorba	27
2.2	Diagram aktivit – Správa sezení, Enterprise architect, vlastní tvorba	36
2.3	Diagram aktivit – Plnění lahve, Enterprise architect, vlastní tvorba .	37
2.4	Diagram nasazení aplikace na Apache httpd, Enterprise architect, vlastní tvorba	41
2.5	Doménový model perzistentní vrstvy, Enterprise architect, vlastní tvorba	43
2.6	Základní rozložení stránky, Enterprise architect, vlastní tvorba	51
2.7	Hlavní horní lišta aplikace, Provozní deník, vlastní tvorba	51
2.8	Postranní lišta s navigací, Provozní deník, vlastní tvorba	52
2.9	Vyskakovací okno s formulářem plnění láhve, Provozní deník, vlastní tvorba	53
A.1	Detail oprav, revizí a provozních kontrol ocelové tlakové lahve, IKIS II, Ing. Petr Koleňák	60
B.1	Roční provozní kontrola prostředku, IKIS II, Ing. Petr Koleňák	61
C.1	Plánované kontroly (upraveno), IKIS II, Ing. Petr Koleňák	62
D.1	Evidenční tabule, SDH Řepy, doc. RNDr. Dušan Knop, Ph.D.	63
E.1	Snímek pořízený z provozního deníku, vlastní tvorba	64

Seznam tabulek

1.1	Vnitřní organizace jednotky a kategorie jednotky, Vyhláška o organizaci a činnosti jednotek požární ochrany [6]	6
-----	---------------------------------------------------------------------------------------------------------------------------	---

1.2	Minimální rozsah OOPP hasiče, Řád výkonu služby v jednotkách hasičských záchranných sborů podniků, sborů dobrovolných hasičů obcí a sborů dobrovolných hasičů podniků [7]	8
1.3	Intervaly provádění údržby u vzduchových kompresorů, Chemická služba [12]	14

Seznam výpisů kódu

2.1	Zakázání změny hodnoty typu	44
2.2	Kontrola stavu uživatele	45
2.3	Anotace entity Plnění	46
2.4	Použití hodnoty wrap u stylu	55
2.5	Změna zdroje obrázku v HTML	56
2.6	Změna hodnot proměnných barev	56

Mé díky patří jak vedoucímu práce doc. RNDr. Dušanu Knopovi, Ph.D., který mi byl velkou oporou, tak i rodičům, především matce, která mi pomohla s editorskou prací této bakalářské práce. Chtěl bych také poděkovat všem lektorům a kamarádům, se kterými jsem mohl jakkoli konzultovat při procesu tvorby práce. Speciální díky pak patří Ing. Petru Kolečákovi, který mi poskytl snímky z aplikace IKIS II.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl 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 užit. Tyto osoby jsou oprávněny Dílo užit jakýmkoli způsobem, který nesnižuje hodnotu Díla a za jakýmkoli účelem (včetně užití k výdělečným účelům). Toto oprávnění je časově, teritoriálně i množstevně neomezené.

V Praze dne 16. května 2024

Abstrakt

Cílem bakalářské práce je vytvoření webového provozního deníku kompresoru. Tato webová aplikace nabízí evidenci kompresorů, tlakových lahví, plnění tlakových lahví a dalších prostředků jednotek sborů dobrovolných hasičů. Nad rámec nabízí také základní evidenci revizí prostředků, jejich umístění a použití hasiči. Jednotlivé záznamy jsou zabezpečeny podle rolí hasiče, který je povinen se do aplikace řádně přihlásit. Rozhraní aplikace je přístupné z mobilního zařízení. Pro její implementaci byly použity technologie nenáročné na běžné hostování. Aplikace je rozdělena na prezentační vrstvu napsanou v Reactu a logickou vrstvu napsanou v Symfony.

Klíčová slova provozní deník kompresoru, tlaková láhev, sbor dobrovolných hasičů, webová aplikace, uživatelská přívětivost, React, Symfony, PHP

Abstract

The aim of the bachelor thesis is to create a web-based compressor operation log. This web application offers a record of compressors, cylinders, cylinder fillings and other resources of volunteer fire brigade units. Beyond the framework, it also offers a basic record of asset revisions, their location and use by firefighters. The individual records are secured according to the role of the firefighter, who is required to properly log into the application. The application interface is accessible from a mobile device. For its implementation, technologies that do not require conventional hosting were used. The application is divided into a presentation layer written in React and a logical layer written in Symfony.

Keywords compressor operating log, pressure cylinder, voluntary fire brigade, web application, user friendliness, React, Symfony, PHP

Seznam zkratek

SDH	Sbor dobrovolných hasičů
JSDH	Jednotka sboru dobrovolných hasičů
JSDHO	Jednotka sboru dobrovolných hasičů obcí
PO	Požární ochrana
JPO	Jednotka požární ochrany
CAS	Cisternová automobilová stříkačka
OOPP	Osobní ochranné pracovní prostředky
HZS	Hasičský záchranný sbor
JS	Javascript
JWT	JSON Web Token
HTML	HyperText Markup Language

Úvod

Sbory dobrovolných hasičů se dělí na několik kategorií. Je zde například kategorie sboru, kde nejsou její členové nuceni vyjíždět na různé poplachy vůbec. Jedná se tak spíše o volnočasovou aktivitu či sport, kde dobrovolní hasiči cvičí na soutěže rozdělené také podle věku, jako tomu je u jiných sportů.

Jsou zde ale také sbory v takové kategorii, kde je časový limit na výjezd. Pro takové jednotky je vhodné, aby disponovaly lepší výbavou, jelikož tak zvýší hodnotu majetku, nebo i lidských životů, které ochrání.

Financování je ale vždy překážkou, která je správně, že existuje. U dobrovolných hasičů tomu není výjimkou. Každý sbor musí platit údržbu, nové prostředky, nebo svůj provoz. Často tak získat finance na dražší výbavu – jakou je například kompresor na vzduchové lahve – se povede jen některým sborům.

Tyto sbory musí evidovat různé údaje o provozu a použití své výbavy, na kterou dostaly finanční dotaci od veřejného subjektu. U kompresoru se měří hlavně takzvané motohodiny, které představují čas, který strávil kompresor plněním vzduchových lahví. Jelikož lahve jsou tlakované na vysoké hodnoty a zároveň dochází k zahřívání, plní se jednotlivé lahve dlouhou dobu. Po určité době musí být kompresor zkontrolován externím technikem na objednávku.

Aby hasič věděl, kdy musí poslat kompresor na technickou kontrolu, je nutné sečíst všechny motohodiny od poslední revize. Popř. je nutná revize také po určité době, pokud nebyl maximální počet motohodin naplněn. Dnes je praxí u dobrovolných hasičů evidovat všechny údaje písemně. Finančně dostupné nebo použitelné aplikace pro dobrovolné hasiče prakticky neexistují.

Tématem bakalářské práce je právě tvorba takové aplikace, která bude nabízet jednoduchou evidenci informací o předmětech, které daná jednotka spravuje. Zaměří se na evidenci plnění vzduchových lahví, a tedy i počtu motohodin a revizí kompresoru. Důležitým aspektem je jednoduchost aplikace, na instalaci a i samotné její používání. Instalace by měla být možná na klasický webový hosting, který není finančně náročný, jelikož si uvědomuji fakt, že sbory dobrovolných hasičů nedisponují financemi na dražší hostingové služby, které jsou nabízeny pro aplikace do komerční sféry. Samotný zdrojový kód

bude veřejně dostupný a zdarma, který bude nabízet jasnou dokumentaci pro instalaci, aby nebylo nutné zvláštní odbornosti v oblasti IT.

Webový provozní deník kompresoru má být webová aplikace, dostupná z veřejného internetu. Na fakultě informačních technologií studuji právě obor Webové inženýrství, proto jsem rád, že jsem si mohl zvolit toto téma. U aplikace je důležité, aby byla zaměřena na uživatelskou použitelnost a přehlednost, to jsou aspekty, které mi jsou blízké. Osobně nejsem členem sboru dobrovolných hasičů, ale někteří příbuzní jsou členy dobrovolných nebo i profesionálních hasičů.

Bakalářská práce je rozdělena na teoretickou a praktickou část. V teoretické části se věnuji tématu hasičských dobrovolných sborů. Povinností evidence jejich prostředků. Rozebírám také téma samotné aplikace resp. vysokotlaké vzduchové kompresory a plnění tlakových lahví. V praktické části popisuji, jak jsem dané téma vypracoval a problémy, které bylo nutné řešit. To znamená jeho požadavky a následnou implementaci. Na závěr popisuji design uživatelského rozhraní aplikace.

Teoretická část

1.1 Dobrovolné hasičské sbory obecně

Nejdříve je třeba si definovat rozdíl mezi Sbory dobrovolných hasičů a Jednotkami sboru dobrovolných hasičů. SDH neboli Sbory dobrovolných hasičů se liší především tím, že jsou právně samostatné organizace orientované na sport a kulturu hasičů. Jedná se tedy o organizace jako nejpočetnější Sdružení hasičů Čech, Moravy a Slezska [1] nebo občanské sdružení Moravská hasičská jednota [2].

Každá obec má ze zákona (Zákon č. 133/1985 Sb. [3]) povinnost zabezpečit Požární ochranu. Zákon vypisuje definice jednotek, které mohou tuto ochranu naplňovat.

Vzor smlouvy SHCMS obec [4].

Jednotkami požární ochrany jsou

1. jednotka hasičského záchranného sboru zřízená jako jednotka hasičského záchranného sboru kraje, jednotka generálního ředitelství nebo jednotka záchranného útvaru hasičského záchranného sboru, která je složena z příslušníků hasičského záchranného sboru (dále jen „příslušník“) určených k výkonu služby na stanicích hasičského záchranného sboru,
2. jednotka hasičského záchranného sboru podniku, která je složena ze zaměstnanců právnické osoby nebo podnikající fyzické osoby, kteří vykonávají činnost v této jednotce jako své zaměstnání, (dále jen „zaměstnanec podniku“),
3. jednotka sboru dobrovolných hasičů obce, která je složena z fyzických osob, které nevykonávají činnost v této jednotce požární ochrany jako své zaměstnání,

4. jednotka sboru dobrovolných hasičů podniku, která je složena ze zaměstnanců právnické osoby nebo podnikající fyzické osoby, kteří nevykonávají činnost v této jednotce požární ochrany jako své zaměstnání.

Když to tedy shrneme, existuje jednotka hasičského záchranného sboru kraje, sboru dobrovolných hasičů obce, hasičského záchranného sboru podniku a sboru dobrovolných hasičů podniku nebo také vojenská hasičská jednotka.

V rámci jednotek sboru dobrovolných hasičů jsou často převážně členové SDH. Podle velikosti, a tedy často typu dané jednotky, kterou doplňují členové místní organizace, lze uzavřít takzvanou smlouvu o vzájemné spolupráci SDH s obcí [4]. Dle jejího vzoru Příloha k zákonu České národní rady o požární ochraně [5] lze vyvodit hlavní právní vztahy.

Pomáhá při plnění úkolů na úseku PO zejména tím, že:

1. Pomáhá vyhledávat členy jednotek SDH obce.
 2. Podílí se na odborné přípravě členů jednotky SDH obce.
 3. Podílí se se souhlasem vlastníka na provádění údržby, oprav požární techniky, věcných prostředků PO a objektů PO, včetně vodních zdrojů. Při této činnosti dbá na dodržování ochrany životního prostředí.
 4. Podílí se na činnosti směřující předcházení požárům, zejména na preventivně výchovné činnosti mezi občany a mládeží.
 5. Podílí se na ediční a publikační činnosti a na dokumentaci historie PO a hasičstva.
 6. Spolupůsobí při zabezpečování úkolů ochrany obyvatelstva.
 7. Vychovává a připravuje mladé hasiče a dorost k dodržování zásad požární bezpečnosti, občanské bezúhonnosti a připravuje je k plnění úkolů PO a člena jednotky SDH obce.
 8. Pomáhá při přípravě dokumentace PO obce, zejména Požárního řádu obce, Řádu ohlašovacího požáru apod.
 9. Zabezpečuje odborně způsobilými osobami preventivní požární hlídku na akcích pořádaných se souhlasem obce jako jsou např. pořádání divadla, estrády, plesy a zábavy a jejich v prostorách.
 10. Plní další úkoly pomoci své obci na základě smluvního ujednání mezi obcí a SDH jako výraz projevu svobodné vůle a vlastního rozhodnutí.
-

Samotná smlouva se může samozřejmě lišit. SDH tedy často pomáhá obci při různých kulturních akcích, připravuje právní dokumenty nebo pomáhá s dostatečným doplněním počtu členů jednotky pomocí různých náborových akcí.

1.1.1 Kategorie Jednotek dobrovolných hasičských sborů

Je praktické hasičské jednotky kategorizovat tak, aby bylo jasné, s čím můžeme u dané jednotky počítat, nebo jak velkými prostředky disponuje. Samotná definice taky právně zavazuje jednotky. Lze je tak zpětně různě penalizovat například za pozdní výjezd, nebo pokud neplní jiné požadavky, jakými může být správa a používání technických prostředků. Podle Přílohy k zákonu č. 133/1985 Sb. [5] je definováno několik kategorií JPO I-V. Příloha nám také definuje územní a místní působnost. Kde JPO I až JPO III spadají do územní působnosti a zasahují tak i mimo území zřizovatele, zbylé spadají pouze do místní působnosti, tedy zasahují v katastrálním území dané obce, která je zřizovatelem této jednotky.

Kategorie právně definují časový limit na výjezd pro danou kategorii jednotky. Definuje také základní počet členů, nebo technickou připravenost, kterou musí daná jednotka splňovat.

Tabulka 1.1 uvádí vnitřní organizaci jednotky a tedy i požární techniku, kterou dané druhy JPO disponují. Pro vývoj aplikace je důležité, aby zahrnovala možnost evidovat takovou požární techniku a o takovém počtu, který je pro daný typ techniky očekávatelný.

JPO II zahrnuje jak hasiče z povolání na hlavní pracovní poměr, tak hasiče na vedlejší pracovní poměr. Zatímco JPO III a výše už jsou převážně hasiči na dobrovolné bázi. Je u nich také vyšší časový limit na výjezd. U JPO II je časový limit pouze 5 minut zatímco pro JPO III a výše je limit 10 minut.

Kategorie tedy nabízejí poměrně flexibilní volbu pro obec, tak aby si mohla zvolit podle dostupných počtu členů a zároveň podle jejich pracovní nasazenosti, která lze očekávat. Obec pak splní svou právní povinnost. Zároveň lze vyčíst, že velikosti jednotek nejsou nijak závratné a počty předmětů, které musejí jako povinné minimum vlastnit, také ne. V aplikaci může tento fakt ovlivnit způsob dotazování na data z databáze. Pokud by počty byly velké, pak by bylo vhodné odpovědi vracet stránkovaně. Pokud se neočekává, že počty záznamů nebudou přesahovat tisíce, pak pro tento způsob není důvod.

1.1.2 Základní položky hasičského inventáře

Podle Tabulky 1.1 je povinností jednotek požární ochrany, aby disponovaly několika vozidly. Zároveň je nutné, aby vlastnila jednotka dostatek izolačních dýchacích přístrojů, motorové stříkačky a radiové stanice. Často se setkáme také s tím, že JPO disponují několika vodními čerpadly, aby bylo možné při povodních pomáhat při vypouštění vody ze sklepů, nebo jiných míst. SDH

■ **Tabulka 1.1** Vnitřní organizace jednotky a kategorie jednotky, Vyhláška o organizaci a činnosti jednotek požární ochrany [6]

Vnitřní organizace jednotky	Kategorie jednotky				
	II/1	II/2	III/1	III/2	V
Druh jednotky JPO					
Celkem základní početní stav členů ⁵⁾	12	24	12	24	9
Počet členů v pohotovosti pro výjezd	4	8	4	8	4
Funkce					
Velitel	1	1	1	1	1
Velitel družstva	2	5	2	5	2
Strojník	3	6	4	6	2
Hasič, starší hasič	6	12	5	12	4
Požární technika a věcné prostředky požární ochrany³⁾					
Cisternová automobilová stříkačka v základním provedení (CAS)	1	1	1	1	1 ¹⁾
Dopravní automobil	1	1	1	1	1 ¹⁾
Automobilový žebřík do 30 m ²⁾	1 ²⁾	1 ²⁾	1	1 ²⁾	-
Automobilová plošina do 30 m	1 ²⁾	1 ²⁾	-	-	-
Odsavač kouře nebo přetlakový ventilátor	1 ¹⁾	1 ¹⁾	1 ¹⁾	1 ¹⁾	1 ¹⁾
Motorová stříkačka	1 ¹⁾	1 ¹⁾	1 ¹⁾	1 ¹⁾	1
Izolační dýchací přístroj ⁴⁾	4	8 ¹⁾	4	8 ¹⁾	4 ¹⁾
Vozidlová radiostanice požární ochrany	2	2	2	2	-
Přenosná radiostanice požární ochrany	2	4	2	4	1 ¹⁾
Mobilní telefon	1 ¹⁾	1 ¹⁾	1 ¹⁾	1 ¹⁾	1 ¹⁾

Poznámky:

1. JSDHO je vybavena uvedenou požární technikou, věcnými prostředky požární ochrany, jen pokud je to odůvodněno plošným pokrytím, dokumentací zdolávání požárů objektů, jejichž ochranu před požáry a mimořádnými událostmi jednotka zabezpečuje. Do počtu izolačních dýchacích přístrojů, přetlakových ventilátorů a motorových stříkaček se započítávají také prostředky umístěné v CAS nebo v jiné požární technice jednotky.
2. Vybavení jednotky sboru dobrovolných hasičů obce výškovou technikou se provádí podle obdobné zásady, na které je vybavena stanice typu P2 hasičského záchranného sboru kraje (příloha č.3 a č.6).
3. Minimální vybavení může být až o jedenapůlnásobek navýšeno, pokud je jednotka předurčena pro systém záchranných prací při dopravních nehodách, při haváriích, při živelních pohromách a pro ochranu obyvatelstva, nebo pokud je tak odůvodněno havarijním plánem kraje nebo dokumentací zdolávání požárů objektů, jejichž ochranu jednotka zabezpečuje.
4. Autonomní dýchací přístroj na stlačený vzduch s otevřeným okruhem, typ s plicní automatikou pro použití v plynném prostředí.
5. Základní početní stav členů jednotky sboru dobrovolných hasičů obce se zvyšuje o potřebný počet členů stanovený právním předpisem, je-li tato jednotka současně zařízením civilní ochrany.

Zličín také disponuje několika elektrocentrálami, párem plovoucích čerpadel anebo jedním kalovým čerpadlem. Aplikace bude nabízet evidenci těchto prostředků alespoň omezeně na revize a jejich použití. Jelikož se jedná o stejný typ předmětu, jakým je například vysokotlaký vzduchový kompresor.

Článek 9 Řádu výkonu služby JSDH pak také udává viz. Tabulka 1.2, s jakými osobními ochrannými pracovními prostředky (OOPP) každého hasiče musí jednotka vybavit.

Jedná se ale spíše o výčet pracovního oblečení, který by měly jednotky vlastnit, ale není příliš finančně nákladný, ani se v článku dále nespecifikují další nutné vlastnosti, kterými by měly kusy OOPP disponovat. Takový je pouze zmíněný zásahový oděv I a II, který musí splňovat hlavně ochranné vlastnosti, pro likvidaci požárů v otevřeném terénu nebo při jiných likvidačních činnostech.

Článek dále zmiňuje další doporučení nebo zákazy, jako například, že je zakázáno kombinovat civilní oblečení se zásahovými OOPP. Také zmiňuje, že je každá jednotka by měla disponovat takovou výbavou OOPP, podle svého zaměření jednotky. Takové kusy výbavy mohou být společné, je mezi nimi například protichemický ochranný oděv nebo respirátor.

V aplikaci by mělo být tedy možné vytvářet obecné kategorie pro předměty, které chce jednotka evidovat. Pokud by dostala finanční podporu pro nějaký kus OOPP, který je například více finančně nákladný, pak je vhodné, aby šlo o takový předmět, který lze filtrovat podle kategorie. Jako příklad bych uvedl pracovní stejnokroj, který pokud je nechán ušít na míru, lze očekávat, že se cena bude pohybovat v několika tisících Kč. Některé finanční dotace vyžadují, aby byly tyto prostředky používány a jejich použití zaevidováno, proto by měla aplikace také nabízet evidenci použití členem sboru.

1.1.3 Pražské sbory dobrovolných hasičů

Chtěl bych uvést pro kontext aktuální stav k roku 2024 v Praze. Zde se nachází podle portálu Praha.eu [8] celkem 36 JSDH JPO III. Mezi které patří podle zdroje: Běchovice, Březiněves, Dolní Měcholupy, Horní Měcholupy, Chodov, Cholupice, Klánovice, Kolovraty, Kunratice, Letňany, Lipence, Lysolaje, Písnice, Radotín, Řeporyje, Řepy, Satalice, Stodůlky, Suchdol, Zbraslav, Zličín, Újezd nad Lesy, Ďáblice, Nebušice. K JPO V pak Benice, Čakovice, Dubeč, Kbely, Koloděje, Libuš, Lochkov, Újezd, Praha 1, Třebonice, Újezd nad Lesy, Velká Chuchle a Praha 8. Ještě pro kontext v České republice podle výroční zprávy SH ČMS za rok 2022 [9] bylo spolupracujících SDH s JSDHO (jednotka sboru dobrovolných hasičů obce) celkem 6290, takže samotná Praha zahrnuje pouze maximálně půl procenta celkového počtu jednotek JPO III a JPO V. Jako zajímavost uvedu ještě, že existují JPO IV, které představují podnikové JSDH. Takovými jsou například dílny DPP.

Podle stránek Sboru dobrovolných hasičů Praha – Zličín [10] těchto sborů pak například JPO III Zličín disponuje vozidly CAS 30 Scania P440 6x6, CAS

■ **Tabulka 1.2** Minimální rozsah OOPP hasiče, Řád výkonu služby v jednotkách hasičských záchranných sborů podniků, sborů dobrovolných hasičů obcí a sborů dobrovolných hasičů podniků [7]

Druh os. ochranného pracovního prostředku	Počet na hasiče
Pracovní stejnokroj II (PS II) blůza	1
Pracovní stejnokroj II (PS II) kalhoty	1 (2) ^I
Čepice k pracovnímu stejnokroji II	1
Zásahový oděv I (kabát, kalhoty)	1 ^{III}
Zásahový oděv II (blůza, kalhoty)	1 ^{II}
Přilba pro hasiče	1
Obuv pro hasiče	1
Ochranné rukavice pro hasiče (zásahové rukavice)	1
Ochranné rukavice proti mechanickým rizikům	1
Kukla pro hasiče	1 ^{III}
Hasičský opasek s karabinou s minimální pevností 22 kN v podélném směru se sekyrou	1 ^{IV}
Svítilna	1 ^{VI}
Spodní prádlo pro hasiče - triko	2 ^V
Doplňky zásahového oděvu spodky s dlouhými nohavicemi	1 ^{III}
Doplňky zásahového oděvu - nátělník	2 ^{III}

Poznámky:

- I 2 ks kalhot PS II jsou minimálním požadavkem pro hasiče z povolání. PS II je stejnokrojem hasiče dobrovolné jednotky a v provedení ve variantě B není určen pro zásahy u požárů.
- II Jako alternativu k tomuto oděvu lze až do vymoření požívat PS II ve variantě A.
- III Minimální požadavek pro hasiče v jednotce HZS podniku nebo v jednotce SDH podniku.
- IV Uvedený prostředek může být u dobrovolných jednotek používán i jako společná ochranná pomůcka v množství odpovídajícím počtu hasičů jednoho družstva (6 ks).
- V Spodní prádlo může být nahrazeno doplňky zásahového oděvu.
- VI Pokud není ve výbavě požární techniky ve stejném počtu jako minimální početní stav členů podle počtu výjezdů družstev jednotky.

24 Scania P 340 4x4, BUS Mercedes Benz Sprinter a DA 12 Ford Transit. Z chemických prostředků to jsou: IDP Dräger PA 94 Basic, masky k IDP Dräger FPS 7000, záchranná kukla Dräger a GasAlert Quattro. Prostředky strojní služby je pak několik čerpadel a motorových pil.

Pro porovnání s ostatními JPO III jednotkami se dá považovat taková výbava jako standard. Rozdíl je pak hlavně v množství investic od obce nebo jiných veřejných subjektů. Uvádím tento příklad jako důležitý pro následný vývoj aplikace, aby bylo možné očekávat, v jakém rozsahu budou počty techniky. To pak ovlivní například, jaké rozhraní použít pro jejich zobrazení, tedy také uživatelskou přívětivost aplikace. Je také možné méně čitelné položky zobrazit jednotlivě v menu, pokud by se jednalo o cisternové automobilové stříkačky CAS.

Na většině webových stránek SDH bohužel není informace o počtech vzduchových kompresorů a tlakových lahvích dostupná. Buď není aktuální webová stránka (platí hlavně pro vzduchové lahve), nebo jednotky dobrovolných hasičů nedisponují kompresory, jelikož jejich cena se pohybuje ve vyšších řádech tisíců Kč. Může zde hrát také roli bezpečnost resp. prevence proti loupežím, kdy sbory nechtějí zveřejnit informace o prostředcích, kterými disponují. Lze očekávat, že sbory budou disponovat kompresory výjimečně a počty tlakových lahví se pohybují maximálně v desítkách.

1.1.4 Zdroje financí JPO

Finance jsou bezpochyby důležitým aspektem pro provoz JPO. Odvíjí se od nich kvalita resp. rok výroby většiny technické výbavy. Pro vysokotlaké vzduchové kompresory jsou finance tou první podmínkou, jestli si jej JPO pořídí. Ceny takových kompresorů jsou v řádech stovek tisíců Kč. Tomuto tématu se budu věnovat více v další sekci kapitoly. Profesionální hasiči, jelikož jsou financováni ministerstvem vnitra a musejí splňovat vysoké nároky a standardy, používají software pro evidenci svých technických prostředků s nutnými licencemi a certifikáty od komerčních firem na zakázky. Příkladem je program IKIS, kterému se budu částečně věnovat také v další sekci kapitoly.

Zdroje financí jsou různé. Hasičský záchranný sbor České republiky zveřejnil pro Jednotky požární ochrany dokument, jaké jsou možnosti zdroje příjmů pro získání dostatečných financí na provoz a techniku sboru [11]. Tento dokument je platný pro rok 2015-2017, novější alternativu jsem nebyl schopen nalézt, očekával bych ale, že dokument bude stejný, lišit se bude pouze obnos peněz, které sbory čerpaly z různých zdrojů na projekty. V dokumentu jsou totiž přítomné příklady pro jednotlivé typy příjmů.

Zdroje pro financování těchto jednotek tvoří:

1. výdaje obcí, které jednotky SDH obcí zřizují, jako základní zdroj financování,
2. dary právnických nebo fyzických osob na základě zákona o daních z příjmu,
3. dotace, které může poskytnout obcím kraj na základě zákona o požární ochraně v samostatné působnosti kraje¹ – dotace může být investiční i neinvestiční; pokud kraj poskytne tuto dotaci obcím, jedná se o výdaj z rozpočtu kraje do rozpočtu obce za podmíněných stanovených zákonem o rozpočtových pravidlech územních rozpočtů²,
4. účelová neinvestiční dotace, kterou poskytuje kraj na základě zákona o požární ochraně na tzv. přenesený výkon státní správy obcím³. Tato dotace se nyní poskytuje jako tzv. průběhová a finanční prostředky obcím poskytuje Ministerstvo vnitra – generální ředitelství Hasičského záchranného sboru ČR (dále jen „MV-GŘ HZS ČR“) z rozpočtu Ministerstva vnitra (dále jen „MV“) prostřednictvím krajů v souladu se zákonem o rozpočtových pravidlech⁴. Přeneseným výkonem státní správy se dle zákona o požární ochraně⁵ rozumí:
 - (a) zásah jednotky SDH obce mimo území svého zřizovatele,
 - (b) akceschopnost jednotky SDH obce zasahovat mimo území tzv. jednotkou kategorie JPO II a JPO III,
 - (c) odborná příprava velitelů a strojníků nebo specializační kurzy členů jednotky.
5. účelová investiční dotace na požární techniku jednotek SDH obcí, která je poskytována MV-GŘ HZS ČR obcím přímo z rozpočtu MV⁶,
6. ostatní dotační tituly – např. Evropské strukturální a investiční fondy – IROP 2014+.

¹ § 27 odst. 3 zákona č. 133/1985 Sb., o požární ochraně ve znění pozdějších předpisů

² zákon č. 250/2000 Sb., o rozpočtových pravidlech územních rozpočtů ve znění pozdějších předpisů

³ § 27 odst. 1 písm. d) zákona č. 133/1985 Sb., o požární ochraně ve znění pozdějších předpisů

⁴ § 14 zákona č. 218/2000 Sb., o rozpočtových pravidlech a o změně některých souvisejících zákonů (rozpočtová pravidla)

⁵ § 29 zákona č. 133/1985 Sb., o požární ochraně ve znění pozdějších předpisů

⁶ § 24 odst. 1 písm. d) zákona č. 133/1985 Sb., o požární ochraně ve znění pozdějších předpisů

Dokument dále rozepisuje Účelové neinvestiční dotace pro JSDHO, které rozděluje na výdaje zabezpečení akceschopnosti, výdaje na odbornou přípravu, výdaje za uskutečněný zásah a výdaje na věcné vybavení neinvestiční povahy. V roce 2017 bylo celkem poskytnuto 100 mil. Kč na neinvestiční dotace, kde téměř polovina byla vynaložena na akceschopnost JSDHO.

Účelové investiční dotace jsou ze dvou zdrojů. Z Fondu zábrany škod České kanceláře pojistitelů a ze Státního rozpočtu. Dělí se převážně na tyto cíle:

1. Investiční dotace na cisternovou automobilovou stříkačku jednotek SDH vybraných obcí
2. Účelová investiční dotace na pořízení dopravního automobilu
3. Účelová investiční dotace pro výstavbu nebo rekonstrukci požárních zbrojnic

Také jsou v dokumentu zmíněné investice z Evropských strukturálních a investičních fondů. Ty jsou zaměřené na hasičské zbrojnice s cílem nové dislokace anebo modernizace a také na techniku. Tyto dotace jsou financovány s 85 % z Evropských fondů. Pro porovnání byla za rok 2017 byla celková výše dotací 158 mil. Kč.

Finance tedy JPO může získat z několika zdrojů. Zákon o Požární ochraně zavazuje obce, které tak jsou často hlavním zdrojem financí pro JPO. Obce často získávají finance právě z různých dotačních titulů. Ve městech, kde se nachází profesionální jednotky, pak obce resp. městské části nejsou nuceny financovat takové JPO, jelikož jim požární ochranu zajišťují právě tyto profesionální jednotky.

Z dokumentu se lze také dočíst, že jsou dotace zaměřeny na větší položky, kterými jsou hlavně investice do CAS, které tvoří základní technický arsenál jednotky, nebo oprav jednotlivých zbrojnic a garáží. Na větší položky jakou by byl software pro evidenci je nutné výběrové řízení na vyšší úrovni a samotný software se pak instaluje na všechny stanice. Pokud ale byl vysoutěžen pouze pro profesionální jednotky, třeba kvůli nutné odbornosti hasičů k zacházení s tímto programem, pak JPO III a níže si nemohou dovolit financovat takové výběrové řízení a jsou odkázáni na vlastní zdroje a osobní inovaci.

1.2 Povinnosti evidence předmětů

O povinné evidenci se zmiňuje Řád výkonu v jednotkách [7], který lze stáhnout na internetových stránkách Hasičského záchranného sboru ČR. Rozděluje povinnou evidenci na několik typů:

1. Základní dokumentace prostředku
2. Záznam o vyřazení prostředku z užívání

3. Záznam o kontrole prostředku
4. Protokol o revizi prostředku
5. Kalibrační protokol a ověřovací list

Základní evidence rozděluje prostředky na ty, u kterých je povinné evidovat více údajů, jelikož se považují za více důležité, ostatní jich nemají tolik. Tyto prostředky, ke kterým je povinné evidovat více údajů jsou například: automatizované externí defibrilátory, vakuové fixační prostředky a dlahy, plnicí soupravy k pneumatickým těsnícím a zvedacím prostředkům, pneumatické stabilizační tyče a další.

Povinné údaje pro vyjmenované prostředky: název prostředku, typ prostředku, kmenová jednotka, výrobce, typ podle výrobce, evidenční číslo (pokud není možné opatřit prostředek evidenčním číslem, evidence se provede např. na základě výrobního čísla), datum výroby, datum zavedení do evidence, aktuální stav (zařazení) prostředku (v provozu/mimo provoz), identifikaci osoby, která provedla záznam.

Povinné údaje pro ostatní prostředky jsou: název prostředku, typ prostředku, kmenová jednotka, aktuální počet kusů a identifikace osoby, která provedla záznam. V Příloha D.1 jsem uvedl fotku aktuálního způsobu evidence takových informací.

1.2.1 Důsledky na implementaci aplikace

Samotná aplikace se zaměřuje na evidenci použití vysokotlakého vzduchového kompresoru, předpokládám tedy, že není nutné evidovat všechny údaje. Jelikož ale v aplikaci bude nutné držet záznam i o několika kompresorech a několika lahví, pak údaje pro ostatní prostředky jsou ideální základ pro kvalitní identifikaci kompresoru nebo lahve. V aplikaci lze také očekávat, že bude možné přidávat záznamy o dalších prostředcích, ke kterým by mělo být možné vytvořit jejich nový typ. Zároveň hasič, který jakkoli manipuluje s kompresorem, musí držet dostatečnou odbornost, je vhodné evidovat, kdo kompresor používal. Nabízí se tak evidovat pověřenou osobu, která vkládá buď nový záznam o prostředku, nebo eviduje její použití. To pak následně splňuje poslední položku údaje pro ostatní prostředky, tedy že je veden údaj o osobě, která provádí záznam.

Řád dále specifikuje záznam o vyřazení prostředku z užívání. Pro aplikaci může takový záznam představovat atribut u prostředku, který identifikuje stav. Pokud by bylo nutné prostředek vyřadit, pak se stav přepne do false hodnoty, která by definovala právě vyřazení. Pro kompresor by bylo ale vhodné nechat záznam v deníku přítomen, jelikož je vhodné, aby bylo možné zpětně dohledat informaci, jaké lahve a kdy daný kompresor plnil, nebo kdo kompresor používal. Jelikož je aplikace ale zaměřená na používání kompresorů a ne na

evidenci prostředků, pak není nutné evidovat datum změny stavu, důvod vyřazení a identifikaci osoby, jak stanovuje Řád výkonu služby v jednotkách. To mi alespoň ušetří práci při implementaci.

Také je možné podle Řádu evidovat záznam o kontrole prostředku. Takový záznam by měl specifikovat druh kontroly, datum, název prostředku, jeho typ, evidenční nebo výrobní číslo prostředku, hodnoty parametrů, pokud jsou měřeny a výsledek se záznamem o osobě, která kontrolu provedla. Po konzultaci s vedoucím bakalářské práce je pro praxi dostačující alespoň na jejich jednotce evidence tzv. revize, kde je specifikován datum, kdy byla provedena a datum následující revize, zároveň by revize mohli vykonávat pouze dostatečně odborní hasiči. Kontrola by tedy byla možná zaznamenávat pomocí této revize, kdy by výsledek byl přepnutí stavu prostředku, pokud by kontrola našla závadu. Tato revize by pak suplovala i záznam o revizi, který je dalším typem záznamu.

Záznam o revizi by měl evidovat údaje jako název, typ a výrobní číslo prostředku, dále pak datum poslední revize, výsledek revize a datum splatnosti revize. Tyto údaje by byly pokryté revizí, kterou zmiňuji v předchozím odstavci.

1.2.2 Vzduchové kompresory

Pro plnění tlakových lahví plněných vzduchem se používají takzvané vysokotlaké kompresory. Takovými kompresory jsou i takzvané kompresory s písty, kdy se plnění dělí na dvě nebo více fází podle počtu stupňů zařízení. První stupeň je plnění okolním vzduchem a následující stupně jsou jeho postupně stlačení nad hodnotu tlaku v lahvi, kam se následně vzduch vpustí. Kompresory jsou často třístupňové, kde při každém stupni je vzduch stlačen na vyšší hodnotu tlaku a tlak je pojištěn ventilem na danou tlakovou hodnotu. Protože při plnění stoupá teplota lahve a vzduchu, je kompresor chlazen vodou nebo i pouze okolním vzduchem. Je také nutné filtrovat vlhkost ze vzduchu takzvanými odlučovači. Stlačený vzduch se tedy filtruje i od dalších nečistot, jakými jsou prachové částí nebo různé oleje.

Kompresory lze dělit do několika druhů. První je kompresor stabilní, kdy je větší s vyšším výkonem umístěn na stanovišti, kam se vozí tlakové lahve pro plnění. Existují také mobilní kompresory, které mají nižší výkon a hmotnost. Používají se hlavně při delším zásahů v terénu. Pak jsou i kompresorové stanice, které se skládají z místností pro plnění, uložení a zásobníku tlakových lahví a místností pro samotné kompresory. Jedná se o komplexní strukturu. Pro jednotky sboru dobrovolných hasičů a tedy i téma této bakalářské práce je hlavní první typ. Tedy kompresor, který je stabilní na jedné zbrojnici, kam si ostatní jednotky svážejí vlastní vzduchové lahve pro plnění.

Příkladem takového kompresoru může být Poseidon, který je vzduchový třístupňový tříválcový vzduchem chlazený kompresor pro plnění TL o plicním tlaku 20 nebo 30 MPa. Pro jeho evidenci je nutné vést záznam o údržbě, která

■ **Tabulka 1.3** Intervaly provádění údržby u vzduchových kompresorů, Chemická služba [12]

Interval	Druh údržby
Denně	Kontrola stavu oleje, zkouška pojistného ventilu – natlakovat kompresor na maximum a pojistný ventil musí odpustit po 30 min ruční přezkoušení funkce ventilů
Týdně	Zkouška automatického vypouštění kondenzátu po 25 Mh kontrola napnutí klínových řemenů po 50 Mh sací filtr Mikronic po 250 Mh pojistné ventily
Po 1000 Mh	Zkouška těsnosti rozvodů
Po 2000 Mh	Vyčistění ventilů (výměna), výměna oleje a sintrových vložek
Po 3000 Mh	Výměna oleje, syntetické oleje, výměna ventilů
Ročně	Kontrola pístů a pístních kroužků, kontrola otevírání pojistného ventilu, výměna oleje, i když je natočeno méně než 1000 Mh, kontrola počítadla cyklů
Po 2 letech	Výměna oleje

se řídí podle této Tabulky 1.3.

1.2.3 Tlakové lahve

Tlakové lahve jsou hasiči používány pro různé účely. Hlavní účel je plnění vzduchem pro izolační dýchací přístroje. Také lze lahve plnit vzduchem pro jiné účely, například pro zvedací vaky, kanálové ucpávky, dekontaminační sprchy. Plnění kyslíkem se používají pro kříšení. Jsou i TL plněné technickými plyny. Webový provozní deník ale bude vést záznamy plnění lahví pouze pro lahve plněné vzduchem, které se následně využívají s ostatním příslušenstvím jako izolační dýchací přístroje.

Lahve se dělí na typy podle materiálu a to na ocelové, lehčené ocelové, kompozitní a hliníkové. Podle průzkumu z roku 2010 vzduchové lahve byly tvořeny ze 45 % ocelovými a z 55 % kompozitními. Jejich rozdíl je také v tom, že kompozitní lahve v ohni nevybuchují, zatímco ocelové ano. Kompozitní mají desetinásobnou rezervu pevnosti, jsou ale více náchylné na mechanické poškození. Koroze uvnitř ocelové lahve je problém. Ke zvýšené vlhkosti může dojít při úplném vypuštění. Provozní deník by měl počítat s faktem, že láhev bude vždy obsahovat alespoň nějaký tlak a nikdy by nemělo dojít k úplnému vypuštění. Pokud by k němu došlo, musí láhev automaticky na revizi.

Životnost se liší podle typu lahve a také podle normy, kterou splňuje. U kompozitních lahví je životnost určena také podle roku výroby a měla by být vyznačena, pokud tomu tak není, je výchozí životnost 15 let. Prodloužení

životnosti se nedoporučuje, je to ale možné pod vedením autorizované osoby na tomu určené zkoušce. Prodloužení životnosti je pak až na 3 roky. U ocelových lahví vyrobených podle ČSN EN 1964-2 a zkoušených podle ČSN EN 1968 platí tzv. neomezená životnost, tedy že není důvod pro překročení životnosti, kromě těch lahví, které byly vyrobeny před rokem 2002, kde je životnost omezená na 40 let.

1.2.4 Masky, nosiče, ostatní předměty

Vysokotlaké vzduchové kompresory a tlakové lahve jsou důležité, aby bylo možné využívat sestavu izolačního dýchacího přístroje, kterou by měla jednotka zajistit pro všechny členy v pohotovosti pro výjezd viz. Tabulka 1.1. Ta se skládá z nosiče, lahve, masky, plicního držáku a automatiky. Komplety lze například objednat na stránkách vyzbrojna.cz, kde nejlevnější k dubnu 2024 stojí 38 638 Kč s DPH [13]. Obsahuje buď lahev z oceli nebo kompozitu. Pro menší jednotku JPO III o dvanácti členech, existují i jednotky JPO III viz. Tabulka 1.1 o čtyřiaadvaceti členech, je minimální cena za izolační dýchací přístroje 154 552 Kč.

Tyto další prostředky by mělo být možné také evidovat v aplikaci. Vyskytují se ideálně na jednom místě jako sada. Kdyby provozní deník nabízel správu místa každého prostředku, měl by nabízet i možnost zobrazit, jestli je sada připravena, nebo je láhev ještě plněna. Hasiči by tak poznali, jestli jsou například všechny sady na CAS připravené, nebo jakou sadu si mohou vzít na výjezd. Typicky ale jednotka disponuje více tlakovými lahvemi než nosiči, a proto by sada mohla být doplněna vždy jinou tlakovou lahví. [12]

1.3 Plnění lahví

Proces plnění lahví je důležité popsat také proto, že webový provozní deník bude zaměřený hlavně na jeho evidenci. Samotný proces plnění závisí na typu kompresoru a každý kompresor má svá specifika, která jsou uvedena v návodu. Cituji z učebních skript chemické služby ISBN 978-80-87544-09-9. Ta uvádí příklad kompresoru Poseidon, který je stabilní kompresor a představuje tak kompresor, který se vyskytuje na zbrojnicích jednotek dobrovolných hasičů. Tento kompresor plní tlakové lahve buď na 20 MPa nebo 30 MPa.

Tabulka Vybrané technické normy k tlakovým lahvím, která je v citovaném dokumentu [12] uvádí normy, kterými je potřeba se řídit při plnění lahve, anebo které se obecně pojí k tlakovým lahvím. Tabulku jsem převedl do seznamu, který uvádím níže.

Vybrané technické normy k tlakovým lahvím, Chemická služba [12]

- ČSN 07 8304:2008 – Tlakové nádoby na plyny – Provozní pravidla

- **ČSN 07 8305:1976** – Kovové tlakové nádoby k dopravě plynů – Technická pravidla
- **ČSN 07 8602:1987** – Kovové láhve na plyny. Uzavírací ventily pro láhve na plyny s plnicím přetlakem do 20 MPa. Všeobecná ustanovení.
- **ČSN 07 8631:1987** – Kovové láhve na plyny. Uzavírací ventily pro láhve na plyny s plnicím přetlakem do 20 MPa. Rozměry.
- **ČSN EN 1089-3** – Láhve na přepravu plynů – Označení lahví (kromě lahví na LPG) – Část 3: Barevné značení
- **ČSN EN 12245:2009** – Láhve na přepravu plynů – Plně ovinuté kompozitové láhve
- **ČSN EN 12257:2002** – Láhve na přepravu plynů – Bezešvé částečně ovinuté kompozitové láhve
- **ČSN EN 144-2:1999** – Ochranné prostředky dýchacích orgánů – Ventily lahví na plyny – Část 2: Závítové spojení na výstupu
- **ČSN EN 144-3:2003** – Ochranné prostředky dýchacích orgánů – Ventily lahví na plyny – Část 3: Závítové spojení na výstupu pro plyny Nitrox a kyslík určené pro potápění
- **ČSN EN 1968:2002** – Láhve na přepravu plynů – Periodická kontrola a zkoušení bezešvých ocelových lahví
- **ČSN EN ISO 10286:2008** – Láhve na plyny – Terminologie
- **ČSN EN ISO 11623:2003** – Láhve na přepravu plynů – Periodická kontrola a zkoušení lahví na plyny z kompozitních materiálů
- **ČSN EN ISO 13769:2009** – Láhve na plyny – Značení ražením

1.3.1 Průběh plnění lahve

Samotný postup je složen z několika kroků. Začíná se kontrolou ventilů, koncovky na tlakové lahvi, uzavření odpouštěcího ventilu na koncovce. Po zapnutí kompresoru lze již tlakovat i více lahví najednou. Při plnění je nutné dbát na maximální tlak na kompresoru i tlakové lahvi a je nutné pravidelně odkalovat.

1.3.2 Znalostní požadavky pro hasiče

Plnění lahve může provádět pouze hasič, který je dostatečně oprávněn, to znamená seznámen s předpisy pro obsluhu a také se všemi bezpečnostními předpisy. Takovými jsou poplachové směrnice nebo požární řád pracoviště a hlavně musí být zaškolen k obsluze kompresoru a tlakových lahví podle požadavků specifikovaných od provozovatele zařízení.

Ověření znalostí ověřuje revizní technik, který disponuje osvědčením odborné způsobilosti, které je nutné obnovit každé 3 roky.

1.4 Design uživatelského rozhraní

Nejdříve uvedu samotnou definici uživatelského rozhraní od Freda Churchilla v anglickém originále a také její překlad. [14]

The user interface (UI) is the point of human-computer interaction and communication in a device. This can include display screens, keyboards, a mouse and the appearance of a desktop. It is also the way through which a user interacts with an application or a website.

Uživatelské rozhraní (UI) je místo interakce a komunikace mezi člověkem a počítačem v zařízení. Může zahrnovat obrazovky, klávesnice, myš a vzhled pracovní plochy. Je to také způsob, jakým uživatel komunikuje s aplikací nebo webovou stránkou.

1.4.1 Požadavky na uživatelské rozhraní

Správný design uživatelského rozhraní je důležitým aspektem této aplikace, hlavně protože požadavek na téma této bakalářské práce a tedy i na vytvoření této aplikace vznikl ze skutečnosti, kdy se všechny zápisy o plnění lahví kompresory psaly písemně do sešitu, kde je uživatelské rozhraní téměř nulové a představuje mnoho potenciálních chyb pro uživatele. Pro příklad takového sešitu viz. Obrázek 1.1. Počítat motohodiny kompresoru, podle kterých se odvíjí jeho následující revize, znamenalo projít všechny zapsané zápisy a sečíst jejich dobu trvání.

Mohlo tak vzniknout několik druhů chyb, které by mělo nové uživatelské rozhraní eliminovat.

1. Přehlednutí záznamu, který se následně nezaznamenal.
2. Špatný součet motohodin.
3. Nemožnost čistě opravit špatný záznam, který tak byl příliš nepřehledný.
4. Mechanické zničení nebo ztráta samotného zápisového archu.

NO.	DATA	TRAVNÍ	PLNĚNÍ REZERV	OBJEM LÁHVE	TRAVNÍ TLAKA	JEDNOTKA Z ČÍSLA LÁHVE	POŘÍČÍ
450	24.9.23	0,62	VZDUCH	6,9	300	NEBUSICE - 1,4,5,6,8,11,13,14	56
451	26.9.23	0,15	VZDUCH	6,9	300	HORONĚVICE - 1,4,5,6,8,11,13,14	56
452	26.9.23	0,12	VZDUCH	6,9	300	ZLICHŮV - 6,9,13	56
453	26.9.23	0,03	VZDUCH	6,9	300	ZLICHŮV - 1,7	56
454	3.10.23	1,0	VZDUCH	6,9	300	ZLICHŮV - 1,4,5,6,8,11,13,14	56
455	5.10.23	0,33	VZDUCH	6,9	300	ŘEPA - 12,13,14,23,27	56
456	4.10.23	0,15	VZDUCH	6,9	300	ZLICHŮV - 1,4,5,6,8,11,13,14	56
457	4.10.23	0,08	VZDUCH	6,9	300	ŘEPA - 1,4,5,6,8,11,13,14	56
458	8.10.23	0,60	VZDUCH	6,9	300	ŘEPA - 1,4,5,6,8,11,13,14	56
459	8.10.23	0,07	VZDUCH	6,9	300	ZLICHŮV - 1,6	56
460	10.10.23	1,08	VZDUCH	6,9	300	ŘEPA - 1,4,5,6,8,11,13,14	56
461	10.10.23	0,17	VZDUCH	6,9	300	ŘEPA - 1,4,5,6,8,11,13,14	56
462	11.10.23	0,33	VZDUCH	6,9	300	ZLICHŮV - 1,4,5,6,8,11,13,14	56
463	11.10.23	0,09	VZDUCH	6,9	300	ZLICHŮV - 7,12	56
464	11.10.23	0,15	VZDUCH	6,9	300	NEBUSICE - 1,4,5,6,8,11,13,14	56
465	11.10.23	0,14	VZDUCH	6,9	300	NEBUSICE - 1,4,5,6,8,11,13,14	56
466	12.11.23	0,33	VZDUCH	6,9	300	ŘEPA - 3,13,16,24	56
467	15.11.23	0,27	VZDUCH	6,9	300	ŘEPA - 6,14,17,25	56
468	15.11.23	0,38	VZDUCH	6,9	300	NEBUSICE - 8,11,7,14	56
469	15.11.23	0,47	VZDUCH	6,9	300	NEBUSICE - 3,10, BOUTHEAN - 13,14,15	56
470	15.11.23	0,17	VZDUCH	6,9	300	ZLICHŮV - 1,13,14,15	56
471	15.11.23	0,08	VZDUCH	6,9	300	NEBUSICE - 3,10, BOUTHEAN - 13,14,15	56
472	15.11.23	0,33	VZDUCH	6,9	300	ZLICHŮV - 2,3,4,5,6,8,9,11,13,14	56
473	25.11.23	0,75	VZDUCH	6,9	300	ZLICHŮV - 1,4,5,6,8,11,13,14	56
474	25.11.23	0,74	VZDUCH	6,9	300	ZLICHŮV - 1,4,5,6,8,11,13,14	56
475	25.11.23	0,73	VZDUCH	6,9	300	ZLICHŮV - 1,4,5,6,8,11,13,14	56
476	25.11.23	0,72	VZDUCH	6,9	300	ZLICHŮV - 1,4,5,6,8,11,13,14	56
477	25.11.23	0,71	VZDUCH	6,9	300	ZLICHŮV - 1,4,5,6,8,11,13,14	56
478	25.11.23	0,70	VZDUCH	6,9	300	ZLICHŮV - 1,4,5,6,8,11,13,14	56
479	25.11.23	0,69	VZDUCH	6,9	300	ZLICHŮV - 1,4,5,6,8,11,13,14	56
480	25.11.23	0,68	VZDUCH	6,9	300	ZLICHŮV - 1,4,5,6,8,11,13,14	56
481	25.11.23	0,67	VZDUCH	6,9	300	ZLICHŮV - 1,4,5,6,8,11,13,14	56
482	25.11.23	0,66	VZDUCH	6,9	300	ZLICHŮV - 1,4,5,6,8,11,13,14	56
483	25.11.23	0,65	VZDUCH	6,9	300	ZLICHŮV - 1,4,5,6,8,11,13,14	56
484	25.11.23	0,64	VZDUCH	6,9	300	ZLICHŮV - 1,4,5,6,8,11,13,14	56
485	25.11.23	0,63	VZDUCH	6,9	300	ZLICHŮV - 1,4,5,6,8,11,13,14	56
486	25.11.23	0,62	VZDUCH	6,9	300	ZLICHŮV - 1,4,5,6,8,11,13,14	56
487	25.11.23	0,61	VZDUCH	6,9	300	ZLICHŮV - 1,4,5,6,8,11,13,14	56
488	25.11.23	0,60	VZDUCH	6,9	300	ZLICHŮV - 1,4,5,6,8,11,13,14	56
489	25.11.23	0,59	VZDUCH	6,9	300	ZLICHŮV - 1,4,5,6,8,11,13,14	56
490	25.11.23	0,58	VZDUCH	6,9	300	ZLICHŮV - 1,4,5,6,8,11,13,14	56
491	25.11.23	0,57	VZDUCH	6,9	300	ZLICHŮV - 1,4,5,6,8,11,13,14	56
492	25.11.23	0,56	VZDUCH	6,9	300	ZLICHŮV - 1,4,5,6,8,11,13,14	56
493	25.11.23	0,55	VZDUCH	6,9	300	ZLICHŮV - 1,4,5,6,8,11,13,14	56
494	25.11.23	0,54	VZDUCH	6,9	300	ZLICHŮV - 1,4,5,6,8,11,13,14	56
495	25.11.23	0,53	VZDUCH	6,9	300	ZLICHŮV - 1,4,5,6,8,11,13,14	56
496	25.11.23	0,52	VZDUCH	6,9	300	ZLICHŮV - 1,4,5,6,8,11,13,14	56
497	25.11.23	0,51	VZDUCH	6,9	300	ZLICHŮV - 1,4,5,6,8,11,13,14	56
498	25.11.23	0,50	VZDUCH	6,9	300	ZLICHŮV - 1,4,5,6,8,11,13,14	56
499	25.11.23	0,49	VZDUCH	6,9	300	ZLICHŮV - 1,4,5,6,8,11,13,14	56
500	25.11.23	0,48	VZDUCH	6,9	300	ZLICHŮV - 1,4,5,6,8,11,13,14	56

Obrázek 1.1 Zápisový sešit záznamů o plnění lahví, SDH Řepy, doc. RNDr. Dušan Knop, Ph.D.

5. Špatné zabezpečení, kdy zápis mohl provést kdokoli, bez kontroly dostatečné odbornosti.

Nové uživatelské rozhraní by tedy mělo zvládat předejít těmto nedostatkům. Záznamy by měly být dostatečně čitelné, to znamená, aby nezabíraly příliš místa na obrazovce. Mělo by být možné filtrovat a řadit, podle nějakého údaje jako např. podle kompresoru, který plnil danou láhev.

Mělo by být možné jednoduše zjistit počet motohodin, kdy daný kompresor plnil láhve od poslední revize. Jednoduše upravit špatně zanesený záznam o plnění a také by měla aplikace nabízet možnost spravovat pro uživatele jejich role, podle kterých by se omezil přístup k vytvoření nového záznamu.

Dále by mělo být jednoduše možné zobrazit u každého prostředku jeho poslední revizi. Současně s přítomným počtem motohodin, by tak bylo jednoduché zjistit, kdy je nutné kompresor znovu zkontrolovat. Stejně tak u lahví, kde by po každém plnění byla nabízena informace, jaký tlak měla při posledním plnění. Tato informace je důležitá, protože se lahve plní několikrát, než splňují cílovou hladinu tlaku a mohou být použité při zásahu.

Přirozené případy užití, jako přesun více prostředků nebo plnění více lahví zároveň, by mělo být možné vybráním všech prostředků najednou a provedením jedné akce. Pokud by se jednotlivé lahve v aplikaci plnily po jedné, pak by používání aplikace bylo po delší době nepříjemné pro uživatele, jelikož se tyto případy užití vztahují na více prostředků zároveň. Jako příklad uvedu situaci, kdy po příjezdu ze zásahu je potřeba přemístit všechny lahve ke kompresoru, kde se následně i společně plní.

1.4.2 Typický uživatel aplikace

Mít alespoň základní definici typického uživatele je důležité pro vývoj aplikace. Protože je nutné určit, jaké technologie je možné použít a jaké postupy a nové prvky lze očekávat, že bude uživatel aplikace znát a umět využívat. Pokud by například byla aplikace pouze v angličtině, pak bude sice mezinárodní, nelze ale od běžného občana České republiky očekávat vyšší úroveň jazyka než B1, a tedy by byla pro většinu nepoužitelná.

Typický uživatel v kontextu této aplikace představuje člen JSDH. Takový uživatel je středního věku, který je s moderními technologiemi v denním kontaktu. Vlastní dotykový telefon, který představuje hlavní a nejčastější kontakt s moderními technologiemi. Zná základní principy přihlašování. Zvládá interakci s běžnou aplikací, tedy dokáže očekávat, jak jsou prezentované odkazy, tlačítka nebo jaké objekty je možné scrollovat. Preferuje kontakt pomocí myši, protože na klávesnici nezná rozpoložení kláves, a tedy je pro něj nepříjemné časté a nutné psaní. Vztah k technologiím má spíše negativní, podle něj sice dokáží v mnohém hodně pomoci, jsou ale často náchylné na malé změny, které mají za následek nepochopitelné stavy aplikace nebo neočekávatelné reakce. Nepříjemná je pro něj dlouhá odezva aplikace.

Takový typ uživatele samozřejmě nepokrývá všechny typy uživatelů aplikace. Je důležité ale brát v reálný potaz, že existuje a není v menšině, a proto by aplikace měla být v mnohém z pohledu uživatelského rozhraní konzervativní.

1.4.3 Možnosti zobrazení

Aplikace bude především zobrazována na stolních počítačích, které jsou již dnes přítomny na jednotkách sborů dobrovolných hasičů. Při manipulaci prostředků nebo plnění lahví by bylo vhodné, aby bylo možné využívat aplikaci také na mobilním zařízení, ušetřila by se tak práce při označování prostředků na počítači, který by byl v jiné místnosti a hasič by si tak nemusel pamatovat označení pro každý prostředek a ten následně vyhledat na počítači.

Aplikace by měla také být přizpůsobena i velikosti tabletů, které jsou dnes častým zařízením. Proto by měla disponovat designem uživatelského rozhraní responzivním pro různé šířky a výšky displejů. I na takových obrazovkách, které jsou menší šířky je potřeba, aby bylo možné zobrazit seznamy prostředků přehledně, tedy například nabízet přirozené posunutí, aby nebylo nutné stránku oddalovat.

V roce 2021 podle veřejného průzkumu přístupů na internet v Evropě přes počítač je kolem 50 % a na telefonu kolem 46 % [15]. Proto by aplikace měla počítat do budoucna s přístupem přes mobilní zařízení.

Důležitou možností zobrazení je také zobrazení webového provozního deníku pro tisk. Tisk může být potřeba například pro doložení nějakého záznamu písemně, nebo může být potřeba vytisknout libovolný seznam z aplikace pro jeho vyvěšení na nástěnku ve zbrojnici.

1.5 Existující alternativy aplikace

V České republice existují dvě hlavní alternativy – IKIS II a PORT.ALL. IKIS II je dlouhodobě vyvíjen na zakázky pro jednotlivé moduly. Je velmi komplexní. Nabízí spolehlivou aplikaci, která dnes zahrnuje téměř všechny případy užití. Druhá PORT.ALL je aplikace, která je určena pro užití jednotkami sborů dobrovolných hasičů, dnes je zastaralá a nenabízí všechny případy užití. Disponuje také zastaralým uživatelským rozhraním.

1.5.1 Internetový portál PORT.ALL

Tato aplikace resp. internetový portál je již staršího data, soudíc ze vzhladu uživatelského rozhraní. Nabízí ale dostačující základ. Skládá se ze tří modulů – ekonomika, lidé a strojní služba. Bohužel ale nedisponuje evidencí plnění tlakových lahví. Celé rozhraní je zahrnuté v horním menu, kde si uživatel zvolí s jakým modulem chce pracovat. Aplikace nabízí import a export položek, evidenci včetně vyřazení prostředku a uživatelská práva. Přístupná je

přes internetový portál. Samotný server je hostovaný na Generálním ředitelství HZS. [16]

Nevýhody jsou hlavně uživatelské rozhraní a nedostatečné pokrytí případů užití. Rozhraní je nepřehledné a nabízí mnoho funkcí, které jsou ale všechny zobrazené na jedné obrazovce, a pro nového uživatele je časově náročné, než se zorientuje ve vysoké nabídce možností. Další nevýhodou je nutné časté a přesné klikání na menší tlačítka a jiné záložky. Také občas aplikaci nefunguje přístup do databáze. Poslední nevýhodou, soudě ze screenshotů aplikace, je pak nepříznivé působení různým šířkám displejů. Aplikace byla designovaná na jednu šířku a na telefonu je naprosto nepoužitelná. Příklad z aplikace okna pro přidání nového prostředku Obrázek 1.2.

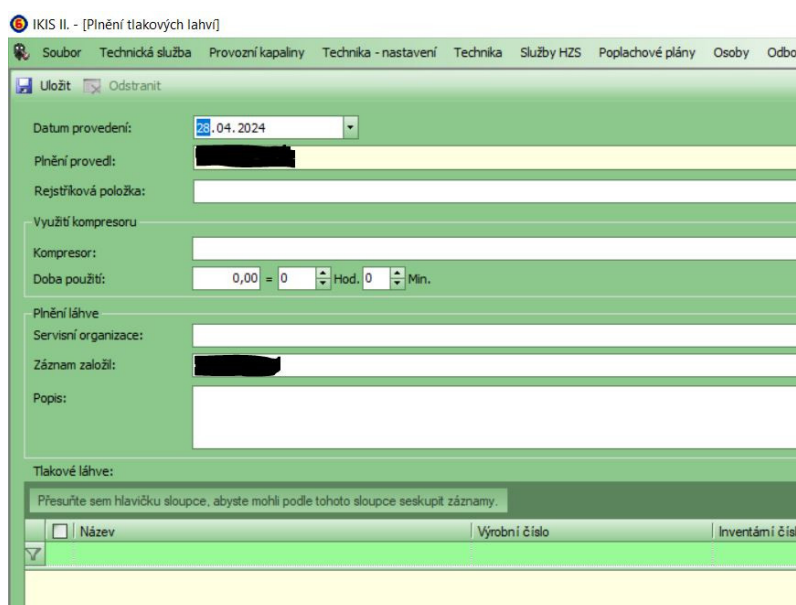
■ Obrázek 1.2 PORT.ALL přidání prostředku, Modul Chemická služba [17]

1.5.2 Aplikace profesionálních hasičů IKIS II

Na úvod uvedme, že touto aplikací nedisponují neprofesionální jednotky, tedy jednotky dobrovolných hasičských sborů. Je ale poměrně komplexní a lze se jí proto inspirovat. Na stránkách HZS můžeme jednoduše vyhledat smlouvy k dalším modulům, kterými se IKIS II rozšířil. Lze například nalézt smlouvu o modulu k chemické službě [18]. Ta pro inspiraci může sloužit pouze okrajově, nabízí pouze seznam funkcionalit, kterými bude nový modul disponovat.

Získal jsem několik snímků ze samotné aplikace od Ing. Petra Kolečáka, který pracuje u profesionálních hasičů v chemické službě. Snímek s formulářem pro plnění tlakové lahve je zobrazen níže v této sekci, ostatní snímky jsou pak v přílohách. Jedná se o výstřižky obrazovky, které nezobrazují celé rozhraní aplikace. Slouží tedy hlavně pouze pro inspiraci.

Na Obrázku 1.3 je zachyceno okno s formulářem pro editaci nebo vytvoření nové evidence plnění tlakové lahve. Formulář nabízí evidenci data, plnicího kompresoru, servisní organizace, osoby, která provedla plnění nebo která uložila tento záznam. Rozhraní je podobné portálu PORT.ALL. Na první pohled si lze všimnout, že IKIS II není aplikace přístupná z webového prohlížeče, nýbrž se jedná o aplikaci nainstalovanou přímo do systému počítače. Tento výstřížek zobrazuje nové okno aplikace s formulářem. V prohlížeči je možné pomocí Javascriptu také vyvolat otevření nového okna. Tato funkcionality není ale přívětivá pro uživatele, jelikož nelze dokonale předpokládat chování a je nutné stáhnout celý obsah stránky v tomto okně znovu. Proto je vhodné takové okno v aplikaci nasimulovat jinak. Lze použít například vyskakovací formulář, který překryje aktuální okno aplikace.



IKIS II. - [Plnění tlakových lahví]

Soubor Technická služba Provozní kapaliny Technika - nastavení Technika Služby HZS Poplachové plány Osoby Odbo

Uložit Odstranit

Datum provedení: 04. 04. 2024

Plnění provedl: [redacted]

Rejstříková položka: [redacted]

Využití kompresoru

Kompresor: [redacted]

Doba použití: 0,00 = 0 Hod. 0 Min.

Plnění láhve

Servisní organizace: [redacted]

Záznam založil: [redacted]

Popis: [redacted]

Tlakové láhve:

Přesuňte sem hlavičku sloupce, abyste mohli podle tohoto sloupce seskupit záznamy.

<input type="checkbox"/>	Název	Výrobní číslo	Inventární číslo
--------------------------	-------	---------------	------------------

■ **Obrázek 1.3** Snímek z aplikace IKIS II, Petr Kolečák

Formulář na snímku nabízí pole pro evidenci doby plnění. Toto pole pro jednotky času je ale neobvykle implementováno, kdy je nutné zadat hodiny a minuty zvlášť. Dnešní prohlížeče nabízejí jedno časové pole pro obě hodnoty, které je na jednotlivých typech operačních systémů a prohlížečích implementováno samostatně, takže je například způsob plnění hodnot na mobilních zařízeních jiný než na ostatních zařízeních a tedy i uživatelsky příjemnější. Zároveň tlačítko pro uložení je situováno nad formulářem. Uživatele tak může mást, kdy po postupném vyplnění polí očekává, že bude tlačítko uložení následovat za posledním polem. Osobně preferuji rozložení formuláře takové, kdy jsou popisky jednotlivých polí umístěny nad poli pro vstup a ne vedle, jako tomu je na tomto snímku. Není potřeba řešit odsazení mezi popisky a poli tak, že jsou všechny pole posunuté až za nejdelším popiskem. Jsou pak také popisky blíže

k polím a pro uživatele je formulář přehlednější.

Údaje formuláře jako servisní organizace nebo rejstříková položka mají svůj smysl, pro aplikaci menšího rozsahu, kterou je Webový provozní deník kompresoru, nenabízí velkou přidanou hodnotu a do samotné aplikace bych je nezahrnul.

Další snímek detailu oprav, revizí a provozní kontrol prostředku resp. ocelové tlakové lahve je přiložen v Příloze A. Snímek okna pro uložení nového záznamu o roční provozní kontrole prostředku je v Příloze B. A poslední snímek seznamu plánovaných kontrol všech prostředků je v Příloze C.

Ze snímků lze vypořadovat, že má IKIS II dobře odladěné tabulky. Záznamy lze filtrovat podle všech sloupců zároveň. Je možné editovat záznamy přímo z tabulky nebo řadit záznamy podle hodnot sloupce. Delší záznamy jsou schované za okrajem obrazovky a je nutné tabulku posunout. Zvýrazněné záznamy jsou přehledně obarvené. Hodnoty jsou správně naformátované a delší hodnoty jsou skryté.

IKIS II představuje komplexní aplikaci, která dobře splňuje její požadavky. Pro menší Webový provozní deník představuje vhodnou inspiraci v mnoha ohledech. Není ale přístupná přes internet a nemá stejně jako PORT.ALL podporu uživatelského rozhraní pro mobilní zařízení. Předpokládám, že je vázána na jeden operační systém. Její architektura je, soudě z přiložených snímků, vysoce kvalitní. Nabízí intuitivní správu. Jednotlivé entity jsou dobře propojené. Hodnoty jsou správně zobrazovány a celkové rozhraní je přehledné.

1.6 Open-source požadavky

Cíle práce je implementace open-source aplikace. Proto je důležité brát v potaz licenci, pod kterou bude aplikace distribuována. Představa je taková, že by bylo možné aplikaci dále vyvíjet kýmkoli a případně také zpoplatnit, ideálně při přidání dalších funkcionalit. Aplikace spadá pod licenci bakalářské práce, je proto nutné řešit, jakou licenci bude mít samotná bakalářská práce, a také licenci samotného softwaru.

1.6.1 Licence bakalářské práce

Práva, která určí licenci bakalářské práce, lze zvolit z několika možných. Tato aplikace bude tzv. volná pro libovolné užití kýmkoli. Proto jsem zvolil prohlášení, které neomezuje její kopírování a pokračování ve vývoji. Je ale nutné pohlídat licenci, se kterou se daný software šíří, protože se spíše podobá licenci MIT a není možné licencovat software s licenci GPL. Autor této bakalářské práce má právo dále stavět a nabízet daný software se striktnější licenci.

1.6.2 Open-source licence

Obecně známými otevřenými licencemi jsou GPL (General Public License) a MIT Licence (Massachusetts Institute of Technology Licence). Jejich hlavní rozdíl je v přístupu k derivátům softwaru. Kde GPL zavazuje software, aby zůstal distribuovaný také s touto otevřenou licencí. Proto není možné použít tuto licenci s tímto projektem, jelikož volné prohlášení o právech dovoluje autorovi stavět dále se striktnější licencí. MIT Licence umožňuje následné prodání softwaru.

Začátek Preambule GPL verze 3. [19]

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the *GNU General Public License* is intended to guarantee your freedom to share and change all versions of a program—to make sure it remains free software for all its users. We, the *Free Software Foundation*, use the *GNU General Public License* for most of our software; it applies also to any other work released this way by its authors. You can apply it to your programs, too.

Tedy přímo v preambuli stojí toto.

Naproti tomu obecná veřejná licence GNU má zaručit svobodu sdílení a změny všech verzí programu – zajistit, aby zůstal svobodným softwarem pro všechny uživatele.

MIT Licence byla vytvořena v osmdesátých letech. Je tzv. „permisivní“ licencí, která opravňuje software používat, kopírovat, upravovat, slučovat publikovat, distribuovat, poskytovat sublicence a prodávat jeho kopie [20].

Praktická část

2.1 Funkční a nefunkční požadavky

Rozepíší funkční a nefunkční požadavky, které jsou přímo v zadání, následně doplním další požadavky, které by měla aplikace i na základě teoretické části splňovat. Protože je aplikace zaměřená pouze na plnění tlakových lahví, uvedu právě požadavky, které cílí na téma bakalářské práce, ale neuvedu požadavky na potencionální možné rozšíření.

2.1.1 Funkční požadavky

- **F1:** Aplikace bude zabezpečená přihlášením. Všechna evidence bude tak přístupná pouze po přihlášení uživatelským jménem a heslem. Zobrazení nebude omezené podle rolí uživatele, proto bude možné se vždy přihlásit nezávisle na roli uživatele.
- **F2:** Po přihlášení do aplikace pod úvodním uživatelem a defaultním heslem bude umístěn formulář, který představí aplikaci uživateli a pobídne ho k aktualizaci hesla.
- **F3:** Aplikace bude nabízet evidenci kompresorů a tlakových lahví. U těchto prostředků budou dostatečné informace – typ, sériové číslo, označení, výrobce, popis, zavedení do provozu, datum výroby, stav, jednotka – tak, aby hasiči mohli převést své dosavadní zápisy a jednotlivé prostředky bylo možné jednoznačně identifikovat.
- **F4:** Aplikace bude nabízet evidenci a správu jednotek SDH. Zároveň bude možné přiřazovat tlakové lahve k těmto jednotkám.
- **F5:** V aplikaci bude možné zavést a evidovat údaj o plnění lahve daným vzduchovým kompresorem. Tento údaj bude možné přidat pouze administrátorem nebo uživatelem s právy technika. Záznam o plnění bude dispo-

novat minimálně údajem o uživateli, kompresoru, láhvi, aktuálním tlaku, začátkem a koncem plnění.

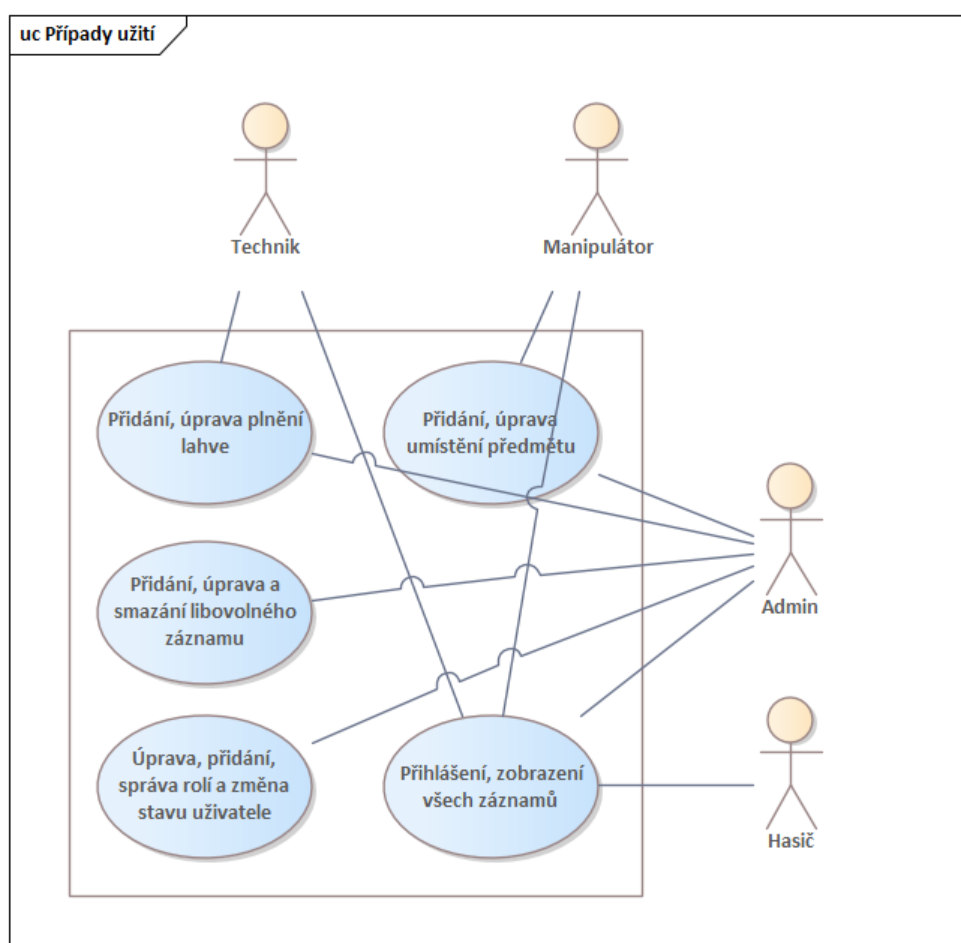
- **F6:** Admin bude disponovat právem upravit přihlašovací údaje ostatních uživatelů. Bude možné pomocí formuláře přidat další uživatele. Uživatele bude možné v rámci úprav deaktivovat tak, aby nebylo možné se za něj přihlásit.
- **F7:** Aplikace bude responzivní k šířkám obrazovek. Všechn obsah bude postupně s měnící šířkou tak transformovaný, aby nedocházelo k přetékaní. Pokud by nešlo jinak zabránit přetékaní, upřednostní se posunutí pomocí posuvníku.
- **F8:** Bude k dispozici tmavý režim takový, že se aplikace automaticky přepne podle aktuálního stavu zařízení. V tomto režimu budou stejné informace a obrázky stejně jako ve světlém režimu.
- **F9:** Funkčností se bude inspirovat, jak to bude možné, systémem IKIS II. Pokud se nevztahuje licence na tuto aplikaci nebo by bylo příliš složité získat dostatečné informace.
- **F10:** Pokud nastane logická chyba v aplikaci, bude prezentována uživateli jednoduchá stránka s chybovou hláškou a možnost vrácení zpět na domovskou stránku.
- **F11:** Uživatelské rozhraní bude zaměřené na srozumitelnost a přehlednost. Font aplikace bude podporovat český jazyk. V rozhraní se nebudou překrývat informace a tlačítka. Nebudou ani schovaná za okraji stránky.

2.1.2 Nefunkční požadavky

- **N1: Platforma** – Nasazení aplikace do produkčního prostředí nebude záviset na platformě zařízení.
- **N2: Instalace** – Aplikace bude dostupná internetu nebo intranetu po domluvě se zadavatelem práce. Bude k dispozici dokumentace, která nabídne několik možností instalace s jejími postupy. Samotná instalace bude intuitivní a nebude vyžadovat přílišné znalosti v oblasti IT.
- **N3: Dostupnost** – Bude možné aplikaci jednoduše hostovat na službách, které jsou finančně dostupné pro JSDH, to znamená do 1 000 Kč za rok. Zároveň bude možné volit mezi různými službami webhostigů, takže instalace nebude možná pouze na jedné.
- **N4: Přístupnost** – Stránky aplikace budou splňovat základy přístupnosti tak, aby bylo možné aplikaci používat z různých zařízení a pomocí různých příslušenství jako je myš, klávesnice, touchpad nebo pomocí čtečky, které podporují načtení stránky s Javascriptem.

2.2 Případy užití

Uvádím diagram případů užití, který popisuje role uživatelů – Admin, Technik, Manipulátor, a Hasič – a jejich práva. Shrnul jsem případy užití tak, aby byly vidět hlavní rozdíly práv jednotlivých rolí. Uživatel může zároveň sdílet několik rolí zároveň a každou rolí může disponovat více uživatelů zároveň. Neuvádím nepřihlášeného uživatele, který nedisponuje žádnými právy.



■ **Obrázek 2.1** Diagram případů užití, Enterprise architect, vlastní tvorba

2.2.1 Role uživatelů

Uživatele lze dělit na role podle práv, kterými disponují. Tyto druhy použiji pro popis případů užití.

1. **Nepřihlášený uživatel** – Uživatel, který není nijak autentizovaný.

- 2. Přihlášený uživatel** – Představuje hasiče sboru, který nedisponuje žádnou rolí. Smí zobrazovat informace. Je základně autorizovaný ke čtení.
- 3. Technik** – Přihlášený uživatel, který plní roli technika v hasičském sboru. Může tak evidovat plnění lahví.
- 4. Administrátor** – Uživatel, který je přihlášený a zároveň disponuje nejvyššími právy v aplikaci.

2.2.2 Případy užití

V této sekci popíšeme případy užití, které představují většinu interakcí s aplikací. Každý případ užití je rozepsán na jeho popis, aktéra a samotný postup (path) případu užití. Většina vychází z funkčních požadavků.

2.2.2.1 UC1: Úvodní přihlášení

Popis: První přihlášení po nasazení aplikace do produkce pod výchozím uživatelským jménem a heslem.

Aktér: Nepřihlášený uživatel

Basic path:

- Uživatel si zobrazí libovolnou stránku aplikace. Aplikace zobrazí přihlašovací formulář, protože detekuje nepřihlášeného uživatele.
- Uživatel vyplní výchozí uživatelské jméno a heslo dle dokumentace pro instalaci.
- Odešle dotaz na server po stisknutí na tlačítko *Přihlásit*. Aplikace zobrazí adekvátní odezvu a blok pro načítání, aby bylo zřetelné v jaké je aplikace stavu.
- Po úspěchu přihlášení zmizí přihlašovací formulář. Uživatel zůstává na stejné stránce bez přesměrování.

2.2.2.2 UC2: Přihlášení

Popis: Přihlášení libovolného uživatele do aplikace. Stejný případ užití platí pro expiraci sezení při delším používání aplikace.

Aktér: Nepřihlášený uživatel

Basic path:

- Uživatel zobrazí stránku aplikace.
- Aplikace vyzve uživatele k přihlášení přes vyskakovací okno s formulářem.
- Uživatel vyplní své přihlašovací údaje a odešle požadavek.
- Aplikace po správném zadání zruší vyskakovací okno.

2.2.2.3 UC3: Úvodní formulář

Popis: Interakce s úvodním formulářem po přihlášení s výchozími hodnotami. Formulář lze vždy zavřít.

Aktér: Administrátor

Basic path:

1. Administrátor je přihlášen s výchozími hodnotami z dokumentace instalace.
2. Aplikace identifikuje použití výchozích hodnot.
3. Na obrazovce je přítomen úvodní formulář ve vyskakovacím okně v kroku s formulářem na změnu hesla.
4. Administrátor vyplní nové heslo a odešle požadavek.
5. Aplikace automaticky skočí na další krok úvodního formuláře s popisem aplikace.
6. Administrátor si přečte popis aplikace a překlikne na další krok s přidáním uživatele.
7. Administrátor přidá nového uživatele, nebo přejde na další krok.
8. Poslední krok nabídne vytvoření prostředku.
9. Administrátor uzavře formulář, nebo se automaticky uzavře po vytvoření prostředku.

2.2.2.4 UC4: Úvodní formulář v aplikaci bez záznamů uživatelů a prostředků

Popis: Interakce s úvodním formulářem po přihlášení již bez výchozích hodnot, ale bez žádného záznamu uživatele nebo prostředku v aplikaci. Formulář lze vždy zavřít.

Aktér: Administrátor

Basic path:

1. Aplikace identifikuje neexistující uživatele a prostředky v aplikaci.
2. Na obrazovce je přítomen úvodní formulář ve vyskakovacím okně v kroku s popisem aplikace.
3. Příklad užití pokračuje 6. krokem UC3.

2.2.2.5 UC5: Přidání prostředku libovolného typu

Popis: Přidání jak kompresoru, tak tlakové lahve nebo libovolného jiného typu.

Aktér: Administrátor

Basic path:

1. Administrátor otevře stránku s prostředky.
2. Klikne na tlačítko přidat prostředek.

3. Aplikace otevře formulář.
4. Administrátor zadá libovolné údaje o prostředku. Jediný povinný je jeho typ, který může být vlastní.
5. Administrátor klikne na tlačítko přidat.
6. Aplikace zavolá asynchronní dotaz na server.
7. Po úspěchu se stránka automaticky aktualizuje bez přenačtení stránky. Nový prostředek je přítomen v seznamu prostředků.

2.2.2.6 UC6: Úprava prostředku

Popis: Úprava hodnot prostředku kromě jeho typu. Zahrnuje přiřazení lahve k Jednotce SDH.

Aktér: Administrátor

Basic path:

1. Administrátor otevře stránku s prostředky.
2. Vybere záznam, který chce upravit.
3. Klikne na tlačítko pro editaci.
4. Aplikace otevře formulář pro úpravu s informacemi o prostředku. Pro přiřazení k jednotce nabízí formulář kolonku Jednotka, kde si Administrátor může zvolit libovolnou Jednotku z aplikace.
5. Administrátor upraví hodnoty prostředku a klikne na tlačítko *Upravit*.
6. Aplikace zavolá asynchronní dotaz na server.
7. Po úspěchu se stránka automaticky aktualizuje bez přenačtení stránky. Aktualizované hodnoty jsou již obsaženy v seznamu prostředků.

Alternate:

1. Administrátor otevře stránku s prostředky.
2. Vybere prostředek, který chce upravit a otevře jeho detailní stránku.
3. Klikne na tlačítko pro úpravu v horní části.
4. Aplikace otevře formulář pro editace s informacemi o prostředku.
5. Administrátor upraví hodnoty prostředku a klikne na tlačítko *Upravit*.
6. Aplikace zavolá asynchronní dotaz na server.
7. Po úspěchu se stránka automaticky aktualizuje s novými hodnotami bez přenačtení stránky.

2.2.2.7 UC7: Smazání prostředku

Popis: Smazání prostředku.

Aktér: Administrátor

Basic path:

1. Administrátor otevře stránku s prostředky.
2. Vybere záznam, který chce smazat.
3. Klikne na tlačítko pro mazání.
4. Potvrdí dotaz pro ověření.
5. Aplikace zavolá asynchronní dotaz na server.
6. Po úspěchu se stránka automaticky aktualizuje bez přenačtení stránky. Záznam o prostředku již není přítomen. Jsou i kaskádově smazány ostatní informace, které se váží na daný prostředek.

2.2.2.8 UC8: Přidání záznamu o plnění lahve

Popis: Přidání záznamu o plnění libovolné tlakové lahve.

Aktér: Administrátor, Technik

Basic path:

1. Uživatel s dostatečnými právy otevře stránku záznamů o plnění lahví.
2. Klikne na tlačítko *Přidat plnění*.
3. Aplikace otevře formulář.
4. Uživatel vyplní všechny povinné údaje formuláře. Vybere kompresor a láhev.
5. Administrátor klikne na tlačítko *Plnit*.
6. Aplikace zavolá asynchronní dotaz na server.
7. Po úspěchu se stránka automaticky aktualizuje bez přenačtení stránky. Nové plnění je obsažené v seznamu. Jsou i automaticky aktualizované záznamy lahve a kompresoru.

Alternate:

1. Administrátor otevře stránku s lahvemi.
2. Postupným klikáním na záznamy lahví vybere ty, které chce plnit kompresorem.
3. Klikne na tlačítko pro *Plnit* v horní části vedle nadpisu.
4. Aplikace otevře formulář pro plnění lahví ve vyskakovacím okně s předvyplněnými lahvemi.
5. Administrátor vyplní všechny povinné údaje formuláře. Vybere kompresor.

6. Administrátor klikne na tlačítko *Plnit*.
7. Aplikace zavolá asynchronní dotaz na server.
8. Po úspěchu se stránka automaticky aktualizuje, zavře se pop-up formulář.

2.2.2.9 UC9: Úprava záznamu o plnění lahve

Popis: Úprava plnění lahve v aplikaci.

Aktér: Administrátor, Technik

Basic path:

1. Uživatel s dostatečnými právy otevře stránku záznamů o plnění lahví.
2. Vybere záznam, který chce upravit.
3. Klikne na tlačítko pro úpravu.
4. Aplikace otevře formulář pro úpravu s informacemi o plnění.
5. Uživatel upraví hodnoty prostředku a klikne na tlačítko *Upravit*.
6. Aplikace zavolá asynchronní dotaz na server.
7. Po úspěchu se stránka automaticky aktualizuje s upravenými hodnotami bez přenačtení stránky.

2.2.2.10 UC10: Smazání plnění

Popis: Smazání záznamu o plnění tlakové lahve.

Aktér: Administrátor, Technik

Basic path:

1. Administrátor otevře stránku s plněním lahví.
2. Vybere záznam, který chce smazat.
3. Klikne na tlačítko pro mazání.
4. Potvrdí dotaz pro ověření.
5. Aplikace zavolá asynchronní dotaz na server.
6. Po úspěchu se stránka automaticky aktualizuje bez přenačtení stránky. Záznam o prostředku již není přítomen.

2.2.2.11 UC11: Přidání uživatele

Popis: Přidání dalšího uživatele do aplikace včetně vytvoření jeho hesla pro přihlášení.

Aktér: Administrátor

Basic path:

1. Administrátor otevře stránku s uživateli.
2. Klikne na tlačítko *Přidat uživatele*.

3. Aplikace otevře formulář.
4. Administrátor vyplní alespoň uživatelské jméno a heslo.
5. Administrátor klikne na tlačítko *Vytvořit*.
6. Aplikace zavolá asynchronní dotaz na server.
7. Po úspěchu se stránka automaticky aktualizuje bez přenačtení stránky. V aplikaci je již evidován uživatel, který se smí přihlásit, pokud Administrátor zaškrtnl položku Aktivní ve formuláři.

2.2.2.12 UC12: Úprava záznamu uživatele

Popis: Úprava uživatele včetně zneaktivnění.

Aktér: Administrátor

Basic path:

1. Administrátor otevře stránku s uživateli.
2. Vybere záznam, který chce upravit.
3. Klikne na tlačítko pro úpravu.
4. Aplikace otevře formulář pro úpravu s informacemi o uživateli.
5. Administrátor upraví hodnoty a klikne na tlačítko *Upravit*. Pokud chce Administrátor uživatele zneaktivnit, pak vynuluje kolonku Aktivní.
6. Aplikace zavolá asynchronní dotaz na server.
7. Po úspěchu se stránka automaticky aktualizuje s upravenými hodnotami bez přenačtení stránky.

2.2.2.13 UC13: Přepnutí tmavého a světlého režimu

Popis: Příklad užití aplikace, který popisuje, jak bude fungovat automatické přepínání režimu světlosti.

Aktér: Všechny role

Basic path:

1. Uživatel otevře libovolnou stránku v libovolném stavu aplikace.
2. Na svém zařízení přepne režim světlosti systému.
3. Aplikace se automaticky přepne do adekvátního stavu. Nedojde k žádné jiné změně stavu kromě této změny světlosti. Ani přenačtení obsahu nebo obnovy libovolného elementu.

2.2.2.14 UC14: Přidání jednotky

Popis: Přidání záznamu jednotky do aplikace.

Aktér: Administrátor

Basic path:

1. Administrátor otevře stránku s jednotkami.
2. Klikne na tlačítko *Přidat jednotku*.
3. Aplikace otevře formulář.
4. Administrátor zadá libovolné údaje o jednotce. Jediný povinný je její název.
5. Administrátor klikne na tlačítko *Vytvořit*.
6. Aplikace zavolá asynchronní dotaz na server.
7. Po úspěchu se stránka automaticky aktualizuje bez přenačtení stránky. Nová jednotka je v seznamu.

2.2.2.15 UC15: Úprava jednotky

Popis: Úprava údajů o jednotce.

Aktér: Administrátor

Basic path:

1. Administrátor otevře stránku s jednotkami.
2. Vybere záznam, který chce upravit.
3. Klikne na tlačítko resp. ikonku pro editaci.
4. Aplikace otevře formulář pro úpravu s vyplněnými informacemi o jednotce.
5. Administrátor upraví hodnoty jednotky a klikne na tlačítko *Upravit*.
6. Aplikace zavolá asynchronní dotaz na server.
7. Po úspěchu se stránka automaticky aktualizuje bez přenačtení stránky. Aktualizované hodnoty jsou již v seznamu.

2.2.2.16 UC16: Smazání jednotky

Popis: Smazání záznamu jednotky.

Aktér: Administrátor

Basic path:

1. Administrátor otevře stránku s jednotkami.
2. Vybere záznam, který chce smazat.
3. Klikne na tlačítko resp. ikonu pro mazání.
4. Potvrdí dotaz pro ověření.
5. Aplikace zavolá asynchronní dotaz na server.
6. Po úspěchu se stránka automaticky aktualizuje bez přenačtení stránky. Záznam jednotky již není přítomen, automaticky se smažou vazby prostředků na jednotky.

2.2.3 Diagramy aktivit

Uvádím dva diagramy aktivit, které navazují na případ užití Přihlášení a na případ užití Přidání a Úpravu záznamu o plnění lahve.

První diagram Obrázek 2.2 popisuje správu sezení v aplikaci. Sezení neboli Session v aplikaci je nastaveno pomocí JWT. Kdy tento token obsahuje informaci o validní autorizaci podepsanou soukromým klíčem, kterou si klient nebo server následně ověřuje pomocí veřejného klíče. Tento token expiruje již po jedné hodině. Proto je potřeba řádně popsat aktivitu, kdy token expiruje například při plnění formuláře.

Jsou zde dva scénáře. První je automatická kontrola tokenu na straně klienta, který po jeho expiraci zobrazí Přihlašovací formulář. Druhý je při odeslání požadavku na server s již prošlým tokenem. Server pak odpoví chybou klienta se status kódem 400-499. V obou případech je nutné se znovu přihlásit. Klient ale zobrazí formulář přihlášení přes aktuální stránku, proto uživatel nepřijde o rozdělanou práci.

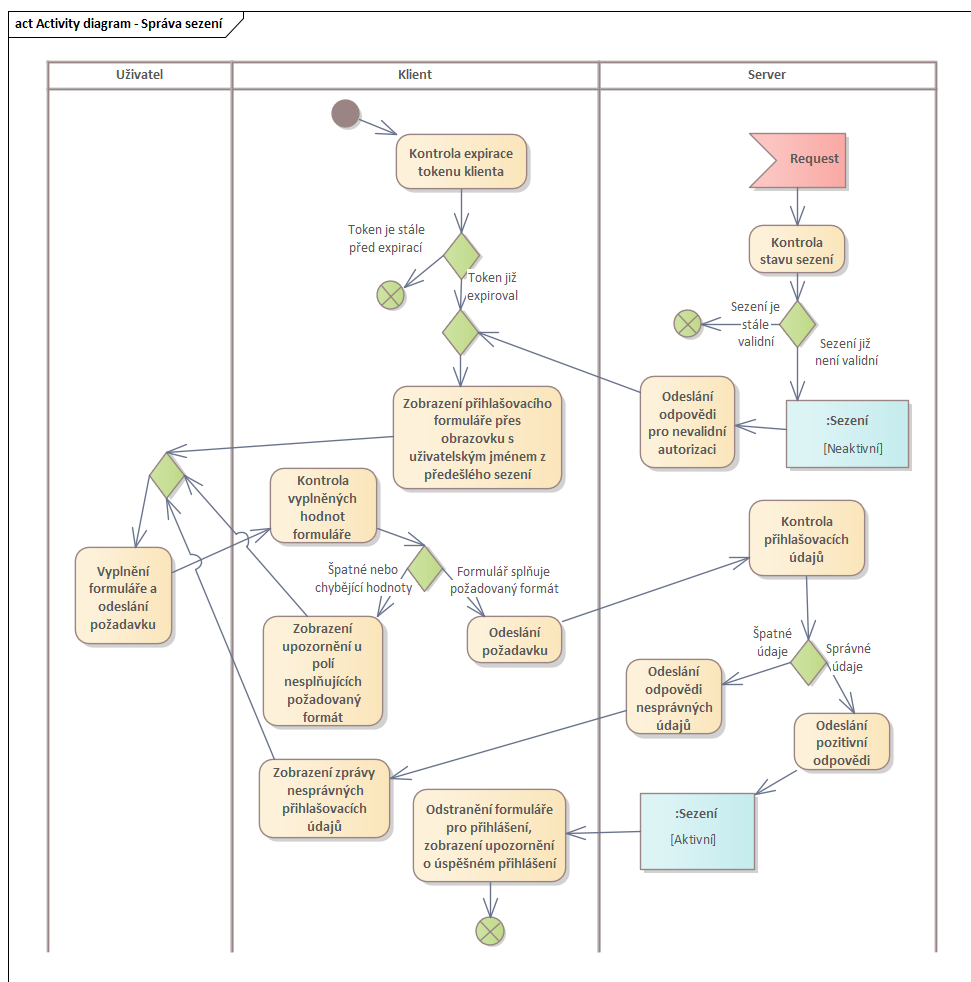
Druhý diagram Obrázek 2.3 popisuje aktivitu vytvoření nebo editaci záznamu Plnění lahve. Samotná aktivita je poměrně přímočará. Hodnoty požadavku jsou kontrolovány na straně klienta i serveru. Práva přihlášeného uživatele jsou také kontrolována na obou stranách. Čtenář si může všimnout vyvolané události po odeslání požadavku na server. Tato událost odkazuje na Obrázek 2.2 kdy, pokud je token již prošlý, odpoví chybou a přes obrazovku je zobrazen formulář přihlášení. Podobně lze definovat i aktivity pro přidání nebo úpravu ostatních typů entit.

2.3 Volba technologií

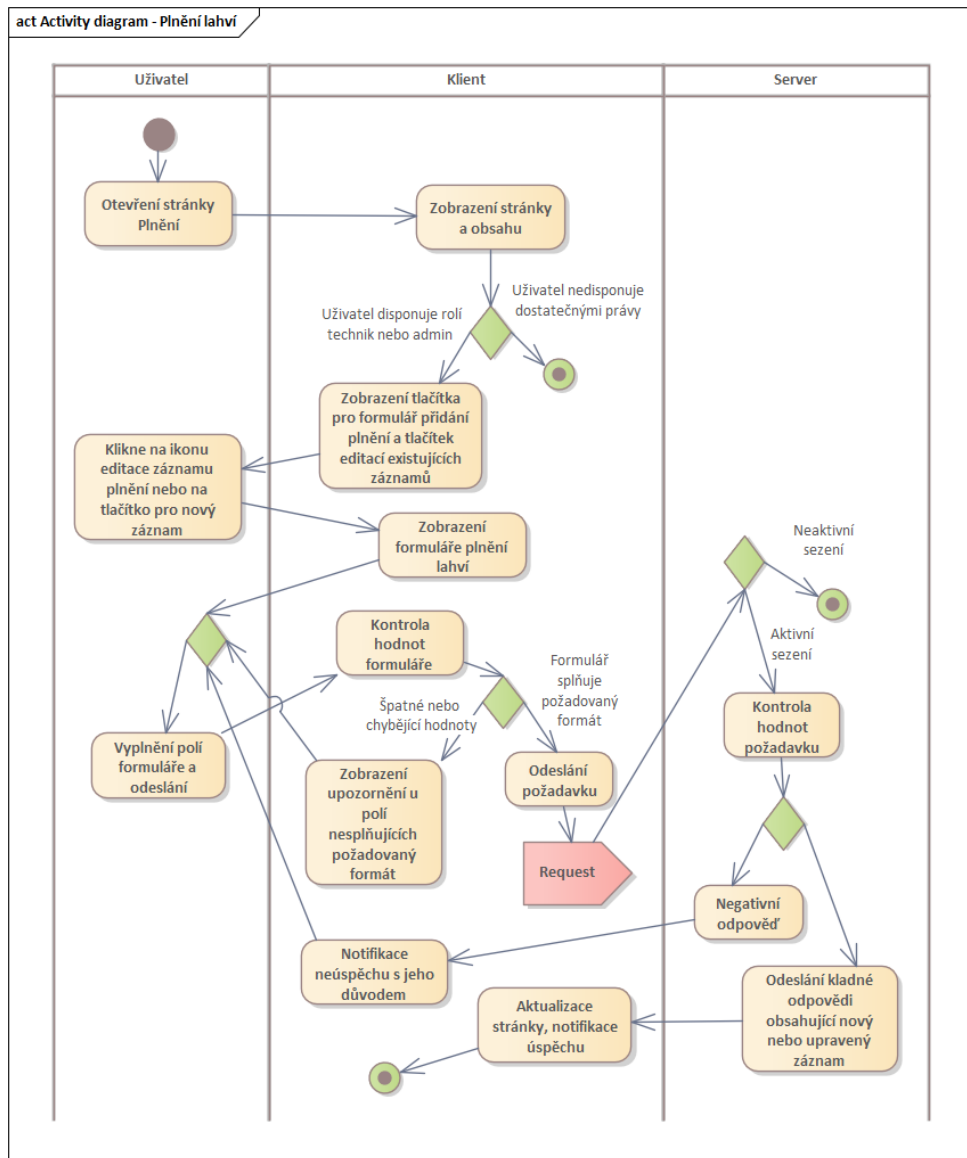
Klasická webová aplikace se dělí na několik vrstev. Perzistentní, logickou a prezentační. Pro každou z těchto vrstev je potřeba zvolit vhodnou technologii, která se může i doplňovat s dalšími v správně fungující celek. Je vždy mnoho faktorů, které volbu ovlivňují. Může to být finanční náročnost prostředků, jako licence za databázi, nebo nějaké předplatné. Další hosting, resp. hardware pokud je hostování na vlastním zařízení, a od něj se odvíjející cena. Jako další faktor uvádím framework nebo programovací jazyk, který ovládají programátoři. Postupně rozvádím, jaká jsem volil rozhodnutí s jejich odůvodněním v závislosti na těchto faktorech.

2.3.1 Volba technologie pro perzistentní vrstvu

Samotný webový provozní deník je určený jednotkám dobrovolných hasičských sborů, které nedisponují většími financemi v porovnání s komerčními firmami, které si mohou dovolit svoji aplikaci hostovat na různých cloudových službách nebo vlastním dedikovaném serveru za stovky tisíc korun. Předpoklad je jednoduchý hosting založený na Apache httpd s PHP, ke kterému je následně



■ Obrázek 2.2 Diagram aktivit – Správa sezení, Enterprise architect, vlastní tvorba



■ Obrázek 2.3 Diagram aktivit – Plnění lahví, Enterprise architect, vlastní tvorba

k dispozici historicky zpravidla MySQL nebo nověji PostgreSQL. Protože je cílem jednoduchá instalace, která nevyžaduje vyšší zkušenosti plynoucí z různých možností nastavení a připojení databáze, které jsou nabízeny, rozhodl jsem se zvolit technologii SQLite. Tato databáze je vhodná na malé projekty. Je zdarma a její kompletní instalace představuje jeden soubor, se kterým komunikuje framework. Je tedy nezávislá na hostingu, a také proto je poměrně oblíbená. Pro mě tedy představuje ideální volbu.

2.3.2 Důvody pro PHP

PHP je programovací jazyk používaný převážně pro vytváření webových aplikací. Jedná o interpretovaný jazyk, a je tedy nezávislý na platformě. Na cenově dostupných webhostingových službách je PHP standard. Jeho nevýhoda je, že je poměrně pomalý. Při každém dotazu se musí načíst všechny nutné knihovny a tento celkový kód následně převést do strojového kódu, který není tak optimalizovaný jako je například kód zkompilovaný z jazyka C++. Proto je PHP nevhodné pro enterprise aplikace s deseti tisíci uživateli jako jsou e-shopy, nebo jiné služby otevřené lidem.

Počet členů jednotlivých SDH nepřesahuje ani stovky a očekává se, že budou aplikaci hostovat lokálně a jen pro vlastní sbor. Trochu větší latence je sice nepříjemná, ale není větší omezení. Finanční výhoda ji převyšuje.

S PHP mám kladnou zkušenost. Není nijak složitý na provoz, jako například Java nebo Node.js, u kterých je nutné, aby běžely v aktivní procesu, a proto jsou náročnější na korektní instalaci, nastavení a správu. To by limitovalo potencionální jednotky na rozběhnutí aplikace, pokud by musely hledat správce systémů, kteří jsou jim ochotni pomoci s provozem a instalací takového runtime systému.

2.3.3 Framework Symfony

Logická vrstva aplikace, pro kterou jsem zvolil jazyk PHP, potřebuje zvolit vhodný framework. Ten zajistí základní funkce a skeleton, na kterém lze stavět nové funkce. V tomto případě hlavně rozhraní pro komunikaci s klientem.

Framework Symfony je populární framework, který tak nabízí možnost pro ostatní vývojáře k připojení k tomuto projektu. Při jeho instalaci lze zvolit skeleton pro REST API, kde je obsažena komunikace s perzistentní vrstvou. Zároveň obsahuje zabezpečení rozhraní a nabízí mnoho knihoven třetích stran, které lze použít například pro autentizaci a autorizaci uživatelů.

Jeho použití je přímočaré a dává vývojáři prostor se soustředit na prezentační vrstvu aplikace, protože mnoho podstatných funkcionalit stačí pouze nakonfigurovat a definovat. Knihovna Doctrine [21] mapuje objekty a ukládá je do databáze. Pro autentizaci jsem zvolil LexikJWTAuthenticationBundle [22], který nabízí správu JWT¹. Pro tvorbu API jsem použil Api-platform [23].

Zbylé použité knihovny byly převážně přímo z frameworku Symfony.

2.3.4 Knihovna Api-platform

V logické vrstvě, jak jsem zmínil v předešlém odstavci, jsem zvolil pro definici rozhraní knihovnu Api-platform. Tato knihovna automaticky vytváří koncové body pro jednotlivé CRUD² operace nad definovanými entitami. Prezentační vrstva je napsaná v JS frameworku. Díky němu jsem schopný umístit většinu logiky právě tam. Chtěl jsem také, aby byla aplikace rychlejší na načítání, to mi nabízí právě JS framework, jako React nebo Angular, který nenačítá stránky ze serveru, ale pouze se asynchronně dotazuje serveru na data a všechny změny stránky jsou vytvořeny pomocí Javascriptu.

Všechny koncové body rozhraní jsou vytvořené automaticky podle očekávaného chování správného REST API. Knihovna Api-platform umí pracovat s entitami automaticky. Nabízí také možnost zabezpečení podle rolí uživatele a to i pro jednotlivé metody zvlášť. V aplikaci jsem například rozdělil role pro admina, technika, manipulátora nebo jen přihlášeného uživatele, kde pouze admin smí například zavolat POST na entitu User, a tím vytvořit nového uživatele. Jednoduché je také propojení s JWT. LexikJWTAuthenticationBundle vystaví vlastní koncový bod pro přihlášení. Api-platform následně pro každý dotaz, kde je povinná role přihlášeného uživatele, jej zkontroluje automaticky. O autorizaci se ale stará samotná Symfony.

Všechny definice zveřejněných koncových bodů se definují pomocí anotací u definic entit. Záleží podle verze PHP a frameworku, jestli podporují anotace. V aplikaci jsem tedy například zakázal PUT operace nad entitami tak, že jsem definoval pouze povolené a PUT mezi ně nepřidal. Příjemné je také používání anotací u atributů třídy. Lze tak kontrolovat minimální hodnoty, nebo vyvolávat výjimky u prázdných hodnot povinných atributů.

2.3.5 JS framework

Pro prezentační vrstvu jsem využil Javascript framework. Jak jsem zmínil v odstavci výše, jako hlavní benefit je rychlost načítání a vytváření změn. Kdy se všechno odehrává na straně klienta a na server se pouze odesílají požadavky se změnami nebo dotazy pro data. Jsou případy, kdy na straně klienta není možné odeslat asynchronní dotaz a nečekat na odpověď, jako při přihlášení nebo přidání nového záznamu. Tam pak odpadá výhoda asynchronního dotazování. I tak lze ale očekávat rychlejší odezvu, neboť není nutné generovat novou stránku, která je často téměř identická.

¹JSON Web Token (JWT) je otevřený standard (RFC 7519), který definuje kompaktní a samostatný způsob bezpečného přenosu informací mezi stranami ve formě objektu JSON. [24]

²Create, Read, Update, Delete

Další pro mě důležitou výhodou je samotný jazyk Javascript, který je také interpretovaný. Není nutné pro každou změnu stránku znovu kompilovat. Moderní frameworky dokáží automaticky aktualizovat téměř ihned stránku při každé změně v kódu. Programování je tedy velmi intuitivní. Je dynamicky typovaný a je silně objektově orientovaný. Vím, že aplikace nebude mít velký rozsah a nebude náročná na výpočetní výkon, proto jsou tyto aspekty pro mne výhodou, která mi pomáhá v rychlosti vývoje.

Nabízí se několik možností volby. Zmínil bych hlavně React [25], Angular, nebo pak deriváty Reactu jako Next.js. Osobně mám dobrou zkušenost s Reactem, ke kterému se následně instalují různé knihovny s dalšími funkcionalitami.

Další důležitou otázkou je, zda budu psát aplikaci v Typescriptu. Ten dle mého názoru slouží hlavně pro udržitelnost kódu do budoucna. Je také dobrým pomocníkem pro vývoj ve větším počtu vývojářů. Pro mne by ale představovat zpomalení vývoje a málo výhod. Do budoucna lze typy do kódu kdykoli přidat.

2.4 Diagram nasazení

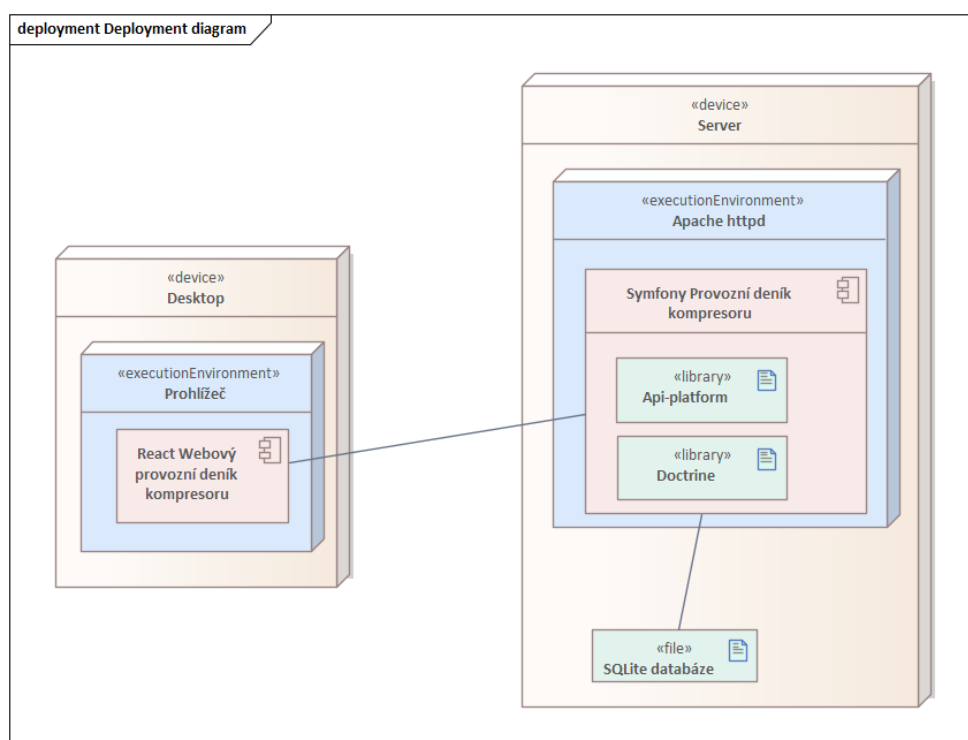
Jak jsem již zmínil v předchozí Sekci 2.3, zvolil jsem framework Symfony a React hlavně kvůli požadavku na finančně dostupný hosting pro JSDH. Klasicky se aplikace v jazyce PHP nasazují na Apache httpd, ten je často používán i u většiny poměrně levných hostingů pro jednoduché statické nebo dynamické webové stránky. Dnes je populární také technologie Nginx. Soubory klienta jsou dostupné jako statické soubory, které si uživatel stáhne a načte v prohlížeči. Lze je jednoduše „cachovat“.

Vytvořil jsem diagram nasazení Obrázek 2.4, kde jsem zvolil technologii Apache httpd, která má povolený module PHP verze 8.2 a výše. Popisuje základní představu možného nasazení aplikace. Desktop představuje na tomto diagramu stolní počítač. Webový provozní deník je ale možné používat z libovolného zařízení, které disponuje internetovým prohlížečem podporující Javascript. Jedná se tedy o základní představu možného nasazení aplikace. Databáze je uložena na stejném systému jako běží Apache httpd. Jelikož se jedná o samostatný soubor, je nutné k němu definovat cestu v souboru prostředí frameworku Symfony, pokud je uložen na jiném, než výchozím místě. Například pokud je zálohován.

2.5 Architektura perzistentní vrstvy

Zvolil jsem databázi SQLite [26]. Na tuto databázi se napojí framework tedy Symfony, který s ní komunikuje. V závislých knihovnách, které jsem nainstaloval se Symfony, je již zahrnutý ovladač pro SQLite, konfigurace tedy představuje pouze v souboru s proměnnými prostředí zadefinovat cestu k souboru.

```
DATABASE_URL="sqlite:///kernel.project_dir%/var/data.db"
```



■ **Obrázek 2.4** Diagram nasazení aplikace na Apache httpd, Enterprise architect, vlastní tvorba

Symfony nabízí interaktivní možnost pro vytvoření jednotlivých entit. Tyto entity jsou následně nahrány do databáze pomocí takzvané migrace. Programátor se tedy nepotká se psaním definicí jednotlivých tabulek, pokud se sám nerozhodne upravit jejich definice. Nechal jsem si pomocí této interaktivní metody vygenerovat celou architekturu, pro jednotlivé entity doménového modelu, které jsem si nechal schválit vedoucím práce.

2.5.1 Doménový model

Samotný doménový model si můžete prohlédnout na Obrázku 2.5.

Hlavní entitou je Předmět. Ten představuje jakýkoliv prostředek, který chtějí hasiči evidovat. Láhev či Kompresor jsou ISA³ tohoto Předmětu, tedy jsou vždy a pouze jeden Předmět. Nejdůležitější entita je pak Plnění, která je navázána právě na Kompresor a na Láhev.

Osoba představuje uživatele. Chtěl bych podotknout, že uživatele nebude možné smazat, ale pouze deaktivovat. K tomu slouží jeho atribut *aktivní*. Důvod je hlavně ten, aby bylo možné zobrazit historické záznamy s údaji, kdo je provedl.

Doménový model následně nabízí několik vazeb mezi Osobou a Předmětem. Je to Revize, tedy že uživatel provedl revizi daného předmětu. Kompresory a další prostředky mohou mít různé revize. Aplikace tak zde nabízí volné místo pro rozvoj, tak aby šlo například revize různě diverzifikovat. Další je Použití. JSDH jsou často povinny určité prostředky pravidelně používat. Proto je praktické evidovat jejich použití s datem. Poslední je pak Přemístění.

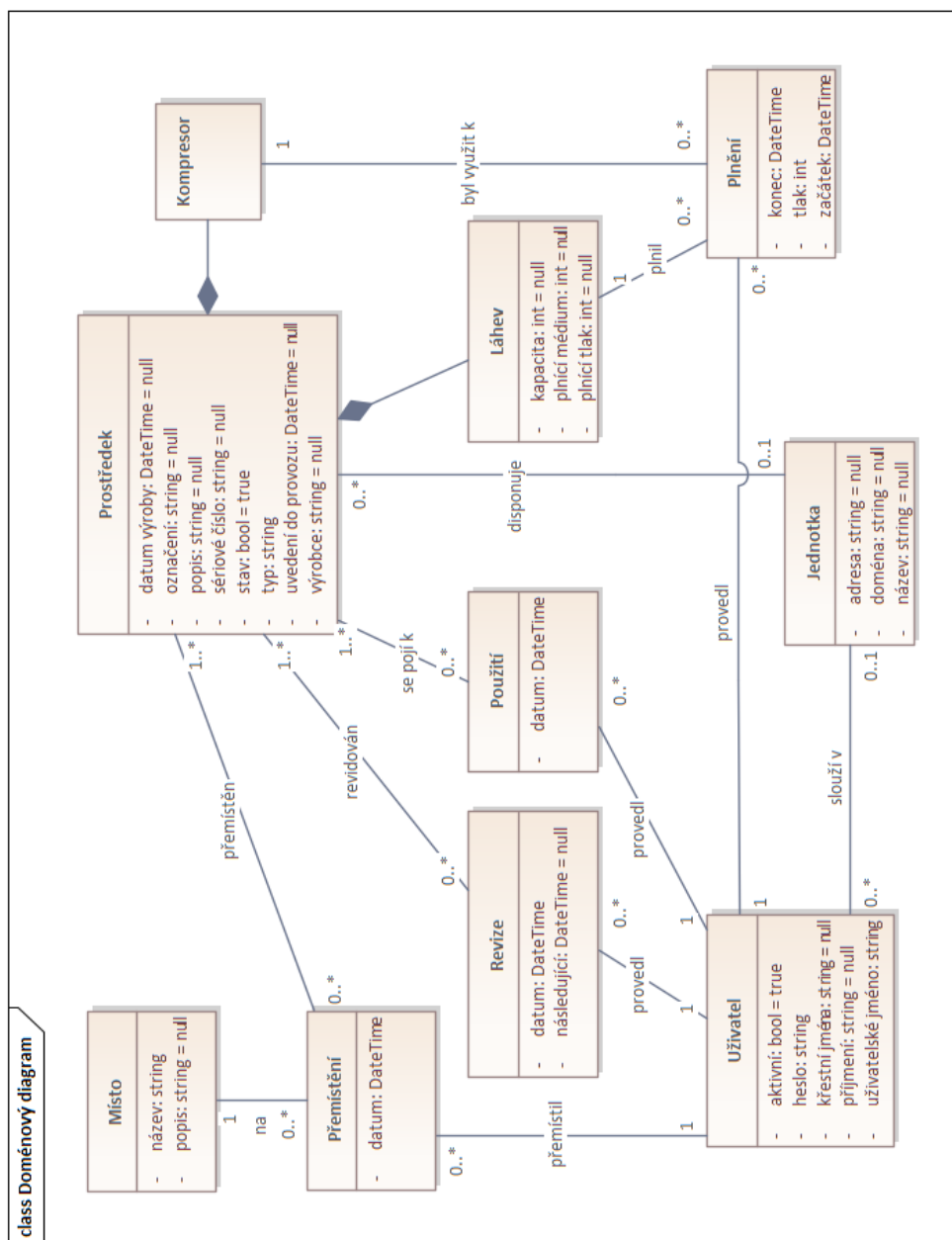
Entita Přemístění slouží pro popsání aktuálního místa prostředku. Navrhl jsem doménový model tak, aby bylo vidět historické umístění prostředku. Proto právě to poslední umístění představuje místo, kde se prostředek momentálně nachází. Místa mohou být různá. Například revize, nebo jednotlivé názvy jednotek, pokud jednotka plní lahve pro ostatní jednotky, nebo prostě názvy praktické jako jednotlivé CAS, místnosti atp.

Poslední entitou je entita Jednotka, ta pouze pomáhá v evidenci a přehlednosti. Podle názvu Jednotky lze například záznamy různě řadit nebo filtrovat. Předměty jsou vždy vlastněny Jednotkou a zároveň každý Uživatel by měl spadat do nějaké Jednotky.

2.6 Logická vrstva

V této kapitole rozepisuji jaké jsem vytvořil rozhraní logické vrstvy. Pro definici rozhraní jsem zvolil knihovnu Api-platform viz. Sekce 2.3.4. Tato knihovna nabízí velmi intuitivní tvorbu API, která je založená na definicích entit, pro které automaticky vygeneruje rozhraní. Začnu tedy rozepsáním definicí entit.

³Zkratka z anglického "Is a". Podtyp, který nepotřebuje identifikátor.



Obrázek 2.5 Doménový model perzistentní vrstvy, Enterprise architect, vlastní tvorba

■ Výpis kódu 2.1 Zakázání změny hodnoty typu

```
/**
 * @throws AttributeImmutableException
 */
public function setType(string $type): static
{
    if ($this->type !== null) {
        throw new AttributeImmutableException();
    }

    $this->type = $type;

    return $this;
}
```

2.6.1 Definice entit

V Symfony je každá entita definována pomocí interaktivního terminálu zavo-láním příkazu.

```
php bin/console make:entity
```

Výhodou je také pomoc při vytváření vazeb. Definice jednotlivých atributů nabízejí klasické možnosti párování včetně ManyToMany. Pro ty se následně automaticky vytvoří potřebné tabulky. U párování jsem řešil, jak správně defi-novat kaskádové mazání. Pokud chci smazat Místo, pak očekávám, že automa-ticky dojde ke smazání i Umístění na tomto místě. Naštěstí existuje anotace, která u atributu vazby definuje kaskádové mazání. Interaktivní terminál mi takovou možnost nenabídl. Pro ISA⁴ mezi Předmětem a Lahví nebo Kompresorem, jsem použil tzv. LifecycleCallback, který, pokud je mazán Předmět, smaže i entity Lahve nebo Kompresoru, u kterých je definované kaskádové mazání pro Plnění.

Pro různé atributy jsem také požadoval, aby nebyly nikdy prázdné (při změně nebo vložení). Řídil jsem se podle vážnosti atributu v aplikaci, proto jsem například zakázal prázdnou hodnotu atributu u entity Plnění, název u Místa nebo datum u Použití. U ostatních entit jsem byl poměrně benevo-lentní k povinnosti vyplnění hodnoty atributů. Není tedy nutné vyplnit Jed-notku u uživatele, nebo jakýkoli atribut u Předmětu kromě typu. Ten jsem podmínkou v setType funkci zakázal upravovat Výpis kódu 2.1. Lze nastavit pouze pokud je rovný null hodnotě. Ta je ale povolena pouze jako iniciační hodnota, jelikož je typ povinný atribut při vytváření.

⁴Zkratka z anglického "Is a". Podtyp, který nepotřebuje identifikátor.

■ Výpis kódu 2.2 Kontrola stavu uživatele

```
public function checkPreAuth(UserInterface $user): void
{
    // Check if the user is active
    if (!$user instanceof User) {
        return;
    }

    if ($user->isActive() === false) {
        throw new UserInactiveException();
    }
}
```

Pro uživatele, kteří byli zneaktivněni adminem, je potřeba zamezit jejich přihlášení do aplikace. Pro autentizaci používám knihovnu LexikJWTAuthenticationBundle, která se stará o kontrolu validního tokenu. Symfony umožňuje implementovat rozhraní `UserCheckerInterface`. To slouží právě pro kontrolu objektu `Uživatel`. Lze tak definovat libovolná práva. Přidal jsem tedy jednoduchou podmínku, která vyvolá výjimku, pokud je příslušný atribut objektu nastaven na `false`. Výpis kódu 2.2

2.6.2 Role uživatelů

U jednotlivých entit bylo nutné omezit přístup podle rolí aktuálně přihlášeného uživatele. Všechny koncové body API jsem zakázal pro nepřihlášeného uživatele, protože se jedná o interní aplikaci. Jediný povolený koncový bod je logicky pro přihlášení. Následně jsem povolil všechny GET metody pro přihlášeného uživatele. Odbornost se vztahuje hlavně na úpravy a přidávání, proto není čtení záznamů omezené na vyšší práva než je přihlášený uživatel. Bylo tak třeba vyřešit roli technika a roli manipulátora. Manipulátor je role, která nesplňuje specifickou odbornost, je to spíše osoba, která má na starosti umístění jednotlivých prostředků, aby si všichni nepřesouvali věci pod rukama. Technik má pak navíc povoleno přidávat a upravovat Plnění. Manipulátor to stejné ale u Umístění. Povedlo se mi tak poměrně jednoduše rozdělit jejich pravomoci pouze pomocí jednoduchých anotací u entit viz. Výpis kódu 2.3. Poslední rolí je admin, který nemá omezení. V aplikaci je výchozí jeden uživatel admin, který je po přihlášení vyzván na změnu hesla.

2.6.3 Povolené metody

U každé entity je tedy povolena metoda GET. Api-platform nabízí také `Get-Collection`, která vrátí instance všechny nebo stránkovane v jedné odpovědi.

■ Výpis kódu 2.3 Anotace entity Plnění

```
#[ApiResponse(security: "is_granted('ROLE_ADMIN')")]
#[GetCollection(security: "is_granted('ROLE_USER')")]
#[Get(security: "is_granted('ROLE_USER')")]
#[Post(security: "is_granted('ROLE_TECHNICIAN')",
    validationContext: ['groups' => ['Default', 'create']])]
#[Patch(security: "is_granted('ROLE_TECHNICIAN')")]
#[Delete]
#[ORM\Entity(repositoryClass: FillingRepository::class)]
#[ORM\HasLifecycleCallbacks]
```

POST, PATCH a DELETE jsou vždy povoleny pro admina. U Plnění je povolena metoda POST a PATCH pro roli Technik a u Umístění pro roli Manipulátor. Metodu PUT jsem zakázal. POST a PATCH jsou její dostatečnou náhradou, u klienta tak nemám potřebu používat metodu PUT.

2.6.4 Testovací data

Osobně při vývoji považuji za velmi praktické pracovat s aplikací, která již obsahuje sice vymyšlená, ale alespoň nějaká data. Vývojář má díky tomu lepší představu, jak se aplikace bude chovat. Lze ihned jednoduše testovat případy užití. Knihovna Doctrine nabízí rozhraní Fixture, které slouží právě pro toto. Třída, která toto rozhraní rozvíjí, lze následně zavolat jako příkaz a databáze se naplní vymyšlenými hodnotami podle definice.

Vytvořil jsem tedy třídu pro každou entitu. Navzájem se seřadí, tak aby se do databáze vkládaly nejdříve hodnoty entit, které nejsou na jiných závislé, takovou je například entita Jednotka. Většinu hodnot jsem předdefinoval, jejich vazby, většina numerických hodnot a prázdné hodnoty jsou ale čistě náhodné. Lze proto pro každé zavolání vytvoření těchto vymyšlených hodnot vytvořit zdánlivě jedinečný set.

2.7 Prezentační vrstva

Klient je hlavní část aplikace, kam jsem umístil i většinu logiky. Zvolil jsem framework React s nativním Javascriptem. Chtěl bych zmínit, že jsem klienta psal v programu Visual Studio Code [27]. Ten nabízí instalaci různých rozšíření, která pomáhají s prací. Jako hlavní je rozšíření ES7+ React/Redux/React-Native snippets pro usnadnění práce s Reactem, ESLint pro kontrolu kódu JS, GitLens pro práci s verzovacím nástrojem Git. Prettier je výborné rozšíření, které formátuje prakticky všechny typy souborů, se kterými jsem se setkal při vývoji. Poslední je rozšíření Sass, které slouží pro formátování a napovídání

.sass souborů, tedy stylů aplikace.

2.7.1 Volba fontu

Volba fontu je dle mého názoru jedna z nejdůležitějších faktorů ovlivňující celkový vzhled aplikace. V České republice je potřeba splnit podmínku, aby font nabízel znaky pro rozšířenou latinku. Fonty se rozlišují, jestli jsou placené nebo s volnou licencí. Podle těchto podmínek jsem volil na webové aplikaci Google Fonts [28], která nabízí uživatelky příjemnou orientaci mezi různými třídami. Nejvíce vhodný mi přišel Poppins, který je momentálně velmi populární. Samotnou instalaci fontu jsem ale provedl přes import souborů .woff2, které obsahují jeho definice. Zajistil jsem tak rychlejší načítání aplikace a zároveň neposkytuje informace Googlu, při každém stažení této třídy.

2.7.2 Použité knihovny

V rámci projektu jsem integroval několik klíčových knihoven třetích stran, které mi umožnily efektivně implementovat různé funkcionality v aplikaci.

Axios [29] – Tuto knihovnu jsem použil pro asynchronní dotazování na server. Axios nabízí příjemné rozhraní pro dotazování a využívá Promises pro zpracování asynchronních operací.

Formik [30] – Pro validaci a vytváření formulářů jsem zvolil knihovnu Formik. Tato knihovna usnadňuje validaci formulářů na straně klienta a poskytuje nástroje pro zpracování a manipulaci s hodnotami jednotlivých polí ve formuláři.

React Redux [31] – Jedná se o klíčovou knihovnu pro správu stavu aplikace na straně klienta. React Redux umožňuje centralizované ukládání a správu dat, díky čemuž mohou různé části aplikace jednoduše číst a upravovat data. Všechny změny dat se automaticky propagují do všech částí aplikace.

React Tables [32] – Klíčová knihovna pro renderování tabulek z dat. Této knihovně se předávají data, definice sloupců a popřípadě funkce pro řazení hodnot ve sloupcích. Knihovna se následně postará o jejich vykreslení. Nabízí další funkce, například vybírání záznamů po kliknutí.

React Router [33] – Ve webovém prohlížeči zpracovává jednotlivé stránky aplikace tak, že prohlížeč pouze vyměňuje obsah, ale k reálnému načtení jednotlivých URL nedochází. Lze tak držet stav aplikace bez nutnosti neustálého dotazování serveru při přeskokování zpět a vpřed.

Moment [34] – Moment pracuje s daty, která umí načíst z textu nebo je zapsat. Data lze přeformátovat, porovnávat nebo různě zobrazovat.

2.7.3 Struktura kódu aplikace

Celou aplikaci jsem rozdělil do několika stránek – Domovská stránka, Předměty, Kompresory, Láhve, Plnění, Místa, Jednotky, Uživatelé. Stránka Předměty a Uživatelé má i podstránku, resp. detail jednoho Předmětu nebo jednoho Uživatele. Vytvořil jsem také stránku pro chybu, která vykreslí text s odkazem na domovskou stránku.

Existuje několik globálních komponent, které nejsou vázány na žádnou specifickou stránku. Těmi jsou navigace, komponenta vykreslující načítání, a pak pomocné komponenty odkazu, tabulky nebo položky ve formuláři pro výběr hodnoty ze sady možností.

Definoval jsem také několik podstatných tříd. Jednou je definice tzv. Redux Storů, tedy míst, kde ukládá knihovna Redux data. Další třída je k usnadnění vytváření dotazů. Tato třída interně používá knihovnu Axios. Přidává například hlavičku s autorizačním tokenem. Poslední takovou externí třídou je spíše definice pro React Router, která udává jednotlivé cesty pro komponenty stránek.

Některé stránky využívají různé vlastní komponenty. Ty jsou často právě nějakou nadstavbou nad globálními komponentami. Snažil jsem se recyklovat jednotlivé komponenty tak, že například komponenta formuláře pro nové Plnění je stejná jak pro stránku Předmětu tak pro stránku Uživatele.

2.7.4 Struktura stránky aplikace

Každá stránka zobrazuje jednu nebo více tabulek. U těchto tabulek může, pokud má uživatel dostatečné práva, být umístěn i formulář, který zpravidla vytvoří nový záznam, nebo nějaký upravuje. U stránek detailu předmětu nebo uživatele lze také editovat jejich informace.

Domovská stránka obsahuje tři tabulky. Slouží pro zobrazení důležitých informací, které kontrolují různé podmínky, které musí hasičský sbor plnit.

1. První zobrazuje nejbližší revize prostředků.
2. Každá láhev nesmí být plná déle než půl roku bez použití. K tomu slouží druhá tabulka.
3. Poslední zobrazuje uživatele, kteří nepoužili láhev za poslední 3 měsíce.

Takto aplikace plní její předpoklad, tedy webový provozní deník. Fyzický deník je zpravidla sešit, který obsahuje jednotlivé záznamy. Aplikace tedy je takový online interaktivní sešit. Nabízí také více obecné informace, které lze vyčíst ze záznamů. U předmětu jsou to poslední místo, tedy místo posledního umístění, nebo datum poslední revize.

2.8 Testování aplikace

Testovat bylo potřeba jak prezentační vrstvu, tak i logickou vrstvu. Testy přinášejí mnoho benefitů. Může se tak předejít vložení nechtěné chyby při budoucí implementaci. Na logické vrstvě jsem otestoval zabezpečení koncových bodů rozhraní. Samotná logika je zde téměř nepřítomná díky použití knihovny Api-platform. U Klienta jsem otestoval případy užití.

2.8.1 Testy logické vrstvy

Zde jsem využil rozhraní ApiTestCase od knihovny Api-platform. Otestoval jsem několik případů. Tyto testy pracují s proměnnými testovacího prostředí. Celkem testy zahrnují přes 200 assercí.

Testy autorizace. Tento test je přímočarý. Nejdříve vytvoří uživatele, pokud již není přítomen. Tento uživatel následně odešle autorizační požadavek na koncový bod rozhraní pro přihlášení. Je očekávána odpověď, která obsahuje autorizační token. S tímto tokenem se následně pošle GET požadavek, a očekává se validní odpověď. GET požadavky jsou vždy povoleny.

Testování rozhraní podle práv uživatele. Při tomto testu se zkouší různé kombinace metod, koncových bodů a rolí uživatele pro každou entitu. Celkem se tak otestuje 230 kombinací. Každý test, který je rozdělen podle role uživatele. Nejprve nastaví uživateli patřičnou roli, se kterou následně testuje koncový bod. Kontroluje se status kód odpovědi.

Test přihlášení neaktivního uživatele. Při tomto testu je kontrolováno, jestli uživatel má možnost se přihlásit, pokud je nastaven jeho stav jako neaktivní. Následně se otestuje znovu, ale již s aktivním stavem.

2.8.2 Testy prezentační vrstvy

Kontrola jednotlivých funkcionalit u klienta je kontrolována testy prostředí, kdy testeři vyzkouší danou funkci a kladou důraz na její dostatečně přehlednou a srozumitelnou odpověď. Je tak otestováno například automatické přepínání do tmavého režimu, jestli jsou texty čitelné a obrázky srozumitelné. Také je otestována responzivita a použitelnost aplikace na mobilním zařízení. Operace přidání a odebrání záznamu s kontrolou domovské stránky.

V aplikaci jsem definoval unit testy, které testují základní funkce klienta. Pro tyto testy jsem použil knihovnu Jest. Je potřeba řádně nakonfigurovat, aby správně načítala jednotlivé skripty. Otestoval jsem tak tzv. helper funkce. Ty především zpřehledňují kód, jsou importovány a použity několika komponentami. Dále jsem otestoval funkci, která vytváří dotazy na server. Zkontroloval jsem, jestli dotaz obsahuje správné hlavičky a autentizační token. Dále jsem také otestoval Redux Store, kde jsem se soustředil na funkce, které vyvolávají

dotazy na server a na jejich následné reakce k Redux Storu. Simuloval jsem funkci dispatch a kontroloval, jak byla volána.

Pro statickou analýzu kódu používám ESLint, ten kontroluje syntax, formátování anebo potencionální zanesení bugy. Primárně jsem se zaměřil na samotný vývoj a uživatelské rozhraní, protože se jedná o projekt, který se neočekává, že bude spravován více lidmi, resp. více jak jedním. Zároveň je aplikace velmi čitelná svojí strukturou.

2.9 Uživatelské rozhraní

Při vytváření jsem se výrazně soustředil na samotné uživatelské rozhraní. Cílem bylo, aby byla aplikace dobře použitelná na různých zařízeních a aby všechny akce byly dobře predikovatelné a uživatel vždy věděl, v jakém stavu se aplikace nachází. Chtěl bych zde vyzdvihnout několik příkladů, jak jsem se toho snažil docílit.

2.9.1 Základní rozdělení stránky

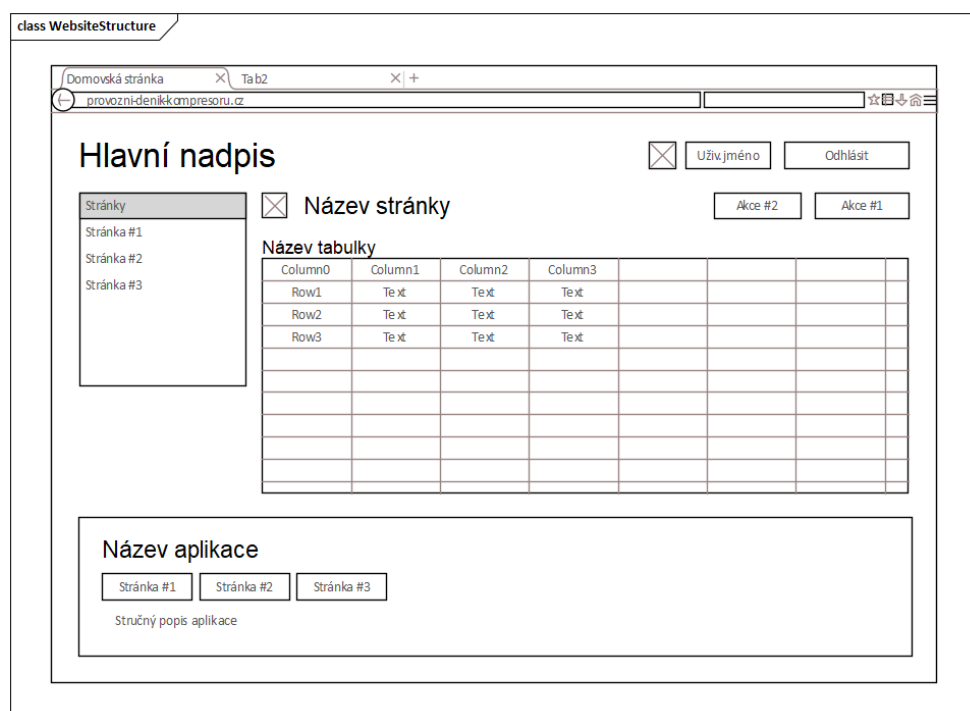
Téměř každá webová stránka nebo aplikace následuje typické rozdělení. Docílí se tak toho, že uživatel, který ještě stránku nenavštívil, dokáže rozpoznat, co jednotlivé části představují. Těmi myslím například hlavní navigaci, postranní menu, nebo patičku. V HTML 5 mají tyto hlavní a časté prvky i vlastní značky. Tento fakt také podporuje směr, kdy by každý element na stránce měl vypovídat o sobě podle názvu značky a použití div elementů, tedy elementů, které nenesou nějaký určitý smysl, mělo být minimální.

Bylo by správné, aby Webový provozní deník kompresoru, také tento směr následoval. Rozhodl jsem se stránku rozdělit podle následujícího Obrázku 2.6.

Hlavní horní lišta Neposouvá se s posunem stránky dolů. Nabízí odkaz na domovskou stránku přes Název aplikace. Je důležité vzhledově rozdělit aktivní odkaz a neaktivní, resp. když je uživatel na domovské stránce, pak je název plný, jinak je barva názvu světlejší. Na pravé straně horní lišty je pak odkaz pro odhlášení. Přihlášení bude zajišťovat vyskakovací formulář. Pokud je uživatel přihlášený, pak je vedle tlačítka na odhlášení také username a ikonka, která reprezentuje nejvyšší roli přihlášeného uživatele. Obrázek 2.7

Postranní lišta Slouží jako hlavní navigace. Obsahuje stránky jednotlivých entit. Jedná se o navigaci stránek druhé úrovně. Zde je důležité, aby aktuální stránka měla viditelně neaktivní odkaz. Pro jednotlivé stránky jsem přidal ikony tak, aby pokud je uživatel již po několikáté na stránce, ihned rozpoznal, o jakou stránku se jedná. Obrázek 2.8

Hlavní obsah stránky Je rozdělen na nadpis s tlačítky povolených akcí a samotný obsah. Tato část stránky je omezená na maximální šířku. To je dů-



■ **Obrázek 2.6** Základní rozložení stránky, Enterprise architect, vlastní tvorba

ležitě, pokud uživatel navštívil aplikaci na širokém monitoru, aby nebyl nucený při prohlížení stránky otáčet hlavu.

Patička Tato část je také důležitá pro uživatelskou čitelnost a orientaci. Pokud se uživatel dostane na konec, barevný kontrast ho upozorní, že je již v jiné části a pravděpodobně na konci. Patička shrnuje aplikaci, obsahuje odkazy na podstránky, název aplikace a popis aplikace.

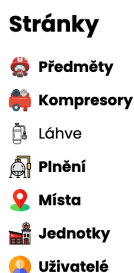
Webový provozní deník kompresoru



■ **Obrázek 2.7** Hlavní horní lišta aplikace, Provozní deník, vlastní tvorba

2.9.2 Vzdušnost stránky

Pro správnou čitelnost stránek existuje několik způsobů rozložení. Sekce lze rozdělit tak, že uživatel na obrazovce vidí pouze jednu a další se vyměňují podle toho, jak uživatel na stránce sjezdí nebo vyjíždí. Takové rozložení ale podle mého názoru není praktické, pokud se očekává, že uživatel bude potřebovat často mezi sekcemi přecházet a číst. Proto volba rozložení sekcí záleží



■ **Obrázek 2.8** Postranní lišta s navigací, Provozní deník, vlastní tvorba

na tom, co se od stránky očekává a jaký má účel. Pokud by se sekce zobrazovaly jednotlivě na obrazovce, pak by jejich rozložení preferovalo zaujetí uživatele před praktičností. Jednotlivé sekce lze také skrýt a zobrazit po jejich rozevření. Často se také používá barevné rozdělení, kde má každá sekce jinou barvu pozadí.

V aplikaci se jednotlivé sekce zobrazují hned pod sebou. Vzdušnost aplikace spočívá v tom, že velikost odsazení je vždy podle toho, v jaké kontextu resp. na jaké úrovni se dva elementy nacházejí. Je tak důležité toto odsazení kontrolovat a nedopustit chybu. Ta je pak velmi viditelná a může vést ke zmatení uživatele. Barvy jsem nepoužil pro rozdělení obsahu vůbec. Tímto způsobem tak obsah dobře vynikne a zároveň uživatele neruší animace, barvy nebo tlačítka pro otevření.

Pro lepší orientaci jsem také použil malé horizontální oddělovače. Ty jsou slabě viditelné, takže uživatele neruší při čtení obsahu. Uživatel ale rozpozná, že se nachází v jiné části. Slouží také pro uchycení elementů do kontextu. Pokud by byla stránka příliš vzdušná, například pokud je nadpis příliš krátký, pak je tlačítko na konci nadpisu volné a horizontální oddělovač jej spáruje s nadpisem.

2.9.3 Vyskakovací formulář

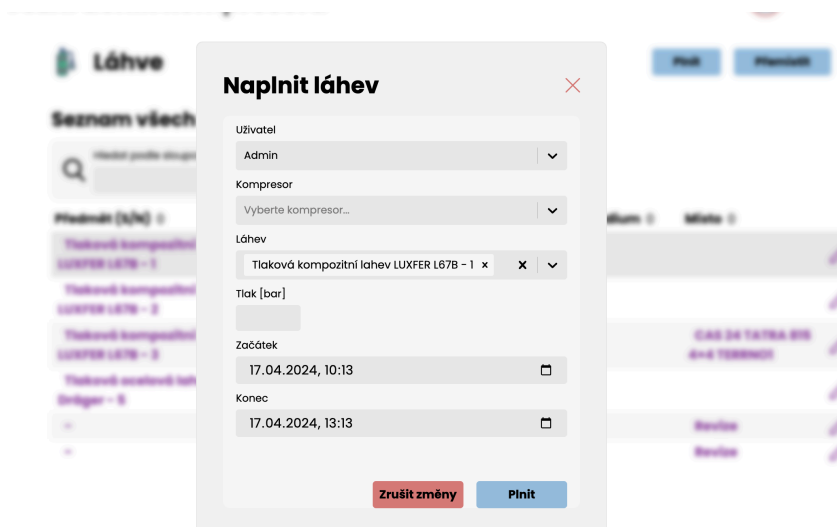
Vyskakovací formuláře jsem v aplikaci použil na více místech. První myšlenka byla, neboť JWT pro autentizaci expiruje již po jedné hodině, zamezit ztrátě rozpracovaných dat. Proto když token expiruje, vyskočí formulář přes okno. To, co je na pozadí, zůstane. U formuláře pro přihlášení ale není možné přejít zpět k rozpracované práci, dokud se uživatel nepřihlásí.

Vyskakovací okno tak představuje povinnou nebo dobrovolnou cestu, kterou se uživatel může vydat. Lze z ní ale vždy (kromě expirace sezení) sejít na tu hlavní. Ve velkém počtu aplikacích by takové okno bylo implementováno pomocí jednotlivých stránek. Pokud je aplikace renderována již na straně serveru, může průchod takovou cestou představovat nepříjemnou zkušenost, protože uživatel při každém kroku čeká na odpověď.

Design jsem zvolil takový, aby byla vidět aplikace stále za formulářem.

Vše kromě samotného formuláře je rozmazané, uživatel tak dostane představu, že formulář je před samotnou aplikací. Plně se koncentruje na samotné okno a obsah stránky za ním ho nijak nevyrušuje.

Některé vyskakovací formuláře nabízejí jejich zavření již po kliknutí mimo samotný formulář. Toto jsem neimplementoval, protože by na telefonu mohlo takové kliknutí být posun stránkou. Chci, aby uživatel zavřel formulář pouze úmyslně. Tlačítko pro zavření je vždy přítomné vedle nadpisu a je dostatečně barevně a vzdáleností oddělené.



■ **Obrázek 2.9** Vyskakovací okno s formulářem plnění láhve, Provozní deník, vlastní tvorba

2.9.4 Postranní lišta

Postranní lišta resp. navigace (Obrázek 2.8) představuje několik prvků a rozhodnutí, které nejsou na první pohled viditelné. Rozcestí odkazů, které nabízí, může být také umístěno pod hlavní lištou. Může také být pevně umístěné na obrazovce. Volba zobrazovat postranní navigaci vedle hlavního obsahu byla hlavně proto, že dnes desktopy disponují často širokými monitory. Maximální šířku hlavního obsahu jsem omezil, takto toto postranní místo není zbytečně nevyužité. Zároveň pokud by byla navigace pod hlavní lištou, zabírala by zbytečně velký poměr vertikální výšky obrazovky.

Upravil jsem pozici tak, aby pokud přejede uživatel níže, zůstala navigace stále vidět. Toho jsem docílil díky poměrně nové hodnotě pozice ve stylech viz. ukázka níže. Je také žádoucí nastavit pozici od vrchu obrazovky.

```
position: sticky; top: 2rem;
```

Webový provozní deník je také dostupný pro mobilní zařízení. Zde je potřeba, aby postranní lišta nezabírala pevnou část obrazovky. Povedlo se mi nastavit styly tak, aby navigace byla schovaná za levou stranou obrazovky a dostupná po kliknutí na štítek, který ji označuje. Pro uživatele tato akce simuluje reálnou záložku například v poznámkovém bloku. Po kliknutí na libovolný odkaz nebo znovu na štítek se navigace zase schová.

2.9.5 Použití barev a ikon

Barvy jsou důležité prvky uživatelského rozhraní, ale jejich nadměrné použití vede ke zmatení uživatele a zneřehlednění aplikace. Proto jsem vybral pět barev, které jsem použil. Je mezi nimi několik stupňů světlosti bílé barvy, pak modrá, zelená, červená a fialová barva. Každá má svůj význam.

Modrá

- Zvýrazňuje libovolnou akci, kterou může vyvolat uživatel.
- Neutrální barva, která se nesnaží vybídnout uživatele, aby na ni klikl, ale nabízí mu tuto možnost.

Zelená

- Slouží jako symbol úspěchu.
- Používám ji jako barvu textu notifikace po úspěšném dotazu na server.

Červená

- Symbolizuje opak zelené barvy.
- Pokud je v aplikaci něco špatně. Varuje uživatele, že po kliknutí se stane nevratná událost, jako je smazání záznamu nebo odhlášení.

Fialová

- Slouží pouze pro odkazy.
- Uživatel tak ví, že pokud klikne na text s touto barvou, dostane se na jinou požadovanou stránku nebo část stránky.

Odstíny bílé

- Slouží hlavně pro oddělení kontextů.

Jednotlivé barvy také rozlišují role uživatele. Ikona pro uživatele s administrátorskými právy je ohraničena červenou barvou. Pro uživatele technika modrou a pro manipulátora fialovou. Ostatní uživatele sboru mají okraj ikony šedou a nejnudnější barvou.

Ikony v aplikaci používám z několika důvodů. První je uživateli nabídnout jinou možnost memorizace stránky. Jsou tak označeny všechny stránky nebo

■ Výpis kódu 2.4 Použití hodnoty wrap u stylu

```
.description_list
  display: inline-flex
  flex-wrap: wrap
```

samotná aplikace. Ikony jsou také určitým atraktivním prvkem. Lze s nimi „lákat“ uživatele. V postranní navigaci jsou ikony barevné pro ostatní stránky, než je ta aktuální. Aktuální stránka má ikonu černobílou.

Další užití ikon je odstranění opakujícího se textu, například u tlačítka „editovat“. Stránka se tak odlehčí a zároveň láká uživatele, aby vyzkoušel editaci záznamu. Červená ikona koše zase smaže po potvrzení záznam. Využívám ikonu i pro změnu stavu uživatele. Ta ukazuje, jaký stav chci, aby uživatel měl při kliknutí na ikonu.

2.9.6 Optimalizace pro různé šířky displejů

Responzivita aplikace je definována pouze pomocí stylů. Při vývoji aplikace jsem styly jednotlivých objektů psal tak, aby byly automaticky responzivní. Využíval jsem k tomu tzv. grid nebo flex. Hodnota grid nabízí možnost automaticky sloupce zalomit na další řádek. Hodnota flex má atribut pro automatické zalomení. Ukazuji příklad použití flexu ve Výpis kódu 2.4.

Jak jsem zmínil výše, postranní navigace je automaticky schována na obrazovkách telefonů. Ostatní objekty se seřadí na vertikálu místo horizontály. Tabulky je vhodné neřešit a pouze nastavit jejich přetékaní tak, aby bylo možné je posunout a sloupce nechat přirozeně vedle sebe.

Je také důležité zmínit, že určitá šířka obrazovky je také při tisku stránky aplikace. Při tisku je ale spíše důležité smazat prvky, které není potřeba mít na papíře, a tím ušetřit místo pro obsah. Zároveň je vhodné změnit barvu pozadí, aby tisk šetřil toner.

2.9.7 Tmavý režim

Hlavním benefitem tmavého režimu je, že aplikace je více příjemná pro uživatele ve tmě. V Česku je v zimním období tma již odpoledne, proto lze očekávat, že aplikace bude používána na hasičské stanici také ve tmě. Nehledě na fakt, že mohou být jednotky povolány i uprostřed noci.

Již v roce 2024 se stává standardem, aby libovolné prostředí nabízelo tmavý režim. Proto pro budoucnost aplikace je extra žádoucí, aby také tento režim nabízela. Osobně mi přijde hodně nepříjemné, když v pokoji ve tmě používám delší dobu libovolnou aplikaci ve tmavém režimu a následně navštívím třeba jen jednu stránku na internetu ve světlém režimu. Dovolím si predikovat, že tyto radikální změny tak budou s nastupujícím trendem stále více nepříjemné.

■ Výpis kódu 2.5 Změna zdroje obrázku v HTML

```
<picture>
  <source media="(prefers-color-scheme: dark)"
    srcSet={`/images/ui/edit_white.svg`} />
  <img width="24px" src={`/images/ui/edit.svg`}
    alt="Ikona úpravy záznamu" />
</picture>
```

■ Výpis kódu 2.6 Změna hodnot proměnných barev

```
@media (prefers-color-scheme: dark)
:root
  --theme-ui-colors-text: white
  --clr-black: white
  --clr-almost-black: rgb(220, 220, 220)
  --clr-white: black
  --clr-almost-white: rgb(35, 35, 35)
  --clr-semi-white: rgb(55, 55, 55)
  --clr-almost-white-transparent: rgba(35, 35, 35, .7)
  --clr-darker-white: rgb(85, 85, 85)
  --clr-blue: rgb(76, 105, 179)
  --clr-lighter-blue: rgb(37, 76, 108)
  --clr-lighter-red: #b75050
  --clr-green: #7fc279
  --clr-active: #ba7bc8
```

Samotná implementace tmavého režimu je přímočará. Je ale nutný předpoklad, aby byly všechny barvy a obrázky, které se v aplikaci vyskytují a očekává se, že se budou měnit, správně definované. Například ve stylech lze již jednoduše definovat proměnné, které jsou následně propagované přes celou aplikaci. V kombinaci s tzv. media query, lze jednoduše nahradit jejich hodnoty při tmavém režimu. U obrázku lze použít poměrně nový způsob změny zdroje v závislosti také na media query. Oba případy ukazují ve Výpis kódu 2.6 a Výpis kódu 2.5.

Tmavý režim aplikace je tedy závislý na media query, který se dotazuje, jaký režim je nastavený v prostředí zařízení. Je tedy možné, aby aplikace byla v tmavém režimu uprostřed dne. Zároveň ale není možné nastavit režim ručně v aplikaci.

2.9.8 Přístupnost aplikace

Posledním ale neméně důležitým faktorem ovlivňující uživatelské rozhraní je jeho samotná přístupnost. Tím se myslí jak přístupnost pro zdravotně tělesně postižené, tak přístupnost pro webové boty, které si indexují stránky. Bohužel je aplikace napsána v JS frameworku, a proto je potřeba pro její zobrazení Javascript. Dnes se ale stává standardem jej podporovat.

Při psaní aplikace jsem dodržoval několik zásad, aby byla aplikace dostatečně přístupná.

- **Správná volba HTML tagů.** Jak jsem již zmínil, je vhodné volit značky jednotlivých elementů správně, tak aby nebyly obecné, pokud to je sebevíc možné. Bot pak dokáže lépe pochopit význam obsahu a čtečky zase přistupovat k elementům, tak jak se to očekává.
- **Popisky obrázků.** Každý obrázek je označen atributem, který obsahuje popis jeho obsahu. To je zásadní pro čtečky, které logicky nedokážou zobrazit jejich obsah. Zároveň pokud se stránka resp. obrázek pomaleji načítá, tak se zobrazí jeho popis.
- **Dostatečně výrazné odkazy.** Každý odkaz by měl být dostatečně vizuálně odlišitelný od ostatního obsahu. Toho jsem dosáhl především změnou barvy textu a použitím ikony. Každý odkaz by měl být odlišitelný alespoň dvěma způsoby. Například barvou a style fontu nebo jeho velikostí. To se mi bohužel nepovedlo všude.
- **Zobrazovat obrázky jako HTML elementy.** Někdy se obrázky zobrazují jako pozadí nastavené pomocí stylu. Pak ale při tisku stránky obrázek zmizí, proto jsem všechny ikony a obrázky zobrazil pomocí HTML elementu.
- **Všechny funkce aplikace jsou dostupné pouze po kliknutí.** Všechny funkce, které aplikace nabízí, musí být dostupné po sekvenci kliknutí. Pokud by bylo zapotřebí například podržení nebo přetažení, pak nastanou situace, kdy toho uživatel nebude schopen.

Závěr

Cílem práce bylo vytvořit webovou aplikaci, která pomáhá v evidenci provozu vysokotlakého vzduchového kompresoru jednotkám sborů dobrovolných hasičů. V rámci toho bylo zapotřebí řádně analyzovat současný stav, povinnosti evidence a obecně provoz typické JSDH, tak aby aplikace odpovídala jejich požadavkům. Požadavky zahrnovaly nenáročnost na hostování a nasazení aplikace, uživatelskou přívětivost aplikace a nízké nároky na platformu a s ní související omezené finanční prostředky na provoz. Výsledná práce tyto požadavky splňuje.

Analýza stavu a provozu jednotek sborů dobrovolných hasičů byla zahrnuta v teoretické části, kde jsou přítomny jednotlivé tabulky převzaté z řádů, doporučení a směrnic SDH. Jsou zde rozepsány také popisy jednotlivých akcí, jako plnění lahví nebo revize hasičských prostředků. Následně jsem se věnoval uživatelskému rozhraní a již existujícím alternativám, ze kterých lze přebrat určitou inspiraci. V neposlední řadě důležitý požadavek na otevřenou licenci tohoto projektu, aby mohly jednotlivé sbory tuto aplikaci volně využívat a popřípadě také rozšiřovat o další funkcionality.

Praktická část uvádí funkční a nefunkční požadavky, které vzešly ze zadání a analýzy. Na ně navazují základní a nutné případy užití, které aplikace musí v závislosti na funkčních a nefunkčních požadavcích nabízet. Následuje argumentace pro volbu technologií jednotlivých vrstev aplikace. Pak již je rozepsána samotná implementace, jsou zde vyzdvižená řešení problémů, které se vyskytly při vývoji, nebo zajímavé řešení požadavků. V poslední sekci této kapitoly se věnuji uživatelskému rozhraní. Popisuji prvky rozhraní, které jsem použil. Proč jsem je volil a celkově jaká byla myšlenka za tvorbou uživatelského rozhraní.

Aplikace je dostupná v příloze bakalářské práce včetně souborů s dokumentací pro nasazení na různé typy hostingových služeb. Je zpřístupněna na veřejné doméně automatickou integrací z verzovacího systému pro její testování a diskuzi nad novými funkcionalitami.

Výsledná práce stále nabízí prostor pro další vývoj. Prvním takovým je rozdělení revizí prostředků na jednotlivé typy. Většina prostředků hasičské

zbrojnice je revidována techniky podle různých zaměření. Lze například kontrolovat jen určité komponenty nebo provést revizi většího zaměření jednou za delší časový úsek, nebo menší a častěji. Od těchto typů revizí se odvíjejí i následující data takových revizí. Jako druhý volný prostor je podpora více jazyků. Členové jednotek sborů dobrovolných hasičů mohou být cizinci a rozumět česky jen částečně. Při vývoji aplikace již existovaly poměrně pohodlné možnosti pro zavedení multijazyčnosti. Do budoucna lze také aplikaci doplnit o zbylé funkcionality, kterými disponuje webový portál PORT.ALL, tak aby byly jednotky schopné kompletně přejít na tento webový provozní deník.

Detail oprav a kontrol TL

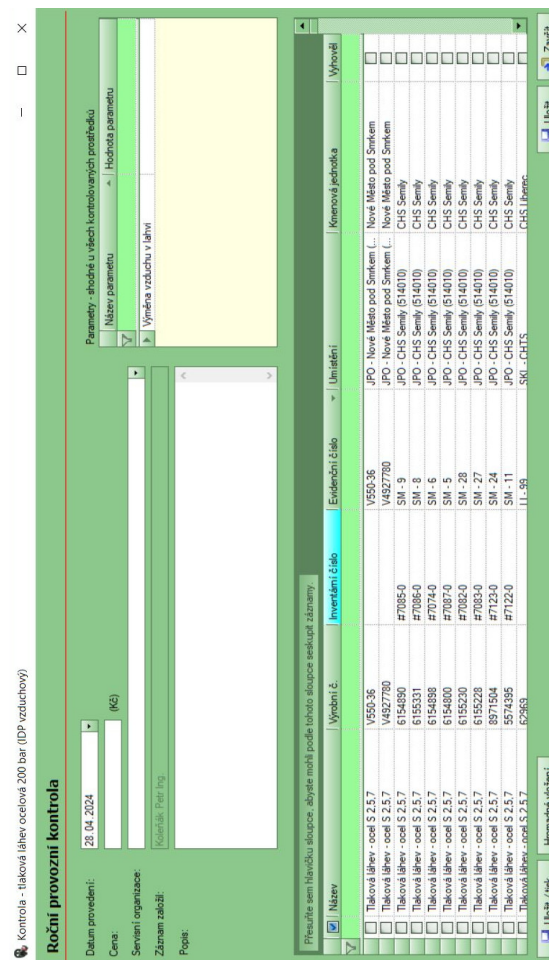
Opravy/Provozní kontroly/Revize

Přesuňte sem hlavičku sloupce, abyste mohli profile tohoto sloupce seskupit záznamy.

Název	Evidenční číslo	Výrobní číslo	Cena celkem			
tlaková lahev ocelová 3...		FO 2537	0			
DETAIL	Typ záznamu	Název	Název servisu o... Pops	Datum provedení	Cena	Vyhoví
Ediřace	Provozní kontrola	Počíní provozní ...	SDH Cvikov	12.11.2023	0	<input checked="" type="checkbox"/>
Ediřace	Revize	5ti letá revize dle...	Dräger Safety - P...	26.10.2023	0	<input checked="" type="checkbox"/>

Obrázek A.1 Detail oprav, revizí a provozních kontrol ocelové tlakové lahve, IKIS II, Ing. Petr Koleňák

Roční provozní kontrola



■ Obrázek B.1 Roční provozní kontrola prostředku, IKIS II, Ing. Petr Kolečák

Příloha C

Plánované kontroly

Název proskvětku	Název	Evidenční č.	Výrob.	Kmenová jednotka	Typ provozní kont.	Název	Datum od kdy je	Přít. i provozní	Zbývá do	Datum poslední
Tlaková šňev	tlaková		PO 2362	Cvikov	Revize	Šň lea	2028.08.27	2028.10.26	1641	2023.10.26
Tlaková šňev	tlaková		PO 2364	Cvikov	Provozní kontrola	Roční	2024.10.13	2024.11.12	197	2023.11.12
Tlaková šňev	tlaková		PO 2364	Cvikov	Revize	Šň lea	2018.08.31	2018.10.30	-3007	2018.10.30
Tlaková šňev	tlaková		PO 2370	Cvikov	Provozní kontrola	Roční	2024.10.13	2024.11.12	197	2023.11.12
Tlaková šňev	tlaková		PO 2370	Cvikov	Revize	Šň lea	2018.08.31	2018.10.30	-5007	2018.10.30
Tlaková šňev	tlaková		PO 2384	Cvikov	Provozní kontrola	Roční	2024.10.13	2024.11.12	197	2023.11.12
Tlaková šňev	tlaková		PO 2384	Cvikov	Revize	Šň lea	2028.08.27	2028.10.26	1641	2023.10.26
Tlaková šňev	tlaková		PO 2537	Cvikov	Provozní kontrola	Roční	2024.10.13	2024.11.12	197	2023.11.12
Tlaková šňev	tlaková		PO 2537	Cvikov	Revize	Šň lea	2028.08.27	2028.10.26	1641	2023.10.26
Tlaková šňev	tlaková		PO 2544	Cvikov	Provozní kontrola	Roční	2024.10.13	2024.11.12	197	2023.11.12
Tlaková šňev	tlaková		PO 2544	Cvikov	Revize	Šň lea	2028.08.27	2028.10.26	1641	2023.10.26
Tlaková šňev	tlaková		PO 2547	Cvikov	Provozní kontrola	Roční	2024.10.13	2024.11.12	197	2023.11.12
Tlaková šňev	tlaková		PO 2547	Cvikov	Revize	Šň lea	2023.08.02	2023.10.01	-210	2018.10.01
Tlaková šňev	tlaková			Cvikov	Provozní kontrola	Roční	2024.10.13	2024.11.12	197	2023.11.12
Tlaková šňev	tlaková			Cvikov	Revize	Šň lea	2027.09.10	2027.11.09	1289	2022.11.09
Tlaková šňev	tlaková		3657601	Cvikov	Provozní kontrola	Roční				
Tlaková šňev	tlaková		3657601	Cvikov	Revize	Šň lea				
Tlaková šňev	tlaková		4044800	Cvikov	Provozní kontrola	Roční				
Tlaková šňev	tlaková		4044800	Cvikov	Revize	Šň lea				
Tlaková šňev	tlaková		4303700	Cvikov	Provozní kontrola	Roční				
Tlaková šňev	tlaková		4303700	Cvikov	Revize	Šň lea				
Tlaková šňev	tlaková		4303804	Cvikov	Provozní kontrola	Roční				
Tlaková šňev	tlaková		4303804	Cvikov	Revize	Šň lea				
Tlaková šňev	tlaková		4304210	Cvikov	Provozní kontrola	Roční				
Tlaková šňev	tlaková		4304210	Cvikov	Revize	Šň lea				
Tlaková šňev	tlaková		4650298	Cvikov	Provozní kontrola	Roční				
Tlaková šňev	tlaková		4650298	Cvikov	Revize	Šň lea				
Tlaková šňev	tlaková		4653785	Cvikov	Provozní kontrola	Roční				
Tlaková šňev	tlaková		4653785	Cvikov	Revize	Šň lea				

■ Obrázek C.1 Plánované kontroly (upraveno), IKIS II, Ing. Petr Koleňák

Příloha E

Snímek z finální verze aplikace








Webový provozní deník kompresoru



Admin

Odhlásit

Stránky

-  Předměty
-  Kompresory
-  Láhve
-  Plnění
-  Místa
-  Jednotky
-  Uživatelé



Láhve

Plnit

Přemístit

Přidat láhev

Seznam všech lahví



Hledat podle sloupce Předmět (S/N) ▾

Předmět (S/N) ⇅	Kapacita [l] ⇅	Plnicí tlak [bar] ⇅	P
Tlaková kompozitní lahev LUXFER L67B - 1	463		▼
Tlaková kompozitní lahev LUXFER L67B - 2	399		
Tlaková kompozitní lahev			

■ Obrázek E.1 Snímek pořízený z provozního deníku, vlastní tvorba

Bibliografie

1. SDRUŽENÍ HASIČŮ ČECH, MORAVY A SLEZSKA. *Úvodní stránka* [online]. SH ČMS, 2024 [cit. 2024-04-09]. Dostupné z: <https://www.dh.cz/>.
2. MILAN, Zdeněk. *Vše o MHJ - Moravská hasičská jednota* [online]. MHJ, 2011-01-01 [cit. 2024-04-09]. Dostupné z: <https://mhj.hasicovo.cz/cs/o-nas>.
3. ČESKO. Zákon č. 133/1985 Sb. ze dne 17. 12. 1985 České národní rady o požární ochraně. *Sbírka zákonů České republiky* [online]. 1985, č. 34 [cit. 2024-04-09]. ISSN 1211-1244. Dostupné z: <https://www.zakonyprolidi.cz/cs/1985-133>.
4. SDRUŽENÍ HASIČŮ ČECH, MORAVY A SLEZSKA. *Vzor smlouvy SHCMS obec* [online]. SH ČMS, 2024 [cit. 2024-04-09]. Dostupné z: www.dh.cz/images/Dokumenty/smlouvy/vzor-smlouvy-shcms-obec.docx.
5. ČESKO. Příloha k zákonu č. 133/1985 Sb. ze dne 17. 12. 1985 České národní rady o požární ochraně. *Sbírka zákonů České republiky* [online]. 1985, č. 34 [cit. 2024-04-09]. ISSN 1211-1244. Dostupné z: <https://www.podnikatel.cz/zakony/zakon-c-133-1985-sb-zakon-o-pozarni-ochrane/f2805562/#aktualni-zneni>.
6. ČESKO. Vyhláška č. 247/2001 Sb. ze dne 23. 07. 2001 Ministerstva vnitra o organizaci a činnosti jednotek požární ochrany. *Sbírka zákonů České republiky* [online]. 2001, č. 95 [cit. 2024-04-09]. ISSN 1211-1244. Dostupné z: <https://www.zakonyprolidi.cz/cs/2001-247#f5765993>.
7. MV – GENERÁLNÍ ŘEDITELSTVÍ HASIČSKÉHO ZÁCHRANNÉHO SBORU ČR. *Řád výkonu služby v jednotkách hasičských záchranných sborů podniků, sborů dobrovolných hasičů obcí a sborů dobrovolných hasičů podniků* [online]. Ministerstvo vnitra, 2021-12-22 [cit. 2024-04-09]. Dostupné z: <https://www.hzscr.cz/clanek/vykon-sluzby.aspx?q=Y2hudW09Ng%3D%3D>.

8. MAGISTRÁT HL. M. PRAHY, ODBOR BEZPEČNOSTI. *Detail článku / Bezpečnostní portál Magistrátu hl. m. Prahy* [online]. Magistrát HMP, 2024 [cit. 2024-04-09]. Dostupné z: <https://bezpecnost.praha.eu/clanky/jednotky-sboru-dobrovolnych-hasicu>.
9. SDRUŽENÍ HASIČŮ ČECH, MORAVY A SLEZSKA. *Výroční zpráva 2022* [online]. SH ČMS, 2022 [cit. 2024-04-11]. Dostupné z: https://www.dh.cz/images/Dokumenty/vyrocní-zprava/2022/vz-2022_SH-CMS_online.pdf.
10. NONAME. *Prostředky chemické služby* [online]. Sbor dobrovolných hasičů Praha - Zličín, 2024 [cit. 2024-04-11]. Dostupné z: www.sdhzlicin.cz/index.php/jednotka/prostredky-chemicke-sluzby.
11. HASIČSKÝ ZÁCHRANNÝ SBOR ČR. *Finanční zdroje na obměnu a nákup nové techniky pro jednotky požární ochrany a vodní záchrannou službu v letech 2015 - 2017* [online]. Praha: HZS ČR, 2016-03-10 [cit. 2024-04-11]. Dostupné z: <https://www.hzscr.cz/soubor/financni-zdroje-pro-jpo-a-vzs-pdf.aspx>.
12. MATĚJKA, Jiří. *Chemická služba: učební skripta* [online]. Praha: Ministerstvo vnitra - generální ředitelství Hasičského záchranného sboru ČR, 2012 [cit. 2024-04-11]. ISBN 978-80-87544-09-9. Dostupné z: <https://www.hzscr.cz/soubor/chemicka-sluzba-2020-pdf.aspx>.
13. NONAME. *sady dýchacích přístrojů / Požární bezpečnost s.r.o.* [online]. CORA computer s.r.o., 2024 [cit. 2024-04-11]. Dostupné z: <https://www.vyzbrojna.cz/cz/2501/sady.html>.
14. CHURCHVILLE, Fred. *What is user interface (UI)? Definition from SearchAppArchitecture* [online]. TechTarget, 2021 [cit. 2024-04-11]. Dostupné z: <https://www.techtarget.com/searchapparchitecture/definition/user-interface-UI>.
15. NONAME. *Mobile vs. Desktop Market Share and Usage Statistics* [online]. Kinsta Inc., 2023-11-27 [cit. 2024-04-11]. Dostupné z: <https://kinsta.com/mobile-vs-desktop-market-share/>.
16. RCS KLADNO, S. R. O. *Uživatelská příručka aplikace PORT.ALL JSDH* [online]. HZS ČR, 2024 [cit. 2024-04-10]. Dostupné z: https://jsdh.izscr.cz/Resources/Uzivatelaska_prirucka_aplikace_PortAll.pdf.
17. HASIČSKÝ ZÁCHRANNÝ SBOR ČR. *Modul Chemická služba* [online]. HZS ČR, 2024 [cit. 2024-04-10]. Dostupné z: <https://www.hzscr.cz/soubor/modul-chemicka-sluzba-pdf.aspx>.
18. MINISTERSTVO VNITRA, RCS Kladno, s. r. o. *36 Smlouva SW modulu IKIS* [online]. HZS ČR, 2011 [cit. 2024-04-10]. Dostupné z: <https://www.hzscr.cz/soubor/36-smlouva-sw-modulu-ikis-pdf.aspx>.

19. FREE SOFTWARE FOUNDATION, INC. *The GNU General Public License v3.0 - GNU Project - Free Software Foundation* [online]. Free Software Foundation, Inc., 2007-06-29 [cit. 2024-04-10]. Dostupné z: <https://www.gnu.org/licenses/gpl-3.0.en.html>.
20. ULS LIBRARIAN. *MIT License - Open Licenses: Creative Commons and other options for sharing your work - Guides at University of Pittsburgh* [online]. University of Pittsburgh, 2023-09-13 [cit. 2024-04-10]. Dostupné z: <https://pitt.libguides.com/openlicensing/MIT>.
21. NONAME. *Doctrine* [online]. The Doctrine Project [cit. 2024-04-14]. Dostupné z: <https://www.doctrine-project.org/>.
22. CHALAS, Robin. *LexikJWTAuthenticationBundle* [online]. GitHub Inc., 2024-03-04 [cit. 2024-04-14]. Dostupné z: <https://github.com/lexik/LexikJWTAuthenticationBundle>.
23. KÉVIN DUNGLAS, Les-Tilleuls.coop. *API Platform* [online]. API Platform, 2023 [cit. 2024-04-14]. Dostupné z: <https://api-platform.com/>.
24. NONAME. *JSON Web Token Introduction - jwt.io* [online]. Okta, Inc., 2024 [cit. 2024-04-23]. Dostupné z: <https://jwt.io/introduction>.
25. NONAME. *React* [online]. Meta Platforms, Inc., 2024 [cit. 2024-04-15]. Dostupné z: <https://react.dev/>.
26. NONAME. *SQLite Home Page* [online]. SQLite Consortium, 2024-03-09 [cit. 2024-04-15]. Dostupné z: <https://sqlite.org/>.
27. NONAME. *Visual Studio Code - Code Editing. Redefined* [online]. Microsoft Corporation [cit. 2024-04-16]. Dostupné z: <https://code.visualstudio.com/>.
28. NONAME. *Browse Fonts - Google Fonts* [online]. Google Ireland Limited [cit. 2024-04-16]. Dostupné z: <https://fonts.google.com/>.
29. JASONSAAYMAN. *GitHub - axios/axios: Promise based HTTP client for the browser and node.js* [online]. GitHub, Inc. [cit. 2024-04-16]. Dostupné z: <https://github.com/axios/axios>.
30. PALMER, Jared. *GitHub - jaredpalmer/formik: Build forms in React, without the tears* [online]. GitHub, Inc. [cit. 2024-04-16]. Dostupné z: <https://github.com/jaredpalmer/formik>.
31. NONAME. *React Redux / React Redux* [online]. Dan Abramov, Redux documentation authors [cit. 2024-04-16]. Dostupné z: <https://react-redux.js.org/>.
32. WIERUCH, Robin. *GitHub - table-library/react-table-library: :bento: React Table Library* [online]. GitHub, Inc.s [cit. 2024-04-16]. Dostupné z: <https://github.com/table-library/react-table-library>.

33. REMIX SOFTWARE, INC. *Home v6.22.3 / React Router* [online]. Remix Software, Inc. [cit. 2024-04-16]. Dostupné z: <https://reactrouter.com/en/main>.
34. CHERNEV, Iskren Ivov. *GitHub - moment/moment: Parse, validate, manipulate, and display dates in javascript.* [online]. GitHub, Inc.s [cit. 2024-04-16]. Dostupné z: <https://github.com/moment/moment>.

Obsah příloh

readme.txt	stručný popis obsahu média
archive	adresář s distribucemi aplikace podle způsobu
├─ docker-compose.tar.gz	distribuce pomocí docker-compose
├─ managed-httpd.tar.gz	distribuce na spravovaný Apache httpd
├─ not-managed-httpd.tar.gz	distribuce na hosting Apache httpd
src	
├─ impl	zdrojové kódy implementace
│ ├─ server	zdrojové kódy serveru
│ └─ client	zdrojové kódy klienta
└─ thesis	zdrojová forma práce ve formátu \LaTeX
text	text práce
└─ ctufit-thesis.pdf	text práce ve formátu PDF